

Oracle® Database

Net Services 管理者ガイド

19c

F16136-08(原本部品番号:E96313-10)

2022年9月

タイトルおよび著作権情報

Oracle Database Net Services管理者ガイド, 19c

F16136-08

[Copyright ©](#) 2002, 2022, Oracle and/or its affiliates.

原著者: Binika Kumar

原協力著者: Douglas Williams、Prakash Jashnani

原協力者: Abhishek Dadhich、Alan Williams、Anita Patel、Bhaskar Mathur、Christopher Jones、David Lin、Feroz Khan、Kant Patel、Kevin Neel、Krishna Itikarlapalli、Misaki Miyashita、Murali Purayathu、Norman Woo、Robert Achacoso、Santanu Datta、Saravanakumar Ramasubramanian、Sarma Namuduri、Scot McKinley、Sharad Chandran R、Srinivas Pamu、Steve Ding、Sudeep Reguna、Thanigai Nallathambi、Yi Ouyang

目次

- [タイトルおよび著作権情報](#)
- [はじめに](#)
 - [対象読者](#)
 - [ドキュメントのアクセシビリティについて](#)
 - [ダイバーシティ&インクルージョン](#)
 - [関連ドキュメント](#)
 - [表記規則](#)
- [『Oracle Database Net Services管理者ガイド』のこのリリースでの変更点](#)
 - [新機能](#)
 - [非推奨となった機能](#)
- [第I部 Oracle Net Servicesの理解](#)
 - [1 Oracle Net Servicesの概要](#)
 - [1.1 Oracle Net Servicesについて](#)
 - [1.1.1 接続の理解](#)
 - [1.1.1.1 クライアント/サーバー・アプリケーション接続について](#)
 - [1.1.1.1.1 Javaクライアント・アプリケーション接続](#)
 - [1.1.1.2 Webクライアント・アプリケーション接続について](#)
 - [1.1.1.2.1 JavaアプリケーションWebサーバーからのWebクライアント接続](#)
 - [1.1.1.2.2 アプリケーションWebサーバーを使用しないWebクライアント接続](#)
 - [1.1.2 管理性の理解](#)
 - [1.1.2.1 位置の透過性について](#)
 - [1.1.2.2 集中構成と管理について](#)
 - [1.1.2.3 クイック・インストールおよび構成について](#)
 - [1.1.3 共有サーバー・アーキテクチャの理解](#)
 - [1.1.4 パフォーマンスの理解](#)
 - [1.1.4.1 リスナー・キュー・サイズ](#)
 - [1.1.4.2 セッション・データ・ユニットのサイズによるデータ転送の最適化](#)
 - [1.1.4.3 TCP/IPの固定バッファ・フラッシング](#)
 - [1.1.4.4 Sockets Direct Protocol](#)
 - [1.1.4.5 データベースの可用性](#)
 - [1.1.5 ネットワーク・セキュリティの理解](#)
 - [1.1.5.1 ファイアウォール・アクセス制御](#)
 - [1.2 データベース・インスタンスの理解](#)
 - [1.3 Oracle Net Servicesのコンポーネント](#)
 - [1.3.1 Oracle Netについて](#)
 - [1.3.1.1 Oracle Net Foundationレイヤー](#)
 - [1.3.1.2 Oracle Protocol Support](#)
 - [1.3.2 Oracle Net Listenerについて](#)
 - [1.3.3 Oracle Connection Managerについて](#)

- [1.3.4 ネットワーキング・ツールについて](#)
 - [1.3.5 Oracle Advanced Securityについて](#)
 - [2 データベースの識別とアクセス](#)
 - [2.1 データベース・インスタンスの理解](#)
 - [2.2 データベース・サービスの理解](#)
 - [2.3 データベース・サービスへの接続](#)
 - [2.3.1 接続記述子について](#)
 - [2.3.1.1 接続記述子のIPv6アドレスについて](#)
 - [2.3.2 プロトコル・アドレスについて](#)
 - [2.3.3 サービス登録について](#)
 - [2.3.3.1 インスタンス名の指定](#)
 - [2.3.3.2 サービス・ハンドラの指定](#)
 - [2.4 サービス・ハンドラの理解](#)
 - [2.4.1 ディスパッチャについて](#)
 - [2.4.2 専用サーバー・プロセスについて](#)
 - [2.4.3 データベース常駐接続プーリングについて](#)
 - [2.5 ネーミング・メソッドの理解](#)
 - [2.5.1 ネーミング・メソッドの選択](#)
 - [2.5.2 ネーミング・メソッドを使用したクライアント・セッションの確立](#)
 - [2.5.3 接続文字列の入力](#)
 - [2.6 複数リスナーを使用したサービスのアクセス可能性の拡張](#)
 - [2.6.1 接続時フェイルオーバーについて](#)
 - [2.6.2 透過的アプリケーション・フェイルオーバーについて](#)
 - [2.6.3 クライアント・ロード・バランシングについて](#)
 - [2.6.4 接続ロード・バランシングについて](#)
 - [3 ネットワーク・アドレス情報の管理](#)
 - [3.1 ローカル管理の使用](#)
 - [3.2 集中管理用のディレクトリ・サーバーの使用](#)
 - [3.2.1 ディレクトリ情報ツリーの理解](#)
 - [3.2.1.1 ドメイン・コンポーネント・ネームスペースの完全修飾名](#)
 - [3.2.1.2 X.500ネームスペースの完全修飾名](#)
 - [3.2.1.3 エントリの相対名の使用](#)
 - [3.2.1.4 エントリの完全修飾名の使用](#)
 - [3.2.2 Oracleコンテキストの理解](#)
 - [3.2.3 ネット・サービス別名エントリの理解](#)
 - [3.2.4 ディレクトリ・サーバーのエントリを追加または変更できるユーザー](#)
 - [3.2.5 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](#)
 - [3.2.6 ディレクトリ・サーバーを使用するときの考慮事項](#)
 - [3.2.6.1 パフォーマンスに関する考慮事項](#)
 - [3.2.6.2 セキュリティに関する考慮事項](#)
 - [3.2.6.2.1 認証方式](#)
 - [3.2.6.2.2 アクセス制御リスト](#)
 - [3.2.6.3 オブジェクト・クラス](#)

- [3.2.7 Microsoft Active Directoryにおけるディレクトリ・ネーミング・サポートの制約](#)
 - [4 通信レイヤーの理解](#)
 - [4.1 クライアント/サーバー・アプリケーションのOracle Netスタック通信の理解](#)
 - [4.1.1 クライアント通信スタックについて](#)
 - [4.1.1.1 クライアント・アプリケーション・レイヤー](#)
 - [4.1.1.2 プレゼンテーション・レイヤー](#)
 - [4.1.1.3 Oracle Net Foundationレイヤー](#)
 - [4.1.1.4 Oracle protocol supportレイヤー](#)
 - [4.1.2 サーバー通信スタックについて](#)
 - [4.2 JavaアプリケーションのOracle Netスタック通信の使用](#)
 - [4.3 WebクライアントのOracle Netスタック通信の使用](#)
 - [4.4 Oracle protocol supportレイヤーの理解](#)
 - [4.4.1 TCP/IPプロトコルについて](#)
 - [4.4.1.1 IPv6アドレスの表記](#)
 - [4.4.1.1.1 CIDR表記](#)
 - [4.4.1.1.2 URLのIPv6アドレス](#)
 - [4.4.1.1.3 IPv4マップ・アドレス](#)
 - [4.4.1.2 IPv6インタフェースとアドレス構成](#)
 - [4.4.1.3 IPv6のネットワーク接続](#)
 - [4.4.1.4 Oracle Database 12cでのIPv6のサポート](#)
 - [4.4.2 TLSプロトコル付きTCP/IPについて](#)
 - [4.4.3 Named Pipesプロトコルについて](#)
 - [4.4.4 Sockets Direct Protocol \(SDP\)について](#)
 - [4.4.5 Exadirectプロトコルについて](#)
 - [4.4.6 Websocketプロトコルについて](#)
 - [5 Oracle Netアーキテクチャの理解](#)
 - [5.1 サービス登録について](#)
 - [5.2 リスナーおよび接続要求について](#)
 - [5.3 Oracle Restartについて](#)
 - [5.4 ブロックされた接続要求について](#)
 - [5.5 データベース・サーバー・プロセス・アーキテクチャの理解](#)
 - [5.5.1 共有サーバー・プロセスについて](#)
 - [5.5.2 専用サーバー・プロセスについて](#)
 - [5.6 Oracle Connection Managerのアーキテクチャの理解](#)
 - [5.7 完全なアーキテクチャ](#)
- [第II部 Oracle Net Servicesの構成と管理](#)
 - [6 Oracle Net Services接続のクイック・スタート](#)
 - [6.1 接続確立の前提条件](#)
 - [6.2 ネットワークの可用性の確認](#)
 - [6.3 Oracle Net ListenerとOracle Databaseサーバーの起動](#)
 - [6.4 Oracle Connection Managerの起動](#)
 - [6.5 データベースへの接続](#)
 - [6.6 データベースに接続するための簡易接続の使用](#)

- [7 Oracle Net Servicesの管理](#)
 - [7.1 ユーザー・インタフェース・ツールの使用](#)
 - [7.1.1 Oracle Enterprise Manager Cloud Controlを使用したOracle Net Servicesの構成](#)
 - [7.1.1.1 「Net Services管理」ページへのアクセス](#)
 - [7.1.2 Oracle Net Managerを使用したOracle Net Servicesの構成](#)
 - [7.1.2.1 Oracle Net Managerの起動](#)
 - [7.1.2.2 Oracle Net Managerのナビゲート](#)
 - [7.1.2.3 Oracle Net Managerのウィザードの使用](#)
 - [7.1.2.3.1 Netサービス名ウィザードの使用](#)
 - [7.1.2.3.2 ディレクトリ・サーバー移行ウィザードの使用](#)
 - [7.1.3 Oracle Enterprise Manager Cloud ControlおよびOracle Net Managerの使用時期の決定](#)
 - [7.1.4 Oracle Net Configuration Assistantを使用したネットワーク・コンポーネントの構成](#)
 - [7.2 OracleNetAdminsグループについて](#)
 - [7.2.1 OracleNetAdminsグループへのユーザーの追加](#)
 - [7.2.2 OracleNetAdminsグループからのユーザーの削除](#)
 - [7.2.3 OracleNetAdminsグループの所有権の変更](#)
 - [7.3 リスナー制御ユーティリティを使用したリスナーの管理](#)
 - [7.4 共通ネットワーク・タスクの実行](#)
- [8 ネーミング・メソッドの構成](#)
 - [8.1 簡易接続ネーミング・メソッドの構成](#)
 - [8.1.1 簡易接続ネーミング・メソッドについて](#)
 - [8.1.2 簡易接続プラスについて](#)
 - [8.1.3 簡易接続ネーミング・メソッドの例](#)
 - [8.1.4 クライアントでの簡易接続ネーミングの構成](#)
 - [8.1.5 DNS別名を使用するための簡易接続ネーミングの構成](#)
 - [8.2 ローカル・ネーミング・メソッドの構成](#)
 - [8.2.1 インストール中のtnsnames.oraファイル構成](#)
 - [8.2.2 インストール後のtnsnames.oraファイル構成](#)
 - [8.3 ディレクトリ・ネーミング・メソッドの構成](#)
 - [8.4 ディレクトリ・ネーミング・サーバーでの複数のデフォルト・コンテキストの作成](#)
 - [8.5 ローカル・ネーミング・エントリのディレクトリ・ネーミング・サーバーへのエクスポート](#)
 - [8.6 ディレクトリ・ネーミング・エントリのtnsnames.oraファイルへのエクスポート](#)
 - [8.7 外部ネーミング・メソッドの構成](#)
- [9 Oracle Net Listenerの構成と管理](#)
 - [9.1 Oracle Net Listenerの概要](#)
 - [9.2 動的なサービス登録の構成](#)
 - [9.2.1 サービス登録の初期化パラメータの設定](#)
 - [9.2.2 ローカル・リスナーへの情報の登録](#)
 - [9.2.3 リモート・リスナーへの情報の登録](#)
 - [9.2.4 ネットワーク内の全リスナーへの情報の登録](#)

- [9.2.5 ネーミング・メソッドの構成](#)
 - [9.3 インストール中のOracle Net Listenerの構成](#)
 - [9.4 Oracle Net Listener構成のカスタマイズ](#)
 - [9.4.1 リスニング用プロトコル・アドレスの構成](#)
 - [9.4.1.1 Oracle Enterprise Manager Cloud Controlを使用したリスニング用プロトコル・アドレスの構成](#)
 - [9.4.1.2 Oracle Net Managerを使用したリスニング用プロトコル・アドレスの構成](#)
 - [9.4.2 多数の同時接続要求の処理](#)
 - [9.4.3 Oracle Net Listenerのセキュリティの管理](#)
 - [9.4.3.1 有効なノードおよびサブネットの指定](#)
 - [9.5 リスナーの管理](#)
 - [9.5.1 リスナーの起動と停止](#)
 - [9.5.1.1 リスナー制御ユーティリティを使用したリスナーの起動または停止](#)
 - [9.5.1.2 Oracle Enterprise Manager Cloud Controlを使用したリスナーの起動または停止](#)
 - [9.5.2 Oracle Restart構成のリスナーの管理](#)
 - [9.5.2.1 SRVCTLユーティリティを使用した構成済のリスナーの表示](#)
 - [9.5.2.2 SRVCTLユーティリティを使用したリスナーの追加または削除](#)
 - [9.5.2.3 SRVCTLユーティリティを使用したリスナーの起動または停止](#)
 - [9.5.3 リスナーの現在のステータスの確定](#)
 - [9.5.3.1 リスナー制御を使用したステータスの表示](#)
 - [9.5.3.2 Oracle Enterprise Manager Cloud Controlを使用したステータスの表示](#)
 - [9.5.4 リスナーのサービスの監視](#)
 - [9.5.5 リスナー・ログ・ファイルの監視](#)
 - [9.6 リスナーのリダイレクトの理解](#)
 - [10 Oracle Connection Managerの構成](#)
 - [10.1 cman.oraファイルについて](#)
 - [10.2 Oracle Connection Managerの構成方法](#)
 - [10.2.1 Oracle Connection Managerホスト用のcman.oraファイルの構成](#)
 - [10.2.1.1 Oracle Connection Manager \(CMAN\)でのTransport Layer Securityの構成](#)
 - [10.2.1.2 アクセス制御の有効化](#)
 - [10.2.2 Oracle Connection Managerを使用する場合のクライアントの構成](#)
 - [10.2.3 Oracle Connection Managerを使用する場合のOracle Databaseサーバーの構成](#)
 - [10.2.3.1 Oracle Connection Managerで使用するためのサービス登録の構成](#)
 - [10.2.3.2 Oracle Connection Managerのセッションの多重化の有効化](#)
 - [10.2.4 Traffic DirectorモードのOracle Connection Managerの構成](#)
 - [10.2.4.1 Traffic DirectorモードのOracle Connection Managerの使用について](#)

- [10.2.4.2 Traffic DirectorモードのOracle Connection Manager用のcman.oraファイルの構成](#)
- [10.2.4.3 Traffic DirectorモードのOracle Connection Managerのプロキシ認証用のウォレットの構成](#)
 - [10.2.4.3.1 Traffic DirectorモードのOracle Connection Managerでの外部パスワード・ストアの使用](#)
- [10.2.4.4 Traffic DirectorモードのOracle Connection Managerのプロキシ認証用のデータベースの構成](#)
- [10.2.4.5 Traffic DirectorモードのOracle Connection Managerでのサービス登録の構成](#)
- [10.2.4.6 Traffic DirectorモードのOracle Connection Managerのプロキシ常駐接続プールの構成](#)
- [10.2.4.7 計画外イベントのためのTraffic DirectorモードのOracle Connection Managerの構成](#)
- [10.2.4.8 計画済停止イベントのためのTraffic DirectorモードのOracle Connection Managerの構成](#)
- [10.2.4.9 サービス・アフィニティのためのTraffic DirectorモードのOracle Connection Managerの構成](#)
- [10.2.4.10 Traffic DirectorモードのOracle Connection ManagerでのTransport Layer Securityの構成](#)
- [10.2.4.11 Traffic DirectorモードのOracle Connection Managerの制限事項](#)
- [10.3 IPv4およびIPv6のブリッジとしてのOracle Connection Managerの使用](#)
- [10.4 Oracle Connection Managerの起動および停止](#)
- [10.5 Oracle Connection Managerの機能強化](#)
- [11 共有サーバー・アーキテクチャの構成](#)
 - [11.1 ディスパッチャについて](#)
 - [11.1.1 ディスパッチャ単位のグループ化](#)
 - [11.1.2 ディスパッチャの監視](#)
 - [11.2 セッション多重化の有効化](#)
 - [11.3 共有サーバーと専用サーバーの両方を使用する環境でのクライアントの構成](#)
- [12 プロファイルの構成](#)
 - [12.1 プロファイル構成の概要](#)
 - [12.2 インストール中のプロファイルの構成](#)
 - [12.3 名前解決のクライアント属性の理解](#)
 - [12.3.1 クライアントのデフォルト・ドメインについて](#)
 - [12.3.1.1 デフォルト・ドメインの指定](#)
 - [12.3.2 ネーミング・メソッドの優先順位](#)
 - [12.3.3 プロセスへの接続要求のルーティング](#)
 - [12.4 データベース・アクセス制御の設定](#)
 - [12.4.1 データベース・アクセス制御の構成](#)
 - [12.5 プロファイルの詳細情報について](#)
 - [12.5.1 Oracle Net Servicesを使用したsqlnet.oraファイルでの拡張機能の設定](#)

- [12.6 外部ネーミング・メソッドの構成](#)
- [12.7 Oracleネットワーク・セキュリティの構成](#)
- [13 Oracle Net Servicesの拡張機能の使用](#)
 - [13.1 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](#)
 - [13.1.1 リスナー・プロトコル・アドレスのリスト作成](#)
 - [13.1.2 アドレス・リスト・パラメータについて](#)
 - [13.1.2.1 アドレス・リスト・パラメータの構成](#)
 - [13.1.3 拡張接続データ・パラメータについて](#)
 - [13.2 接続ロード・バランシングの理解](#)
 - [13.2.1 共有サーバーの構成のための接続ロード・バランシングの例](#)
 - [13.2.2 専用サーバーの構成のための接続ロード・バランシングの例](#)
 - [13.2.3 クライアント接続のCOLOCATION_TAG](#)
 - [13.3 透過的アプリケーション・フェイルオーバーの構成](#)
 - [13.3.1 透過的アプリケーション・フェイルオーバーについて](#)
 - [13.3.2 透過的アプリケーション・フェイルオーバーのリストア対象](#)
 - [13.3.3 FAILOVER_MODEパラメータについて](#)
 - [13.3.4 透過的アプリケーション・フェイルオーバーの実装](#)
 - [13.3.4.1 接続時フェイルオーバーおよびクライアント・ロード・バランシングのTAF](#)
 - [13.3.4.2 接続再試行のTAF](#)
 - [13.3.4.3 接続事前確立のTAF](#)
 - [13.3.5 透過的アプリケーション・フェイルオーバーの確認](#)
 - [13.4 プライマリおよびセカンダリ・インスタンス構成のインスタンス・ロールの指定](#)
 - [13.5 静的サービス登録の構成](#)
 - [13.5.1 リスナーの静的サービス情報の構成](#)
 - [13.6 サード・パーティのデータベース・サービスへの接続の構成](#)
 - [13.6.1 外部プロシージャのデフォルト構成](#)
 - [13.6.1.1 外部プロシージャを使用する場合のOracle Net Servicesの構成](#)
 - [13.6.1.1.1 外部プロシージャのデフォルト構成の変更](#)
 - [13.6.1.1.2 外部プロシージャを実行するための新規リスナーの作成](#)
 - [13.6.2 Oracle異機種間サービスのOracle Net Servicesについて](#)
 - [13.6.2.1 エージェントに接続するためのOracle Databaseの構成](#)
 - [13.6.3 Oracle Rdbデータベースを使用する場合のOracle Net Servicesの構成](#)
- [14 パフォーマンスの最適化](#)
 - [14.1 ネットワーク・データの圧縮化の利点の理解](#)
 - [14.2 セッション・データ・ユニットの構成](#)
 - [14.2.1 データベースのSDUサイズの設定](#)
 - [14.2.2 クライアントのSDUサイズの設定](#)
 - [14.3 帯域幅遅延積の求め方](#)
 - [14.4 I/Oバッファ・スペースの構成](#)
 - [14.4.1 サーバー上のI/Oバッファ・サイズの構成](#)
 - [14.4.1.1 共有サーバー・プロセスのバッファ・サイズ・パラメータの設定](#)
 - [14.4.2 クライアント上のI/Oバッファ・スペースの構成](#)
 - [14.5 InfiniBand接続をサポートするSDPの構成](#)

- [14.5.1 SDP利用の前提](#)
 - [14.5.2 サーバー上のSDPの構成](#)
 - [14.5.3 クライアント上のSDPの構成](#)
- [14.6 InfiniBand接続をサポートするExadirectの構成](#)
 - [14.6.1 Exadirectを使用するための前提条件](#)
 - [14.6.2 サーバー上のExadirectの構成](#)
 - [14.6.3 クライアント上のExadirectの構成](#)
- [14.7 権限のないユーザーによるリソース使用の制限](#)
- [14.8 クライアント/サーバー接続のためのキーベース・ルーティングの構成](#)
 - [14.8.1 キーベースのルーティングについて](#)
 - [14.8.2 接続文字列でのシャーディング・キーの指定](#)
- [第III部 Oracle Net Servicesのテストおよびトラブルシューティング](#)
 - [15 接続のテスト](#)
 - [15.1 ネットワークのテスト](#)
 - [15.2 TNSPINGユーティリティを使用したクライアントからの接続テスト](#)
 - [15.3 TRCROUTEユーティリティを使用したクライアントからの接続テスト](#)
 - [16 Oracle Net Servicesのトラブルシューティング](#)
 - [16.1 自動診断リポジトリの理解](#)
 - [16.1.1 ADRCI: ADRコマンド・インタプリタ](#)
 - [16.2 Oracle Net Servicesの診断](#)
 - [16.2.1 サーバーの問題の診断](#)
 - [16.2.2 クライアントの問題の診断](#)
 - [16.3 Oracle Net Servicesの最も一般的なエラー・メッセージの解決](#)
 - [16.3.1 ORA-03113: 通信チャンネルでend-of-fileが検出されました](#)
 - [16.3.2 ORA-12154: TNS: 指定された接続識別子を解決できませんでした](#)
 - [16.3.2.1 ローカル・ネーミング・メソッドの使用時のORA-12154エラー](#)
 - [16.3.2.2 ディレクトリ・ネーミング・メソッドの使用時のORA-12154エラー](#)
 - [16.3.2.3 簡易接続ネーミング・メソッドの使用時のORA-12154エラー](#)
 - [16.3.2.4 外部ネーミング・メソッドの使用時のORA-12154エラー](#)
 - [16.3.3 ORA-12170: TNS: 接続タイムアウトが発生しました](#)
 - [16.3.4 TNS-12500またはORA-12500: TNS: リスナーが専用サーバー・プロセスの起動に失敗しました](#)
 - [16.3.5 ORA-12514: TNS: リスナーは現在、接続記述子でリクエストされているサービスを認識していません](#)
 - [16.3.6 ORA-12520: TNS: リスナーは、リクエストしたサーバー・タイプに使用可能なハンドラを検出できませんでした](#)
 - [16.3.7 ORA-12521: TNS: リスナーは接続記述子でリクエストされたインスタンスを現在認識していません](#)
 - [16.3.8 ORA-12525: TNS: リスナーは、クライアントのリクエストを許容時間内に受信しませんでした](#)
 - [16.3.9 ORA-12533: TNS: ADDRESSパラメータが不正です](#)
 - [16.3.10 TNS-12540またはORA-12540: TNS:内部制限を超えましたおよびTNS-00510: TNS:内部制限を超えました](#)

- [16.3.11 TNS-12541またはORA-12541: TNS: リスナーがありません](#)
- [16.3.12 TNS-12549またはORA-12549: TNS:オペレーティング・システム・リソース割当て制限を超えましたおよびTNS-00519: TNS:オペレーティング・システム・リソース割当て制限を超えました](#)
- [16.3.13 TNS-12560またはORA-12560: TNS: プロトコル・アダプタ・エラーが発生しました](#)
- [16.3.14 ディレクトリ・ネーミング・エラー](#)
- [16.4 Oracle Net Servicesのトラブルシューティングのヒント](#)
 - [16.4.1 Oracle Net Servicesのトラブルシューティングのためのチェック項目](#)
- [16.5 TNS-12154エラーのトラブルシューティングの例](#)
- [16.6 Oracle Net Servicesのエラー情報のロギング](#)
 - [16.6.1 Oracle Netエラー・スタック](#)
 - [16.6.1.1 エラー・スタック・メッセージの理解](#)
 - [16.6.2 Oracle Net Servicesのログ・ファイル名](#)
 - [16.6.3 Oracle Networkログ・ファイルのセグメンテーション](#)
 - [16.6.4 ロギング・パラメータについて](#)
 - [16.6.4.1 sqlnet.oraのログ・パラメータ](#)
 - [16.6.4.2 listener.oraのログ・パラメータ](#)
 - [16.6.4.3 cman.oraのログ・パラメータ](#)
 - [16.6.5 構成ファイルのロギング・パラメータの設定](#)
 - [16.6.5.1 Oracle Net Managerを使用したsqlnet.oraファイルのパラメータの設定](#)
 - [16.6.5.2 Oracle Enterprise Manager Cloud Controlを使用したlistener.oraファイルのパラメータの設定](#)
 - [16.6.5.3 Oracle Net Managerを使用したlistener.oraファイルのパラメータの設定](#)
 - [16.6.6 制御ユーティリティ実行時のロギング設定](#)
 - [16.6.7 ログ・ファイルの使用](#)
 - [16.6.8 リスナー・ログの分析](#)
 - [16.6.8.1 リスナー・ログ監査証跡](#)
 - [16.6.8.1.1 リスナー・ログ監査証跡情報](#)
 - [16.6.8.1.2 リスナーのログ監査証跡の書式](#)
 - [16.6.8.2 リスナーのサービス登録イベント](#)
 - [16.6.8.2.1 リスナーのサービス登録イベント情報](#)
 - [16.6.8.2.2 リスナーのサービス登録情報の書式](#)
 - [16.6.8.3 リスナーのダイレクト・ハンドオフ情報](#)
 - [16.6.8.4 ONSノード停止イベント情報のリスナー・サブスクリプション](#)
 - [16.6.8.5 リスナーのOracle Clusterware通知情報](#)
 - [16.6.9 Oracle Connection Managerログの分析](#)
- [16.7 Oracle Net Servicesのエラー情報のトレース](#)
 - [16.7.1 Oracle Net Servicesのトレース・ファイル名の理解](#)
 - [16.7.2 トレース・パラメータの設定](#)
 - [16.7.2.1 cman.oraのトレース・パラメータ](#)

- [16.7.2.2 listener.oraのトレース・パラメータ](#)
 - [16.7.2.3 sqlnet.oraのトレース・パラメータ](#)
 - [16.7.2.4 構成ファイルのトレース・パラメータの設定](#)
 - [16.7.2.4.1 Oracle Net Managerを使用したsqlnet.oraファイルのトレース・パラメータの設定](#)
 - [16.7.2.4.2 Oracle Enterprise Manager Cloud Controlを使用したリスナーのトレース・パラメータの設定](#)
 - [16.7.2.4.3 Oracle Net Managerを使用したリスナーのトレース・パラメータの設定](#)
 - [16.7.3 制御ユーティリティ実行時のトレース設定](#)
 - [16.7.4 Oracle Net Servicesのトレース・ファイルの評価](#)
 - [16.7.4.1 ネットワーク・ノード間のデータ・パケットの流れ](#)
 - [16.7.4.2 Oracle Netのデータ・パケットのフォーマット](#)
 - [16.7.4.3 Oracle Netの関連トレース・エラー出力](#)
 - [16.7.5 トレース・アシスタントを使用したトレース・ファイルの検証](#)
 - [16.7.5.1 トレース・アシスタントの構文](#)
 - [16.7.5.2 パケットに関する出力例](#)
 - [16.7.5.3 Two-Task Common \(TTC\)パケットの例](#)
 - [16.7.5.4 接続に関する出力例](#)
 - [16.7.5.5 統計に関する出力例](#)
 - [16.8 Oracleサポート・サービスへの連絡](#)
- [用語集](#)
 - [索引](#)

『Oracle Database Net Services管理者ガイド』のこのリリースでの変更点

Oracle Database 19cのOracle Database Net Services管理者ガイドの変更内容を確認してください。

- [新機能](#)

Oracle Database 19cで使用可能な新機能および拡張機能は次のとおりです。

- [非推奨となった機能](#)

次の機能は、このリリースでは非推奨であり、将来のリリースではサポートされなくなる可能性があります。

新機能

Oracle Database 19cで使用可能な新機能および拡張機能は次のとおりです。

Identity and Access Managementと追加のOracle Database環境との統合

Oracle Databaseリリース19.16で使用可能であり、Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM)ユーザーが追加のOracle Database環境にログインできます。

サポートされているOracle Database環境のリストは、[Oracle Databaseセキュリティ・ガイド](#)を参照してください。

IAMユーザー名およびIAMデータベース・パスワードを使用してデータベース・トークンを取得する機能

IAMユーザー名とIAMデータベース・パスワードまたはセキュアな外部パスワード・ストア(SEPS)を使用したIAMデータベース・トークンの取得のほうが、データベース・アクセスのパスワード検証方法を使用するよりも安全です。このトークンをOCI IAMエンドポイントから直接リクエストするようにデータベース・クライアントを構成できます。

新しいsqlnet.oraまたはtnsnames.oraパラメータを使用すると、この認証方法を構成し、追加のメタデータとともにIAMエンドポイントを指定できます。これらのパラメータは、PASSWORD_AUTH、OCI_IAM_URL、OCI_TENANCYと、オプションのOCI_COMPARTMENTおよびOCI_DATABASEです。

[データベースへの接続](#)および[Oracle Databaseセキュリティ・ガイド](#)を参照してください。

Microsoft Azure Active Directoryと追加のOracle Database環境との統合

Oracle Databaseリリース19.16で使用可能であり、Microsoft Azure Active Directory (Azure AD)ユーザーがAzure AD OAuth2アクセス・トークンを使用して追加のOracle Database環境にログインできます。

サポートされているOracle Database環境のリストは、[Oracle Databaseセキュリティ・ガイド](#)を参照してください。

Azure ADとOracle Autonomous Cloud Databaseの統合

2022年6月にOracle Autonomous Databaseで使用可能となり、Azure ADユーザーがAzure AD OAuth2アクセス・トークンを使用してOracle Cloud Infrastructure (OCI) Autonomous Database (共有インフラストラクチャ)にログインできます。

OCI Oracle Autonomous Databaseは、Azure AD OAuth2トークンを受け入れてデータベースにアクセスできるように

なりました。Azure ADユーザーはAzure ADトークンを使用してデータベースに直接アクセスでき、アプリケーションはサービス・トークンを使用してデータベースにアクセスできます。

[データベースへの接続](#)および[Oracle Databaseセキュリティ・ガイド](#)を参照してください。

IAMとOracle Autonomous Cloud Databaseの統合

Oracle Databaseリリース19.13で使用可能であり、IAMユーザーがデータベース・パスワードまたはトークンベース認証のいずれかを使用してOracle Autonomous Databaseにログインできます。

IAM ADMINユーザーは、IAMユーザーおよびIAMグループの認証と認可の両方を構成できます。IAMユーザーは、SQL*PlusやSQLclなどのツールを使用して、Oracle Autonomous Cloud Databasesにログインできます。

[データベースへの接続](#)および[Oracle Databaseセキュリティ・ガイド](#)を参照してください。

一方向Transport Layer Security (TLS)

この機能を使用すると、一方向TLS (サーバー認証)を構成できます。この方法では、認証局(CA)によって発行された証明書を提示することで、データベース・サーバーのみがクライアントに対して認証を実行し、クライアントはデータベース・サーバー証明書が有効かどうかを検証します。

ローカル・システムのデフォルト証明書ストアにすでにインストールされている信頼できる共通ルート証明書によってデータベース・サーバー証明書が署名されている場合、サーバー証明書を含むOracleクライアント・ウォレットは必要ありません。

[TLSプロトコル付きTCP/IPについて](#)を参照してください。

SQL*Net: バンド外ブレイクのサポートの自動検出

この機能は、バンド外サポートのステータスを判別するためにクライアントとサーバー間のネットワーク・パスを自動的に調査し、自動的に有効化または無効化します。

バンド外ブレイクは、過去のリリースではUNIXプラットフォームに対してデフォルトで有効になっていました。ただし、この構成では、クライアントとサーバー間のパス上のネットワーク・デバイスがバンド外のデータの通過を許可しない場合に多数の問題が発生します。このデータは、削除またはインライン化され、TNS (Transparent Network Substrate)エラー、データ破損などのサーバー側の問題につながる可能性があります。多くの場合、こうした問題は診断が非常に困難です。解決策は、sqlnet.oraパラメータを設定することで、手動でバンド外データの使用を無効にすることです。

[プロファイルの詳細情報について](#)を参照してください。

Oracle Networkログ・ファイルのセグメンテーション

この機能を使用すると、Oracle Net Listener、Connection Manager (CMAN)、グローバル・サービス・マネージャなどのOracle Networkコンポーネントのテキスト・ログ・ファイルの最大サイズおよび数を構成できます。[Oracle Networkログ・ファイルのセグメンテーション](#)を参照してください。

データベースのクライアント/サーバー通信でのWebsocketのサポート

安全なWebソケット接続の確立は、HTTPSで動作し、HTTPSプロキシおよび仲介プロキシをサポートするように設計されます。

データベース・クライアント接続は、安全なWebsocketプロトコルをサポートしています。このプロトコルは、最小限のプロトコル・オーバーヘッドを備えたデータベースへのネイティブ接続を提供します。

[Websocketプロトコルについて](#)を参照してください。

簡易接続プラス

アプリケーションがOracle Databaseへの接続に使用する簡易接続プラス構文で、機能が向上しました。新しいバージョンは簡易接続プラスと呼ばれます。

簡易接続プラスでは、Oracle Databaseのアプリケーション構成と一般的なユースケースのデプロイメントが簡略化されます。簡易接続プラスでは、tnsnames.oraやsqlnet.oraなどのOracle Netパラメータ・ファイルを構成する必要がなくなりました。簡易接続プラスでは、TNS_ADMIN環境変数を設定する必要がなくなりました。

[簡易接続プラス](#)に関する項を参照してください

関連項目

- [プロファイルの詳細情報について](#)
- [Oracle Networkログ・ファイルのセグメンテーション](#)
- [Websocketプロトコルについて](#)
- [簡易接続プラスについて](#)

親トピック: [『Oracle Database Net Services管理者ガイド』のこのリリースでの変更点](#)

非推奨となった機能

次の機能は、このリリースでは非推奨であり、将来のリリースではサポートされなくなる可能性があります。

SERVICE_NAMES初期化パラメータの非推奨

Oracle Database 19c以降、SERVICE_NAMESパラメータをお客様が使用することは非推奨になりました。今後のリリースでサポートが終了する可能性があります。

SERVICE_NAMESパラメータの使用は、積極的にはサポートされなくなりました。高可用性(HA)デプロイメントでは使用しないでください。HA操作ではサービス名パラメータの使用はサポートされません。この制限には、FAN、ロード・バランシング、FAILOVER_TYPE、FAILOVER_RESTORE、SESSION_STATE_CONSISTENCYなどの用途が含まれます。

サービスを管理するには、SRVCTLまたはGDSCCTLコマンドライン・ユーティリティ、またはDBMS_SERVICEパッケージを使用することをお勧めします。

ノート:



非推奨の SERVICE_NAMES パラメータは、Oracle Net 接続文字列の SERVICE_NAME パラメータとは異なります。SERVICE_NAME パラメータは引き続き有効です。

脆弱なネイティブ・ネットワーク暗号化アルゴリズムおよび整合性アルゴリズムの非推奨

DES、DES40、3DES112、3DES168、RC4_40、RC4_56、RC4_128、RC4_256およびMD5アルゴリズムは、このリリースでは非推奨で

す。

この非推奨化に伴い、ネットワーク暗号化および整合性の構成を確認して、非推奨になった脆弱なアルゴリズムのいずれかを使用するように指定されているかどうかを確認することをお勧めします。

より強力なアルゴリズムを使用するようにOracle Database環境を移行するには、My Oracle Supportノート[2118136.2](https://support.oracle.com/epos1?id=2118136.2)で説明されているパッチをダウンロードしてインストールします。

関連項目

- [『Oracle Databaseセキュリティ・ガイド』](#)

親トピック: [『Oracle Database Net Services管理者ガイド』のこのリリースでの変更点](#)

第I部 Oracle Net Servicesの理解

第I部では、Oracle Net Servicesの概念、製品およびツールの概要を説明します。

この部は、次の章で構成されています。

- [Oracle Net Servicesの概要](#)
Oracle Net ServicesアーキテクチャおよびOracle Net Foundationレイヤーの基本要素を理解します。
- [データベースの識別とアクセス](#)
- [ネットワーク・アドレス情報の管理](#)
- [通信レイヤーの理解](#)
- [Oracle Netアーキテクチャの理解](#)

1 Oracle Net Servicesの概要

Oracle Net ServicesアーキテクチャおよびOracle Net Foundationレイヤーの基本要素について説明します。

- [Oracle Net Servicesについて](#)
Oracle Net Servicesによって、ネットワークの構成および管理の複雑さが軽減され、パフォーマンスが最大化され、ネットワーク診断機能が改善されます。
- [データベース・インスタンスの理解](#)
- [Oracle Net Servicesのコンポーネント](#)

親トピック: [Oracle Net Servicesの理解](#)

1.1 Oracle Net Servicesについて

[Oracle Net Services](#)は、異機種間分散コンピューティング環境にある企業全体への接続ソリューションを提供します。

Oracle Net Servicesは、複雑なネットワーク構成や管理を簡略化し、パフォーマンスを最大化して、ネットワーク診断機能を向上させます。

ノート:



用語の「SQL*Net」と「Net Services」は、Oracle ドキュメント全体を通じて同じ意味で使用され、どちらの用語も同じ機能を指します。

この項では、代表的なネットワーク構成に関係する基本的なネットワーキングの概念を説明します。

- [接続の理解](#)
- [管理性の理解](#)
- [共有サーバー・アーキテクチャの理解](#)
- [パフォーマンスの理解](#)
- [ネットワーク・セキュリティの理解](#)

親トピック: [Oracle Net Servicesの概要](#)

1.1.1 接続の理解

Oracle Net Servicesのコンポーネントの1つである[Oracle Net](#)では、クライアント・アプリケーションからOracle Databaseサーバーへのネットワーク・セッションを可能にします。一度ネットワーク・セッションが確立されると、Oracle Netは、クライアント・アプリケーションおよびデータベースのデータ送信手段として機能します。クライアント・アプリケーションとデータベース間でのメッセージの交換に加え、これらの間の接続を確立および維持します。このようなジョブの実行は、Oracle Netがネットワーク上の各コンピュータに配置されることにより可能になります。

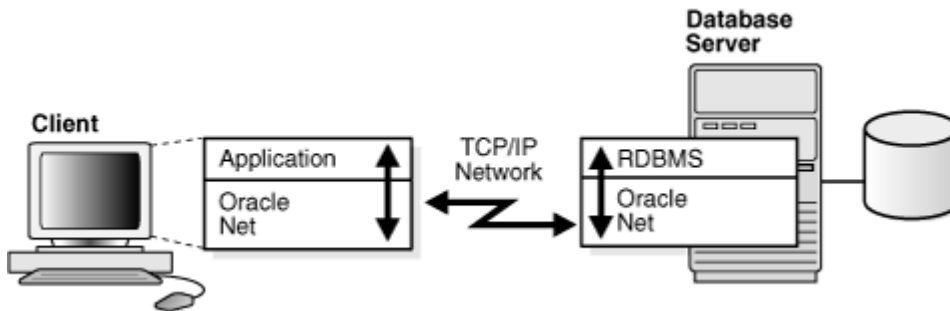
- [クライアント/サーバー・アプリケーション接続について](#)
- [Webクライアント・アプリケーション接続について](#)

親トピック: [Oracle Net Servicesについて](#)

1.1.1.1 クライアント/サーバー・アプリケーション接続について

Oracle Netを使用すると、従来のクライアント/サーバー・アプリケーションからOracle Databaseサーバーへの接続が可能になります。[図1-1](#)は、Oracle Netがクライアントとデータベース・サーバー間でネットワーク接続を可能にする方法を示しています。Oracle Netは、クライアントおよびデータベース・サーバーの両方に存在するソフトウェア・コンポーネントです。Oracle Netはネットワーク[Oracle protocol support](#)の最上位レイヤーになります。つまり、アプリケーションがネットワークにアクセスする方法およびネットワーク上でデータを転送するためにパケットに分割する方法を決定するルールです。[図1-1](#)では、Oracle Netは[TCP/IP](#)と通信するので、コンピュータ・レベルの接続およびクライアントとデータベース間のデータ転送が可能になります。

図1-1 クライアント/サーバー・アプリケーション接続



特に、Oracle Netは接続の確立と維持を行う[Oracle Net Foundationレイヤー](#)と、そのテクノロジーを業界標準のプロトコルにマップする[Oracle protocol support](#)で構成されています。

- [Javaクライアント・アプリケーション接続](#)

親トピック: [接続の理解](#)

1.1.1.1.1 Javaクライアント・アプリケーション接続

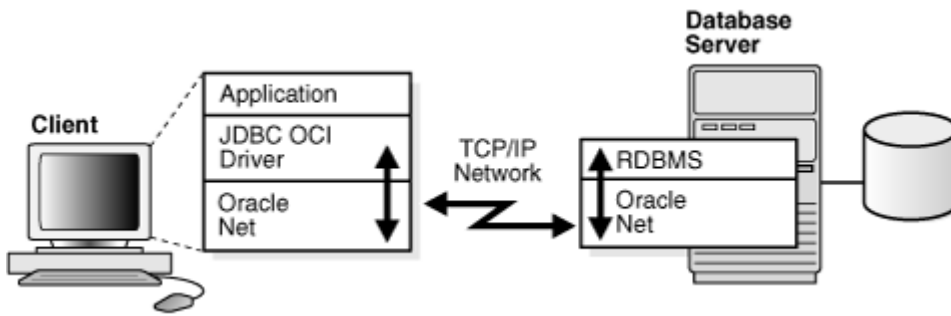
Javaクライアント・アプリケーションは、Oracleデータベースに、Javaからリレーショナル・データベースへの接続を提供する標準のJavaインタフェースである [Java Database Connectivity \(JDBC\)ドライバ](#)を介してアクセスします。オラクル社は次のドライバを提供します。

- [JDBC OCIドライバ](#)。Oracleクライアント・インストールとともにクライアント側で使用します。
- [JDBC Thinドライバ](#)。Oracleインストールを必要とせず、主にアプレットとともにクライアント側で使用するPure Javaドライバ。

これらのドライバはOracle Netを使用するため、クライアント・アプリケーションとOracleデータベース間の接続を確立できます。

[図1-2](#)は、JDBC OCIドライバおよびOracle Databaseサーバーを使用するJavaクライアント・アプリケーションを示しています。JavaクライアントはJDBC OCIドライバへのコールを作成します。これにより、JDBCコールはOracle Netレイヤーに直接変換されます。次にクライアントは、Oracle Netで構成されたOracle Databaseとの通信にOracle Netを使用します。

図1-2 Javaアプリケーション接続



関連項目:

- [『Oracle Database JDBC開発者ガイド』](#)
- [Oracle Database JDBC Java APIリファレンス](#)

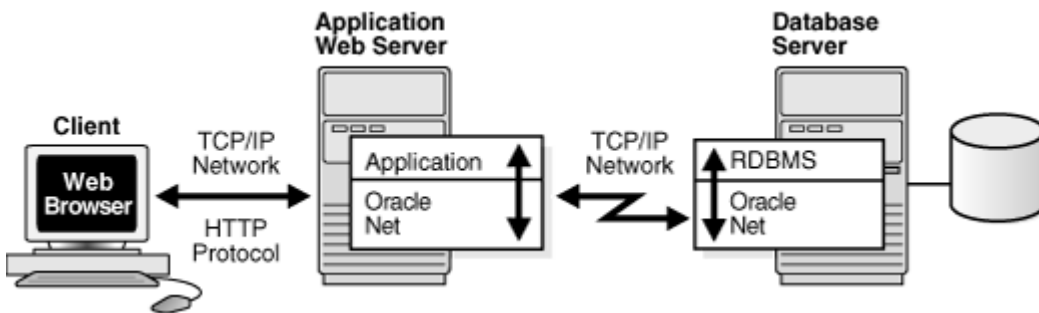
親トピック: [クライアント/サーバー・アプリケーション接続について](#)

1.1.1.2 Webクライアント・アプリケーション接続について

クライアントのWebブラウザからOracle Databaseサーバーへのインターネット接続は、接続要求がアプリケーションWebサーバーに送られる点を除いて、クライアント/サーバー・アプリケーションへの接続と同じです。

図1-3では、クライアントWebブラウザ、アプリケーションWebサーバー、Oracle Databaseサーバーを含むWebクライアント接続の基本アーキテクチャを示します。クライアント上のブラウザは、[HTTP](#)を使用してWebサーバーと通信し、接続要求を作成します。Webサーバーは、処理を行うアプリケーションに要求を送信します。次にアプリケーションは、Oracle Netで構成されたOracle Databaseサーバーとの通信にOracle Netを使用します。

図1-3 アプリケーションWebサーバーからのWebクライアント接続



基本コンポーネントの特長は次のとおりです。

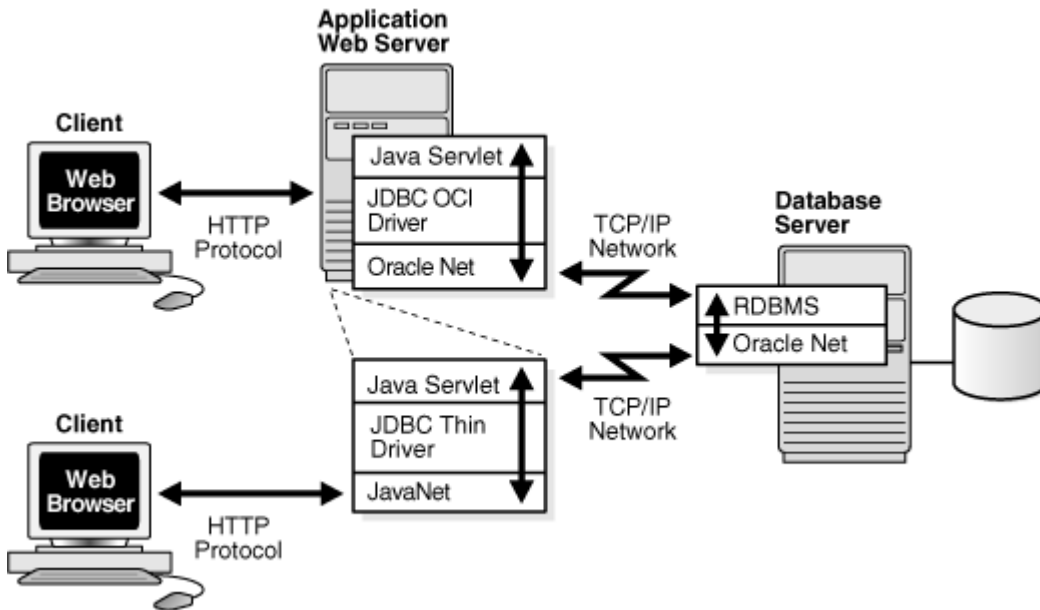
- **HyperText Transfer Protocol(HTTP)**
HTTPは、WebブラウザとアプリケーションWebサーバーの通信を可能にする言語を提供します。
- **アプリケーションWebサーバー**
アプリケーションWebサーバーは、Webサイトのデータの管理とアクセス制御、Webブラウザからの要求への応答を実行します。Webサーバー上のアプリケーションは、データベースと通信し、Webサーバーから要求されたジョブを実行します。
- [JavaアプリケーションWebサーバーからのWebクライアント接続](#)
- [アプリケーションWebサーバーを使用しないWebクライアント接続](#)

親トピック: [接続の理解](#)

1.1.1.2.1 JavaアプリケーションWebサーバーからのWebクライアント接続

図1-4で示すように、アプリケーションWebサーバーは、Javaアプリケーションおよびサーブレットをホストできます。Webブラウザは、HTTPを介してアプリケーションWebサーバーに通信することによって、接続要求を作成します。アプリケーションWebサーバーは要求をアプリケーションまたはサーブレットに送信し、アプリケーションまたはサーブレットは、JDBC OCIまたはJDBC Thinドライバを使用して要求を処理します。次にドライバは、Oracle Netで構成されたOracle Databaseサーバーとの通信にOracle Netを使用します。

図1-4 JavaアプリケーションWebサーバーからのWebクライアント接続



親トピック: [Webクライアント・アプリケーション接続について](#)

1.1.1.2.2 アプリケーションWebサーバーを使用しないWebクライアント接続

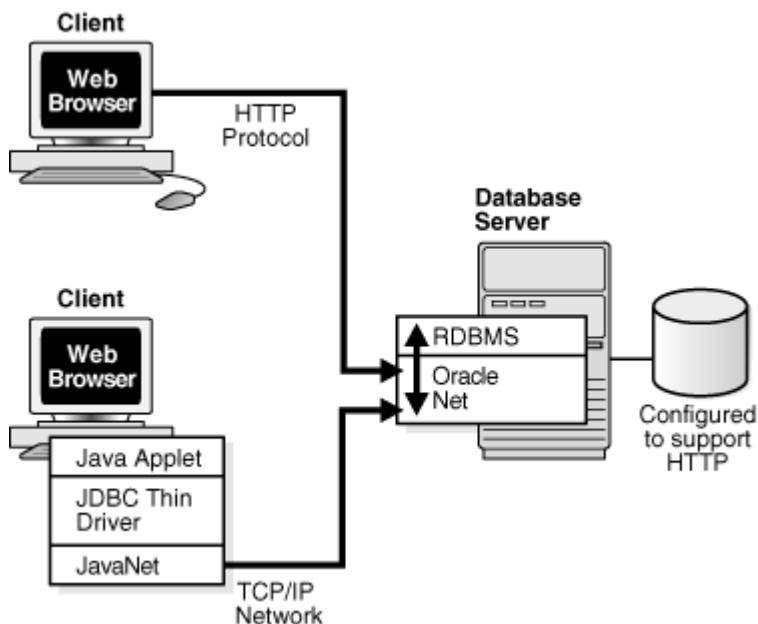
アプリケーションにアクセスするためにアプリケーションWebサーバーを必要としないWebクライアントは、Oracle Databaseに直接アクセスできます。たとえば、Javaアプレットを使用します。通常の接続に加えて、データベースはHTTPプロトコル、[FTP](#)または[WebDAVプロトコル](#)接続を受け入れるように構成できます。これらのプロトコルは、Oracle Databaseインスタンスの[Oracle XML DB](#)への接続に使用します。

図1-5では、2つの異なるWebクライアントを示します。最初のWebクライアントはデータベースへのHTTP接続を作成します。2番目のWebクライアントは、WebブラウザとJDBC Thinドライバを使用し、次にこのドライバは、JavaNetと呼ばれるJava版のOracle Netを使用して、Oracle Netで構成されたOracle Databaseサーバーと通信します。

関連項目:

『[Oracle XML DB開発者ガイド](#)』

図1-5 Webクライアントの接続のシナリオ



親トピック: [Webクライアント・アプリケーション接続について](#)

1.1.2 管理性の理解

Oracle Net Servicesは、ネットワーキング・コンポーネントを構成および管理する複数の管理機能を提供します。

- [位置の透過性について](#)
- [集中構成と管理について](#)
- [クイック・インストールおよび構成について](#)

親トピック: [Oracle Net Servicesについて](#)

1.1.2.1 位置の透過性について

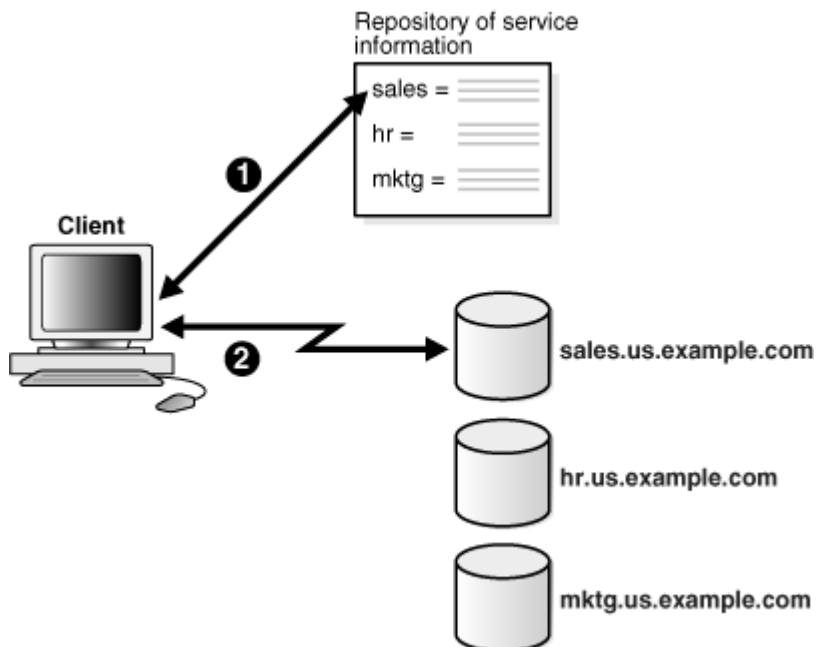
各データベースは、1つ以上のサービスで表されます。[サービス](#)は、[サービス名](#)(sales.us.example.comなど)で識別されます。クライアントはサービス名を使用して、自身がアクセスする必要があるデータベースを識別します。データベース・サービスおよびネットワーク上の位置についての情報は、クライアントに対して透過的です。これは、接続に必要な情報がリポジトリに保存されているためです。

リポジトリは、1つ以上の[ネーミング・メソッド](#)によって表現されます。ネーミング・メソッドとは、データベース・サービスに接続するとき、クライアント・アプリケーションが接続識別子を接続記述子に変換するために使用する解決メソッドです。Oracle Net Servicesは、いくつかの種類のネーミング・メソッドを提供しており、これらは、各クライアント上のローカル構成またはネットワーク上のすべてのクライアントがアクセスできる集中化された構成をサポートしています。

たとえば、[図1-6](#)では、企業にクライアントがアクセスできるデータベースが3つあります。各データベースには、sales.us.example.com、hr.us.example.comおよびmktg.us.example.comなどの個別のサービス名があります。

1. クライアントはリポジトリを使用して、sales.us.example.comに必要な情報を探します。
2. クライアントは必要な情報を得ると、データベースに接続します。

図1-6 サービス情報リポジトリ



親トピック: [管理性の理解](#)

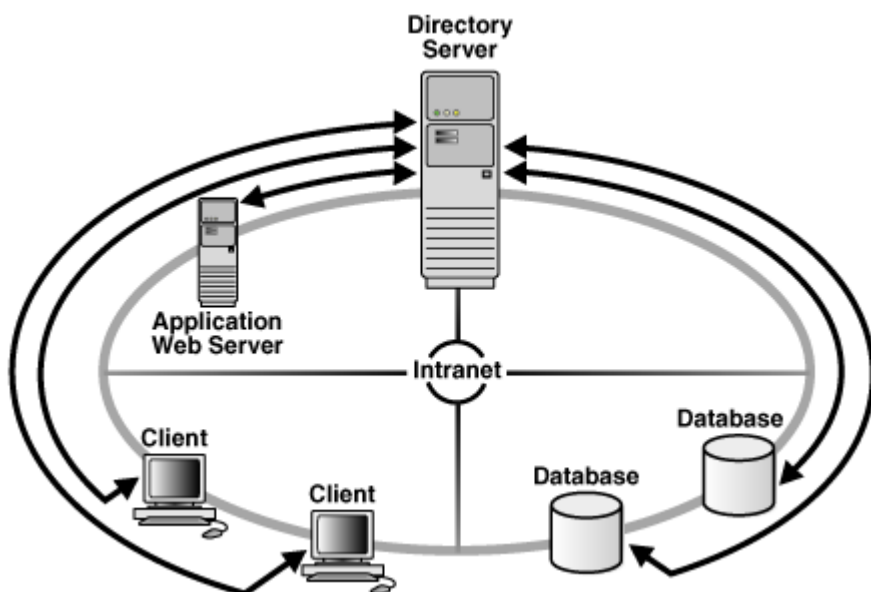
1.1.2.2 集中構成と管理について

大規模なネットワーキング環境を管理するために、管理者は集中化されたリポジトリにアクセスして、容易にネットワーク構成の指定や変更をできる必要があります。このため、Oracle Net Servicesの構成は、LDAP準拠ディレクトリ・サーバーに保存できます。

LDAP準拠のディレクトリ・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のための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構成ファイルに置き換えられます。

ネットワーク上のすべてのコンピュータは、ディレクトリで情報を参照できます。[図1-7](#)では、集中化されたディレクトリ・サーバーに接続するクライアント、Oracle Databaseサーバー、およびその他のサーバー(アプリケーションWebサーバーなど)を示します。

図1-7 ディレクトリ・サーバーを伴うネットワーク構成の集中ストレージ



関連項目:

ディレクトリ・サーバーの概念の詳細は、[「集中管理用のディレクトリ・サーバーの使用」](#)を参照してください

親トピック: [管理性の理解](#)

1.1.2.3 クイック・インストールおよび構成について

ほとんどの環境に対応するように、Oracle Databaseサーバーとクライアントのネットワーク要素が事前構成されています。簡易接続ネーミング・メソッドはデフォルトで使用可能になり、リポジトリを必要としません。クライアントは、データベースのホスト名を使用して接続します。結果として、クライアントとサーバーは簡易接続を使用して即座に接続できる状態にあるため、ユーザーは分散環境の利点を活用できます。

親トピック: [管理性の理解](#)

1.1.3 共有サーバー・アーキテクチャの理解

Oracle Databaseの[共有サーバー](#)・アーキテクチャにより、アプリケーションの拡張性とデータベースへ同時に接続できるクライアント数は増大します。また共有サーバー・アーキテクチャでは、既存のアプリケーションの能力を、アプリケーション自体に何ら変更を加えずにスケールアップできます。

共有サーバーを使用すると、クライアントはデータベース・[サーバー・プロセス](#)と直接通信しません(サーバー・プロセスは、データベース上の1つのプロセスで、データベースにかわってクライアントの要求を処理します)。そのかわり、クライアントの要求は1つ以上の[ディスパッチャ](#)にルーティングされます。ディスパッチャはクライアントの要求を共通のキューに登録します。サーバー・プロセスの共有プールにあるアイドル状態の[共有サーバー](#)は、このキューから要求を取り出して処理します。つまり、サーバー・プロセスの小規模プールによる多数のクライアントの処理が可能になります。

[図1-8](#)および[図1-9](#)は、共有サーバー接続モデルと従来の[専用サーバー](#)接続モデルの基本的な違いを示しています。共有サーバー・モデルでは、ディスパッチャは同時に複数のクライアント接続をサポートします。専用サーバー・モデルでは、各クライアントに対してサーバー・プロセスは1つです。接続要求が受け取られるたびに、サーバー・プロセスが起動され、処理が完了するまでその接続の専用になります。このモデルでは、処理の遅延が起きます。

図1-8 共有サーバー・アーキテクチャ

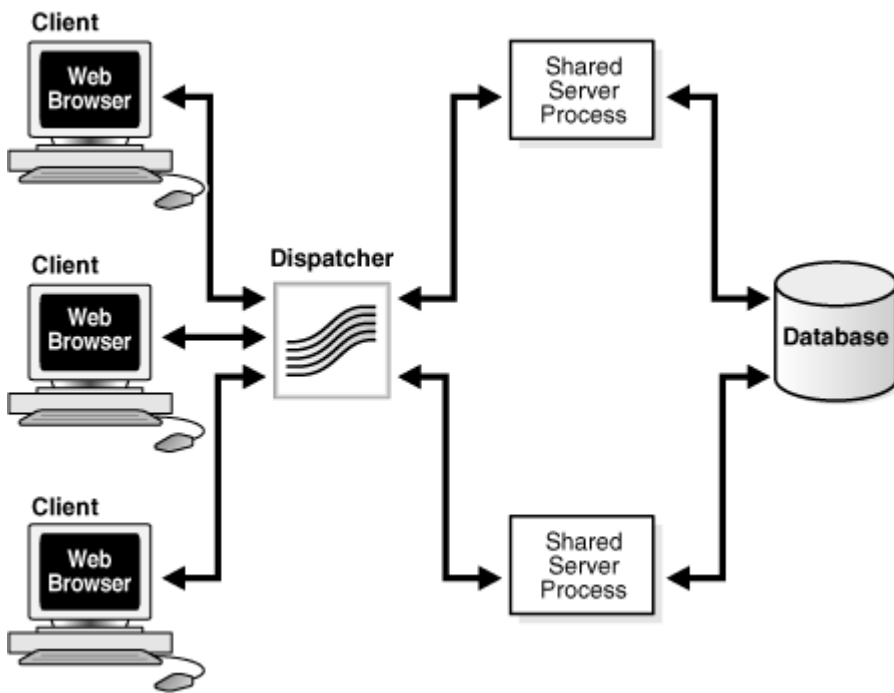
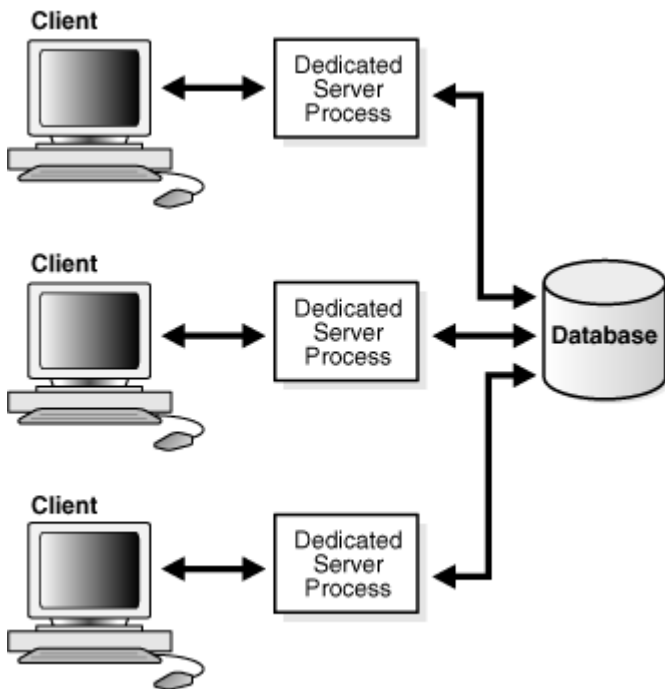


図1-9 専用サーバー・アーキテクチャ



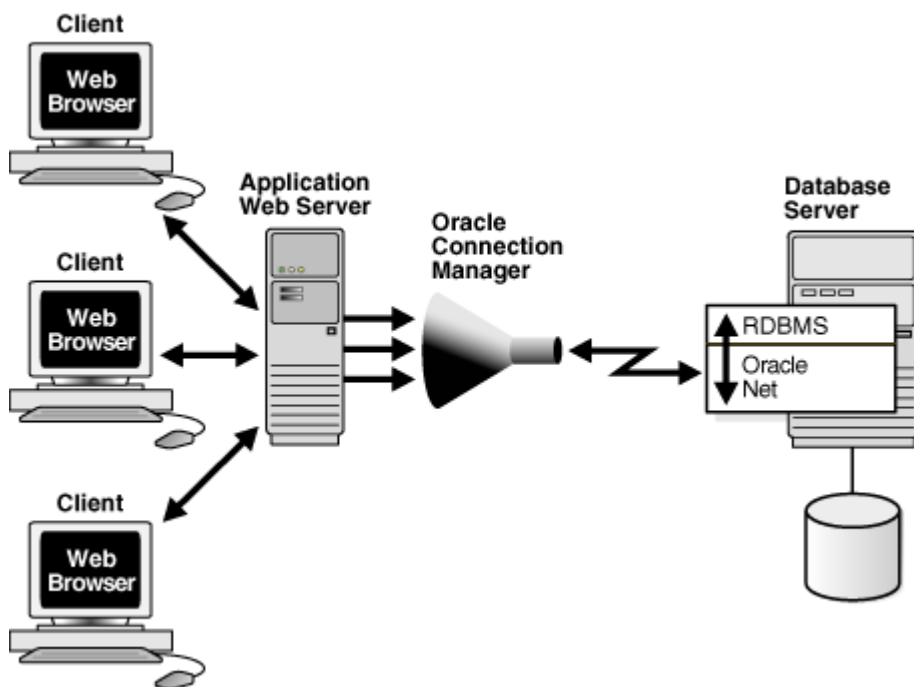
共有サーバーは、接続が多い場合に理想的な構成です。これは、この構成によってサーバーのメモリー要件が軽減されるためです。共有サーバーは、インターネットおよびイントラネットの両方の環境に適しています。

サーバー・リソースの使用率は、Oracle Connection Managerによって、さらに高めることができます。Oracle Net Servicesのコンポーネントである[Oracle Connection Manager](#)では、データベースへのネットワーク接続を単一化することにより、複数のクライアント・ネットワーク・セッションを多重化、つまり集中化できます。

[セッションの多重化](#)機能は、サーバーが着信要求に使用するネットワーク接続のエンドポイント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。これにより、サーバーが処理できるネットワーク・セッションの総数が増加します。1つのOracle Connection Managerを複数のゲートウェイで構成すると、数千のユーザーをサーバーに同時接続できます。

図1-10では、セッションの多重化のWebアーキテクチャにおける使用方法を示します。Oracle Connection ManagerがアプリケーションWebサーバーと同じコンピュータ上で実行されると、アプリケーションWebサーバーは、複数のクライアント・セッションをOracle Connection Managerからルーティングでき、これらのセッションは、継続的にOracle Databaseサーバーにアクセスできます。この機能は特に、セッションの可用性と応答時間が主要課題であるWebアプリケーションで利用価値があります。

図1-10 セッションの多重化



セッションの多重化の長所と短所は、次のとおりです。セッションの多重化は、継続的な接続が必要なネットワークに推奨されません。

セッション多重化の長所

- 各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。
- 多数のクライアントがサポートされます。
- 一定数に制限されたプロセス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。
- リソースの利用を最適化します。
- 実ユーザーの識別および監視が可能になります。
- 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。
- 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。
- 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。

セッション多重化の短所

クライアントはOracle Connection Managerに接続する必要があります。

親トピック: [Oracle Net Servicesについて](#)

1.1.4 パフォーマンスの理解

ユーザーにとってシステム・パフォーマンスは重要です。通常、システムの応答時間が1秒を超えると、ユーザーはパフォーマンスが気にかかるようになります。Oracle Netの構成を変更してシステム・パフォーマンスの向上を図ることができます。

この項では、パフォーマンスに関する考慮事項について説明します。

- [リスナー・キュー・サイズ](#)
- [セッション・データ・ユニットのサイズによるデータ転送の最適化](#)
- [TCP/IPの固定バッファ・フラッシング](#)
- [Sockets Direct Protocol](#)
- [データベースの可用性](#)

親トピック: [Oracle Net Servicesについて](#)

1.1.4.1 リスナー・キュー・サイズ

TCP/IPを介してリスニング・プロセス(リスナー、Oracle Connection Manager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、Oracle Net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。

親トピック: [パフォーマンスの理解](#)

1.1.4.2 セッション・データ・ユニットのサイズによるデータ転送の最適化

ネットワーク上にデータを送信する前にOracle Netはデータをバッファリングして、[セッション・データ・ユニット\(SDU\)](#)にカプセル化します。バッファが一杯になりフラッシュされるか、データベース・サーバーがデータを読み出そうとするとOracle Netはこのバッファに格納されているデータを送信します。大量のデータを転送する場合やメッセージのサイズが一貫している場合は、SDUバッファのサイズを調整することによって、パフォーマンス、ネットワークの使用率、メモリー消費を改善できます。SDUはクライアント、アプリケーションWebサーバーおよびデータベースにデプロイできます。

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も効果的な方法です。これを行った後、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のサイズを変更する際の考慮事項

SDUのサイズは、次の場合に変更します。

- サーバーから戻されるデータが個別のパケットに分かれる場合。
- 遅延の起こる広域ネットワーク(WAN)上にいる場合。
- パケット・サイズが一定している場合。
- 戻されるデータ量が多い場合。

次の場合は、SDUのサイズを変更しないでください。

- 隣の欄に記載されている遅延を回避するためにアプリケーションをチューニングできる場合。
- 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。
- 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。

ノート:



Oracle Database 11g 以降、Oracle Net Services では、バルク・データ転送が Oracle SecureFiles LOB や Oracle Data Guard REDO トランスポート・サービスなどのコンポーネント用に最適化されています。ネットワーク・パラメータ・ファイルで指定されている SDU サイズ制限は、これらのバルク・データ転送には適用されません。バルク・データ転送最適化は、ASO オプションが有効な場合や TLS トランスポートが使用されている場合には適用されません。

関連項目:

[「セッション・データ・ユニットの構成」](#)

親トピック: [パフォーマンスの理解](#)

1.1.4.3 TCP/IPの固定バッファ・フラッシング

TCP/IPを使用するアプリケーションがいくつかある状況では、Oracle Netのパケットがネットワークにすぐに送出されません。この動作は大量のデータを送出するときに頻繁に実行されます。TCP/IPの実装自体がフラッシュの欠如が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します。

関連項目:

TCP.NODELAYパラメータの詳細は、[『Oracle Database Net Servicesリファレンス』](#)を参照してください。

親トピック: [パフォーマンスの理解](#)

1.1.4.4 Sockets Direct Protocol

Oracle Net Servicesでは、InfiniBand高速ネットワークをサポートします。InfiniBandは、高帯域幅のI/Oアーキテクチャで、CPU、サーバー側のデバイスおよびネットワーク・サブシステム間の通信速度を上げるように設計されています。Oracle Net Servicesは、Sockets Direct Protocol ([SDP](#))をサポートしています。SDPは、InfiniBandネットワーク・ピア間の使用を目的とした業界標準のワイア・プロトコルです。

SDPを使用すると、データの中間的なレプリケーションが除去され、メッセージ交換の負荷がCPUからネットワーク・ハードウェアへ移動することにより、TCP/IPのオーバーヘッドが削減されます。その結果、[待機時間](#)の短縮、帯域幅の拡大、接続のスループットの向上につながり、ネットワーク処理に占有されるCPUサイクルが削減されます。

Oracle WebLogic Serverやその他のサード・パーティの中間層クライアントを含めて、クライアント間の通信、およびOracle Database 12cは、高速相互接続の恩恵を受けています。Oracle WebLogic Serverのインストールの一部としてOracle TCP/IPのサポートが含まれています。

Oracle WebLogic Serverサーバーにインストールされたドライバが、TCP/IPサポートをSDPサポートに透過的に変換します。次に、SDP要求はInfiniBandスイッチに送信され、このスイッチによって、要求が処理され、Oracle WebLogic Serverサー

バーからデータベース・サーバーに転送されます。

関連項目:

[「InfiniBand接続をサポートするSDPの構成」](#)

親トピック: [パフォーマンスの理解](#)

1.1.4.5 データベースの可用性

データベースへの可用性を実現することはネットワークでは重大な問題です。複数のリスナーを同じデータベース・サービスへのクライアント接続要求が処理できるように構成できます。これは各インスタンスにリスナーが対応付けられているOracle Real Application Clusters構成で有利な構成です。複数のリスナー構成では、次の機能を利用できます。

- 接続時フェイルオーバーではクライアントは最初のリスナーが失敗した場合に別のリスナー(通常は異なるノード上の)に要求を送ることができます。
- クライアント・ロード・バランシングではクライアントは複数のリスナー(通常は異なるノード上)に要求をランダム化できます。これらの機能は一緒に、または別々に使用できます。一緒に使用するとデータベースへのアクセスの確実性が高められ単一のリスナーが過負荷にならないように負荷を分散できます。

親トピック: [パフォーマンスの理解](#)

1.1.5 ネットワーク・セキュリティの理解

データ・アクセスとデータの安全な転送は、Oracle Databaseをデプロイする際の重要な考慮事項です。データベースへのアクセスの付与や制限は、安全なネットワーク環境を構築するために重要です。Oracle Net Servicesでは、ファイアウォール・アクセス制御と有効なノード登録を使用してデータベース・アクセス制御を実現しています。

- [ファイアウォール・アクセス制御](#)

関連項目:

有効なノード登録の詳細は、[「Oracle Net Listenerのセキュリティの管理」](#)を参照してください

親トピック: [Oracle Net Servicesについて](#)

1.1.5.1 ファイアウォール・アクセス制御

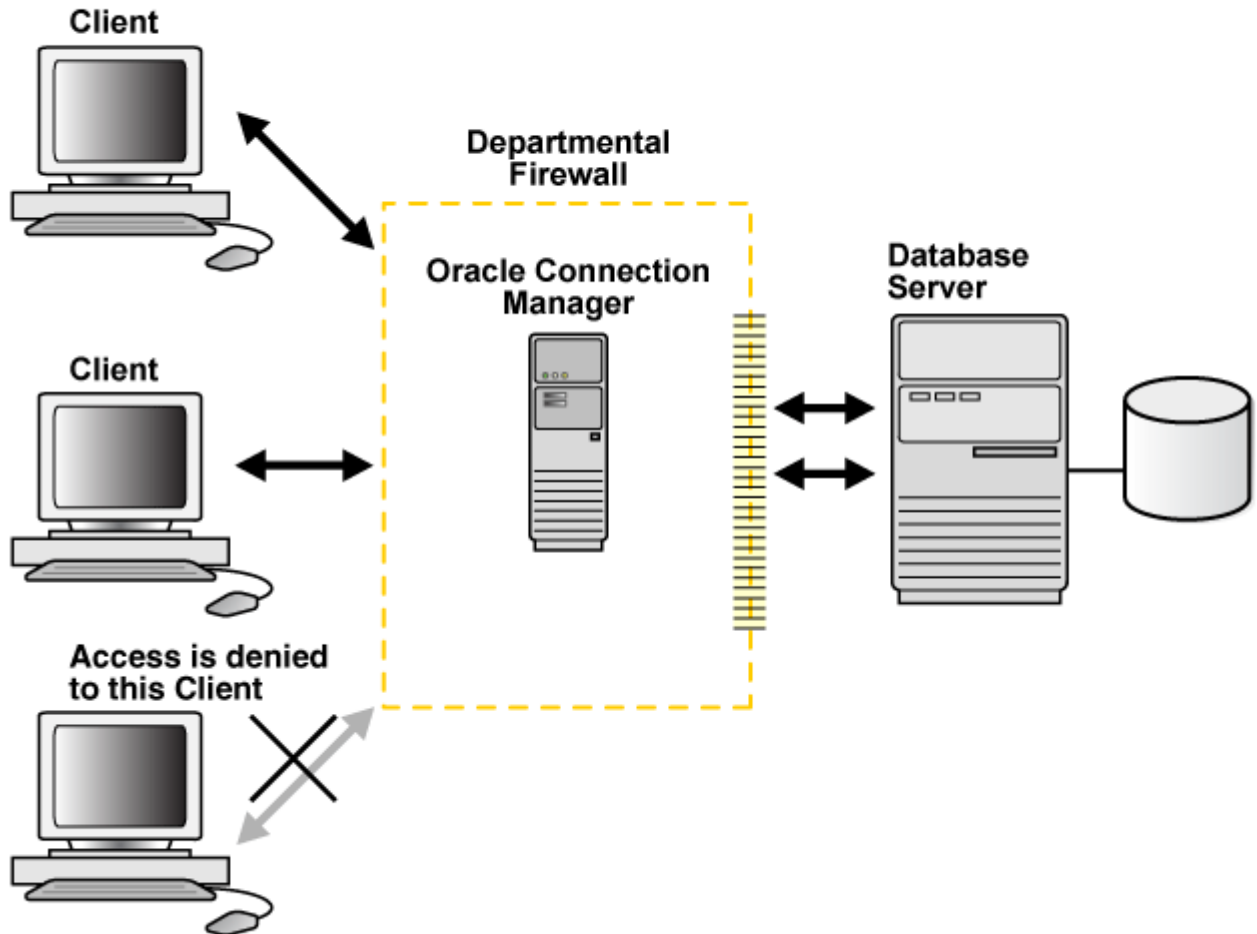
Oracle Connection Managerは、特定のデータベース・サービスやコンピュータに対するクライアントのアクセス権限を付与したり、制限するように構成できます。フィルタリング・ルールを指定すれば、次の基準に基づいて、サーバーへの特定のクライアントのアクセスを可能にしたり、制限したりできます。

- ソース・ホスト名またはクライアントのIPアドレス
- 接続先ホスト名またはサーバーのIPアドレス
- 接続先データベース・サービス名

- Oracle Net Servicesのセキュリティ機能のクライアントでの使用

図1-11では、3つのクライアントとOracle Databaseサーバーの間に位置するOracle Connection Managerを示します。最初の2つのクライアントにはアクセスを許可し、3番目のクライアントのアクセスは制限できるようにOracle Connection Managerを構成します。

図1-11 Oracle Connection Managerによるイントラネット・ネットワークのアクセス制御



Oracle Connection Managerは、サード・パーティのファイアウォール製品と統合できませんが、1つのアプリケーション・ゲートウェイとして機能するようにベンダー独自の製品とパッケージ化できます。

一般にファイアウォールは、着信要求を受け取るように、およびOracle Databaseからの発信コールを許可するように設定する必要があります。フィルタリング・ルールを定義することにより、ネットワークへのアクセスを制限できます。

警告:

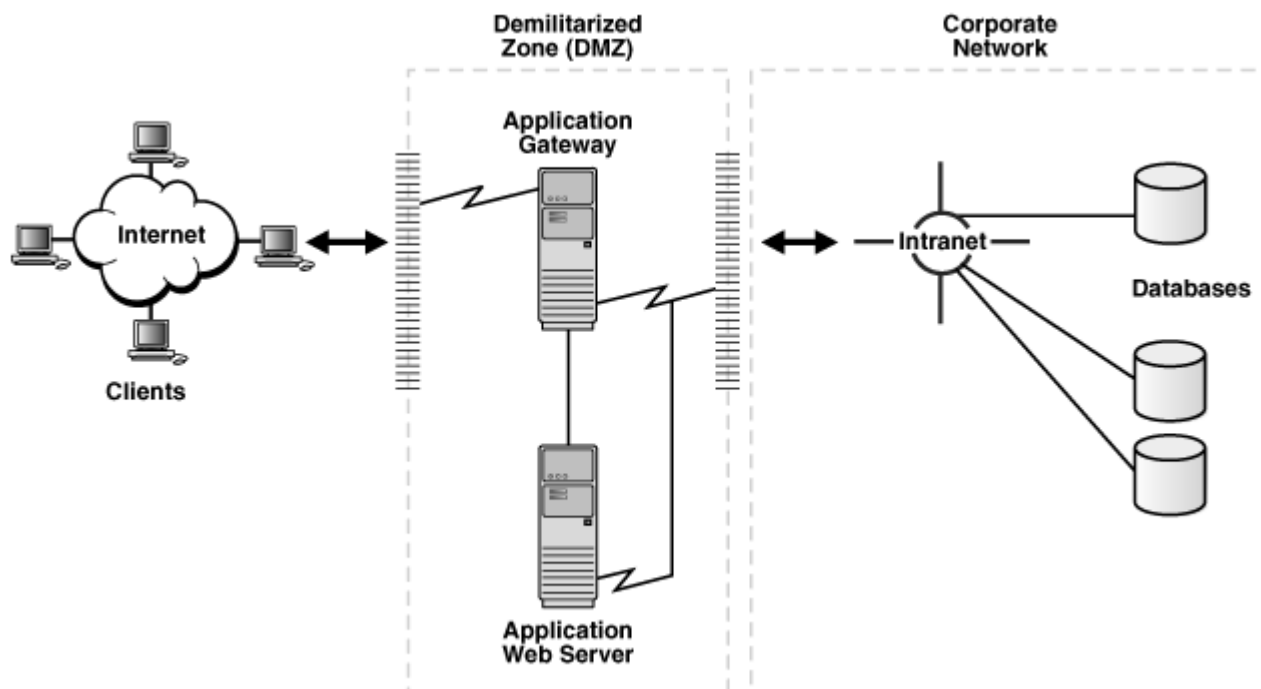


ファイアウォール・オプションの設定が正しくない場合、セキュリティの問題が発生する可能性があります。ファイアウォールの設定を変更する前に、オプションおよびネットワーク・サイトのポリシーについてシステム管理者に確認して

ください。

図1-12は、社内と社外のネットワーク間の通信量を制御し、アクセス制御および監査用に単一のチェックポイントを提供するアプリケーション・ゲートウェイを示しています。結果として、権限のないインターネット・ホストは社内のデータベースに直接アクセスできませんが、権限のあるユーザーは企業ネットワークの外部のインターネット・サービスを使用できます。この機能は、機密データへのリモート・アクセスを制限するために、インターネット環境では重要です。

図1-12 アプリケーション・ゲートウェイによるインターネット・ネットワークのアクセス制御



1つのファイアウォールが停止した場合に備えて、Oracle Connection ManagerファイアウォールまたはOracle Netファイアウォール・プロキシをインターネット・ネットワーク環境に少なくとも2つデプロイすることが重要です。

親トピック: [ネットワーク・セキュリティの理解](#)

1.2 データベース・インスタンスの理解

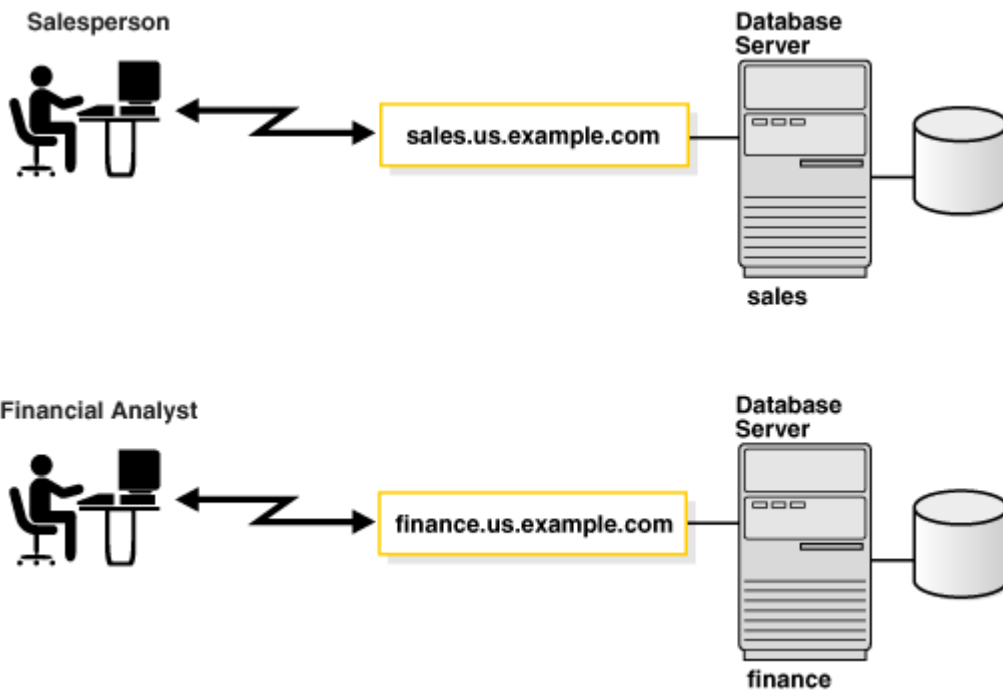
データベースには1つ以上の**インスタンス**があります。インスタンスは、[システム・グローバル領域\(SGA\)](#)と呼ばれるメモリ領域とOracleバックグラウンド・プロセスからなります。インスタンスのメモリおよびプロセスは、関連付けられたデータベースのデータを効率よく管理し、データベース・ユーザーに提供します。

ノート:

インスタンスは、[Oracle XML DB](#) など、他のサービスも管理します。

図2-1では、salesとfinanceの2つのデータベース・インスタンスを示します。各インスタンスは、それぞれデータベースとサービス名に対応付けられています。

図1-13 各データベースに1つのインスタンス

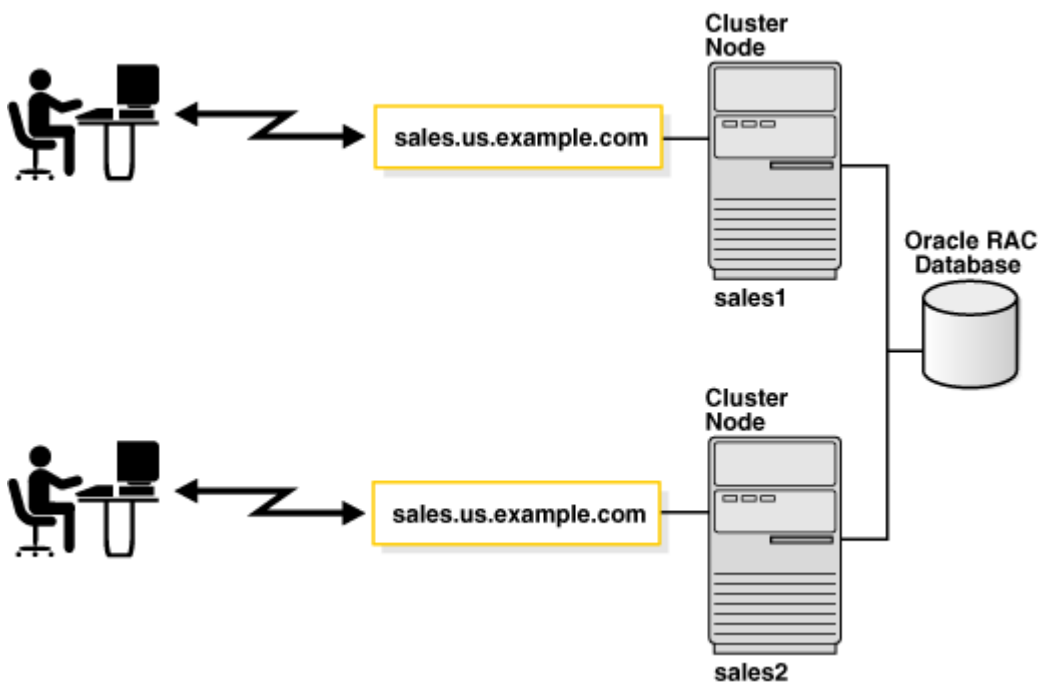


インスタンスは**インスタンス名**で識別されます(この例では、salesやfinanceなどです)。インスタンス名は、INSTANCE_NAME初期化パラメータで指定されます。インスタンス名のデフォルトは、データベース・インスタンスの[Oracleシステム識別子\(SID\)](#)です。

一部のハードウェア・アーキテクチャでは、複数のコンピュータがデータ、ソフトウェアまたは周辺装置へのアクセスを共有できます。[Oracle Real Application Clusters \(Oracle RAC\)](#)では、単一の物理データベースを共有する異なるコンピュータ上で複数のインスタンスを実行することで、このようなアーキテクチャを活用できます。

[図2-2](#)は、Oracle RAC構成を示しています。この例では、2つのインスタンス、sales1とsales2が1つのデータベース・サービス、sales.us.example.comに対応付けられています。

図1-14 1つのOracle RACデータベースへの複数インスタンスの対応付け



親トピック: [Oracle Net Servicesの概要](#)

1.3 Oracle Net Servicesのコンポーネント

この項では、接続性、管理性、スケーラビリティおよびセキュリティの機能について説明します。

- [Oracle Netについて](#)
- [Oracle Net Listenerについて](#)
- [Oracle Connection Managerについて](#)
- [ネットワーキング・ツールについて](#)
- [Oracle Advanced Securityについて](#)

親トピック: [Oracle Net Servicesの概要](#)

1.3.1 Oracle Netについて

Oracle Netは、クライアントおよびOracle Databaseサーバー上に存在するソフトウェア・レイヤードです。クライアント・アプリケーションとサーバー間でのメッセージの交換に加え、業界標準のプロトコルを使用して、これらの間の接続を確立および維持します。Oracle Netには、次の2つのソフトウェア・コンポーネントがあります。

- [Oracle Net Foundationレイヤー](#)
- [Oracle Protocol Support](#)
Oracle Net Foundationレイヤーでは、Oracle protocol supportを使用して、これらの業界標準のネットワーク・プロトコルと通信します。

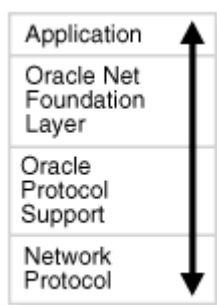
親トピック: [Oracle Net Servicesのコンポーネント](#)

1.3.1.1 Oracle Net Foundationレイヤー

クライアント側でアプリケーションは、Oracle Net Foundationレイヤーと通信して接続を確立し、それを維持します。Oracle Net Foundationレイヤーは、TCP/IPなどの業界標準のネットワーク・プロトコルで通信できるOracle protocol supportを使用して、Oracle Databaseサーバーと通信します。

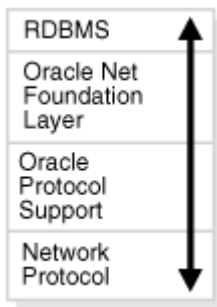
[図1-15](#)では、クライアントの通信スタックを示します。

図1-15 クライアント上のOracle Net



[図1-16](#)で示すように、Oracle Databaseサーバー側は、クライアント側と同じです。ネットワーク・プロトコルによってクライアントの要求情報がOracle protocol supportレイヤーに送信されてから、Oracle Net Foundationレイヤーに送信されます。次に、Oracle Net FoundationレイヤーはOracle Databaseサーバーと通信して、クライアントの要求を処理します。

図1-16 サーバー上のOracle Net



親トピック: [Oracle Netについて](#)

1.3.1.2 Oracle Protocol Support

Oracle Net Foundationレイヤーでは、Oracle protocol supportを使用して、これらの業界標準のネットワーク・プロトコルと通信します。

- TCP/IP(バージョン4およびバージョン6)
- Transport Layer Security (TLS)付きTCP/IP
- 名前付きパイプ
- SDP

Oracle protocol supportは、Oracle Net Foundationレイヤーの機能をクライアント/サーバー接続で使用する業界標準のプロトコルにマップします。

関連項目

- [Oracle protocol supportレイヤーの理解](#)

親トピック: [Oracle Netについて](#)

1.3.2 Oracle Net Listenerについて

Oracle Databaseサーバーは[Oracle Net Listener](#)を通じて初期接続を受け取ります。Oracle Net Listener(このマニュアルでは[リスナー](#)と呼びます)は、クライアント要求を受け取ってサーバーに渡します。リスナーはプロトコル・アドレスで構成されており、同じプロトコル・アドレスで構成されたクライアントは、そのリスナーに接続要求を送信できます。接続が確立されると、クライアントとOracleサーバーは互いに直接通信します。

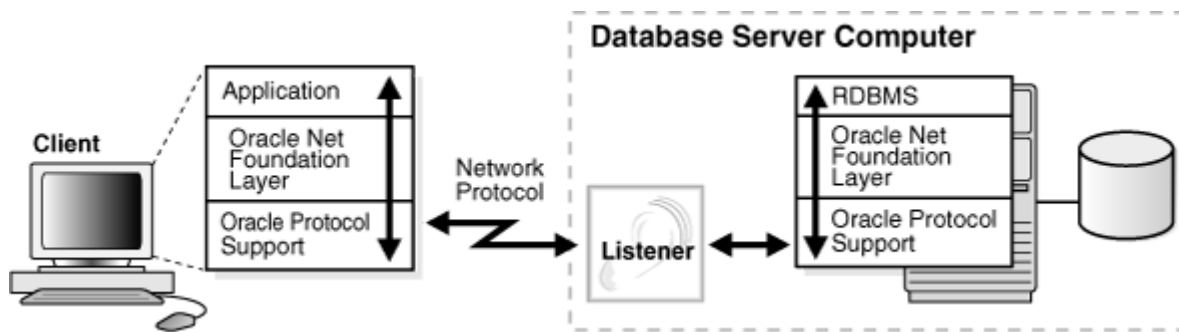
Oracle Net Listenerは、サービスに対応するACL (アクセス制御リスト)をサポートしています。これは、すべてのIPプロトコルでサポートされます。

関連項目:

リスナーのACLの詳細は、『Oracle Database PL/SQLパッケージおよびタイプ・リファレンス』の[DBSFUSER.DBMS_SFW_ACL_ADMINに関する項](#)を参照してください。

[図1-17](#)では、クライアントから接続要求を受け取り、Oracleサーバーにその要求を転送するリスナーを示します。

図1-17 接続要求時のリスナー



関連項目:

リスナーの詳細は、[「Oracle Net Listenerの構成と管理」](#)を参照してください

親トピック: [Oracle Net Servicesのコンポーネント](#)

1.3.3 Oracle Connection Managerについて

Oracle Connection Managerは、クライアント・サーバーやOracle Databaseサーバーとは別に、コンピュータに常駐するソフトウェア・コンポーネントです。これにより、データベース・サーバーに対するリクエストがプロキシ化されて選別されます。さらに、データベース・セッションの多重化も行われます。

Oracle Connection Managerは、セッションの多重化によって、単一のトランスポート・プロトコル接続から特定の接続先に複数のセッションを集中化させます。これにより、Oracle Connection ManagerではOracle Databaseサーバーが着信要求に使用する接続エンドポイントの数を少なくできるため、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が削減されます。

Oracle Connection Managerは、アクセス制御フィルタとしてOracle Databaseへのアクセスを制御します。

ノート:

Oracle Connection Manager を Traffic Director モードの Connection Manager として機能させるには、`cman.ora` で `tdm=yes` を設定します。



Traffic Director モードの Oracle Connection Manager では、高可用性(HA)が改善され(計画済、計画外を問わず)、接続の多重化がサポートされ、ロード・バランシングが提供されます。また、この機能には、Oracle Connection Manager やサービスの停止イベントに関する計画済停止について OCI クライアントに伝えるインバンド・クライアント通知メカニズムが用意されています。

関連項目:

- [「共有サーバー・アーキテクチャの理解」](#)
- フィルタリングについては、[「ファイアウォール・アクセス制御」](#)を参照してください

親トピック: [Oracle Net Servicesのコンポーネント](#)

1.3.4 ネットワーキング・ツールについて

Oracle Net Servicesは、ネットワークを構成、管理および監視するユーザー・インタフェースとコマンドライン・ユーティリティを提供します。

- [Oracle Net Configuration Assistant](#)は、リスナーおよびネーミング・メソッドを構成できるスタンドアロン・ツールです。
- [Oracle Enterprise Manager Cloud Control](#)では、複数ファイル・システムでの構成機能とリスナーの管理機能を結合して、Oracle Net Servicesを構成および管理するための統合環境を提供します。
- [Oracle Net Manager](#)では、ローカル・クライアントやサーバー・ホスト上のOracleホームを構成する機能を提供します。
- また、コマンドライン制御ユーティリティによって、リスナーやOracle Connection Managerなどのネットワーク・コンポーネントを構成、管理および監視できます。

Oracle Enterprise Manager Cloud ControlやOracle Net Managerでは、Oracle Net Configuration Assistantで作成したリスナーやネーミング・メソッドの構成を細かく調整できます。さらに、Oracle Enterprise Manager Cloud ControlおよびOracle Net Managerは、組込みウィザードとユーティリティを提供し、これを使用して接続性をテストしたり、データを特定のネーミング・メソッドから別のネーミング・メソッドへ移行したり、さらに追加ネットワーク・コンポーネントを作成することが可能になります。

関連項目:

[Oracle Net Servicesの管理](#)

親トピック: [Oracle Net Servicesのコンポーネント](#)

1.3.5 Oracle Advanced Securityについて

Oracle Advanced Securityは個別にライセンス供与可能な製品で、Oracle Database透過的データ暗号化(TDE)とOracle Data Redactionを提供します。TDEでは、認可された受取人のみが読めるようにデータが暗号化されます。Oracle Data Redactionでは、管理者は次のリダクション・タイプを使用して列データのリダクション(マスク)を行うことができます。

- 完全なリダクション: 列データのすべてのコンテンツをリダクションします。問合せを行ったユーザーに返される、リダクションされた値は、列のデータ型によって異なります。たとえば、NUMBERデータ型の列はゼロ(0)でリダクションされ、文字データ型は空白でリダクションされます。
- 部分的なリダクション: 列データの一部をリダクションします。たとえば、クレジットカードの番号の大半(最後の4桁以外)をアスタリスク(*)でマスクするなどです。
- 正規表現で、リダクションするデータのパターンを使用できます。たとえば、正規表現を使用して、文字の長さが変化する可能性のある電子メール・アドレスをリダクションできます。これは、文字データのみで使用するように設計されています。
- ランダム・リダクション: リダクションされたデータは、値が表示のたびにランダムに生成されて、問合せを行ったユーザーに表示されます。

- リダクションなし: 管理者は、リダクション・ポリシーが定義された表に対する問合せの結果に影響を与えずに、リダクション・ポリシーの内部動作をテストできます。

関連項目:

[『Oracle Database Advanced Securityガイド』](#)

親トピック: [Oracle Net Servicesのコンポーネント](#)

2 データベースの識別とアクセス

データベースの識別方法とクライアントのデータベースへのアクセス方法について説明します。

- [データベース・インスタンスの理解](#)
- [データベース・サービスの理解](#)
- [データベース・サービスへの接続](#)
- [サービス・ハンドラの理解](#)
- [ネーミング・メソッドの理解](#)
- [複数リスナーを使用したサービスのアクセス可能性の拡張](#)

親トピック: [Oracle Net Servicesの理解](#)

2.1 データベース・インスタンスの理解

データベースには1つ以上の**インスタンス**があります。インスタンスは、[システム・グローバル領域\(SGA\)](#)と呼ばれるメモリー領域とOracleバックグラウンド・プロセスからなります。インスタンスのメモリーおよびプロセスは、関連付けられたデータベースのデータを効率よく管理し、データベース・ユーザーに提供します。

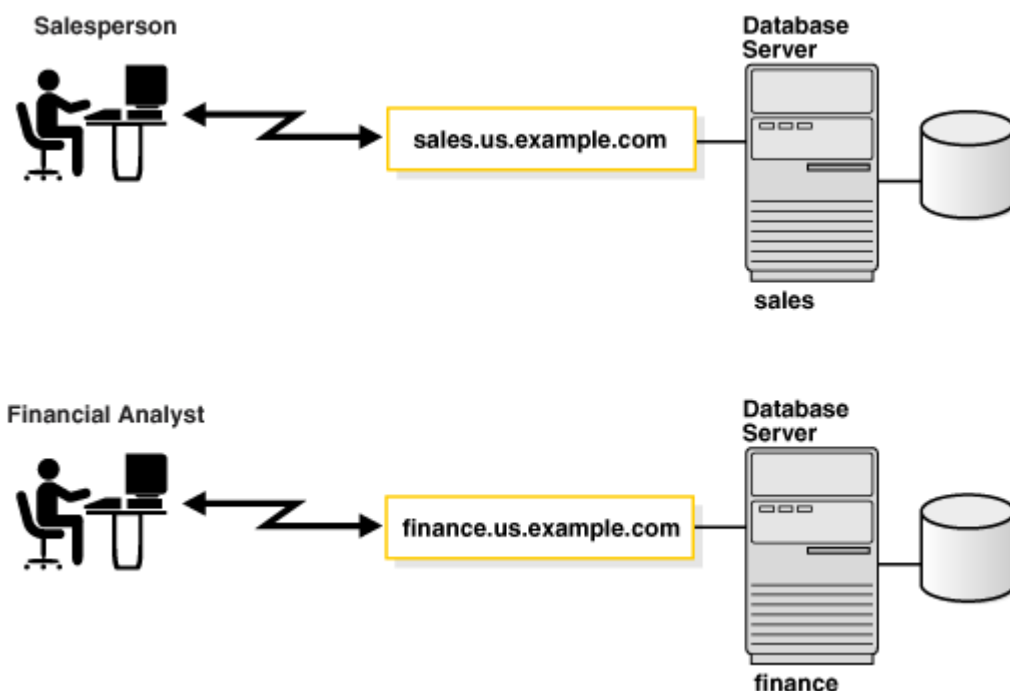
ノート:



インスタンスは、[Oracle XML DB](#) など、他のサービスも管理します。

[図2-1](#)では、salesとfinanceの2つのデータベース・インスタンスを示します。各インスタンスは、それぞれデータベースとサービス名に対応付けられています。

図2-1 各データベースに1つのインスタンス



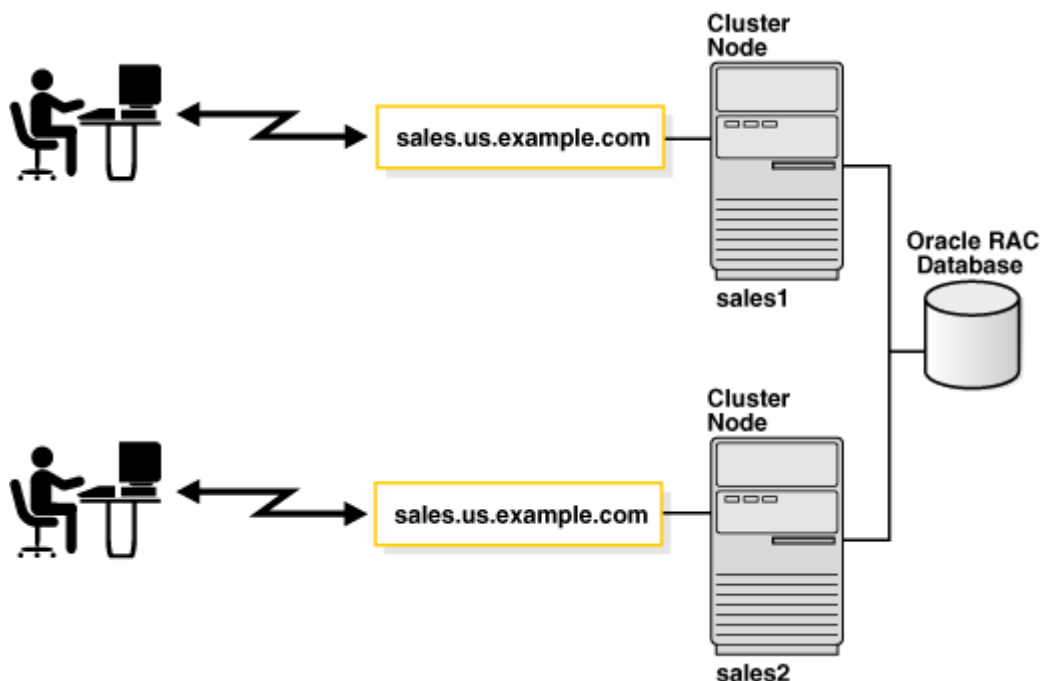
インスタンスは**インスタンス名**で識別されます(この例では、salesやfinanceなどです)。インスタンス名は、

INSTANCE_NAME初期化パラメータで指定されます。インスタンス名のデフォルトは、データベース・インスタンスの[Oracleシステム識別子\(SID\)](#)です。

一部のハードウェア・アーキテクチャでは、複数のコンピュータがデータ、ソフトウェアまたは周辺装置へのアクセスを共有できます。[Oracle Real Application Clusters \(Oracle RAC\)](#)では、単一の物理データベースを共有する異なるコンピュータ上で複数のインスタンスを実行することで、このようなアーキテクチャを活用できます。

[図2-2](#)は、Oracle RAC構成を示しています。この例では、2つのインスタンス、sales1とsales2が1つのデータベース・サービス、sales.us.example.comに対応付けられています。

図2-2 1つのOracle RACデータベースへの複数インスタンスの対応付け



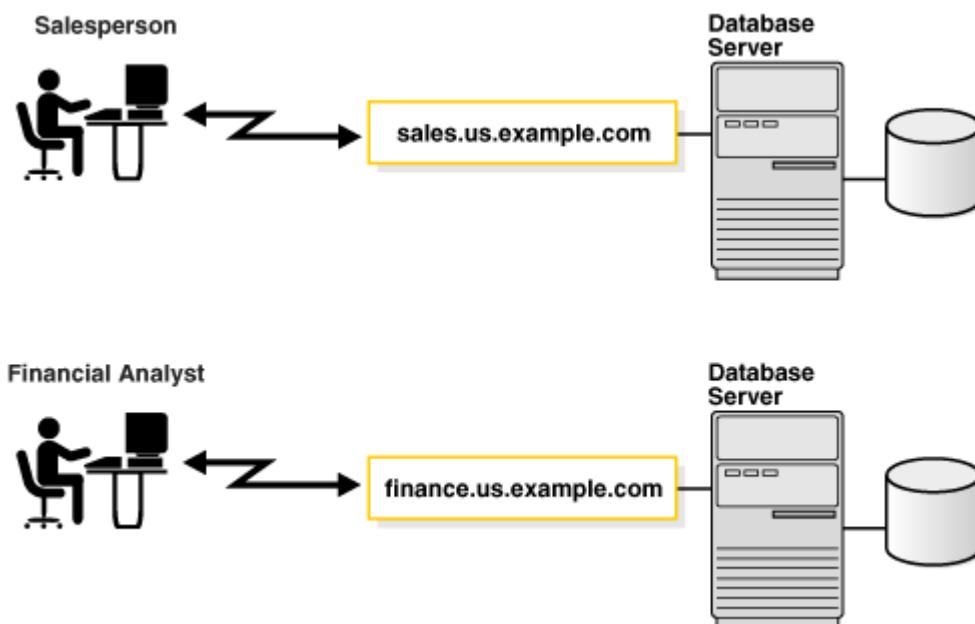
親トピック: [データベースの識別とアクセス](#)

2.2 データベース・サービスの理解

Oracle Databaseは、クライアントに対しては[サービス](#)として表示されます。データベースには、1つ以上のサービスに対応付けることができます。

[図2-3](#)は、2つのデータベースを示しています。各データベースがそれぞれクライアントにデータベース・サービスを提供しています。一方のサービスsales.us.example.comでは、販売担当者が販売データベースにアクセスできます。もう一方のサービスfinance.us.example.comでは、財務アナリストが財務データベースにアクセスできます。

図2-3 各データベースに1つのサービス



販売データベースと財務データベースは、[サービス名](#)、sales.us.example.comおよびfinance.us.example.comによってそれぞれ識別されます。サービス名はデータベースの論理表現です。インスタンスを起動すると、インスタンスはそれ自体を1つ以上のサービス名を使用してリスナーに登録します。クライアント・プログラムまたはデータベースがリスナーに接続すると、これらはサービスへの接続を要求します。

サービス名は複数のデータベース・インスタンスを識別することができ、インスタンスは複数のサービスに属することができます。このため、リスナーはクライアントとインスタンスとの間の仲介役を果し、接続要求を適切なインスタンスに渡します。サービスに接続するクライアントは、必要なインスタンスを指定する必要がありません。

サービス名は、[サーバー・パラメータ・ファイル](#)のSERVICE_NAMES初期化パラメータで指定します。サーバー・パラメータ・ファイルを使用すると、ALTER SYSTEMコマンドを使用して初期化パラメータを変更でき、変更内容は停止して起動した後も維持されます。DBMS_SERVICEパッケージを使用して、サービスを作成することもできます。サービス名のデフォルトは[グローバル・データベース名](#)で、データベース名(DB_NAME初期化パラメータ)およびドメイン名(DB_DOMAIN初期化パラメータ)から構成されています。sales.us.example.comの場合、データベース名はsalesで、ドメイン名はus.example.comです。

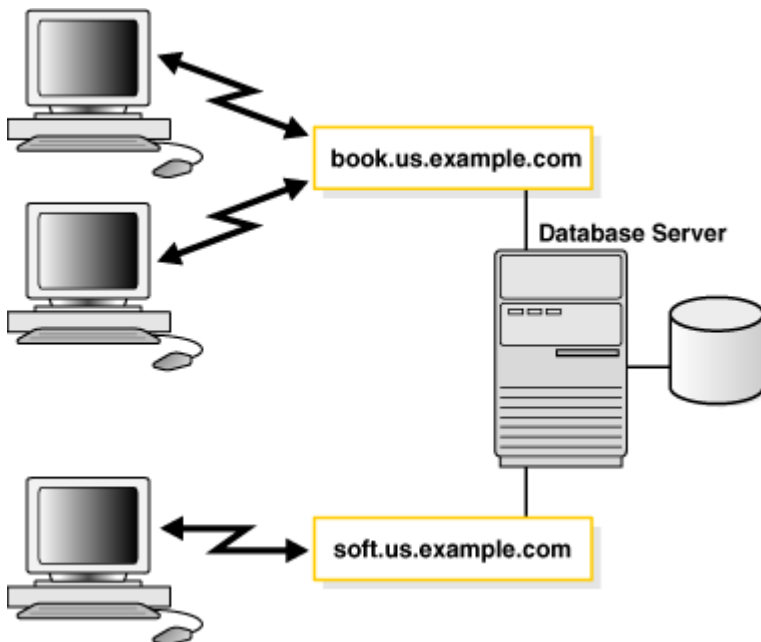
ノート:



Oracle Database 19c以降、SERVICE_NAMESパラメータをお客様が使用することは非推奨になりました。サービスを管理するには、SRVCTL または GDSCTL コマンドライン・ユーティリティ、または DBMS_SERVICE パッケージを使用することをお勧めします。

[図2-4](#)では、1つのデータベースに対応する複数のサービスに接続しているクライアントを示します。

図2-4 1つのデータベースへの複数サービスの対応付け



1つのデータベースに複数サービスを対応付けると、次のような機能が得られます。

- 単一のデータベースを、異なる方法であらゆるクライアントが識別できます。
- データベース管理者は、システム・リソースを制限したり、確保できます。このレベルの制御では、これらのサービスの1つを要求するクライアントに、より適切にリソースを割り当てることが可能です。

関連項目:

- 初期化パラメータの詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。
- ALTER SYSTEM文の詳細は、[『Oracle Database SQLリファレンス』](#)を参照してください。
- SERVICE_NAMESパラメータの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。
- DBMS_SERVICEパッケージの詳細は、[『Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス』](#)を参照してください。

親トピック: [データベースの識別とアクセス](#)

2.3 データベース・サービスへの接続

データベース・サービスに接続するために、クライアントは、データベースの場所とデータベース・サービスの名前を示す[接続記述子](#)を使用します。次の例は、sales.us.example.comというデータベース・サービス、およびホストsales-serverに接続する簡易接続記述子を示しています(デフォルトではポートは1521です)。

```
sales-server/sales.us.example.com
```

次の例は、前述の簡易接続記述子およびデータベース・サービスに対するtnsnames.oraファイル内のエントリを示しています。

```
(DESCRIPTION=
  (ADDRESS= (PROTOCOL=tcp) (HOST=sales-server) (PORT=1521))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.example.com)))
```

- [接続記述子について](#)
- [プロトコル・アドレスについて](#)
- [サービス登録について](#)

関連項目:

[「ネーミング・メソッドの理解」](#)

親トピック: [データベースの識別とアクセス](#)

2.3.1 接続記述子について

接続記述子は、tnsnames.oraファイルに記述されている、リスナーの1つまたは複数のプロトコル・アドレスおよび宛先サービスの接続情報により構成されます。[例2-1](#)は、salesデータベースにマップされた接続記述子を示しています。

例2-1 接続記述子

```
sales=
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521))
  (CONNECT_DATA=
    (SID=sales)
    (SERVICE_NAME=sales.us.example.com)
    (INSTANCE_NAME=sales)))
```

[例2-1](#)に示すように、接続記述子には次のパラメータが含まれています。

- ADDRESSセクションには次のパラメータがあります。
 - PROTOCOLパラメータは、リスナー・プロトコル・アドレスを識別します。プロトコルは、TCP/IPの場合tcpです。
 - HOSTパラメータは、ホスト名を識別します。ホストはsales-serverです。
 - PORTパラメータは、ポートを識別します。ポートは、デフォルトのポート番号1521です。
 - オプションのHTTPS_PROXYパラメータとHTTPS_PROXY_PORTパラメータにより、データベース・クライアントの接続は組織の転送Webプロキシを通過できるようになります。これらのパラメータは、PROTOCOL=TCPSの接続記述子にのみ適用されます。
- CONNECT_DATAセクションには次のパラメータがあります。
 - SIDパラメータは、Oracle Databaseのシステム識別子(SID)を識別します。SIDはsalesです。
 - SERVICE_NAMEパラメータはサービスを識別します。宛先サービス名は、sales.us.example.comという名前のデータベース・サービスです。

この接続記述子パラメータの値は、初期化パラメータ・ファイルのSERVICE_NAMES初期化パラメータ (SERVICE_NAMESでは末尾にSが使用されます)から取得されます。一般的にSERVICE_NAMES初期化パラメータは、データベース名とドメイン名が含まれる[グローバル・データベース名](#)です。例では、sales.us.example.comには、salesのデータベース名とus.example.comのドメインがあります。

ノート:



Oracle Database 19c以降、SERVICE_NAMESパラメータをお客様が使用することは非推奨になりました。サービスを管理するには、SRVCTL または GDSCTL コマンドライン・ユーティリティ、または DBMS_SERVICE パッケージを使用することをお勧めします。

- INSTANCE_NAMEパラメータはデータベース・インスタンスを識別します。インスタンス名はオプションです。
初期化パラメータ・ファイルのINSTANCE_NAMEパラメータは、インストール中またはデータベース作成中に入力されたSIDが、デフォルトで設定されます。

- [接続記述子のIPv6アドレスについて](#)

関連項目:

[「データベース・インスタンスの理解」](#)および[「データベース・サービスの理解」](#)

親トピック: [データベース・サービスへの接続](#)

2.3.1.1 接続記述子のIPv6アドレスについて

ホストは、IP version 4 (IPv4)およびIP version 6 (IPv6)インタフェースを使用できます。IPv6アドレスとIPv6アドレスに解決されるホスト名は、TNS接続アドレスのHOSTパラメータで使用すると便利で、このTNS接続アドレスは、[ネーミング・メソッドの選択](#)にリストされているサポート対象のNetネーミング・メソッドのいずれかを介して取得できます。

Oracle Database 12cでIPv6を使用してエンドツーエンド接続を確立するには、次の構成が必要です。

- クライアントTNS接続アドレスは、IPv6エンドポイント上のOracle Net Listenerに接続する必要があります。
- Oracle Net Listenerに対して構成されているデータベース・インスタンスは、IPv6エンドポイント上の接続要求をリスニングする必要があります。

ホスト名が指定されている場合、正常な接続が確立されるか、すべてのアドレスが試行されるまで、Oracle Netはドメイン・ネーム・システム(DNS)の名前解決によって戻されるすべてのIPアドレスへの接続を試行します。[例2-1](#)で、sales-serverホストはクライアント接続を受け入れるIPv4専用ホストであるとして、DNSはsales-serverを次のIPアドレスにマップします。

1. IPv6アドレス2001:0db8:0:0::200C:417A
2. IPv4アドレス192.0.2.213

この場合、IPv6アドレスがDNSリストの先頭にあるため、Oracle Netはまずこのアドレスで接続を試行します。この例では、sales-serverはIPv6接続をサポートしていないため、この試行は失敗します。Oracle NetはIPv4アドレスへの接続に進み、この試行は成功します。

関連項目:

- [「TCP/IPプロトコルについて」](#)
- [「リスニング用プロトコル・アドレスの構成」](#)
- [「IPv4およびIPv6のブリッジとしてのOracle Connection Managerの使用」](#)

- [IPv6のネットワーク接続](#)

親トピック: [接続記述子について](#)

2.3.2 プロトコル・アドレスについて

接続記述子のアドレスの一部は[リスナー](#)のプロトコル・アドレスです。データベース・サービスに接続するには、クライアントは、まず、データベース・サーバーに常駐しているリスナー・プロセスに接続します。リスナーはクライアントからの着信接続要求を受信し、これらの要求をデータベース・サーバーに送信します。接続が確立された後、クライアントとデータベース・サーバーは直接通信します。

リスナーはクライアントからの要求を受け入れるように[プロトコル・アドレス](#)で設定できます。このアドレスはリスナーがリスニングを実行するプロトコルと、プロトコル固有のその他の情報を定義します。たとえば、リスナーを次のプロトコル・アドレスでリスニングを実行するように設定できます。

```
(DESCRIPTION=
  (ADDRESS= (PROTOCOL=tcp) (HOST=sales-server) (PORT=1521)))
```

前の例では、リスナーのホストとポート番号を指定するTCP/IPプロトコル・アドレスを示しています。これと同じプロトコル・アドレスで構成されたクライアント接続記述子は、このリスナーに接続要求を送信できます。

親トピック: [データベース・サービスへの接続](#)

2.3.3 サービス登録について

接続記述子ではデータベース・サービス名を指定し、これを使用してクライアントは接続の確立を試みます。リスナーは接続要求を処理できるサービスを認識しますが、これは、Oracle Databaseがこの情報をリスナーに動的に登録しているためです。この登録プロセスは、[サービス登録](#)と呼ばれます。登録によって、データベース・インスタンス、および各インスタンスで利用可能なサービス・ハンドラに関する情報がリスナーに提供されます。[ディスパッチャ](#)または[専用サーバー](#)があります。

- [インスタンス名の指定](#)
- [サービス・ハンドラの指定](#)

親トピック: [データベース・サービスへの接続](#)

2.3.3.1 インスタンス名の指定

データベースの特定のインスタンスへの接続が必要な場合、クライアントは特定インスタンスのINSTANCE_NAMEを接続記述子で指定できます。この機能は、Oracle RAC構成を使用する場合に役立ちます。たとえば、次の接続記述子は、sales.us.example.comに対応付けられているsales1のインスタンス名を指定しています。

```
(DESCRIPTION=
  (ADDRESS= (PROTOCOL=tcp) (HOST=sales-server) (PORT=1521))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.example.com)
    (INSTANCE_NAME=sales1)))
```

親トピック: [サービス登録について](#)

2.3.3.2 サービス・ハンドラの指定

特定のタイプのサービス・ハンドラを常に使用するクライアントは、そのサービス・ハンドラのタイプを指定する接続記述子を使用できます。次の例では、接続記述子は (SERVER=shared) を使用して、データベース接続時にディスパッチャを要求します。デフォルトで専用サーバーを使用するようにデータベースを構成できます。

```
(DESCRIPTION=
  (ADDRESS= (PROTOCOL=tcp) (HOST=sales-server) (PORT=1521))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.example.com)
    (SERVER=shared)))
```

リスナーはクライアントの要求を受け取ると、登録されているサービス・ハンドラの1つを選択します。選択したハンドラのタイプ、使用する通信プロトコル、データベース・サーバーのオペレーティング・システムに基づいて、リスナーは次の処理のいずれかを実行します。

- 接続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・プロセスに直接接続します。
- 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。

リスナーがクライアントとの接続処理を完了すると、クライアントはリスナーを介さずに Oracle Database と直接通信します。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。

サービス・ハンドラを指定する場合は、次の点を考慮する必要があります。

- クライアントに専用サーバーを使用する場合は、(SERVER=dedicated) を指定します。SERVERパラメータが設定されていない場合、共有サーバー構成と見なされます。しかし、利用できるディスパッチャがない場合、クライアントは専用サーバーを使用します。
- データベース常駐接続プーリングがサーバーで有効になっている場合、(SERVER=pooled) を指定してプールから接続を取得します。データベース常駐接続プーリングがサーバーで有効になっていない場合、クライアント要求は拒否され、ユーザーはエラー・メッセージを受け取ります。

関連項目:

- リスナーでサービス・ハンドラを使用する方法は、「[リスナーおよび接続要求について](#)」を参照
- 『[Oracle Call Interface プログラマーズ・ガイド](#)』および『[Oracle Database 管理者ガイド](#)』を参照してください。
- グローバル・サービス管理の詳細は、[Oracle Database Global Data Services 概要および管理ガイド](#)を参照してください。

親トピック: [サービス登録について](#)

2.4 サービス・ハンドラの理解

サービス・ハンドラは、Oracle Database への接続ポイントとして機能します。サービス・ハンドラには、[ディスパッチャ](#)または[専用サーバー・プロセス](#)、またはプールがあります。

- [ディスパッチャについて](#)
- [専用サーバー・プロセスについて](#)
- [データベース常駐接続プーリングについて](#)

親トピック: [データベースの識別とアクセス](#)

2.4.1 ディスパッチャについて

共有サーバー・アーキテクチャはディスパッチャ・プロセスを使用して、クライアント接続を共通の要求キューに渡します。サーバー・プロセスの共有プールの中のアイドル状態の共有サーバー・プロセスは、共通キューから要求を取り出します。このアプローチでは、小さいサーバー・プロセス・プールで大量のクライアントを処理することが可能です。専用サーバー・モデルと比較した共有サーバー・モデルの大きな利点は、システム・リソースが少なく済むため、ユーザー数の増加に対応できることです。

リスナーは[サービス・ハンドラ](#)のタイプとしてディスパッチャを使用しますが、これにクライアント要求を渡すことができます。クライアント要求を受け取ると、リスナーは次のいずれかの処理を実行します。

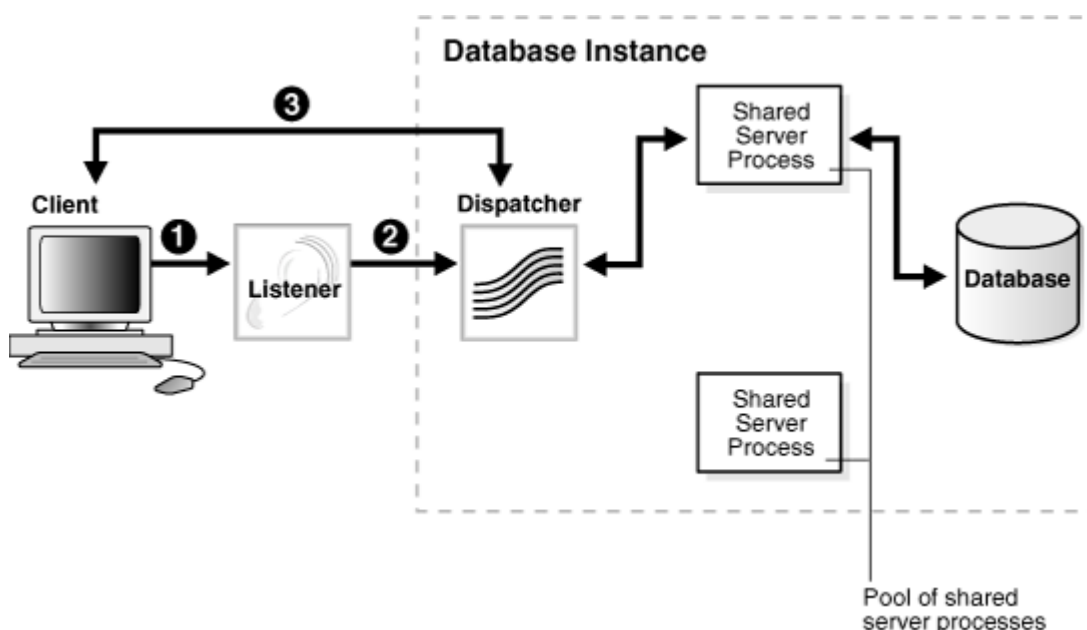
- 接続要求を直接ディスパッチャに渡します。
- ディスパッチャのプロトコル・アドレスを含むリダイレクト・メッセージをリスナーに発行します。次に、クライアントは、リスナーに要求したネットワーク・セッションを終了し、リダイレクト・メッセージで提供されたネットワーク・アドレスを使用して、ディスパッチャとのネットワーク・セッションを確立します。

リスナーは可能な場合は必ずダイレクト・ハンドオフを使用します。リダイレクト・メッセージは、たとえばディスパッチャがリスナーに対してリモートである場合に使用します。

[図2-5](#)では、リスナーが接続要求をディスパッチャに直接渡す様子を示します。

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、接続要求をディスパッチャに直接渡します。
3. クライアントは、ここでディスパッチャに直接接続します。

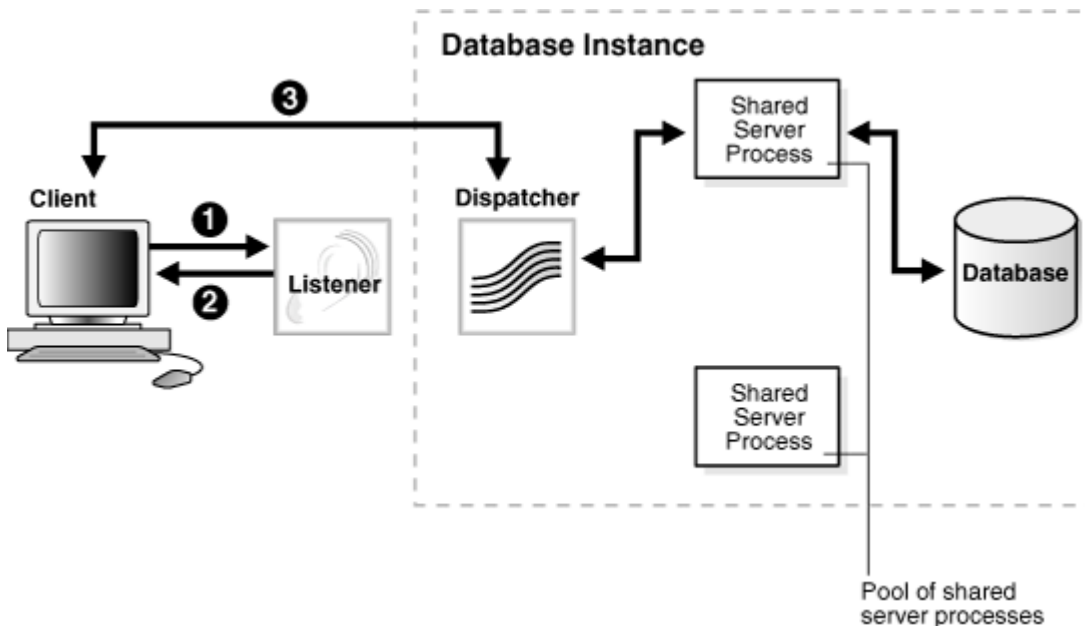
図2-5 ディスパッチャへのダイレクト・ハンドオフ



[図2-6](#)では、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2-6 ディスパッチャにリダイレクトされた接続



親トピック: [サービス・ハンドラの理解](#)

2.4.2 専用サーバー・プロセスについて

専用サーバー構成では、リスナーはクライアントの着信接続要求ごとに専用サーバー・プロセスを個別に起動します。このプロセスはクライアントへのサービス提供のみを行います。セッション完了後、専用サーバー・プロセスは終了します。専用サーバー・プロセスは接続ごとに起動する必要があるため、共有サーバー構成よりも多くのシステム・リソースが構成に必要な場合があります。

専用サーバー・プロセスは、クライアント要求を受け取った時にリスナーが開始するサービス・ハンドラのタイプです。クライアント/サーバー接続を完了するには、次のいずれかの処理が発生します。

- 専用サーバーはリスナーから接続要求を継承します。
- 専用サーバーはリスナーにリスニング・プロトコル・アドレスを通知します。リスナーはリダイレクト・メッセージでプロトコル・アドレスをクライアントに渡し、接続を終了します。クライアントは、そのプロトコル・アドレスを使用して、専用サーバーに直接接続します。

前述の処理のどちらかが選択されるかは、オペレーティング・システムおよび使用中のトランスポート・プロトコルによって決まります。

クライアントとデータベースが同じコンピュータ上に存在する場合、クライアント接続は、リスナーを経由せずに専用サーバー・プロセスに直接渡すことができます。これは、bequeathプロトコルとして知られています。セッションを開始するアプリケーションは、接続要求に対する専用サーバー・プロセスを生成します。データベースの起動に使用されるアプリケーションがデータベースと同じコンピュータ上にある場合、この処理は自動的に実行されます。

ノート:



リモート・クライアントが専用サーバーに接続するためには、リスナーとデータベース・インスタンスを同じコンピュータ上で実行する必要があります。

図2-7では、リスナーがクライアント接続要求を専用サーバー・プロセスに渡す様子を示します。

1. リスナーがクライアント接続要求を受け取ります。
2. リスナーは専用サーバー・プロセスを開始し、専用サーバーはリスナーから接続要求を継承します。
3. クライアントがここで専用サーバーに直接接続します。

図2-7 専用サーバー・プロセスへの接続

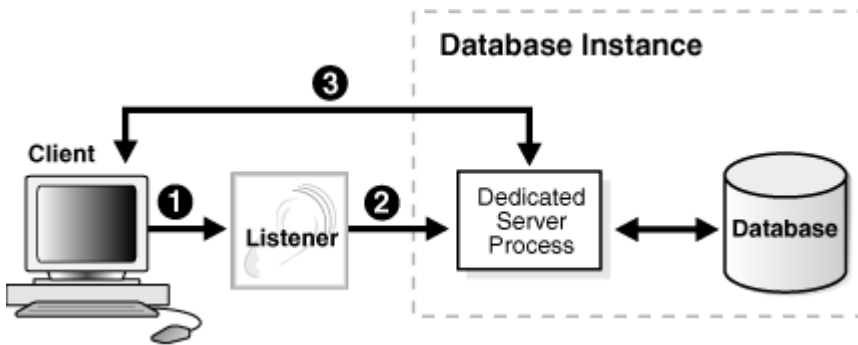
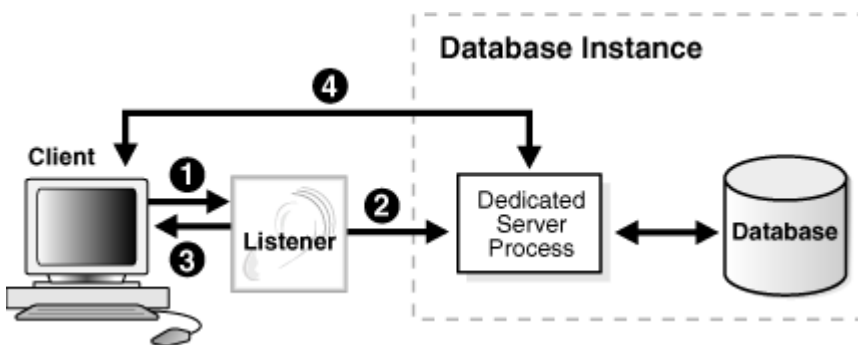


図2-8では、リダイレクト接続における専用サーバーの役割を示します。

1. リスナーがクライアント接続要求を受け取ります。
2. リスナーは専用サーバー・プロセスを開始します。
3. リスナーが専用サーバー・プロセス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2-8 専用サーバー・プロセスへのリダイレクト接続



親トピック: [サービス・ハンドラの理解](#)

2.4.3 データベース常駐接続プーリングについて

データベース常駐接続プーリングは、データベース接続を取得し、比較的短時間の処理を実行した後、データベース接続を解放するような一般的なWebアプリケーション使用シナリオに対して、データベース・サーバーの接続プールを提供します。データベース常駐接続プーリングは、専用サーバーをプールします。プール・サーバーは、サーバー・フォアグラウンド・プロセスとデータベース・セッションの組合せに相当します。データベース常駐接続プーリングは、サーバーとリスナーの間の動的な登録を使用します。静的な登録を使用することはできません。

データベース常駐接続プーリングは、中間層プロセス内のスレッド間で接続を共有する中間層接続プールを補完します。また、これを使用すると、同じ中間層ホスト上の中間層プロセス間、および異なる中間層ホスト上の中間層プロセス間でもデータベース接続を共有できます。この結果、大量のクライアント接続をサポートするために必要となる基本データベース・リソースが大幅に減少するため、データベース層のメモリー・フットプリントが縮小し、中間層とデータベース層の両方のスケールビリティが向上します。すぐに使用できるサーバーのプールを保持することで、クライアント接続の作成と切断のコストが削減されるという利点もあります。

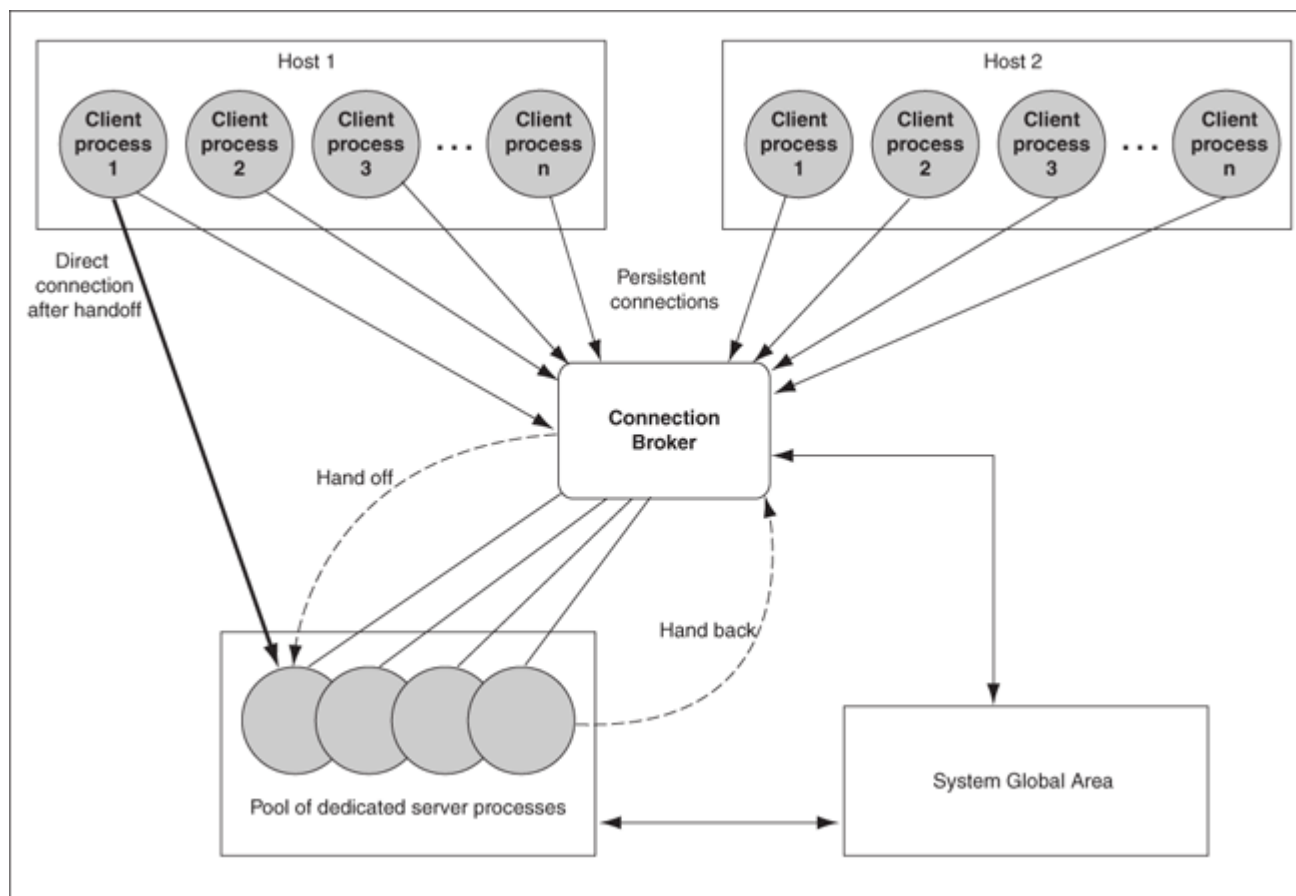
データベース常駐接続プーリングは、クライアント・アプリケーションとプロセス全体に[専用接続](#)のプーリングを提供します。この機能は、データベースへの永続的な接続を維持してサーバー・リソース(メモリーなど)を最適化する必要があるアプリケーションに役立ちます。

データベース常駐接続プールから接続を取得するクライアントは、専用サーバーのかわりにバックグラウンド・プロセス(接続ブローカ)に永続的に接続されます。接続ブローカはプール機能を実装しており、クライアントから専用サーバーのプールへのセッションを使用したインバウンド接続の多重化を実行します。

クライアントがデータベースの作業を実行する必要がある場合、接続ブローカはプールから専用サーバーを選択し、クライアントに割り当てます。その後、クライアントは要求が処理されるまで専用サーバーに直接接続されます。サーバーがクライアント要求の処理を終了した後、サーバーはプールに戻され、クライアントからの接続は接続ブローカに戻されます。

[図2-9](#)は、プロセスを図で示しています。

図2-9 接続ブローカ・プロセスにより接続を処理する専用サーバー・プロセス



親トピック: [サービス・ハンドラの理解](#)

2.5 ネーミング・メソッドの理解

ネーミング・メソッドとは、データベース・サービスに接続するときに、クライアント・アプリケーションが接続識別子を接続記述子に変換するために使用する解決メソッドです。ユーザーは[接続文字列](#)を指定して接続要求を開始します。接続文字列には、ユーザー名、パスワードおよび[接続識別子](#)が含まれます。接続識別子には、接続記述子または接続記述子に解決される名前を使用できます。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。
- データベース・[サービス名](#)または[Oracleシステム識別子\(SID\)](#)。

次のCONNECTコマンドでは、ネットワーク・サービス名のかわりに、接続識別子として完全な接続記述子を持つ接続文字列を使用しています。1つの行として、文字列を入力する必要があります。ページの幅のために2行で表示されます。

```
SQL> CONNECT hr@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sales-server1)
(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))
```

最も一般的な接続識別子の1つは、サービスの単純な名前である[ネットワーク・サービス名](#)です。次のCONNECTコマンドでは、接続文字列に、接続識別子としてネットワーク・サービス名salesを使用しています。

```
SQL> CONNECT hr@sales
```

ネットワーク・サービス名のsales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。このマップ情報には[ネーミング・メソッド](#)でアクセスします。次のネーミング・メソッドを利用できます。

- ローカル・ネーミング
- ディレクトリ・ネーミング
- 簡易接続ネーミング
- 外部ネーミング
- [ネーミング・メソッドの選択](#)
- [ネーミング・メソッドを使用したクライアント・セッションの確立](#)
- [接続文字列の入力](#)

ネットワーク・コンポーネントの起動後に、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、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。

親トピック: [データベースの識別とアクセス](#)

2.5.1 ネーミング・メソッドの選択

接続記述子に名前をマッピングする適切なネーミング・メソッドの選択は組織の規模によって決まります。

- データベースが少ない小規模な組織では、簡易接続ネーミングを使用してデータベース・サーバーのホスト名にTCP/IP接続するか、またはローカル・ネーミングを使用してクライアント上のtnsnames.oraファイルに名前を格納します。
- データベースが多い大規模な組織では、ディレクトリ・ネーミングを使用して集中化されたディレクトリ・サーバーに名前を格納します。
- インターネット・ネットワークではローカル・ネーミング・メソッドでデータベースに接続するために必要なアプリケーションWebサーバーを構成します。

表2-1では、各ネーミング・メソッドの相対的なメリットおよびデメリットをまとめ、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推奨事項を示します。

表2-1 ネーミング・メソッド: メリットおよびデメリット

ネーミング・メソッド	説明	メリット/デメリット	推奨環境:
ローカル・ネーミング	ネットワーク・サービス名とその接続記述子を、tnsnames.ora という名前のローカルに配置された構成ファイルに格納します(この構成ファイルは、デフォルトでは Oracle Database ホームの下の network/admin サブディレクトリにあります)。	<p>メリット:</p> <ul style="list-style-type: none"> ● ネットワーク・サービス名アドレスを解決する簡単な方法です。 ● 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ワーク・サービス名を解決します。 <p>デメリット: すべてのネットワーク・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	接続識別子を集中化された LDAP 準拠の ディレクトリ・サーバー に格納し、データベース・サービスにアクセスします。	<p>メリット:</p> <ul style="list-style-type: none"> ● ネットワーク名とアドレスを 1 箇所に集中させるため、名前の変更と更新が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。 ● ディレクトリはこの他のサービスの名前も格納します。 ● ツールを使用して簡単に構成ができます。 <p>デメリット: 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)
簡易接続ネーミング	クライアントは、ホスト名およびオプションのポート名やサービス名から構成される TCP/IP 接続文字列を使用して、Oracle Database サーバーへ接続できるようになります。	<p>メリット:</p> <ul style="list-style-type: none"> ● 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データベース・ホスト名のみを指定します。 ● 簡略ネーミング・メソッドではクライア 	次に示す条件に合致する単純な TCP/IP ネットワーク <ul style="list-style-type: none"> ● 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

ネーミング・メソッド 説明	メリット/デメリット	推奨環境:
	<p>ント側の構成は必要ありません。</p> <ul style="list-style-type: none"> 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。 <p>デメリット: 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>こと。</p> <ul style="list-style-type: none"> 拡張接続記述子を必要とする機能がないこと。
外部ネーミング	<p>Network Information Service (NIS)外部ネーミングなど、サポートされるサード・パーティ・ネーミング・サービスにネットワーク・サービス名を保存します。</p> <p>メリット: 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 ネットワーク・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>デメリット: Oracle Net 製品では管理できないサード・パーティのネーミング・サービスが必要です。</p>	既存のネーム・サービスを持つネットワーク

ネーミング・メソッドは、次のステップで構成します。

1. ネーミング・メソッドを選択します。
2. 接続記述子を名前にマップします。
3. そのネーミング・メソッドを使用するクライアントを構成します。

親トピック: [ネーミング・メソッドの理解](#)

2.5.2 ネーミング・メソッドを使用したクライアント・セッションの確立

ネーミング・メソッドを使用してクライアント・セッションを確立する代表的なプロセスは、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、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、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、それを該当するデータベース・サーバーに送ります。
5. データベース・サーバーによって、接続が受け入れられます。

ノート:



接続記述子の他にも、ネーミング・メソッドを使用して、接続名をプロトコル・アドレスやプロトコル・アドレス・リストにマッピングできます。

親トピック: [ネーミング・メソッドの理解](#)

2.5.3 接続文字列の入力

ネットワーク・コンポーネントの起動後に、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、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](#)は次のような形式になります。

```
CONNECT username@connect_identifier
```

デフォルトの接続識別子

ほとんどのオペレーティング・システムで、デフォルトの[接続識別子](#)を定義できます。デフォルトを使用する場合、接続文字列で接続識別子を指定する必要がなくなります。デフォルトの接続識別子を定義するには、LinuxおよびUNIXプラットフォームの場合はTWO_TASK環境変数を、Microsoft Windowsの場合はLOCAL環境変数またはレジストリ・エントリを使用します。

たとえば、環境変数TWO_TASKがsalesに設定されている場合は、SQL*Plusから、CONNECT username@salesではなくCONNECT usernameを使用してデータベースに接続できます。Oracle NetはTWO_TASK変数を確認し、接続識別子として値salesを使用します。

TWO_TASKおよびLOCAL環境変数の設定方法は、Oracleのオペレーティング・システム固有のマニュアルを参照してください。

接続識別子と接続記述子の構文特性

接続文字列で使用される接続識別子には、一重引用符(')または二重引用符(")で囲む場合を除いて、空白を含めることはできません。次の例では、空白を含む接続識別子および接続記述子が一重引用符で囲まれています。

```
CONNECT scott@' (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=sales-server)
(PORT=1521)) (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))'
Enter password: password
CONNECT scott@' cn=sales, cn=OracleContext, dc=us, dc=example, dc=com'
Enter password: password
```

接続識別子の中で二重引用符(")が使用されている場合は、一重引用符(')で囲みます。たとえば:

```
CONNECT scott@' sales@"good"example.com'
Enter password: password
```

接続識別子の中で一重引用符(')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:

```
CONNECT scott@"cn=sales, cn=OracleContext, ou=Mary's Dept, o=example"
Enter password: password
```

親トピック: [ネーミング・メソッドの理解](#)

2.6 複数リスナーを使用したサービスのアクセス可能性の拡張

Oracle RACなどの一部の構成では、同じデータベース・サービスへのクライアント接続要求を処理するために、複数のノード上に複数のリスナーを構成できます。次の例では、sales.us.example.comはsales1-serverまたはsales2-serverでリスナーを使用して接続できます。

```
(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.example.com)))
```

複数のリスナー構成では、フェイルオーバーやロード・バランシング機能も、単独で、あるいはそれぞれを組み合わせる有効的に活用できます。

- [接続時フェイルオーバーについて](#)
- [透過的アプリケーション・フェイルオーバーについて](#)
- [クライアント・ロード・バランシングについて](#)
- [接続ロード・バランシングについて](#)

親トピック: [データベースの識別とアクセス](#)

2.6.1 接続時フェイルオーバーについて

[接続時フェイルオーバー](#)機能を使用すると、クライアントは、最初のリスナーへの接続に失敗した場合、別のリスナーに接続できます。接続を試行するリスナーの数は、リスナー・プロトコル・アドレスの数で決まります。接続時フェイルオーバーを実行しない場合、Oracle Netは、1つのリスナーへの接続のみ試行します。

親トピック: [複数リスナーを使用したサービスのアクセス可能性の拡張](#)

2.6.2 透過的アプリケーション・フェイルオーバーについて

[透過的アプリケーション・フェイルオーバー\(TAF\)](#)機能は、Oracle Real Application Clustersなどの高可用性環境のためのランタイム・フェイルオーバーです。TAFでは、アプリケーションとサービス間の接続のフェイルオーバーおよび再確立を行います。これにより、クライアント・アプリケーションは接続障害の発生時にデータベースに自動的に再接続でき、実行中だったSELECT文を再開することも可能です。この再接続は、[Oracle Call Interface\(OCI\)](#)ライブラリ内から自動的に実行されます。

親トピック: [複数リスナーを使用したサービスのアクセス可能性の拡張](#)

2.6.3 クライアント・ロード・バランシングについて

[クライアント・ロード・バランシング](#)機能によって、クライアントは、複数のリスナー間で接続要求をランダム化できます。Oracle Netは、プロトコル・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を均衡化します。クライアント・ロード・バランシングを実行しない場合、Oracle Netは、接続が成功するまでプロトコル・アドレスを順番に試行します。

親トピック: [複数リスナーを使用したサービスのアクセス可能性の拡張](#)

2.6.4 接続ロード・バランシングについて

[接続ロード・バランシング](#)機能を利用すると、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。シングル・インスタンス環境では、リスナーは最も負荷の少ないディスパッチャを選択して、クライアントの着信要求を処理できます。Oracle Real Application Clusters環境では、接続ロード・バランシングによって、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ことも可能です。

サービス登録を動的に行うため、リスナーはすべてのインスタンスとディスパッチャをその場所に関係なく常に認識します。リスナーは、着信したクライアント要求の送信先となるインスタンスを、また共有サーバーが構成されている場合は送信先となるディスパッチャを、ロード情報に応じて判別します。

共有サーバー構成では、リスナーは次の順番でディスパッチャを選択します。

1. ロード量が最小のノード
2. ロード量が最小のインスタンス
3. そのインスタンスのロード量が最小のディスパッチャ

専用サーバー構成では、リスナーは次の順番でインスタンスを選択します。

1. ロード量が最小のノード
2. ロード量が最小のインスタンス

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を持つ場合、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。

親トピック: [複数リスナーを使用したサービスのアクセス可能性の拡張](#)

3 ネットワーク・アドレス情報の管理

Oracle Net Servicesのネットワーク・アドレス情報を、ローカル・ファイルや集中化されたディレクトリ・サーバーに格納する方法について説明します。ネットワーク・アドレス情報は、ローカル管理を使用してネットワーク内の各コンピュータのtnsnames.oraファイルに格納されます。また、集中管理を使用して、集中化した[ディレクトリ・サーバー](#)に格納されます。

- [ローカル管理の使用](#)
- [集中管理用のディレクトリ・サーバーの使用](#)

親トピック: [Oracle Net Servicesの理解](#)

3.1 ローカル管理の使用

ローカル管理を使用するときは、ネットワーク・コンピュータは[表3-1](#)で説明されているファイルで構成されます。このファイルはローカルにコンピュータに格納されます。

表3-1 ローカル管理で使用されるOracle Net構成ファイル

構成ファイル	説明
tnsnames.ora	このファイルは、主にクライアント上にあり、接続記述子にマップされたネットワーク・サービス名を含んでいます。このファイルは、ローカル・ネーミング・メソッドに使用されます。
sqlnet.ora	このファイルは、クライアントとデータベース・サーバーのコンピュータに配置され、内容は次のとおりです。 <ul style="list-style-type: none">● 未修飾サービス名またはネットワーク・サービス名に追加するクライアント・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● 使用するロギング機能とトレース機能● 接続のルート● 外部ネーミング・パラメータ● Oracle セキュリティ・パラメータ● データベース・アクセス制御パラメータ
listener.ora	このリスナーに関する構成ファイルは、データベース・サーバーに配置され、内容は次のとおりです。 <ul style="list-style-type: none">● 接続要求を受け付けるプロトコル・アドレス● リスニングの対象とするデータベース・サービスと非データベース・サービス● リスナーが使用する制御パラメータ
cman.ora	この構成ファイルは、Oracle Connection Manager が実行されるコンピュータ上に

構成ファイル	説明
	<p>ありますが、これには次のコンポーネントが含まれます。</p> <ul style="list-style-type: none"> ● リスニング・エンドポイント ● アクセス制御ルール・リスト ● パラメータ・リスト <p>各 Oracle Connection Manager 構成は、1 つの名前-値(NV)文字列内にカプセル化されており、その文字列は、前述のコンポーネントで構成されています。</p>

構成ファイルは通常、読取り専用OracleホームのORACLE_HOME/network/adminディレクトリとORACLE_BASE_HOME/network/adminディレクトリに作成されます。ただし、構成ファイルは他のディレクトリに作成される可能性があります。Oracle Netは、他のディレクトリに構成ファイルがないか確認します。たとえば、tnsnames.oraファイルの確認順序は次のとおりです。

1. 環境変数TNS_ADMINで指定されているディレクトリ。
2. TNS_ADMIN環境変数が設定されていないか、ファイルがTNS_ADMINディレクトリにない場合、Oracle NetはORACLE_HOME/network/adminディレクトリ内のファイルをチェックします。読取り専用Oracleホームの場合、Oracle NetはORACLE_BASE_HOME/network/adminディレクトリを確認します。

ノート:



Microsoft Windows では、TNS_ADMIN 環境変数はプロセスの環境内で設定されている場合に使用されます。TNS_ADMIN 環境変数が環境内で定義されていない場合、またはプロセスが環境を持たないサービスである場合、Microsoft Windows ではレジストリで TNS_ADMIN パラメータがスキャンされます。

関連項目:

- gsm.oraファイルの詳細は、[Oracle Database Global Data Services概要および管理ガイド](#)を参照してください。
- オペレーティング・システム固有のドキュメント

親トピック: [ネットワーク・アドレス情報の管理](#)

3.2 集中管理用のディレクトリ・サーバーの使用

Oracleネットワークでディレクトリ・サーバーを使用すると、そのディレクトリ・サーバーは、ネットワーク内のOracle Databaseに対するグローバル・データベース・リンクをネットワーク・サービス名として管理します。ユーザーとプログラムは、グローバル・リンクを使用して、対応するデータベース内のオブジェクトにアクセスできます。

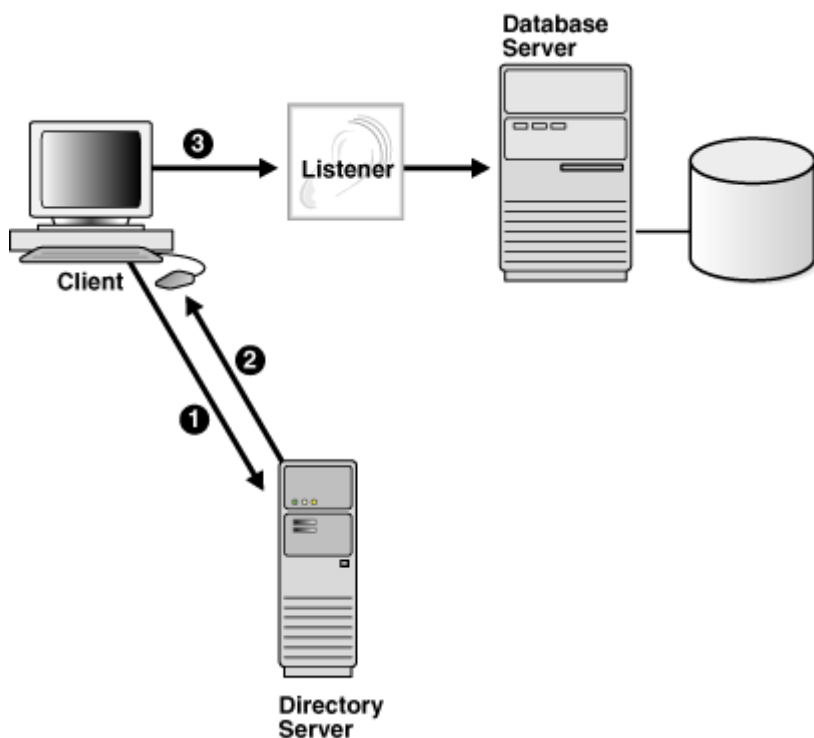
Oracle Net Servicesは、[接続識別子](#)を格納する主な方法の1つとして、集中化されたディレクトリ・サーバーを使用します。クライアントは、接続文字列の中で接続識別子を使用し、ディレクトリ・サーバーはこの接続識別子を接続記述子に解決して、クライアントに戻します。この機能は、[ディレクトリ・ネーミング](#)と呼ばれます。このようなインフラストラクチャによって、ネットワークでのリソースの管理および構成のコストが削減されます。

[Oracle Internet Directory](#)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メカニズムが提供されます。このディレクトリ・サーバーによって、クライアント側とサーバー側にあるローカルのtnsnames.oraファイルを置換できます。

図3-1では、ディレクトリ・サーバー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ます。

1. クライアントは、ディレクトリ・サーバーに接続して接続識別子を接続記述子に解決します。
2. ディレクトリ・サーバーは、この接続識別子を解決して、クライアントに接続記述子を渡します。
3. クライアントは、この接続記述子を使用して、リスナーに接続要求を送信します。

図3-1 ディレクトリ・サーバーを使用して接続識別子を解決するクライアント



ノート:



[Java Database Connectivity\(JDBC\)ドライバ](#)は、ディレクトリ・ネーミングをサポートしています。詳細は、『[Oracle Database JDBC 開発者ガイド](#)』を参照してください。

この項では、次の項目について説明します。

- [ディレクトリ情報ツリーの理解](#)
- [Oracleコンテキストの理解](#)
- [ネット・サービス別名エントリの理解](#)
- [ディレクトリ・サーバーのエントリを追加または変更できるユーザー](#)
- [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](#)
- [ディレクトリ・サーバーを使用するときの考慮事項](#)
- [Microsoft Active Directoryにおけるディレクトリ・ネーミング・サポートの制約](#)

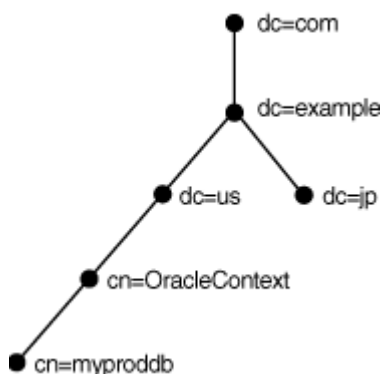
親トピック: [ネットワーク・アドレス情報の管理](#)

3.2.1 ディレクトリ情報ツリーの理解

ディレクトリ・サーバーは、[ディレクトリ情報ツリー\(DIT\)](#)と呼ばれる階層化されたネームスペース構造に情報を格納します。このツリーの各ノードは、[エントリ](#)と呼ばれます。Oracle Net Servicesは、ツリー構造とツリー内の特定のエントリを使用します。DITは、一般にドメイン・ネーム・システム(DNS)構造などの既存のドメイン構造や地理的および組織的な構造と整合されます。

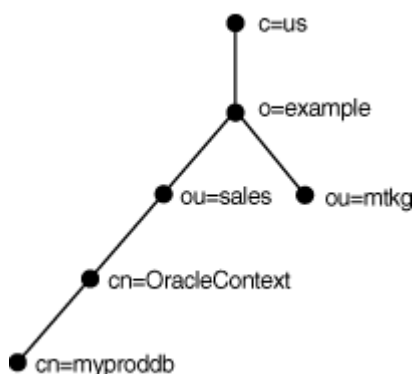
[図3-2](#)では、DNSドメイン・コンポーネントに従って構築されたDITを示します。

図3-2 DNSドメイン・コンポーネントDIT



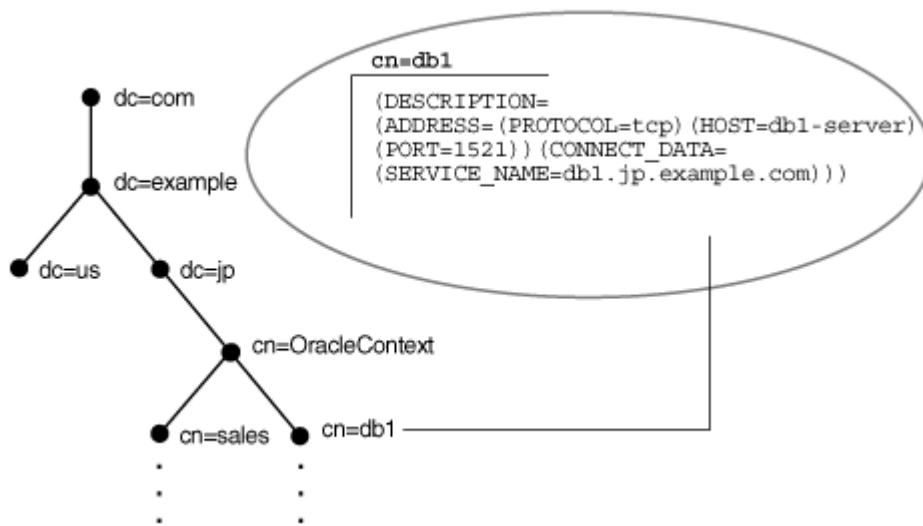
[図3-3](#)では、国、組織および組織単位に従って構築されたDITを示します。この構造は、一般にX.500 DITと呼ばれています。

図3-3 X.500 DIT



たとえば、[図3-4](#)について考えます。cn=salesエントリとcn=db1エントリは、ネットワーク・サービス名とデータベース・サービスをそれぞれ表します。cn=salesとcn=db1の下に存在する別のエントリには、接続記述子情報が含まれます。それらのエントリは、この図では示されていません。cn=salesエントリとcn=db1エントリによって、クライアントは、接続文字列のCONNECT username@salesおよびCONNECT username@db1を使用してデータベースに接続できます。

図3-4 データベース・サービスとDITのネット・サービス・エントリ



各エントリは、[識別名\(DN\)](#)によって一意に識別されます。DNは、ディレクトリ・サーバーの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。db1のDNは、dn:cn=db1,cn=OracleContext,dc=jp,dc=example,dc=comです。salesのDNは、dn:cn=sales,cn=OracleContext,dc=jp,dc=example,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します。

各DNは、[相対識別名\(RDN\)](#)の列で構成されています。RDNは、cnなどの[属性](#)、db1またはsalesなどの値で構成されています。db1のRDNはcn=db1であり、salesのRDNはcn=salesです。属性とその値はエントリを一意に識別します。

- [ドメイン・コンポーネント・ネームスペースの完全修飾名](#)
- [X.500ネームスペースの完全修飾名](#)
- [エントリの相対名の使用](#)
- [エントリの完全修飾名の使用](#)

親トピック: [集中管理用のディレクトリ・サーバーの使用](#)

3.2.1.1 ドメイン・コンポーネント・ネームスペースの完全修飾名

ドメイン・コンポーネント・ネームスペースの場合、クライアントに定義されたデフォルトのディレクトリ・エントリは、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][...]
```

前述の構文で、[dc]はオプションのドメイン・コンポーネントを表し、[...]は追加のドメイン・コンポーネント・エントリを表し、ouは組織単位エントリを表します。

クライアントが接続識別子で使用する完全修飾名は、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][...]
```

前述の構文で、[cn]はOracle Netエントリを表しています。

例3-1 組織単位を使用した完全修飾名の使用

cn=sales,cn=OracleContext,dc=jp,dc=example,dc=comというDNを持つデータベース・オブジェクトsalesのエントリが格納されているディレクトリ・サーバーについて考えます。この例では、クライアントにsales.jp.example.comの接続識別子が必

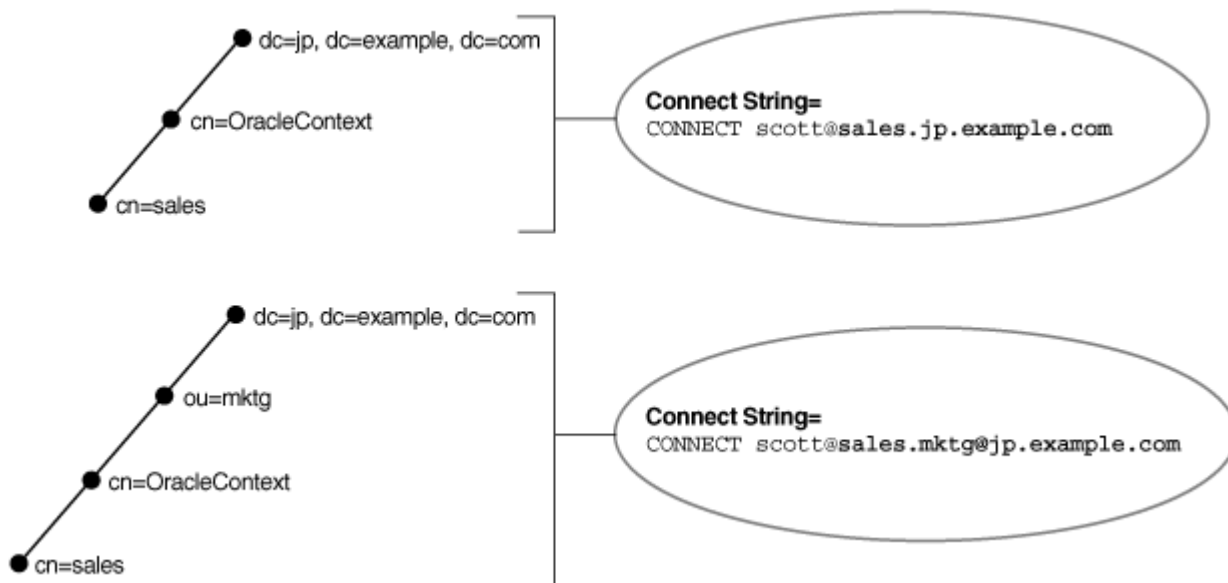
要です。

cn=sales, cn=OracleContext, ou=marketing, dc=jp, dc=example, dc=comというDNを持つデータベース・オブジェクトsalesを含んでいる同様のエントリについて考えます。

ドメイン・コンポーネントは必ず組織単位から分離されている必要があるため、クライアントはcn. ou@dc. dc. dcという形式を使用する必要があります。このクライアントは、接続識別子をsales. mktg@jp. example. comにする必要があります。

[図3-5](#)に前述の例を示します。

図3-5 ドメイン・コンポーネント・ネームスペースの完全修飾名



親トピック: [ディレクトリ情報ツリーの理解](#)

3.2.1.2 X.500ネームスペースの完全修飾名

X.500ネームスペースの場合、クライアントに定義されたデフォルトのディレクトリ・エントリは、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
```

前述の形式で、[ou]はオプションの組織単位名を表し、oは組織を表し、cは国を表します。

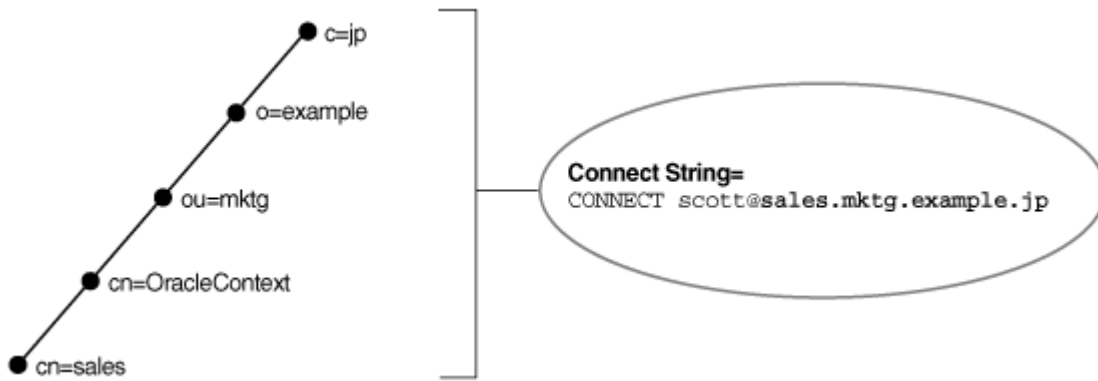
クライアントが接続識別子として使用する完全修飾名は、必ず次のいずれかの形式である必要があります。

```
cn[. ou]. o  
cn[. ou]. o. c
```

前述の形式で、cnはOracle Netエントリを表しています。

たとえば、ディレクトリにcn=sales, cn=OracleContext, ou=marketing, o=example, c=jpというDNを持つデータベース・オブジェクトsalesが格納される場合、クライアントは接続識別子sales. mktg. example. jpを必要とします。[図3-6](#)にこの例を示します。

図3-6 X.500ネームスペースの完全修飾名



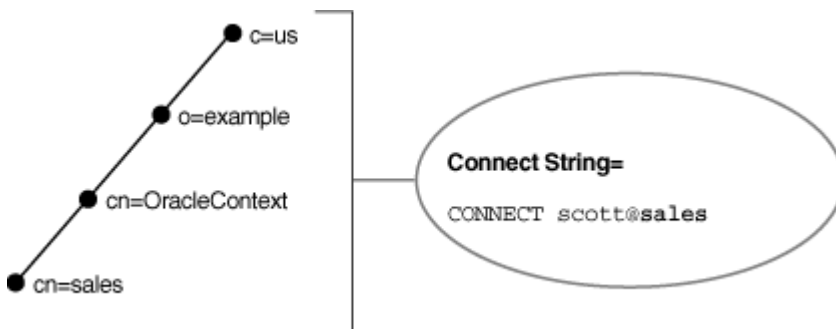
親トピック: [ディレクトリ情報ツリーの理解](#)

3.2.1.3 エントリの相対名の使用

クライアントがデフォルトのレルムOracleコンテキストで構成されている場合、エントリを相対名で識別することが可能であり、サービスを共通名で参照できます。相対名を使用できるのは、クライアントのOracleホームでデフォルトのOracleコンテキストとして構成されたOracleコンテキスト内に、エントリがある場合です。

図3-7に示すように、DNがdn:cn=sales, cn=OracleContext, o=example, c=usで、[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・サーバーについて考えます。クライアントがcn=OracleContext, o=example, c=usというデフォルトのレルムOracleコンテキストで構成されている場合の接続識別子は、salesです。

図3-7 相対ネーミング



ノート:



[JDBC OCI ドライバ](#)は、完全修飾ネーミングと相対ネーミングの両方をサポートしています。[JDBC Thin ドライバ](#)は、完全な DN が使用される場合にのみ、完全修飾ネーミングをサポートします。

関連項目:

- クライアントによるディレクトリの配置方法の詳細は、『[Oracle Fusion Middleware Oracle Internet Directory 管理者ガイド](#)』を参照してください。
- JDBCドライバの詳細は、『[Oracle Database JDBC 開発者ガイド](#)』を参照してください。

親トピック: [ディレクトリ情報ツリーの理解](#)

3.2.1.4 エントリの完全修飾名の使用

[図3-7](#)と同じディレクトリ構造について考えます(ただし、クライアントのOracleホームは、`cn=OracleContext, o=example, c=jp`というデフォルトのレルムOracle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・サーバー内のsalesの位置とは一致しないデフォルトのOracleコンテキストで構成されているため、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次のいずれかの方法を使用して、salesの位置を特定する必要があります。

- エントリの完全なDNを接続文字列に使用します。たとえば:

```
CONNECT username@"cn=sales, cn=OracleContext, o=example, c=us"  
Enter password: password
```

JDBC Thinドライバは、完全なDNが使用される場合にのみ、完全修飾ネーミングをサポートします。ただし、多くのアプリケーションで、DNの使用はサポートされていません。

- エントリは完全修飾DNSスタイル名で参照できます。完全修飾DNSスタイル名は、ディレクトリ・ネーミング・アダプタによってLDAPディレクトリにあるデータベース・オブジェクトの完全x.500 DNにマッピングされます。たとえば:

```
CONNECT username@sales.example.us  
Enter password: password
```

ノート:



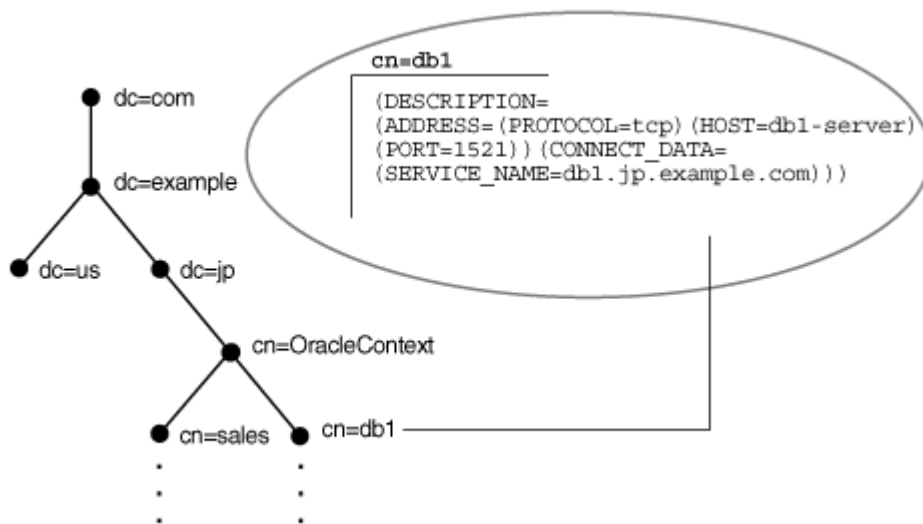
[JDBC OCI ドライバ](#)は、完全修飾ネーミングをサポートしています。[JDBC Thin ドライバ](#)は、完全な DN が使用される場合にのみ、完全修飾ネーミングをサポートします。詳細は、『[Oracle Database JDBC 開発者ガイド](#)』を参照してください。

親トピック: [ディレクトリ情報ツリーの理解](#)

3.2.2 Oracleコンテキストの理解

[図3-8](#)で、エントリ`db1`と`sales`は`cn=OracleContext`の下にあります。このエントリは、[Oracleコンテキスト](#)と呼ばれる特殊なRDNです。Oracleコンテキストの下にあるエントリは、ディレクトリ・ネーミングなど、ディレクトリで利用可能な様々な機能をサポートしています。

図3-8 DITのOracleコンテキスト



ディレクトリの構成時にデフォルトのOracleコンテキストを設定します。クライアントは、このOracleコンテキストをデフォルトの場所として使用し、このディレクトリ・サーバー内で接続識別子を検索します。Oracle Internet Directoryでは、DNが `dn:cn=OracleContext` であれば、DITのルートにあるOracleコンテキストが[アイデンティティ管理レーム](#)にあるデフォルトのOracleコンテキストをポイントします。アイデンティティ管理レームは、同じ管理ポリシーに基づいて集められたアイデンティティです。このOracleコンテキストは、[レームOracleコンテキスト](#)と呼ばれます。別のOracleコンテキストを使用するように設定されていない限り、クライアントはこのレーム固有のOracleコンテキストを使用します。

デフォルトのOracleコンテキストは、接続文字列に影響を与えます。たとえば、クライアントがdb1とsalesのエントリに頻繁にアクセスする必要がある場合、デフォルトとして適切なOracleコンテキストは、dc=jp, dc=example, dc=comなどになります。クライアントのディレクトリ・エントリが、サービスの存在するディレクトリ・エントリと一致しない場合、クライアントは、エントリの完全修飾名を接続文字列で指定する必要があります。詳細は、「[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。

ノート:



接続文字列で、RDN `cn=OracleContext` を明示的に指定する必要はありません。

関連項目:

アイデンティティ管理レームの詳細は、『[Oracle Fusion Middleware Oracle Internet Directory管理者ガイド](#)』を参照してください。

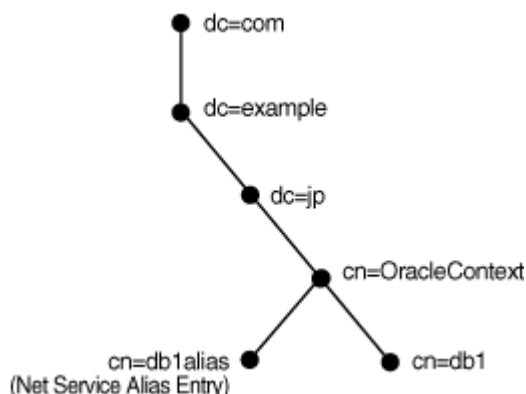
親トピック: [集中管理用のディレクトリ・サーバーの使用](#)

3.2.3 ネット・サービス別名エントリの理解

ディレクトリ・ネーミングでは、データベース・サービスとネットワーク・サービス名のエントリに加えて、[ネットワーク・サービス別名](#)のエントリを作成できます。ネットワーク・サービス別名は、ネットワーク・サービス名またはデータベース・サービスの代替名です。ネットワーク・サービス別名のエントリには、接続記述子情報は含まれていません。ネット・サービス別名が参照するのは、別名のエントリの場所です。クライアントがネットワーク・サービス別名のディレクトリ検索を要求すると、ディレクトリは、エントリがネットワーク・サービス別名であることを判断し、あたかも参照されたエントリであるかのように検索を完了します。たとえば、[図3-9](#)では、ネット

ワーク・サービス別名db1aliasがデータベース・サービスdb1に対して作成されています。db1alias(CONNECT username@db1alias)を使用してデータベース・サービスに接続する場合、実際にはdb1の接続記述子情報に解決されて使用されます。

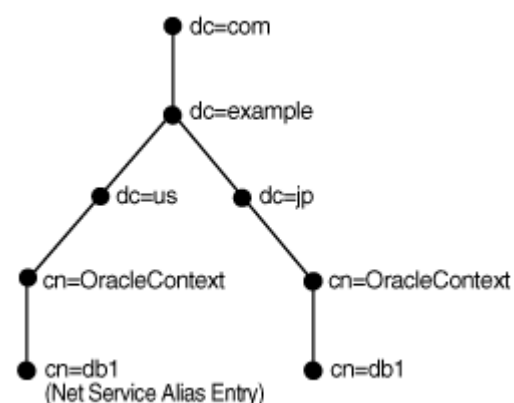
図3-9 ディレクトリ・サーバーでのネット・サービス別名のdb1alias



ネットワーク・サービス別名の使用方法はいくつかあります。図3-9に示すように、ネットワーク・サービス別名は、クライアントがネットワーク・サービス名を別の名前でも参照する方法として役立ちます。もう1つの使用法は、データベース・サービスの1つのOracleコンテキストでネットワーク・サービス別名を使用し、別のOracleコンテキストでネットワーク・サービス名を使用する方法です。これにより、データベース・サービスまたはネットワーク・サービス名を一度ディレクトリ・サーバーに定義できますが、他のOracleコンテキストを使用するクライアントから参照できます。

図3-10では、データベース・サービスdb1はdc=jp, dc=example, dc=comに常駐しています。db1という名前のネットワーク・サービス別名がdc=us, dc=example, dc=com内に作成されます。これによって、日本と米国の両方のクライアントは接続文字列CONNECT username@db1を使用できます。ただし、米国のクライアントはCONNECT username@db1.jp.example.comと指定する必要があります。

図3-10 ディレクトリ・サーバーでのネット・サービス別名のdb1



親トピック: [集中管理用のディレクトリ・サーバーの使用](#)

3.2.4 ディレクトリ・サーバーのエントリを追加または変更できるユーザー

データベース・サービス・エントリは、インストール時またはインストール後に構成されます。次に[Oracle Enterprise Manager Cloud Control](#)または[Oracle Net Manager](#)を使用して、データベース・サービス・エントリのOracle Net属性を変更できます。また、これらのツールを使用すると、ネットワーク・サービス名とネットワーク・サービス別名のエントリを作成できます。

これらの構成ツールを使用するには、[ルートOracleコンテキスト](#)を持つDIT構成とアイデンティティ管理レームが存在する必要があります。ディレクトリ管理者は、Oracle Internet Directoryコンフィギュレーション・アシスタントを使用してこの構成を作成します。デプロイメントによっては、ディレクトリ管理者は、さらに別のOracleコンテキストを作成する必要があります。通常、その他のOracleコンテキストは大規模なサイトの細分化、またはテスト環境からの本番環境の分離に使用されます。

次に示すように、特定のツールは特定のグループによって使用されるため、ツールを使用するにはそのグループのメンバーである必要があります。

- Database Configuration Assistantを使用してデータベース・サービスを作成するには:
 - OracleDBCreatorsグループ(cn=OracleDBCreators, cn=OracleContext...)
 - OracleContextAdminsグループ(cn=OracleContextAdmins, cn=Groups, cn=OracleContext...)
- Oracle Net Managerを使用してネットワーク・サービス名またはネットワーク・サービス別名を作成するには:
 - OracleNetAdminsグループ(cn=OracleNetAdmins, cn=OracleContext...)
 - OracleContextAdminsグループ

OracleNetAdminsグループの所有者はグループ自体です。OracleNetAdminsグループのメンバーには、Oracle Netのオブジェクトと属性に対する作成、変更および読取りのアクセス権限があります。また、グループにメンバーを追加または削除したり、OracleNetAdminsグループの所有者となるグループを追加または削除したりすることもできます。

OracleNetAdminsグループのメンバーは、他のメンバーをOracleNetAdminsグループに追加したり、グループから削除できます。別のグループがOracleNetAdminsのメンバーを追加または削除できるようにする場合は、OracleNetAdminsグループの所有者属性を別のグループに変更します。所有者は、個別のユーザー・エントリではなく、グループ・エントリである必要があります。このグループ・エントリは、LDAPスキーマ・オブジェクト・クラスのGroupOfUniqueNamesとorclPrivilegeGroupで構成されたグループ・エントリです。

OracleContextAdminsグループは、Oracleコンテキストのスーパーユーザーのグループです。OracleContextAdminsグループのメンバーは、サポートされているすべてのタイプのエントリをOracleコンテキストに追加できます。

Oracleコンテキストを作成したディレクトリ・ユーザーは、これらのグループに自動的に追加されます。その他のユーザーは、ディレクトリ管理者によってこれらのグループに追加されます。

関連項目:

- Oracle Net Managerの詳細は、[「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
- Database Configuration Assistantを使用してデータベース・サービスを登録する方法の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。

親トピック: [集中管理用のディレクトリ・サーバーの使用](#)

3.2.5 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

ディレクトリ・サーバーで名前参照を実行する必要がある大半のクライアントは、匿名認証を使用してディレクトリ・サーバーにアクセスします。参照を実行するには、対象のディレクトリ・サーバーで匿名認証が可能である必要があります。通常、ディレクトリ・

サーバーでは、デフォルトで匿名認証が可能ですが、以前のリリースのOracle Internet Directoryなどの一部のディレクトリ・サーバーでは、匿名アクセスが可能になるようにディレクトリの構成が必要です。

エントリを検索するには、クライアントはエントリが常駐するディレクトリ・サーバーを見つける必要があります。クライアントは、次の2つのいずれかの方法でディレクトリ・サーバーを見つけます。

- DNSを使用した動的な方法。この場合、ディレクトリ・サーバーの場所に関する情報は、中央のドメイン・ネーム・サーバーで保存および管理されます。クライアントは要求を処理するときに、DNSからこの情報を取り出します。
- 静的には、Oracle Internet Directoryコンフィギュレーション・アシスタントによって作成され、クライアント・ホストに保存されているディレクトリ・サーバー使用ファイル(ldap.ora)で検出します。

ディレクトリが見つかったら、ルートにあるOracleコンテキストからクライアントは、レルムOracleコンテキストに送られます。

クライアントは、他のネーミング・メソッドを使用する場合と同様に、接続識別子を使用してデータベースに接続します。接続識別子は、データベース・サービス名、ネットワーク・サービス名、ネットワーク・サービス別名のいずれかです。デフォルトのOracleコンテキストがエントリの場所になる場合、これらは共通名(相対名)で参照できます。そうでない場合は、接続識別子に完全修飾名または識別名が必要になります。

関連項目:

- Oracle Fusion Middleware Oracle Internet Directory管理者ガイド
- Microsoft Active Directoryのドキュメント

親トピック: [集中管理用のディレクトリ・サーバーの使用](#)

3.2.6 ディレクトリ・サーバーを使用するときの考慮事項

ディレクトリ・サーバーを使用するときは、次の考慮事項を検討する必要があります。

- [パフォーマンスに関する考慮事項](#)
- [セキュリティに関する考慮事項](#)
- [オブジェクト・クラス](#)

親トピック: [集中管理用のディレクトリ・サーバーの使用](#)

3.2.6.1 パフォーマンスに関する考慮事項

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・サーバーに格納されます。クライアント数によっては、ディレクトリ・サーバーに大量の負荷がかかる場合があります。

接続識別子の参照時には、名前は特定のOracleコンテキストのもとで検索されます。ユーザーは比較的迅速なパフォーマンスで検索を実行できるため、データベースの接続時間は影響を受けません。参照の有効範囲を考慮すれば、参照時間が1秒を超えると、ユーザーは接続時間が長く感じるようになります。

ネットワーク・トポロジを変更したり、レプリケーションを実装すれば、パフォーマンスの問題を解決できます。

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・サーバーのベンダーのマニュアルを参照してください。

親トピック: [ディレクトリ・サーバーを使用するときの考慮事項](#)

3.2.6.2 セキュリティに関する考慮事項

管理クライアントはディレクトリ・サーバー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。この項では、セキュリティに関連する項目を取り上げます。

- [認証方法](#)

クライアントは、接続文字列名の解決やディレクトリ・エントリの管理など、実行されるタスクに応じて異なる認証方式を使用します。

- [アクセス制御リスト](#)

親トピック: [ディレクトリ・サーバーを使用するときの考慮事項](#)

3.2.6.2.1 認証方式

クライアントは、接続文字列名の解決やディレクトリ・エントリの管理など、実行されるタスクに応じて異なる認証方式を使用します。

- ディレクトリ・サーバー内の情報について参照を実行するクライアントは、通常、匿名認証を使用します。名前参照時にLDAPバインドを認証するようにクライアントを構成できます。ネットワーク・サービス・データを保護するか、またはディレクトリへの匿名バインドを無効にする必要があるサイトは、名前参照時の認証にウォレットを使用するようにクライアントを構成する必要があります。

これらのクライアントについて、sqlnet.oraファイルで次のパラメータを設定する必要があります。

```
names.ldap_authenticate_bind = TRUE
wallet_location = location_value
```

- ディレクトリ・サーバー・エントリを管理するクライアントは、ディレクトリ・サーバーで認証を実行します。Database Configuration AssistantやOracle Net Managerを使用すると、エントリの追加や修正が可能です。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。次のいずれかの認証方式を使用します。

- **簡易認証**

クライアントは、DNおよびパスワードを使用して、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・サーバーに格納されているDNおよびパスワードと一致していることを確認します。

- **厳密認証**

クライアントは、Transport Layer (TLS)で使用可能な公開キー暗号によって、ディレクトリ・サーバーに対して自分自身を識別します。公開キー暗号では、メッセージの送信側が受信側の公開キーを使用してメッセージを暗号化します。メッセージが送達されると、受信側は、受信側の秘密キーを使用して、メッセージを復号化します。

関連項目

- [TLSプロトコル付きTCP/IPについて](#)

3.2.6.2.2 アクセス制御リスト

クライアントがディレクトリ・サーバー内の情報に対して読取り、変更または追加をできる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。
- エントリへのアクセスに使用する認証方式。
- 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(読取り、書き込みなど)。

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、OracleDBCreators、OracleNetAdminsおよびOracleContextAdminsの各グループが作成されます。

Oracle Net 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。

[表3-2](#)では、これらのグループ、匿名ユーザー、およびディレクトリ・サーバー内のOracle Netエントリに対する関連内容のACL要件に応じて説明します。

表3-2 ユーザー・グループのACL要件

グループ	ACL要件
匿名ユーザー	匿名ユーザーは、ディレクトリ・サーバー内のすべての Oracle Net 属性とオブジェクトへの読取りアクセス権を持ちます。匿名ユーザーのこれらのオブジェクトに対する読取りアクセスは、Oracle コンテキストにも適用されます。これにより匿名ユーザーは、cn=OracleContext という RDN 内に含まれるディレクトリ・ネーミング・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。 Oracle Net Configuration Assistant は、クライアントのインストール時にこのアクセスをセットアップします。
OracleContextAdmins グループ・ユーザー	OracleContextAdmins(cn=OracleContextAdmins, cn=Groups, cn=OracleContext, ...)のメンバーには、すべてのディレクトリ・ネーミング・オブジェクトに対する作成、変更および読取りのアクセス権限があります。Oracle Net 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。 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 Manager Cloud Control を使用してこのグループに追加できます。
OracleDBCreators グループ・ユーザー	OracleDBCreators(cn=OracleDBCreators, cn=OracleContext, ...)のメンバーには、データベース・サービスのオブジェクトと属性に対する作成と読取りのアクセス権限があ

グループ	ACL要件
	<p>ります。Oracle Net 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 Manager Cloud Control を使用してこのグループに追加できます。</p>
OracleNetAdmins グループ・ユーザー	<p>OracleNetAdmins(cn=OracleNetAdmins, cn=OracleContext, ...)のメンバーには、ディレクトリ・ネーミングのオブジェクトと属性に対する作成、変更および読取りのアクセス権限があります。Oracle Net 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>

Oracle Net Services名および関連データに対する参照または読取りアクセスについて高度なセキュリティが必要とされる場合、管理者はデータの一部またはすべてについて追加の読取りアクセス制御を定義できます。こうしたACL定義を使用して、匿名ユーザーによるOracle Net Servicesデータの読取りを禁止できます。Oracle Net Servicesデータに対する読取りアクセスが制限されている場合、クライアントは名前参照を実行する際に認証済バインドを使用する必要があります。

Oracleの構成ツールには、このデータに対する読取りアクセス制限を定義するための事前定義済グループまたはプロシージャがないため、管理者は、ディレクトリ・システムの標準オブジェクト管理ツールを使用して、必要なグループおよびACLを手動で作成する必要があります。

ldapmodifyおよびLDIF形式ファイルを使用して、ACLをOracle Net Servicesオブジェクトに追加できます。[例3-2](#)に、ユーザーcn=user1のすべてのアクセスを制限する方法を示します。

例3-2 アクセス制御リストを使用したユーザー・アクセスの制限

```
dn: cn=sales, cn=oraclecontext, dc=example, dc=com
replace: orclentrylevelaci
orclentrylevelaci: access to attr=(*)
by dn="cn=user1" (noread, nosearch, nowrite, nocompare)
```

前述の例は、単一オブジェクトのACLの基本的な形式を示しています。この方法は、必ずしもアクセスを定義するための最適な方法ではありません。オブジェクトのアクセス定義は複雑であり、DITの親ノードから継承されるセキュリティ・プロパティに影響する場合がありますためです。管理者はディレクトリ・システムのマニュアルを参照し、Oracle Net Servicesオブジェクトのアクセス管理をディレクトリ全体のポリシーおよびセキュリティ実装に統合することをお勧めします。

Oracle Internet Directoryディレクトリの場合、oidadminには、ユーザーおよびグループを作成し、オブジェクトに対するACLおよび一般的なディレクトリ・セキュリティを定義する機能があります。

関連項目:

- ディレクトリ・エントリにおけるACLの設定方法については、ディレクトリ・サーバーのベンダーのマニュアルを参照してください。
- 名前参照に認証済バインドを使用するようクライアントを構成する方法については、[「認証方式」](#)を参照してください
- ユーザーをOracleNetAdminsグループに追加する方法については、[「OracleNetAdminsグループについて」](#)を参照してください。
- [『Oracle Fusion Middleware Oracle Internet Directory管理者ガイド』](#)

親トピック: [セキュリティに関する考慮事項](#)

3.2.6.3 オブジェクト・クラス

Oracleコンテキスト、データベース・サービスまたはネットワーク・サービス名のエントリを作成する前に、ディレクトリが適切なバージョンのOracleスキーマとともに移入されている必要があります。Oracleスキーマは、[オブジェクト・クラス](#)と呼ばれるオブジェクト・タイプを定義します。これは、ディレクトリ・サーバーおよびその属性に格納できます。[表3-3](#)は、データベース・サービス・エントリ、ネットワーク・サービス名エントリおよびネットワーク・サービス別名エントリのオブジェクト・クラスを示しています。

表3-3 Oracle Net Services LDAPの主要なオブジェクト・クラス

Object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	ネットワーク・サービス名エントリの属性を定義します。
orclNetServiceAlias	ネットワーク・サービス別名エントリの属性を定義します。

[表3-4](#)に、orclDbServer、orclNetServiceおよびorclNetServiceAliasが使用するオブジェクト・クラスを示します。

表3-4 Oracle Net Services LDAPの導出オブジェクト・クラス

Object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。
orclNetAddressList	アドレスのリストを定義します。
orclNetDescription	データベースのプロトコル・アドレスおよびサービスに対する接続情報が含まれた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。

これらのオブジェクト・クラスは、接続記述子の内容を指定する属性を使用します。

関連項目:

オブジェクト・クラスとその属性に関する詳細は、『[Oracle Database Net Servicesリファレンス](#)』を参照してください。

親トピック: [ディレクトリ・サーバーを使用するときの考慮事項](#)

3.2.7 Microsoft Active Directoryにおけるディレクトリ・ネーミング・サポートの制約

Oracle Internet Directoryのみではなく、ディレクトリ・ネーミング・サポートも、[Microsoft Active Directory](#)で提供されていますが、次の制約があります。

- Microsoft Active Directoryは、Microsoft Windowsオペレーティング・システム上でのみサポートしています。したがって、Microsoft Active Directoryのエントリへのアクセスまたはエントリの作成を行うには、クライアント・コンピュータとデータベース・サーバーもMicrosoft Windowsオペレーティング・システム上で実行する必要があります。
- 次の機能は、Microsoft Active Directoryではサポートされていません。
 - 複数のOracleコンテキスト
Microsoft Active DirectoryがサポートするOracleコンテキストは1つです。
 - ネット・サービス別名
Microsoft Active Directoryではネットワーク・サービス別名を作成できません。ただし、ネットワーク・サービス名の作成は可能です。

関連項目:

Microsoft固有の情報については、Microsoft Active Directoryのマニュアルを参照してください。

親トピック: [集中管理用のディレクトリ・サーバーの使用](#)

4 通信レイヤーの理解

Oracle Netの主要な機能は、クライアント・アプリケーションとOracle Databaseサーバー間の接続を確立して維持することです。Oracle Netは複数の通信レイヤーで構成されており、クライアントとデータベース・サーバーはデータの共有、変更、操作が可能です。

- [クライアント/サーバー・アプリケーションのOracle Netスタック通信の理解](#)
- [JavaアプリケーションのOracle Netスタック通信の使用](#)
- [WebクライアントのOracle Netスタック通信の使用](#)
- [Oracle Protocol Support Layerの理解](#)

ネットワーク・プロトコルは、クライアント・コンピュータからデータベース・サーバー・コンピュータまで、データを送る役割があります。この項では、Oracle Net通信スタックのOracle Protocol Supportレイヤーで使用されるプロトコルについて説明します。

関連項目:

Oracle Netアーキテクチャの概要は、[「Oracle Net Servicesの概要」](#)を参照してください

親トピック: [Oracle Net Servicesの理解](#)

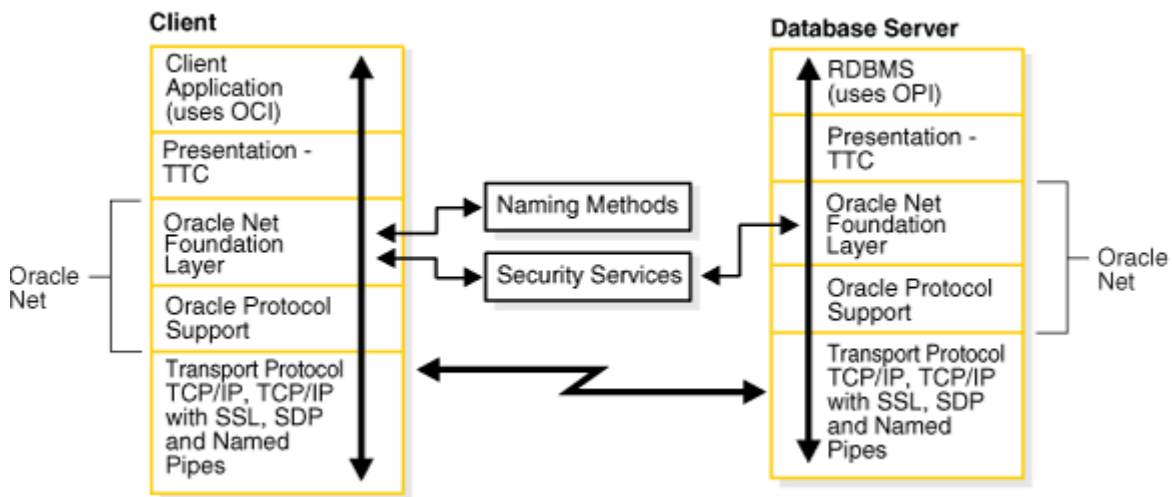
4.1 クライアント/サーバー・アプリケーションのOracle Netスタック通信の理解

データベース・サーバーとはデータベースを管理するOracleソフトウェアのことで、クライアントとはサーバーの情報を要求するアプリケーションのことです。クライアントとサーバーが通信する方法をクライアント/サーバー・スタックと呼びます。

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(クライアント通信スタックによって送信されたもの)は、データベース・サーバー側にある同様の通信スタックで受信されます。データベース・サーバー側でのプロセス・フローは、クライアント側でのプロセス・フローの逆になり、情報は通信レイヤーを通過してさかのぼります。

[図4-1](#)では、接続が確立された後のクライアントとデータベース・サーバーの様々なレイヤーを示します。

図4-1 クライアント/サーバー・アプリケーション接続で使用するレイヤー

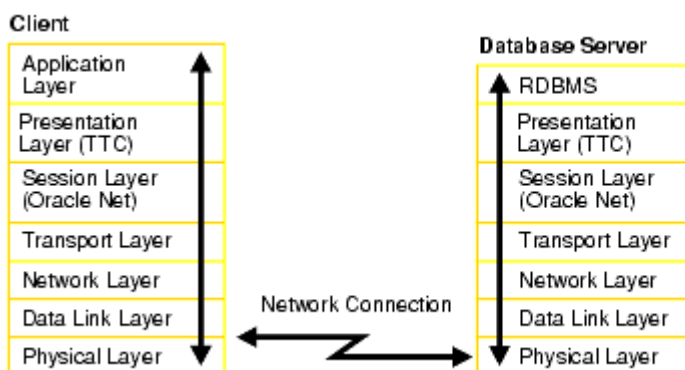


この通信アーキテクチャは、[Open Systems Interconnection \(OSI\)](#)モデルに基づいています。OSIモデルでは、いくつかのコード・レイヤーを介して一方のノードから他方のノードへ情報を渡す、スタックのような形式でコンピュータ間の通信が行われます。次のレイヤーがあります。

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

[図4-2](#)は、Oracle Net FoundationレイヤーとOracle Protocol Supportで構成されるOracle Netソフトウェアが、OSIモデルのセッション・レイヤーにどのように適合しているかを示しています。

図4-2 OSI通信レイヤー



関連項目:

OSIスタックの詳細は、次の場所を参照してください。

<http://www.ietf.org>

この項では、次の項目について説明します。

- [クライアント通信スタックについて](#)
- [サーバー通信スタックについて](#)

親トピック: [通信レイヤーの理解](#)

4.1.1 クライアント通信スタックについて

クライアント通信スタックには次のものが含まれています。

- [クライアント・アプリケーション・レイヤー](#)
- [プレゼンテーション・レイヤー](#)
- [Oracle Net Foundationレイヤー](#)
- [Oracle protocol supportレイヤー](#)

親トピック: [クライアント/サーバー・アプリケーションのOracle Netスタック通信の理解](#)

4.1.1.1 クライアント・アプリケーション・レイヤー

データベースとのセッション時、クライアントは[Oracle Call Interface\(OCI\)](#)を使用して、データベース・サーバーとの対話型操作を実行します。OCIは、アプリケーションとSQL間のインタフェースを提供するソフトウェア・コンポーネントです。

関連項目:

[『Oracle Call Interfaceプログラマーズ・ガイド』](#)

親トピック: [クライアント通信スタックについて](#)

4.1.1.2 プレゼンテーション・レイヤー

クライアントとデータベース・サーバーが異なるオペレーティング・システムで実行されている場合、文字セットの相違が発生します。プレゼンテーション・レイヤーによって、すべての相違が解決されます。接続ごとに最適化され、必要に応じて変換が実行されません。

クライアント/サーバー・アプリケーションが使用するプレゼンテーション・レイヤーは、[Two-Task Common\(TTC\)](#)です。TTCは、クライアントとデータベース・サーバー間の文字セットの相違または形式の相違に対して、文字セットとデータ型の変換を行います。初期の接続時に、TTCは内部データと文字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。

親トピック: [クライアント通信スタックについて](#)

4.1.1.3 Oracle Net Foundationレイヤー

Oracle Net Foundationレイヤーは、クライアント・アプリケーションとデータベース・サーバー間でのメッセージの交換に加え、これらの間の接続を確立および維持します。Oracle Net Foundationレイヤーは、[Transparent Network Substrate\(TNS\)](#)テクノロジーによってこれらのタスクを実行できます。TNSは、すべての業界標準OSIトランスポート・プロトコルおよびネットワーク・レイヤー・プロトコルに単一の共通インタフェースを提供します。TNSによって、ピアツーピア・アプリケーション接続が可能に

なります。複数のコンピュータが相互に直接通信でき、中間デバイスは必要ありません。

クライアント側で、Oracle Net Foundationレイヤーは、クライアント・アプリケーション要求を受け取り、すべての一般的なコンピュータ・レベルの接続性の問題を解決します。次のような問題があります。

- データベース・サーバーや接続先の場所
- 接続に含まれるプロトコルの数
- それぞれの機能に基づいた、クライアントとデータベース・サーバー間の中断の処理方法

サーバー側で、Oracle Net Foundationレイヤーはクライアント側と同じタスクを実行します。また、リスナーとともに機能して着信接続要求を受信します。

接続の確立と維持に加えて、Oracle Net Foundationレイヤーは、ネーミング・メソッドを使用して通信し名前を解決します。また、セキュリティ・サービスを使用して安全な接続を実現します。

親トピック: [クライアント通信スタックについて](#)

4.1.1.4 Oracle protocol supportレイヤー

Oracle protocol supportレイヤーは、Oracle Net Foundationレイヤーの最下位に位置します。これは、クライアント/サーバー接続で使用される業界標準のプロトコルにTNS機能をマッピングする役割を担います。このレイヤーは次のネットワーク・プロトコルをサポートしています。

- TCP/IP([IPv4](#)および[IPv6](#))
- Transport Layer Security (TLS)付きTCP/IP
- 名前付きパイプ
- Sockets Direct Protocol(SDP)
- Exadirect

クライアント/サーバー接続の際、すべてのOracleソフトウェアは、トランスポート・レイヤーの2台のコンピュータ間のコンピュータ・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・スタックが必要です。ネットワーク・プロトコルは、クライアント・コンピュータからデータベース・サーバー・コンピュータまで、データを送る役割があります。ここで、データはサーバー側のOracle protocol supportレイヤーに渡されます。

関連項目

- [Oracle protocol supportレイヤーの理解](#)

親トピック: [クライアント通信スタックについて](#)

4.1.2 サーバー通信スタックについて

サーバー通信スタックはクライアント・スタックと同じレイヤーを使用しますが、例外としてデータベースは[Oracleプログラム・インタフェース\(OPI\)](#)を使用します。OCIから送信される各文に対して、OPIは応答します。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。

親トピック: [クライアント/サーバー・アプリケーションのOracle Netスタック通信の理解](#)

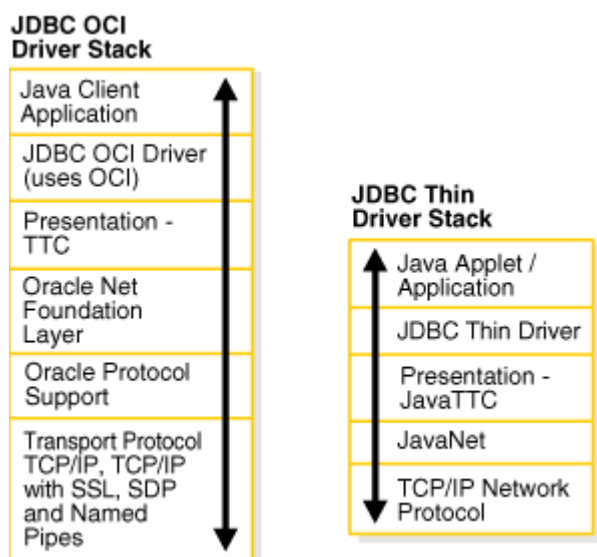
4.2 JavaアプリケーションのOracle Netスタック通信の使用

Oracle [Java Database Connectivity\(JDBC\)ドライバ](#)により、JavaアプリケーションはOracle Databaseにアクセスします。Oracleには、2つのJDBCドライバが用意されています。

- [JDBC OCIドライバ](#)は、クライアント/サーバーのJavaアプリケーションによって使用されるタイプ2のJDBCドライバです。JDBC OCIドライバは、標準のクライアント/サーバー通信スタックと同様の通信スタックを使用します。JDBC OCIドライバは、JDBC起動をOCIの呼出しに変換します。変換された呼出しはOracle Netを介してOracle Databaseサーバーに送信されます。
- [JDBC Thinドライバ](#)は、Javaアプレットが使用するタイプ4のドライバです。JDBC Thinドライバは、Javaソケットを通してOracle Databaseサーバーに直接接続を確立します。JDBC Thinドライバは、JavaNetと呼ばれるOracle Net FoundationレイヤーのJava実装と、JavaTTCと呼ばれるTTCのJava実装を使用してデータベースにアクセスします。

次の図は、JDBCドライバが使用するスタック通信レイヤーを示しています。

図4-3 Javaクライアント・アプリケーションで使用するレイヤー



関連項目:

[『Oracle Database JDBC開発者ガイド』](#)

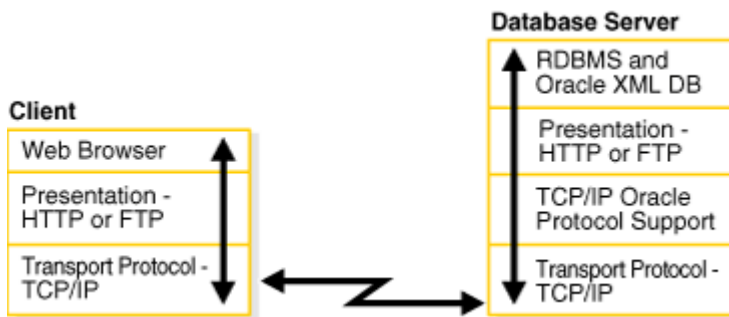
親トピック: [通信レイヤーの理解](#)

4.3 WebクライアントのOracle Netスタック通信の使用

Oracle Databaseサーバーは、Webクライアントがデータベース内の機能にアクセスするためにTTC以外に使用できる、その他の多数の実装をプレゼンテーション・レイヤーに対してサポートしています。リスナーは、これを容易にするため、データベースで要求されるあらゆるプレゼンテーションの実装をサポートします。

[図4-4](#)は、Oracle Databaseインスタンスの[Oracle XML DB](#)へのHTTP接続またはFTP接続で使用するスタック通信レイヤーを示しています。WebDAV接続では、HTTPおよびFTPと同じスタック通信レイヤーが使用されます。

図4-4 Webクライアント接続に使用するレイヤー



関連項目:

[『Oracle XML DB開発者ガイド』](#)

親トピック: [通信レイヤーの理解](#)

4.4 Oracle protocol supportレイヤーの理解

ネットワーク・プロトコルは、クライアント・コンピュータからデータベース・サーバー・コンピュータまで、データを送る役割があります。この項では、Oracle Net通信スタックのOracle Protocol Supportレイヤーで使用されるプロトコルについて説明します。

- [TCP/IPプロトコルについて](#)
TCP/IP (Transmission Control Protocol/Internet Protocol)は、ネットワークを介したクライアント/サーバー通信に使用される標準的な通信プロトコル・スイートです。
- [TLSプロトコル付きTCP/IPについて](#)
Transport Layer Security (TLS)プロトコル付きTCP/IPを使用すると、クライアント上のOracleアプリケーションがTCP/IPとTLSを介してリモートのデータベースと通信できます。
- [Named Pipesプロトコルについて](#)
Named Pipesプロトコルは、分散アプリケーションを使用したクライアントとデータベース・サーバー間で、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。
- [Sockets Direct Protocol \(SDP\)について](#)
Sockets Direct Protocol (SDP)は、InfiniBandネットワーク・ピア間の業界標準のワイア・プロトコルです。InfiniBandネットワークでSDPを使用すると、データの中間的なレプリケーションが除去され、メッセージ交換の負荷がCPUからネットワーク・ハードウェアへ移動することにより、TCP/IPのオーバーヘッドが削減されます。
- [Exadirectプロトコルについて](#)
Exadirectプロトコルは、データベース・アクセスのオーバーヘッドが少ない、革新的なプロトコルです。インフィニバンド環境では、Remote Direct Memory Access (RDMA)の活用によって待機時間とスループットが向上する新しいトランスポートを使用してください。
- [Websocketプロトコルについて](#)
データベース・クライアント接続では、セキュアなWebSocketプロトコルがサポートされます。安全なWebソケット接続の確立は、HTTPS,で動作し、HTTPSプロキシおよび仲介プロキシをサポートするように設計されます。

親トピック: [通信レイヤーの理解](#)

4.4.1 TCP/IPプロトコルについて

TCP/IP (Transmission Control Protocol/Internet Protocol)は、ネットワークを介したクライアント/サーバー通信に使用される標準的な通信プロトコル・スイートです。

TCPは、ホスト間のデータ交換を管理するトランスポート・プロトコルです。IPは、パケット切替えネットワークのためのネットワーク・レイヤー・プロトコルです。

Oracle Netでは、IP version 4 (IPv4)およびIP version 6 (IPv6)の2つのバージョンのIPをサポートしています。IPv6は、現在使用しているIPv4の短所に対応しています。IPv6の主な利点として、128ビット・アドレスの使用から導出される大容量のアドレス空間があります。

- [IPv6アドレスの表記](#)
- [IPv6インタフェースとアドレス構成](#)
- [IPv6のネットワーク接続](#)
- [Oracle Database 12cでのIPv6のサポート](#)

関連項目:

IPv6の仕様は、<http://tools.ietf.org/html/rfc2460>を参照してください。

親トピック: [Oracle protocol supportレイヤーの理解](#)

4.4.1.1 IPv6アドレスの表記

Oracle Databaseでは、RFC 2732で指定されている標準のIPv6アドレス表記をサポートしています。通常、128ビットのIPアドレスは、8つのグループに分かれた4桁の16進数で表され、グループ・セパレータとしてコロン(:)が使用されます。たとえば、次のアドレスは有効なIPv6形式です。

```
2001:0db8:0000:0000:0000:0000:200C:417A
```

アドレスの各16進数は4ビットを表すため、アドレスの各グループは16ビットを表します。次のアドレスは、2001:0db8:0000:0000サブネットの最初と最後のホストを表します。

```
2001:0db8:0000:0000:0000:0000:0000:0000  
2001:0db8:0000:0000:FFFF:FFFF:FFFF:FFFF
```

簡易表記では、連続するゼロのフィールドを2つのコロン(::)セパレータで圧縮できます。次のアドレスは同等の表記になります。

```
2001:0db8:0:0::200C:417A  
2001:0db8::200C:417A  
2001:DB8::200C:417A
```

- [CIDR表記](#)
- [URLのIPv6アドレス](#)
- [IPv4マップ・アドレス](#)

関連項目:

- RFC 2732および表記の表現については、<http://www.ietf.org/rfc/rfc2732.txt>を参照してください。
- RFC 3513および正しいIPv6アドレス指定については、<http://www.ietf.org/rfc/rfc3513.txt>を参照してください。

親トピック: [TCP/IPプロトコルについて](#)

4.4.1.1.1 CIDR表記

Classless Inter-Domain Routing(CIDR)は、IPアドレスをアドレスの値に関係ないサブネットにグループ化する方法です。クラスレス・ルーティングは、IPクラス・システムのアドレス空間の枯渇およびルーティング表の急増を解決するために設計されました。

CIDRは、ネットワーク内の最初のアドレスと、ネットワーク接頭辞のサイズ(10進数で表したビット数)をスラッシュ(/)で区切ることでネットワークを表します。たとえば、2001:0db8::/32は、アドレスの最初の32ビットがネットワークを特定し、残りのビットがネットワーク内のホストを特定することを示します。

CIDRはIPv4アドレスの類似した表記を使用します。たとえば、192.0.2.0/24という表記では、アドレスの最初の24ビットはネットワーク接頭辞を表します。DBMS_NETWORK_ACL_ADMINパッケージは、アクセス制御リストを管理するためのAPIを提供し、IPv6とIPv4の両方のアドレスおよびサブネットのCIDR表記をサポートします。

関連項目:

- DBMS_NETWORK_ACL_ADMINについて学習するには、『[Oracle Database PL/SQLパッケージ・プロシージャおよびタスク・リファレンス](#)』を参照してください。
- RFC 4632については、<http://tools.ietf.org/html/rfc4632>を参照してください。

親トピック: [IPv6アドレスの表記](#)

4.4.1.1.2 URLのIPv6アドレス

URLでは、IPv6アドレスは左括弧([)および右括弧(])文字で囲まれています。たとえば、IPv6アドレス [2001:0db8:0:0:8:800:200C:417A] は、次のURLの一部を形成します。

```
http://[2001:0db8:0:0:8:800:200C:417A]
http://[2001:0db8:0:0:8:800:200C:417A]:80/index.html
```

親トピック: [IPv6アドレスの表記](#)

4.4.1.1.3 IPv4マップ・アドレス

IPv4マップ・アドレスは、次の条件が真の場合のIPv6アドレスのサブクラスです。

- 最初の80ビットは、標準のIPv6表記では0に設定されます。
- 次の16ビットは、標準のIPv6表記では1に設定されます。
- 最後の32ビットはIPv4表記で表されます。

IPv4マップ・アドレスは、IPv4専用ノードのアドレスをIPv6アドレスとして表すことができます。

[例4-1](#)は、同じIPアドレスを異なる表記で示しています。最初のアドレスは標準のIPv6表記を使用しています。2番目のアドレスは、最後の32ビットがピリオドで区切ったIPv4表記を使用しているIPv4マップ・アドレスです。最後のアドレスは、連続するゼロのフィールドを圧縮する簡易表記を使用しています。

例4-1 IPv4マップ・アドレス

```
0000:0000:0000:0000:0000:FFFF:COA8:0226
0000:0000:0000:0000:0000:FFFF:192.0.2.38
::FFFF:192.0.2.38
```

関連項目:

IPv4マップ・アドレスの使用に関連するセキュリティの考慮事項については、<http://tools.ietf.org/html/rfc4942>を参照してください。

親トピック: [IPv6アドレスの表記](#)

4.4.1.2 IPv6インタフェースとアドレス構成

1つのホストで、IPv4とIPv6の2つのインタフェース構成を使用できます。ホストで使用できる構成は次のとおりです。

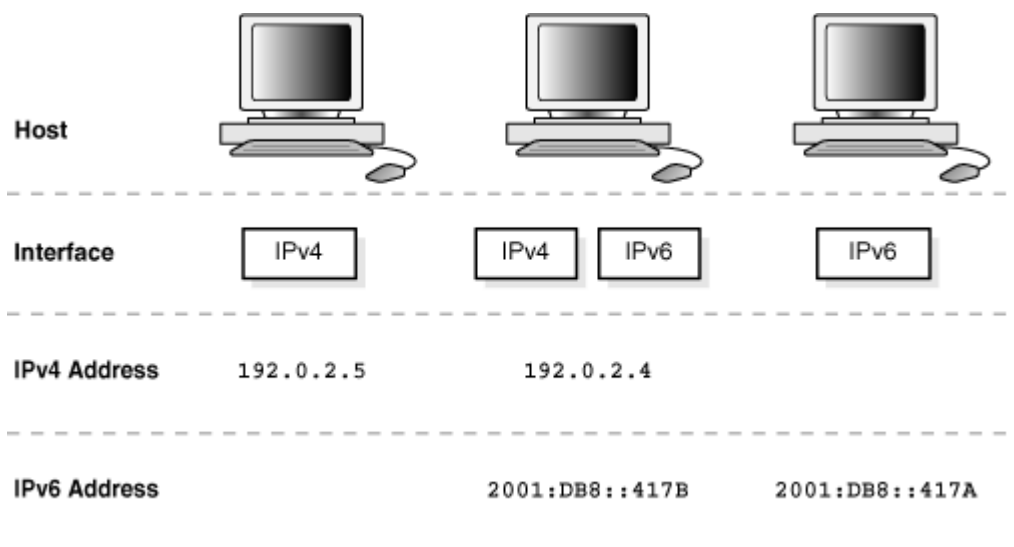
- IPv4インタフェースのみ。この場合、ホストはIPv4専用ホストです。
- IPv6インタフェースのみ。この場合、ホストはIPv6専用ホストです。
- IPv4インタフェースとIPv6インタフェースの両方。この場合、ホストはデュアル・スタック・ホストです。

1つのホストで異なるタイプのIPアドレスを使用することもできます。たとえば、ドメイン名サーバーは、デュアル・スタック・ホストをIPv4アドレスとIPv6アドレスの両方、またはIPv6アドレスのみによって関連付けることができます。サポートされていないIPアドレス構成は次のとおりです。

- IPv4専用ホストはIPv6アドレスを使用できません。
- IPv6専用ホストはIPv4アドレスを使用できません。

[図4-5](#)は、使用できるホストとインタフェース構成を示しています。図の中央のデュアル・スタック・ホストは、IPv4を通じてIPv4ホストと通信し、IPv6を通じてIPv6ホストと通信できます。

図4-5 サポートされているホストおよびインタフェース構成



親トピック: [TCP/IPプロトコルについて](#)

4.4.1.3 IPv6のネットワーク接続

ホストのネットワーク接続とは、ネットワークを通じて別のホストと通信できることを意味します。たとえば、デュアル・スタック・クライアントがIPv6専用サーバーと通信する必要がある場合、ネットワークとルーターはこれらのホスト間のエンドツーエンド通信を可能にする必要があります。

クライアントまたはサーバー・ホストは次の条件を満たす場合、IPv6に対応します。

- 構成済IPv6インタフェースがある。
- IPv6プロトコルを使用して他のホストに接続できる。

ホストのIPv6機能は、一部はネットワーク、一部はインタフェースとアドレス構成に依存します。[図4-6](#)は、クライアント/サーバー・ネットワークにおける接続の可能性を示しています。たとえば、IPv4専用ホストはIPv4専用サーバーまたはデュアル・スタック・サーバーに接続できますが、IPv6専用サーバーには接続できません。専用サーバー・モードと共有サーバー・モードの両方がサポートされています。

図4-6 クライアント/サーバー接続

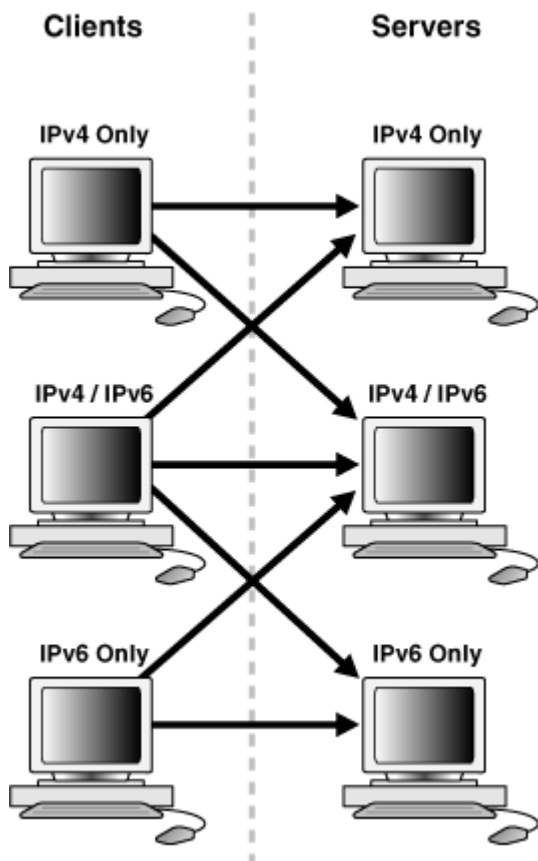


表4-1は、ホストとネットワーク構成が異なるクライアント/サーバー接続に使用されるIPプロトコルをまとめたものです。

表4-1 サポートされているホストおよびネットワーク構成

クライアント	IPv4専用サーバー	デュアル・スタック・サーバー	IPv6専用サーバー
IPv4 専用クライアント	サポート対象(v4)	サポート対象(v4)	サポート対象外
デュアル・スタック・クライアント	サポート対象(v4)	サポート対象(v4、v6)	サポート対象(v6)
IPv6 専用クライアント	サポート対象外	サポート対象(v6)	サポート対象

関連項目:

- [「接続記述子のIPv6アドレスについて」](#)
- [「リスニング用プロトコル・アドレスの構成」](#)
- [「IPv4およびIPv6のブリッジとしてのOracle Connection Managerの使用」](#)

親トピック: [TCP/IPプロトコルについて](#)

4.4.1.4 Oracle Database 12cでのIPv6のサポート

Oracle Database 12cのこのリリースのコンポーネントは、[「IPv6のネットワーク接続」](#)で説明した構成でIPv6をサポートしていますが、次の例外があります。

- プライベート・ネットワークおよびASMネットワークに対するOracle Clusterware

関連項目:

[『Oracle Clusterware管理およびデプロイメント・ガイド』](#)

親トピック: [TCP/IPプロトコルについて](#)

4.4.2 TLSプロトコル付きTCP/IPについて

Transport Layer Security (TLS)プロトコル付きTCP/IPを使用すると、クライアント上のOracleアプリケーションがTCP/IPとTLSを介してリモートのデータベースと通信できます。

Oracle Databaseは、相互TLS (mTLS)およびTLS認証をサポートします。

相互TLS (mTLS)認証

Mutual Transport Layer Security (mTLS)または双方向認証では、両方のノード上のOracleウォレットの証明書や秘密キーなどの暗号化データを使用して、データベース・サーバーとデータベース・クライアント間の暗号化されたセキュア・リンクを構成します。クライアントがデータベース・サーバーへの接続を開始すると、mTLSは、ウォレットに格納されている証明書を使用してサーバーとクライアントの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。

- クライアントとデータベース・サーバーは、交換するメッセージに適用する暗号スイート(認証、暗号化およびデータ整合性タイプのセットで構成)についてネゴシエーションします。
- mTLSでは、データベース・サーバーとクライアントが証明書を交換して、両者を相互に認証します。証明書の信頼性は、両方のパーティによる信頼のルートに対してチェックされます。
- クライアントとデータベース・サーバーは、通信チャネルの暗号化に使用される対称暗号化キーに同意します。

データベースは、ユーザー証明書をチェックして認証局(CA)の署名があることを確認します。

TLS認証

クライアント認証を必要としない場合(Webサーバーにのみ証明書が必要なHTTPS接続と同様)、TLSまたは一方向認証を構成できます。この場合、データベース・サーバーのみがサーバー証明書を提示することでクライアントに対して認証を実行し、クライアントはデータベース・サーバー証明書が有効かどうかを検証します。これにより、データベース・サーバーおよびクライアントは、メッセージを交換する前に暗号化された接続を確立できます。

クライアントは、次のいずれかの方法でデータベース・サーバーのIDを認識および認証します。

- クライアント・ウォレットを使用したTLS認証:
TLSは、信頼できるCA証明書や秘密キーなどの認証データをOracleウォレットに格納します。サーバーがクライアントに証明書を送信すると、クライアントはウォレットに格納されている信頼できるルート証明書を使用して証明書を検証します。
- クライアント・ウォレットを使用しないTLS認証:

データベース・サーバー証明書が、クライアント・システムのデフォルト証明書ストアにすでにインストールされている信頼できるルート証明書によって署名されている場合は、サーバー証明書を含むOracleクライアント・ウォレットは必要ありません。たとえば、Linuxでは/etc/pki/tls/cert.pemです。データベース・サーバーがクライアントに証明書を送信すると、クライアントはデフォルト証明書ストアからの共通ルート証明書を使用してサーバー証明書を検証します。

この方法では、クライアントが信頼できる証明書を使用して独自のウォレットを構成する必要がないため、クライアント構成およびデータベース・クライアント/サーバー通信プロセスが簡略化されます。

関連項目

- [『Oracle Databaseセキュリティ・ガイド』](#)

親トピック: [Oracle protocol supportレイヤーの理解](#)

4.4.3 Named Pipesプロトコルについて

Named Pipesプロトコルは、分散アプリケーションを使用したクライアントとデータベース・サーバー間で、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。

Named Pipesは、特にMicrosoft Windows LAN環境で使用することを念頭に設計されています。サーバー側のプロセスが名前付きパイプを生成し、クライアント側のプロセスが名前によってそれをオープンします。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。

リモートのOracle Database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、Oracle Netにより、クライアント上のアプリケーションはNamed Pipesを使用してOracle Databaseと通信できるようになります。

親トピック: [Oracle protocol supportレイヤーの理解](#)

4.4.4 Sockets Direct Protocol (SDP)について

Sockets Direct Protocol(SDP)は、InfiniBandネットワーク・ピア間の業界標準のワイア・プロトコルです。InfiniBandネットワークでSDPを使用すると、データの中間的なレプリケーションが除去され、メッセージ交換の負荷がCPUからネットワーク・ハードウェアへ移動することにより、TCP/IPのオーバーヘッドが削減されます。

親トピック: [Oracle protocol supportレイヤーの理解](#)

4.4.5 Exadirectプロトコルについて

Exadirectプロトコルは、データベース・アクセスのオーバーヘッドが少ない、革新的なプロトコルです。インフィニバンド環境では、Remote Direct Memory Access (RDMA)の活用によって待機時間とスループットが向上する新しいトランスポートを使用してください。

Exadirectプロトコルは、通信の制御にTCPを使用し、データにIB RCトランスポートを使用します。

このリリースでは、Exadirectプロトコル・アダプタはOracle Linuxでのみサポートされています。

親トピック: [Oracle protocol supportレイヤーの理解](#)

4.4.6 WebSocketプロトコルについて

データベース・クライアント接続は、安全なWebSocketプロトコルをサポートしています。安全なWebソケット接続の確立は、HTTPSで動作し、HTTPSプロキシおよび仲介プロキシをサポートするように設計されます。

このプロトコルは、最小限のプロトコル・オーバーヘッドを備えたデータベースへのネイティブ接続を提供します。安全なWebSocketプロトコルは、データベース・サーバーとクライアント間の接続の確立時にSQL*Netプロトコルをネゴシエートするために使用します。安全なWebSocketプロトコルはTLSを使用します。また、操作用のウォレットが必要です。

WebサーバーはWebソケットのトンネリングをサポートする必要があります。Webサーバー・プロキシ・モジュール(ApacheまたはOracle HTTP Server (OHS)の場合、mod_proxy_wstunnel))は、クライアントとバックエンド・データベース・サーバー間で安全なWebソケット・データベース接続データをプロキシします。データベース・クライアントは安全なWebSocketプロトコルを使用してWebサーバーに接続し、WebサーバーはWebSocketプロトコルを使用してデータベースに接続します。

親トピック: [Oracle protocol supportレイヤーの理解](#)

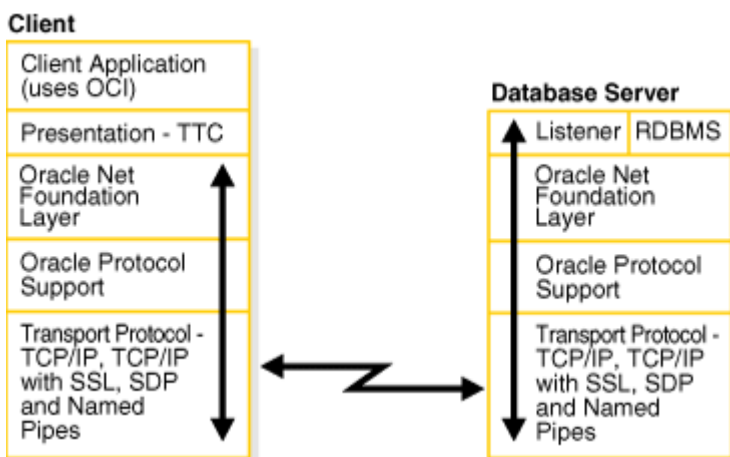
5 Oracle Netアーキテクチャの理解

Oracle NetリスナーはOracle Net Foundationレイヤーの最上位に位置するアプリケーションです。データベースは、クライアント・アプリケーションからの初期接続を、リスナーを通じて受け取ります。

リスナーは、クライアント要求を受け取ってOracle Databaseサーバーに渡します。クライアントがデータベースとのネットワーク・セッションを要求するたびに、リスナーは初期要求を受信します。

図5-1では、初期接続時のクライアントとデータベース・サーバー上にある様々なレイヤーを示します。図に示すように、リスナーはサーバー側のネットワーク・スタックの最上位レイヤーにあります。

図5-1 初期接続で使用するレイヤー



- [サービス登録について](#)
- [リスナーおよび接続要求について](#)
- [Oracle Restartについて](#)
- [ブロックされた接続要求について](#)
- [データベース・サーバー・プロセス・アーキテクチャの理解](#)
- [Oracle Connection Managerのアーキテクチャの理解](#)
- [完全なアーキテクチャ](#)

親トピック: [Oracle Net Servicesの理解](#)

5.1 サービス登録について

リスナーはデータベース・サービスとそのサービス・ハンドラが利用可能かどうかを、[サービス登録](#)から判断します。登録を行うと、[リスナー登録\(LREG\)](#)プロセスからリスナーに次の情報が提供されます。

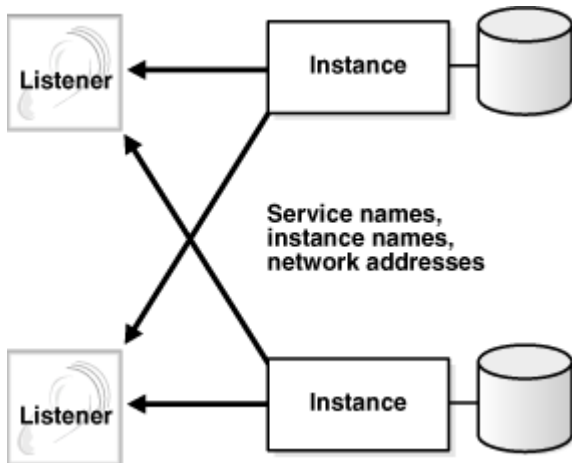
- データベースが提供するデータベース・サービスの名前
- サービスに対応付けられているデータベース・[インスタンス](#)の名前と、その現在および最大のロード情報
- インスタンスから使用可能なサービス・ハンドラ([ディスパッチャ](#)および[専用サーバー](#))とそのタイプ、プロトコル・アドレスおよび現在のロード量と最大ロード量

前述の情報によって、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。

図5-2は、情報を2つのリスナーに登録する2つのデータベース・インスタンスを示しています。登録できるすべての情報を示してい

るわけではありません。たとえば、ポート番号などのリスニング・エンドポイントをリスナーに動的に登録できます。

図5-2 サービス登録



インスタンスの起動時にリスナーが実行していない場合、LREGプロセスはサービス情報を登録できません。LREGは、定期的にリスナーに接続を試みますが、リスナーが起動されてからLREGがリスナーに登録するまで、最大で60秒間遅延することがあります。リスナーの起動後即座にサービス登録を開始するには、SQL文ALTER SYSTEM REGISTERを使用します。この文は特に高可用性が求められる構成で有益です。

親トピック: [Oracle Netアーキテクチャの理解](#)

5.2 リスナーおよび接続要求について

各リスナーは、リスニングするエンドポイントを指定する1つ以上の[プロトコル・アドレス](#)で構成されます。プロトコル・アドレスはリスナーがリスニングを実行するプロトコルと、プロトコル固有のその他の情報を定義します。たとえば、リスナーを次のプロトコル・アドレスでリスニングを実行するように設定できます。

```
(DESCRIPTION=
  (ADDRESS= (PROTOCOL=tcp) (HOST=sales-server) (PORT=1521)))
```

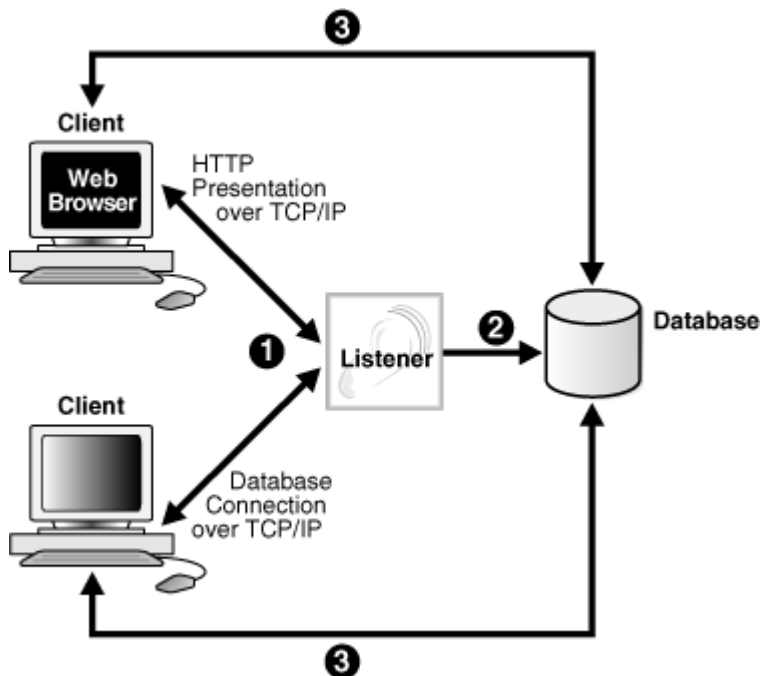
前の例では、リスナーのホスト(sales-server)とポート番号(1521)を指定するTCP/IPプロトコル・アドレスを示しています。

1つのプロトコル・アドレスで構成されたクライアントは、そのリスナーに接続要求を送ります。リスナーはクライアント要求を受信すると、そのクライアント要求を処理する適切な[サービス・ハンドラ](#)を選択してクライアント要求を転送します。サービス・ハンドラは、データベースへの接続ポイントとして機能するディスパッチャまたは専用サーバーです。

[図5-3](#)は、接続確立中のリスナーのロールを示しています。HTTP接続を行うブラウザとデータベース接続を行うクライアントを示しています。

1. ブラウザまたはクライアントがリスナーに接続要求を送信します。
2. リスナーは要求を解析し、要求されたデータベース・サービスのサービス・ハンドラに転送します。
3. ブラウザまたはクライアントはデータベースに接続します。

図5-3 リスナーのアーキテクチャ



親トピック: [Oracle Netアーキテクチャの理解](#)

5.3 Oracle Restartについて

Oracle Restartは、シングル・インスタンス環境におけるOracleデータベースの可用性を向上します。サーバー制御 (SRVCTL)ユーティリティを使用すると、リスナーなどのコンポーネントをOracle Restart構成に追加できます。この構成により、リスナーに障害が発生した場合、または動作していない場合にリスナーを自動的に起動できます。

Oracle Restartを使用するときは、次のことに注意してください。

- リスナーを起動および停止するには、SRVCTLユーティリティを使用します。リスナー制御ユーティリティLSNRCTLは使用しないでください。
- 各リスナーは固有の名前を持つ必要があります。

関連項目:

- [「Oracle Restart構成のリスナーの管理」](#)
- Oracle Restartの構成方法を学習するには、[『Oracle Database管理者ガイド』](#)を参照してください。

親トピック: [Oracle Netアーキテクチャの理解](#)

5.4 ブロックされた接続要求について

ブロックされた接続要求は、対応するインスタンスが登録される前に受信要求が行われた場合、またはデータベースのシャットダウンが進行中のときなどのようにデータベースが制限モードになっている場合に、発生する可能性があります。データベース・インスタンスが制限されたモードの場合、LREGはリスナーにこのインスタンスへの全接続をブロックするよう指示します。クライアントが接続しようとする、次のいずれかのエラーが発生します。

- ORA-12526: TNS:リスナー: 該当するインスタンスはすべて限定モードになっています

- ORA-12527: TNS:リスナー: インスタンスはすべて限定モードになっているか、新規接続をブロックしています
- ORA-12528: TNS:リスナー: 該当するインスタンスはすべて、新規接続をブロックしています

データベース・インスタンスがリスナーにまだ登録されていないと、ORA-12528エラーが発生します。

関連項目:

- エラー・メッセージの詳細は、『[Oracle Databaseエラー・メッセージ](#)』を参照してください。
- ALTER SYSTEM REGISTER文の詳細は、『[Oracle Database SQLリファレンス](#)』を参照してください。
- HTTP、FTPおよびWebDAVのリスニング・エンドポイントの動的な登録の詳細は、『[Oracle XML DB開発者ガイド](#)』を参照してください

親トピック: [Oracle Netアーキテクチャの理解](#)

5.5 データベース・サーバー・プロセス・アーキテクチャの理解

リスナーに登録されたサービス・ハンドラのタイプに基づいて、リスナーは共有サーバー・プロセスまたは専用サーバー・プロセスのいずれかに要求を転送します。データベース・サーバーは、共有サーバー・アーキテクチャを使用して、サーバー・プロセスを多数のクライアント・プロセスが共有できるようにしています。専用サーバー構成では、リスナーはクライアントの着信接続要求ごとに専用サーバー・プロセスを個別に起動します。このプロセスはクライアントへのサービス提供のみを行います。

- [共有サーバー・プロセスについて](#)
- [専用サーバー・プロセスについて](#)

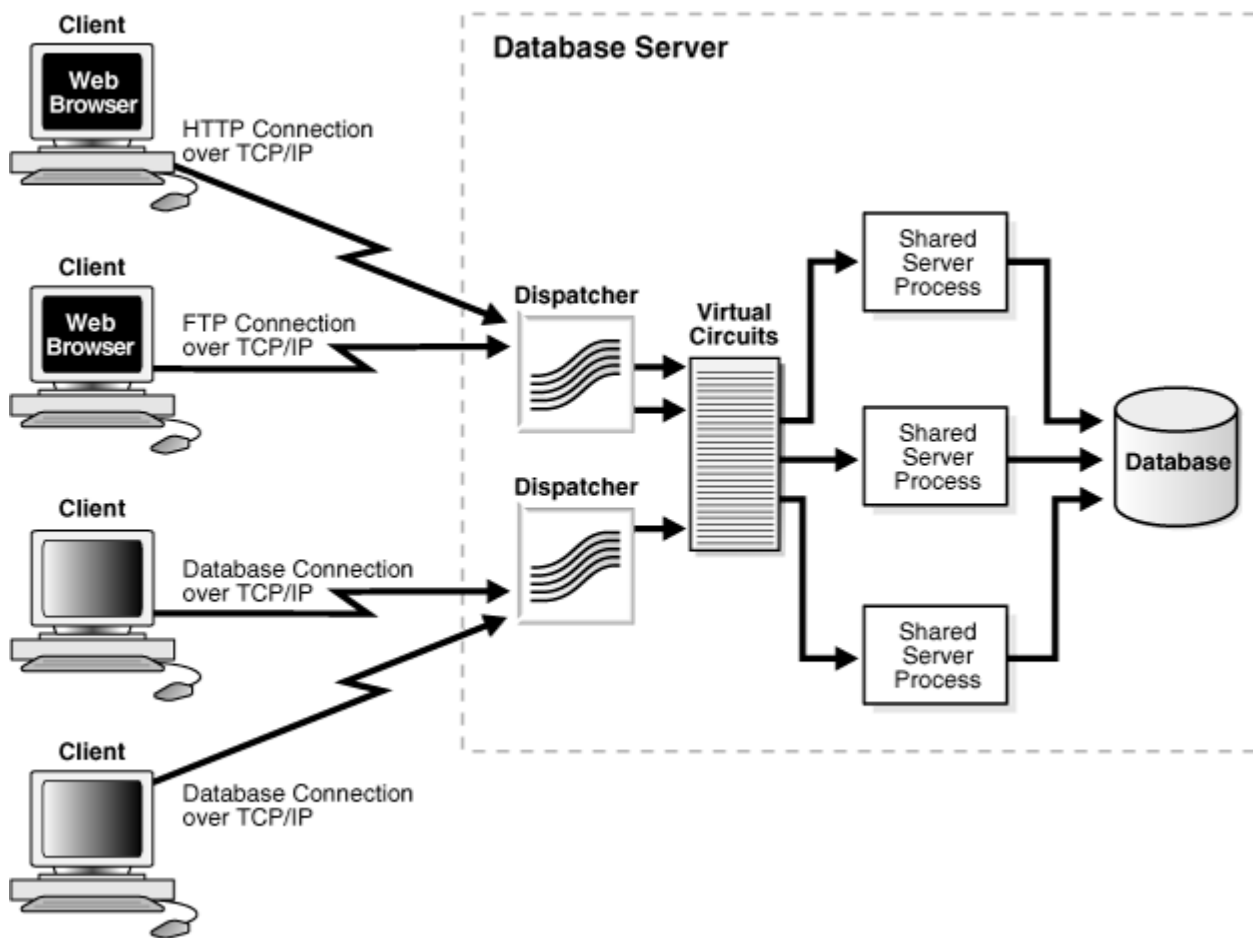
親トピック: [Oracle Netアーキテクチャの理解](#)

5.5.1 共有サーバー・プロセスについて

[図5-4](#)で示すように、共有サーバー・プロセスは、共有サーバー・アーキテクチャで使用されます。共有サーバー・アーキテクチャでは、クライアントは最終的にディスパッチャへの接続を行います。LREGプロセスは、ディスパッチャの場所とロード情報をリスナーに登録するため、リスナーは要求を最もロード量の少ないディスパッチャに転送できます。この登録プロセスは、図には示されていません。

ディスパッチャは、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、[バーチャル・サーキット](#)にバインドされます。バーチャル・サーキットは、ディスパッチャが使用する共有メモリの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、要求が到着するとバーチャル・サーキットを共通要求キューに配置します。アイドル状態の共有サーバーは、要求キューからバーチャル・サーキットを取り出して要求を処理し、要求キューから次のバーチャル・サーキットを取り出す前に、取り出したバーチャル・サーキットを放棄します。共有サーバーは、完了したすべての要求をディスパッチャのレスポンス・キューに格納します。各ディスパッチャには、SGA(システム・グローバル領域)に独自のレスポンス・キューがあります。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。

図5-4 共有サーバー・アーキテクチャ



親トピック: [データベース・サーバー・プロセス・アーキテクチャの理解](#)

5.5.2 専用サーバー・プロセスについて

専用サーバー・アーキテクチャでは、クライアント・プロセスごとに専用サーバー・プロセスへの接続を行います。サーバー・プロセスは、他のいずれのクライアントとも共有されません。[図5-5](#)は、専用サーバー・アーキテクチャを示しています。

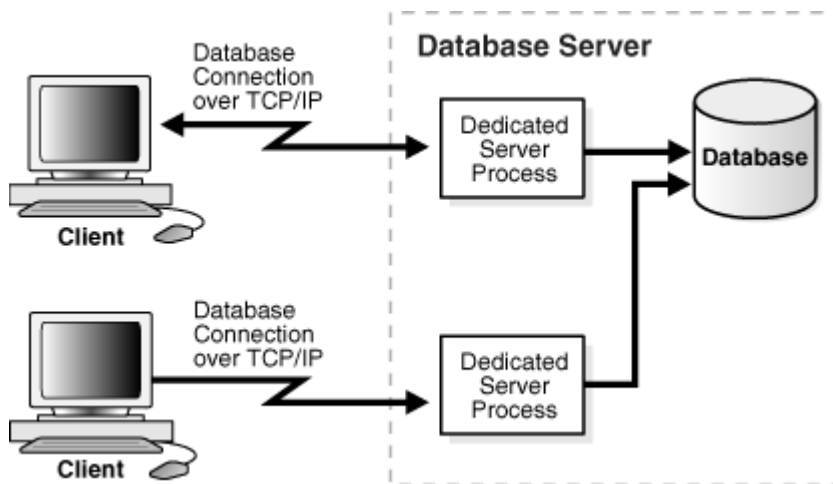
LREGは専用サーバー・プロセスに関する情報をリスナーに登録します。これによってリスナーは、クライアント要求を受け取って転送する際に、専用サーバー・プロセスを開始できます。

ノート:



専用サーバー・アーキテクチャは、HTTP、FTP または WebDAV クライアントをサポートしていません。データベース・クライアントのみサポートします。

図5-5 専用サーバー・アーキテクチャ



親トピック: [データベース・サーバー・プロセス・アーキテクチャの理解](#)

5.6 Oracle Connection Managerのアーキテクチャの理解

Oracle Connection Managerはゲートウェイであり、これを介してクライアントの接続要求が次のホップまたはデータベース・サーバーに直接送信されます。Oracle Connection Managerを通して接続要求を送信するクライアントは、そのOracle Connection Managerに構成されているセッション多重化およびアクセス制御を利用できます。LREGプロセスがサービスを登録するまで、サービス情報は送信されません。

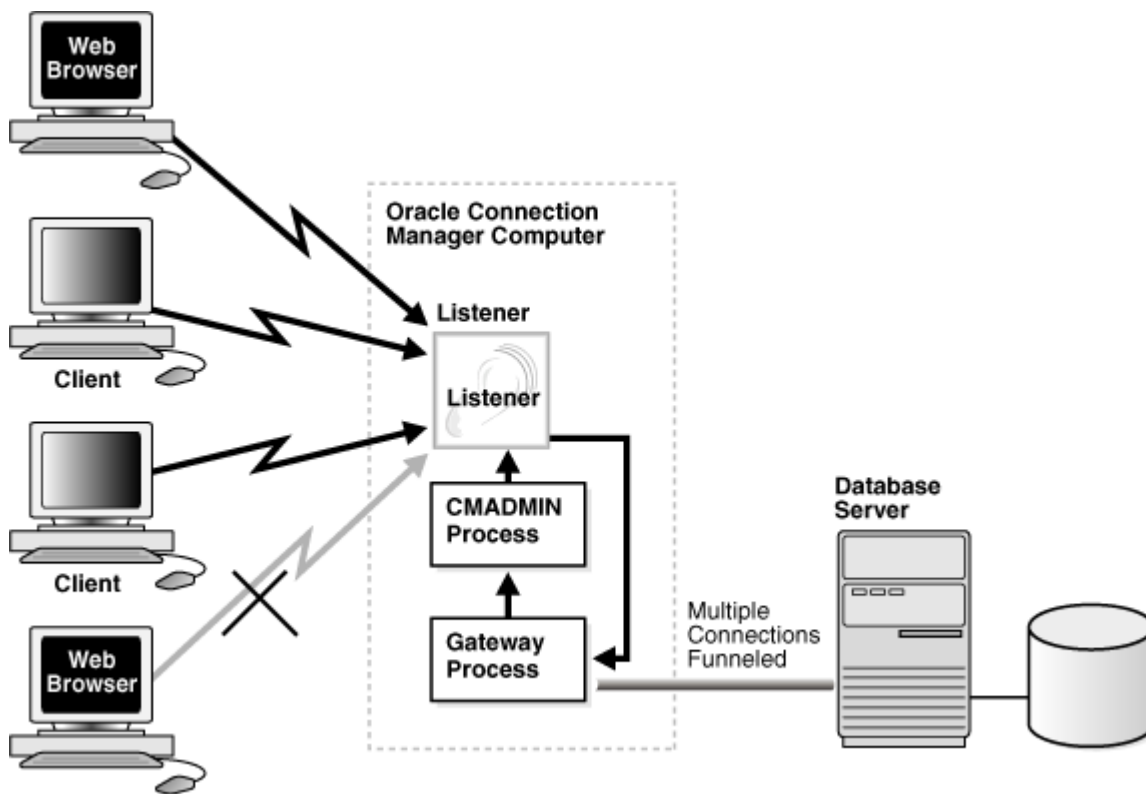
Oracle Connection Managerは3つのコンポーネントで構成されています。

- [リスナー](#)
- [CMGW \(Oracle Connection Manager Gateway\)](#)
- [CMADMIN\(Oracle Connection Manager Administration\)](#)

リスナーは、クライアント接続を受け取り、一連の規則と照合して、アクセスの可否を判断します。アクセスが許可されると、リスナーは最小接続回数の接続を選択してゲートウェイ・プロセスに要求を転送します。CMGWプロセスでは、データをリレーするために接続が終了するまで、この要求を別のOracle Connection Managerに転送するか、またはデータベース・サーバーに直接、転送します。すでに既存のサーバーへの接続がある場合、ゲートウェイは既存の接続を介してこの接続を多重化、または集中化させます。CMADMINは、ゲートウェイ・プロセスとリスナーの状態を監視して、必要に応じてプロセスを停止または開始します。さらに、このリスナーで使用されるゲートウェイ・プロセスの場所とロードを登録し、Oracle Connection Manager Controlユーティリティの要求に応答します。

図5-6では、リスナーは接続要求を選別します。ゲートウェイ・プロセスは、CMADMINプロセスに登録され、CMADMINプロセスはリスナーに登録されます。最後にリスナーは、接続要求をゲートウェイ・プロセスに転送します。ゲートウェイ・プロセスは3つの有効なクライアント接続を受け取り、データベースへの単一のネットワーク・プロトコル接続を介してそれらを多重化します。4番目の接続は一連の規則について評価されて拒否されます。

図5-6 Oracle Connection Managerのアーキテクチャ



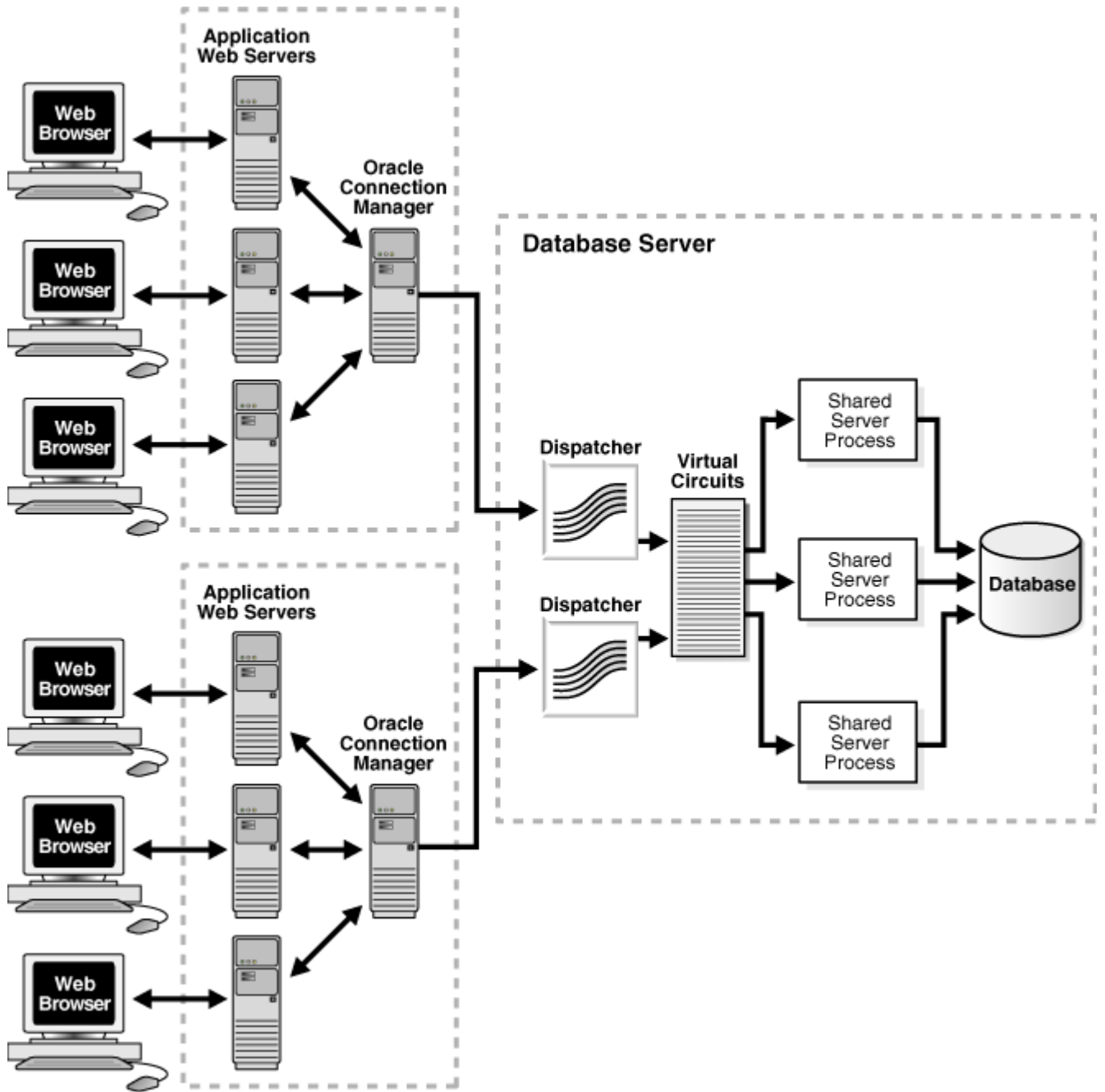
親トピック: [Oracle Netアーキテクチャの理解](#)

5.7 完全なアーキテクチャ

Oracle Netはアーキテクチャ上のソリューションを提供しているため、インターネットやイントラネット環境における拡張性をさらに高めることができます。

図5-7は、Oracle Databaseサーバーへの複数の接続がOracle Connection Managerおよび共有サーバー・アーキテクチャによってさらにスケーラブルになっている様子を示しています。Oracle Connection Managerを使用すると、アプリケーションWebサーバーのネットワークI/Oの負荷が軽減され、共有サーバーを使用すると、さらに多くの同時ユーザーが処理されます。

図5-7 スケーラブルなアーキテクチャ上のソリューション



親トピック: [Oracle Netアーキテクチャの理解](#)

第II部 Oracle Net Servicesの構成と管理

第II部では、Oracle Net Servicesの設定方法と構成方法を説明します。

この部は、次の章で構成されています。

- [Oracle Net Services接続のクイック・スタート](#)
- [Oracle Net Servicesの管理](#)
- [ネーミング・メソッドの構成](#)
- [Oracle Net Listenerの構成と管理](#)
- [Oracle Connection Managerの構成](#)
- [共有サーバー・アーキテクチャの構成](#)
- [プロファイルの構成](#)
- [Oracle Net Servicesの拡張機能の有効化](#)
高度な接続データ・パラメータ、ロード・バランシング、フェイルオーバーおよび非データベース・サービスへの接続など、Oracle Net Servicesの拡張機能を構成する方法について理解します。
- [パフォーマンスの最適化](#)

6 Oracle Net Services接続のクイック・スタート

クライアント・アプリケーションとデータベースのTCP/IPネットワークによる接続のように、単純でありながら一般的なネットワーク構成について、初心者でも設定とテストができるように説明します。

- [接続確立の前提条件](#)
- [ネットワークの可用性の確認](#)
- [Oracle Net ListenerとOracle Databaseサーバーの起動](#)
- [Oracle Connection Managerの起動](#)
- [データベースへの接続](#)
Oracleデータベースに接続するには、いくつかの方法があります。
- [データベースに接続するための簡易接続の使用](#)

親トピック: [Oracle Net Servicesの構成と管理](#)

6.1 接続確立の前提条件

この章のタスクでは、データベース・サーバーとクライアント・コンピュータとのTCP/IP接続について示します。データベース・サーバーとクライアント・コンピュータは次の条件を満たす必要があります。

- データベース・サーバー
 - クライアントにアクセスできるネットワークでサーバーが実行中であること
 - Oracleデータベースがインストールされていること
 - リスナーが構成されていること
 - TCP/IPプロトコル・サポートがインストールされていること
- クライアント・コンピュータ
 - データベース・サーバーにアクセスできるネットワークでクライアント・コンピュータが実行中であること
 - Oracleクライアントがインストールされていること
 - TCP/IPプロトコル・サポートがインストールされていること

TCP/IPネットワークでは、各コンピュータは一意の[IPアドレス](#)を持ちます。[ドメイン・ネーム・システム\(DNS\)](#)などの名前解決サービスは、そのホスト名を持つコンピュータのIPアドレスをマップするために使用します。名前解決サービスを使用しない場合、マッピングは通常hostsと呼ばれるファイルに格納されるため、集約化して保守されます。このファイルは、Linuxでは/etcディレクトリに、Microsoft Windowsでは¥windows¥system32¥drivers¥etcディレクトリにあります。たとえば、sales-serverという名前のデータベース・サーバーのコンピュータのエントリは、次のようになります。

#IP address of server	host name	alias
192.0.2.203	sales-server	sales.us.example.com

親トピック: [Oracle Net Services接続のクイック・スタート](#)

6.2 ネットワークの可用性の確認

Oracle Netをクライアント・コンピュータとデータベース・サーバーとの接続に使用する前に、クライアント・コンピュータがデータベー

ス・サーバーのコンピュータと正常に通信できるかどうかを確認します。ネットワークの接続性を評価しておく、ネットワークに起因するエラーを排除できます。

次の手順では、ネットワーク接続の確認方法について説明します。

1. 次のように、データベース・サーバーのコンピュータが、ループバック・テストによって自分自身と通信できることを確認します。

- a. ハードウェア接続を確認するには、コマンドラインで次のコマンドを入力します。

```
ping ip_address
```

前述のコマンドで、ip_addressはデータベース・サーバー・コンピュータのIPアドレスです。たとえば、次のように入力します。

```
ping 192.0.2.203
```

- b. DNSまたはホスト名が正しく構成されていることを確認するには、コマンドラインで次のコマンドを入力します。

```
ping host_name
```

前述のコマンドで、host_nameはサーバーのホスト名です。

- c. サーバーのTCP/IP設定をテストするには、次のコマンドを入力します。

```
ping 127.0.0.1  
ping6 ::1
```

IPアドレス127.0.0.1は、ループバック・テスト用の標準のIPv4アドレスです。IPアドレス::1 (0: 0: 0: 0: 0: 0: 0: 1)は、ループバック・テスト用の標準のIPv6アドレスです。

2. クライアント・コンピュータがデータベース・サーバーのコンピュータと正常に通信できることを確認します。

検証方法は、ネットワーク・プロトコルによって様々です。TCP/IPでは、PING、FTP、TELNETユーティリティを使用できます。

クライアント・コンピュータがサーバーに接続できない場合、ネットワークのケーブル接続やネットワーク・インタフェース・カードが正しく接続されているかを確認します。これらの問題を解消するには、ネットワーク管理者に連絡してください。

親トピック: [Oracle Net Services接続のクイック・スタート](#)

6.3 Oracle Net ListenerとOracle Databaseサーバーの起動

Oracle Net ListenerとOracle Databaseサーバーは、データベース・サーバーが接続を受け取る順番で実行している必要があります。次の手順では、Oracle Netリスナーを起動する方法について説明します。

1. リスナー制御ユーティリティを使用してリスナーを起動します。コマンド行から、次のように入力します。

```
lsnrctl  
LSNRCTL> START [listener_name]
```

前述のコマンドで、listener_nameは、listener.oraファイルに定義されているリスナーの名前です。デフォルトの名前LISTENERを使用している場合、リスナーを指定する必要はありません。

リスナーが正常に起動したことを示すステータス・メッセージが表示されます。

2. 次のようにデータベースを起動します。

- a. 次のコマンドを使用して、データベースに接続せずにSQL*Plusを起動します。

```
SQLPLUS /nolog
```

- b. 次のコマンドを使用して、SYSDBAとしてデータベースに接続します。

```
SQL> CONNECT username as sysdba
```

パスワードの入力を求めるメッセージが表示されます。

ノート:



単純化するため、この例では、デプロイされたシステムで通常使用されるパスワード管理技術を実行していません。本番環境では、Oracle Databaseのパスワード管理ガイドラインに従い、サンプル・アカウントを無効にしてください。パスワード管理ガイドラインおよびその他のセキュリティ上の推奨事項については、『[Oracle Database セキュリティ・ガイド](#)』を参照してください。

- c. 次のコマンドを使用して、データベースを起動します。

```
SQL> STARTUP database_name
```

前述のコマンドで、database_nameはデータベースの名前です。

関連項目:

データベースの起動の詳細は、『[Oracle Database管理者ガイド](#)』を参照してください。

3. リスナー制御ユーティリティと次のコマンドを使用して、リスナーへのデータベース・[サービス登録](#)が完了していることを確認します。

```
LSNRCTL> SERVICES [listener_name]
```

SERVICESコマンドは、データベースがサポートするサービスと、1つ以上の使用可能な[サービス・ハンドラ](#)の一覧を示します。データベース・サービス登録が示されていない場合は、次のSQLコマンドを入力します。

```
SQL> ALTER SYSTEM REGISTER;
```

関連項目:

SERVICESコマンドの詳細は、『[リスナーのサービスの監視](#)』を参照してください

親トピック: [Oracle Net Services接続のクイック・スタート](#)

6.4 Oracle Connection Managerの起動

Oracle Connection Managerがインストールされている場合は、次の手順に従ってOracle Connection Managerを起動します。

1. 次のコマンドを使用して、Oracle Connection Manager Controlユーティリティ(CMCTL)を起動します。

```
cmctl  
CMCTL> ADMINISTER [instance_name]
```

前述のコマンドで、instance_nameは管理対象のOracle Connection Managerの名前です。cman.oraファイルを表示して、その名前を確認できます。ファイルは、ORACLE_HOME/network/adminディレクトリ内のOracle Connection Managerコンピュータ上にあります。

Oracle Connection Managerは、インスタンス名を示すステータス・メッセージを表示し、まだインスタンスが起動していないことを知らせます。

ノート:



引数にインスタンス名を設定しない場合は、Oracle Connection Manager に完全修飾ホスト名を設定します。これはデフォルトです。ADMINISTER コマンドを発行すると、CMCTL により次のようにインスタンス名が表示されます。

```
CMAN_fully_qualified_host_name
```

2. 次のコマンドを使用して、管理対象として選択したOracle Connection Managerを起動します。

```
cmctl> STARTUP
```

インスタンスが起動したことがOracle Connection Managerで確認され、インスタンスのステータスが表示されます。

3. 次のコマンドを使用して、Oracle Connection Manager Controlユーティリティを終了します。

```
cmctl> EXIT
```

ノート:

Microsoft Windows の場合、Oracle 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」の「サービス」を選択します。
2. 「OracleHOME_NAMECMan」サービスを選択し、「Start」をクリックします。
3. 「サービス」ウィンドウの「閉じる」をクリックします。

親トピック: [Oracle Net Services接続のクイック・スタート](#)

6.5 データベースへの接続

Oracleデータベースに接続するには、いくつかの方法があります。

表6-1 データベース接続方法と構文

接続のタイプ	接続構文	説明
--------	------	----

接続のタイプ	接続構文	説明
コマンドラインから	<p>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。</p> <pre>tool username@connect_identifier</pre> <p>暗号化されたパスワードの入力を求めるメッセージが表示されます。</p> <p>たとえば:</p> <pre>SQLPLUS system@sales Enter password: password</pre>	ほとんどの Oracle のツール製品は、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せん。
ログイン画面から	<pre>username@connect_identifier</pre>	一部のツールには、ログインの代替形式としてログイン画面が用意されています。ユーザーがデータベース・サーバーにログインするには、ツールのログイン画面のユーザー名フィールドにユーザー名と接続識別子の両方を指定し、パスワード・フィールドにパスワードを入力します。
3GL アプリケーションから	<pre>exec sql connect :username identified by :password</pre> <p>前述の接続要求で、:username および:password は 3GL の変数です。これらの変数は、静的にまたはユーザーからの入力によってプログラム内で設定できます。データベース・サーバーに接続する場合、:username 変数の値の書式は次のとおりです。</p> <pre>username@net_service_name</pre> <p>: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せん。</p>	OCI やプリコンパイラなど、3GL で作成されたアプリケーションは、クライアントからの直接データベース・アクセス用に中間層およびデータベース・アプリケーション開発者によって使用されます。
SQL*Plus 内から	<pre>SQLPLUS /nolog SQL> CONNECT username@net_service_name</pre> <p>たとえば:</p> <pre>SQLPLUS /nolog SQL> CONNECT scott@serverx Enter</pre>	<p>一部の Oracle ツールには、ツールを終了せずに、異なるユーザー名を指定してデータベースに接続できるコマンドがあります。</p> <p>他の Oracle のツール製品では、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</p>

接続のタイプ	接続構文	説明
	<p>password: password</p> <p>前述のコマンドで、username と password は、データベース・ユーザーとパスワードで、net_service_name はネットワーク・サービス名です。</p>	<p>CDE Tools では、ログイン・ボタンと、ユーザー名、パスワードおよびリモート・データベース ID のフィールドが使用されます。</p>
<p>次を使用:</p> <p>KERBEROS5_CC_NAME パラメータ</p>	<p>(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sales-svr)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=sales.example.com))(SECURITY=(SQLNET.KERBEROS5_CC_NAME=/usr/tmp/krbcache)))</p>	<p>このパラメータは、複数の Kerberos プリンシパルがデータベース・クライアントを介してログインする必要がある場合に、Kerberos プリンシパル(user)の Kerberos 資格証明キャッシュ・ファイルへの完全パスを指定するために使用します。</p> <p>Kerberos 認証を使用してデータベースに接続する場合は、KERBEROS5_CC_NAME パラメータを使用して資格証明キャッシュへの完全パスを指定する必要があります。単一の Kerberos プリンシパルの場合は、SQLNET.KERBEROS5_CC_NAME パラメータを使用して sqlnet.ora ファイルに資格証明キャッシュ・パスを指定できます。クライアントが複数のデータベース接続を行うために複数の Kerberos プリンシパルを使用する必要がある場合は、個々の接続の CONNECT 文字列または tnsnames.ora ファイルで KERBEROS5_CC_NAME パラメータを指定します。</p> <p>Oracle Database セキュリティ・ガイドを参照してください。</p>
<p>次を使用:</p> <p>KERBEROS5_PRINCIPAL パラメータ</p>	<p>(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sales-svr)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=sales.example.com))(SECURITY=(KERBEROS5_CC_NAME=/tmp/krbuser1/krbcc)(KERBEROS5_PRINCIPAL=krbprincipal1@example.com)))</p>	<p>このパラメータは、データベース・クライアントの Kerberos プリンシパルを指定するために使用されます。</p> <p>これはオプションのパラメータです。ただし、このパラメータを使用する場合は、KERBEROS5_PRINCIPAL が、KERBEROS5_CC_NAME パラメータで指定されている Kerberos 資格証明キャッシュから取得されたプリンシパルと一致することを確認してください。プリンシパル名が一致しない場</p>

接続のタイプ	接続構文	説明
		合、認証は失敗します。
IAM データベース・パスワード検証認証の使用	<pre>SQLPLUS /nolog SQL> CONNECT username@net_service_name Enter password: password</pre> <p>この接続リクエストでは、username は IAM ユーザー名、net_service_name は ネットワーク・サービス名、password は IAM データベース・パスワードです。</p> <p>IAM ユーザーは、パスワードベースのプロキシ認証を使用して、データベース・プロキシ・ユーザー・アカウントを介して接続できます。</p> <pre>SQLPLUS [user1]/password123¥!@cdb1_pdb2 SQL> SHOW USER; select sys_context('USERENV', 'AUTHENTICATION_METHOD') from dual; select sys_context('USERENV', 'PROXY_USER') from dual; select sys_context('USERENV', 'CURRENT_USER') from dual;</pre>	<p>ユーザー認証および認用に Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM)を構成した場合、IAM ユーザーが IAM ユーザー名および IAM データベース・パスワードを使用して OCI Database as a Service (DBaaS)に接続できます。この接続方法では、IAM データベース・パスワード検証(パスワードの暗号化されたハッシュ)がユーザーを認証するために IAM から取得されます。</p> <p>この IAM データベース・パスワードは、OCI コンソール・パスワードとは異なります。IAM ユーザーが、OCI コンソールからこのパスワードを設定できます(OCI IAM パスワードの作成を参照)。</p> <p>必要な認可が付与された後、SQL*Plus などのオンプレミス・クライアント・アプリケーションを使用して、サポートされている任意のデータベース・クライアントからログインできます。</p> <p>IAM データベース検証での IAM ユーザー名および IAM データベース・パスワードの使用はデフォルトの構成であり、クライアントに追加のパラメータを設定する必要はありません。ただし、クライアント側の sqlnet.ora ファイルで PASSWORD_AUTH が OCI_TOKEN に設定されている場合、データベース・クライアントは OCI IAM に接続し、IAM ユーザー名および IAM データベース・パスワードを使用してデータベース・トークンを取得しようとします。この場合、PASSWORD_AUTH=PASSWORD_VERIFIER を設定することで、特定の接続に対してこの設定をオーバーライドできます。</p> <p>プロキシ認証を使用するようにクライアント接続(通常は中間層環境)を構成できます。この場合、パスワードベースのプロキシ認証を使用することで、プロキシ・データベースのユー</p>

接続のタイプ	接続構文	説明
		<p>ザー・アカウントを介して接続するために必要な権限を持つ既存の IAM ユーザーを変更できます。プロキシ・ユーザー・セッションに、IAM ユーザーに付与されたすべての権限がありません。</p> <p>Oracle Database セキュリティ・ガイドを参照してください。</p>
IAM トークンベースの認証 (ベアラー・トークン)の使用	<pre>SQLPLUS /nolog SQL> CONNECT /@connect_identifier</pre> <p>PASSWORD_AUTH 設定により、ベアラー・トークンを使用して IAM トークンベースの認証が強制されます:</p> <pre>(DESCRIPTION= (AADDRESS=(PROTOCOL=tcps) (HOST=saleseserver1) (PORT=1522)) (SECURITY= (SSL_SERVER_DN_MATCH=TRUE) (SSL_SERVER_CERT_DN="C=US, O=example, CN=OracleContext") (PASSWORD_AUTH=OCI_TOKEN) (OCI_IAM_URL=https://auth.us-region-1.example.com/v1/actions/generateScopedAccessBearerToken) (OCI_TENANCY=ocid1.tenancy..12345)) (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))</pre> <p>オプションの OCI_COMPARTMENT および OCI_DATABASE パラメータを使用して、トークン・リクエストの範囲を制限できます。</p> <pre>(DESCRIPTION= (AADDRESS=(PROTOCOL=tcps) (HOST=saleseserver1) (PORT=1522)) (SECURITY= (SSL_SERVER_DN_MATCH=TRUE) (SSL_SERVER_CERT_DN="C=US, O=example, CN=OracleContext") (PASSWORD_AUTH=OCI_TOKEN) (OCI_IAM_URL=https://auth.us-region-1.example.com/v1/actions/generateScopedAccessBearerToken) (OCI_TENANCY=ocid1.tenancy..12345) (OCI_COMPARTMENT=ocid1.compartment..12345) (OCI_DATABASE=ocid1.autonomousdatabase.oc1.12345))</pre>	<p>ユーザー認証および認可用に IAM を構成した場合、IAM ユーザーが IAM データベース・トークン(db-token)を使用して Oracle DBaaS に接続できます。このタイプのトークンはベアラー・トークンであり、秘密キーは付属していません。</p> <p>IAM ユーザー名および IAM データベース・パスワードを使用して、db-token をリクエストするようにデータベース・クライアントを構成できます。アプリケーションはこのタイプのトークンをクライアントに渡せません。この場合、PASSWORD_AUTH パラメータを OCI_TOKEN に設定します。</p> <p>IAM ユーザーが、OCI コンソールから IAM データベース・パスワードを設定できます(OCI IAM パスワードの作成を参照)。</p> <p>データベース・クライアントは、OCI IAM エンドポイントから直接トークンを取得します。データベース・クライアントが追加メタデータとともに IAM エンドポイントを検出できるように、追加パラメータを設定する必要があります。追加のパラメータは OCI_IAM_URL および OCI_TENANCY で、スコープを制限するオプション OCI_COMPARTMENT および OCI_DATABASE があります。</p> <p>IAM ユーザーが/@connect_identifier を使用して IAM ユーザー名および IAM データベース・パスワードでログインすると、PASSWORD_AUTH=OCI_TOKEN 設定および</p>

接続のタイプ	接続構文	説明
	<pre>(CONNECT_DATA=(SERVICE_NAME=sales.us. example.com)))</pre>	<p data-bbox="1029 168 1524 347">/connect_identifier は、REST API リクエストを使用して OCI IAM エンドポイントからトークンを直接取得するようにデータベース・クライアントに指示します。</p> <p data-bbox="1029 392 1524 526">これらのパラメータは、tnsnames.ora ファイル、sqlnet.ora ファイルで指定するか、接続文字列の一部として直接指定できます。</p> <p data-bbox="1029 571 1524 660">Oracle Database セキュリティ・ガイドを参照してください。</p>
<p data-bbox="151 705 446 795">IAM トークンベースの認証 (PoP トークン)の使用</p>	<pre>SQLPLUS /nolog SQL> CONNECT /@connect_identifier</pre> <p data-bbox="478 817 997 929">TOKEN_AUTH 設定により、PoP トークンを使用して IAM トークンベースの認証が強制されます。</p> <pre>(DESCRIPTION= (AADDRESS=(PROTOCOL=tcps) (HOST=sales_db) (PORT=1522)) (SECURITY= (SSL_SERVER_DN_MATCH=TRUE) (SSL_SERVER_CERT_DN="C=US, O=example, CN=OracleContext") (TOKEN_AUTH=OCI_TOKEN)) (CONNECT_DATA=(SERVICE_NAME=sales.us. example.com)))</pre> <p data-bbox="478 1332 997 1512">オプションの TOKEN_LOCATION パラメータを使用して、データベース・トークンおよび秘密キーが格納されているデフォルト・ディレクトリをオーバーライドできます。</p> <pre>(DESCRIPTION= (AADDRESS=(PROTOCOL=tcps) (HOST=sales_db) (PORT=1522)) (SECURITY= (SSL_SERVER_DN_MATCH=TRUE) (SSL_SERVER_CERT_DN="C=US, O=example, CN=OracleContext") (TOKEN_AUTH=OCI_TOKEN) (TOKEN_LOCATION="/home/oracle/.oci/db-token")) (CONNECT_DATA=(SERVICE_NAME=sales.us. example.com)))</pre> <p data-bbox="478 1982 997 2094">IAM ユーザーは、トークンベースのプロキシ認証を使用して、データベース・プロキシ・ユーザー・アカウントを介して接</p>	<p data-bbox="1029 705 1524 1220">ユーザー認証および認可用に IAM を構成した場合、IAM ユーザーは IAM データベース・トークン(db-token)を使用して Oracle DBaaS インスタンスに接続できます。このタイプのトークンは、有効期限と有効範囲を持つ Proof-of-possession (PoP)トークンです。Oracle Cloud Infrastructure (OCI)コマンドライン・インタフェース(CLI)を使用して IAM から、または OCI Software Development Kit (SDK)からプログラム的にトークンをリクエストできます。</p> <p data-bbox="1029 1265 1524 1579">この接続方法では、API キー、セキュリティ・トークン、リソース・プリンシパル、サービス・プリンシパル、インスタンス・プリンシパルまたは委任トークン(委任トークンは Cloud Shell でのみ使用可能)などの IAM ユーザー資格証明のいずれかを使用して、IAM から db-token および秘密キーを取得できます。</p> <p data-bbox="1029 1624 1524 1713">クライアント・アプリケーションは、次のいずれかの方法で接続リクエストを送信できます。</p> <ul data-bbox="1085 1758 1524 2072" style="list-style-type: none"> ● IAM ユーザーが /@connect_identifier を使用してログインした場合(および TOKEN_AUTH が OCI_TOKEN に設定されている場合)、データベース・クライアントは、デフォルト・ディレクトリまたは TOKEN_LOCATION で指定され

接続のタイプ	接続構文	説明
	<p>続きます。</p> <pre>SQLPLUS [user1]/@pop_token_connstr SQL> SHOW USER; select sys_context('USERENV', 'AUTHENTICATION_METHOD') from dual; select sys_context('USERENV', 'PROXY_USER') from dual; select sys_context('USERENV', 'CURRENT_USER') from dual;</pre>	<p>た場所のいずれかから db-token および秘密キーを取得します(IAM トークンベースの認証を使用)。</p> <ul style="list-style-type: none"> ● クライアント・アプリケーションが IAM からトークンを取得するように更新されている場合、IAM は、データベース・クライアント API を使用して、db-token および秘密キーを属性としてデータベース・クライアントに直接渡します。この場合、TOKEN_AUTH および TOKEN_LOCATION パラメータを指定する必要はありません。 <p>これらのパラメータは、tnsnames.ora ファイルまたは sqlnet.ora ファイルで指定することも、接続文字列の一部として直接指定することもできます。</p> <p>プロキシ認証を使用するようにクライアント接続を構成できます。この場合、トークンベースのプロキシ認証を使用することで、プロキシ・データベースのユーザー・アカウントを介して接続するために必要な権限を持つ IAM ユーザーを変更できます。プロキシ・ユーザー・セッションに、IAM ユーザーに付与されたすべての権限があります。</p> <p>Oracle Database セキュリティ・ガイドを参照してください。</p>
<p>Azure AD トークンベース認証の使用</p>	<pre>SQLPLUS /nolog SQL> CONNECT /@connect_identifier</pre> <p>TOKEN_AUTH 設定により、Azure AD トークンベースの認証が強制されます:また、TOKEN_LOCATION パラメータを使用して、アクセス・トークンが格納されるディレクトリ・パスを指定する必要があります。</p> <p>トークン・ファイルの名前が token の場合、クライアントは指定されたディレクトリ・パス(たとえば、/home/dbuser1/access-token)内のファイ</p>	<p>ユーザー認証および認可用に Microsoft Azure Active Directory (Azure AD)を構成した場合、Azure AD ユーザーは Azure AD OAuth2 アクセス・トークンを使用して Oracle Database に接続できます。このアクセス・トークンは、有効期限と範囲を持つベアラー・トークンであり、Azure AD 拡張機能を備えた OAuth2.0 標準に準拠しています。</p> <p>TOKEN_AUTH 設定(TOKEN_AUTH=OAUTH)</p>

接続のタイプ	接続構文	説明
	<p>ルを自動的に検索します。</p> <pre>(DESCRIPTION= (AADDRESS=(PROTOCOL=tcps) (HOST=saleses rver1) (PORT=1522)) (SECURITY= (SSL_SERVER_DN_MATCH=TRUE) (SSL_SERVER_CERT_DN="C=US, O=example, CN=OracleContext") (TOKEN_AUTH=OAUTH) (TOKEN_LOCATION="/home/dbuser1/access -token")) (CONNECT_DATA=(SERVICE_NAME=sales.us. example.com)))</pre> <p>トークン・ファイル名が token と異なる場合は、ディレクトリ・パスとともにファイル名 (mytoken など) を指定する必要があります。</p>	<p>は、トークン・ファイルが token という名前の場合に、TOKEN_LOCATION で指定されたディレクトリからアクセス・トークンを取得するようにデータベース・クライアントに指示します。そうでない場合は、TOKEN_LOCATION パラメータを指定するときに、トークン・ファイル名とディレクトリの場所を使用する必要があります。</p> <p>TOKEN_LOCATION パラメータは、Azure AD トークンベースの認証では必須です。データベース・クライアントはこの場所からアクセス・トークンを取得し、データベース・サーバーに送信します。</p>
	<pre>(DESCRIPTION= (AADDRESS=(PROTOCOL=tcps) (HOST=saleses rver1) (PORT=1522)) (SECURITY= (SSL_SERVER_DN_MATCH=TRUE) (SSL_SERVER_CERT_DN="C=US, O=example, CN=OracleContext") (TOKEN_AUTH=OAUTH) (TOKEN_LOCATION="/home/dbuser1/access -token/mytoken")) (CONNECT_DATA=(SERVICE_NAME=sales.us. example.com)))</pre>	<p>TOKEN_AUTH および TOKEN_LOCATION パラメータを、tnsnames.ora、sqlnet.ora ファイルに指定するか、接続文字列の一部として直接指定します。</p> <p>トークンは、Linux、Microsoft PowerShell またはその他の環境で実行されるツールおよびスクリプトからリクエストできます。これらのトークンは、Microsoft SDK を使用してプログラマ的にリクエストすることもできます。</p> <p>クライアント・アプリケーションは、次のいずれかの方法で接続リクエストを送信できます。</p>
		<ul style="list-style-type: none"> ● Azure AD ユーザーは、サポートされている Microsoft Azure AD 認証フロー (リソース所有者のパスワード資格証明、認可コード、On-Behalf-Of (OBO) フローまたはクライアント資格証明) のいずれかを使用して Azure AD からアクセス・トークンをリクエストし、ローカル・ファイル・ディレクトリに格納できます。 ● クライアント・アプリケーションが Azure AD からトークンを取得するように更新されている場合、アプリ

接続のタイプ	接続構文	説明
		<p>ケーションは Azure AD から直接アクセス・トークンをリクエストし、データベース・クライアント API を使用してそれを属性としてデータベース・インスタンスに渡すこともできます。この場合、TOKEN_AUTH および TOKEN_LOCATION パラメータを指定する必要はありません。</p> <p>Oracle Database セキュリティ・ガイドを参照してください。</p>



ノート:

tnsnames.ora 接続文字列で指定されたパラメータ値は、sqlnet.ora ファイルで指定された値よりも優先されます。

関連項目

- [『Oracle Database Net Servicesリファレンス・ガイド』](#)

親トピック: [Oracle Net Services接続のクイック・スタート](#)

6.6 データベースに接続するための簡易接続の使用

[「ネットワークの可用性の確認」](#)に関する項で説明しているように、ネットワーク接続が確認された場合、簡易接続ネーミング・メソッドを使用してデータベースに接続します。このネーミング・メソッドを利用すると、データベースへのTCP/IP接続を簡単に行えるようになります。これは、クライアントがデータベースのホスト名の他にオプションのポートとサービス名を使用してデータベースに接続できるようにすることで、[ホスト・ネーミング](#)・メソッドの機能を拡張します。簡易接続を使用して接続する構文は、次のとおりです。

```
CONNECT username/password@host[:port][/]service_name[:server_type][/]instance_name]
```



ノート:

Oracle Call Interface のマニュアルでは、server は connect_type と呼ばれています。

Oracle Databaseサーバーのインストールを「標準」モードで実行した場合、Oracleインスタンスで使用されるデフォルトのサービス名はデータベース名であり、次の簡易接続構文を使用してそのインスタンスに接続できます。

```
SQLPLUS /nolog
SQL> CONNECT username@"host/db_name"
SQL> Enter password: password
```

ノート:



Oracle Database では、Oracle Database 10g 時点で、Oracle Names の使用をサポートしていません。Oracle Database 12c のクライアントおよびデータベースでは、LDAP プロキシも含め、Oracle Names を使用して名前を解決できません。Oracle9i のクライアントでは、引き続き Oracle Names を使用して Oracle Database 12c データベースの名前を解決できます。ただし、Oracle Database 12c の新機能を活用するために、LDAP に移行することを強くお勧めします。

関連項目:

このメソッドの詳細は、[「簡易接続ネーミング・メソッドの理解」](#)を参照してください

親トピック: [Oracle Net Services接続のクイック・スタート](#)

7 Oracle Net Servicesの管理

Oracle Net Servicesの各種管理ツールについて説明します。これには、主要管理アプリケーションの[Oracle Enterprise Manager Cloud Control](#)と[Oracle Net Manager](#)が含まれます。また、コマンドライン制御ユーティリティについても説明します。

- [ユーザー・インタフェース・ツールの使用](#)
- [OracleNetAdminsグループについて](#)
- [リスナー制御ユーティリティを使用したリスナーの管理](#)
- [共通ネットワーク・タスクの実行](#)

親トピック: [Oracle Net Servicesの構成と管理](#)

7.1 ユーザー・インタフェース・ツールの使用

Oracle Net Servicesでは、構成および管理タスクの実行を支援するためのツールを提供します。

- [Oracle Enterprise Manager Cloud Controlを使用したOracle Net Servicesの構成](#)
- [Oracle Net Managerを使用したOracle Net Servicesの構成](#)
- [Oracle Enterprise Manager Cloud ControlおよびOracle Net Managerの使用時期の決定](#)
- [Oracle Net Configuration Assistantを使用したネットワーク・コンポーネントの構成](#)

親トピック: [Oracle Net Servicesの管理](#)

7.1.1 Oracle Enterprise Manager Cloud Controlを使用したOracle Net Servicesの構成

Oracle Enterprise Manager Cloud Controlを使用すると、複数のファイル・システムでOracleホーム向けのOracle Net Servicesを構成できます。また、リスナーに共通の管理機能を提供します。Oracle Enterprise Manager Cloud Controlでは、Oracle Net Servicesを構成および管理するための統合環境を提供します。

Oracle Enterprise Manager Cloud Controlを使用すると、複数のOracleホームから次を構成および管理できます。

- リスナー: リスナーを構成して、クライアント接続を受け取ります。
- ネーミング: [接続識別子](#)を定義し、それを[接続記述子](#)にマップし、サービスのネットワークの場所を識別します。Oracle Net Managerでは、ローカルtnsnames.oraファイルや集中化されたディレクトリ・サービスの接続記述子の構成をサポートします。
- ファイルの場所: Oracle Net構成ファイルの場所を指定します。
- [「Net Services管理」ページへのアクセス](#)

関連項目:

Oracle Enterprise Manager Cloud Controlの使用方法の詳細は、Oracle Enterprise Manager Cloud Controlのドキュメント・セットとオンライン・ヘルプを参照してください。

親トピック: [ユーザー・インタフェース・ツールの使用](#)

7.1.1.1 「Net Services管理」ページへのアクセス

次の手順では、Oracle Enterprise Manager Cloud Controlを使用して「Net Services管理」ページにアクセスする方法について説明します。

1. 「データベースにログイン」ページから、データベース資格証明を入力し、次に「ログイン」をクリックします。
「Enterprise Managerホームの選択」ページが表示されます。
2. 「ターゲット」メニューから「すべてのターゲット」を選択します。
3. 「検索の絞込み」リストから「リスナー」を選択します。
4. リスナー名をダブルクリックして、リスナーを選択します。
5. 「Oracleリスナー」メニューから「Net Services管理」を選択します。
「Net Services管理」ページが表示されます。

「Net Services管理」ページでは、リスナー、ネーミング・メソッド、作業環境などを管理できます。管理手順については、本書の他の章で説明しています。

親トピック: [Oracle Enterprise Manager Cloud Controlを使用したOracle Net Servicesの構成](#)

7.1.2 Oracle Net Managerを使用したOracle Net Servicesの構成

Oracle Net Managerでは、Oracle Net Servicesをローカル・クライアントやサーバー・ホスト上のOracleホーム向けに構成できます。

Oracle Net Manager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。

- 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。
- ネーミング: 接続識別子を定義して接続記述子にマップし、サービスのネットワークの場所と識別情報を識別します。Oracle Net Managerでは、ローカルtnsnames.oraファイルや集中化されたディレクトリ・サービスの接続記述子の構成をサポートします。
- 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方法を構成します。
- プロファイル: 作業環境を構成して、クライアントやサーバー上でOracle Net機能の有効化や構成を行います。

この項では、Oracle Net Managerの機能について説明します。ただし、Oracle Net Managerの使用方法の詳細は、オンライン・ヘルプを参照してください。

- [Oracle Net Managerの起動](#)
- [Oracle Net Managerのナビゲート](#)
- [Oracle Net Managerのウィザードの使用](#)

親トピック: [ユーザー・インタフェース・ツールの使用](#)

7.1.2.1 Oracle Net Managerの起動

Oracle Net Managerを起動するには、次のようにします。

- Linuxの場合、ORACLE_HOME/binディレクトリからnetmgrを実行します。
- Microsoft Windowsの場合、「スタート」メニューから「プログラム」を選択し、「Oracle - HOME_NAME」を選択します。次に、「Configuration and Migration Tools」を選択し、「Net Manager」を選択します。

親トピック: [Oracle Net Managerを使用したOracle Net Servicesの構成](#)

7.1.2.2 Oracle Net Managerのナビゲート

Oracle Net Managerのインタフェースには、ツールバー、メニュー項目、およびネットワーク・コンポーネントを構成するプロパティ・シートが含まれています。

ナビゲータ・ペインには、ネットワーク・オブジェクトとそれらが含むオブジェクトがフォルダ階層に編成されたツリー・ビューがあります。フォルダを展開したり折りたたんだりして、接続識別子、リスナー、プロファイルなどのオブジェクトを監視または管理できます。変更を加えるオブジェクトをクリックします。

ナビゲータ・ペインの主要なフォルダは、次のとおりです。

- ローカル
 - ローカルの構成ファイルに構成されている次のネットワーキング要素を表示します。
 - tnsnames.oraファイル内のネット・サービス名
 - listener.oraファイルのリスナー
 - sqlnet.oraファイルのプロファイル
- ディレクトリ
 - ディレクトリ・サーバーに構成されている接続識別子を表示します。

親トピック: [Oracle Net Managerを使用したOracle Net Servicesの構成](#)

7.1.2.3 Oracle Net Managerのウィザードの使用

Oracle Net Managerのウィザードを使用すれば、タスクの処理をステップ・バイ・ステップで実行できます。ウィザードは、複雑なタスクを簡単なステップに分けて、わかりやすく説明します。ウィザードはすべての構成オプションを提供できるようには作成されていません。ウィザードでタスクを完了したら、Oracle Net Managerのその他のコンポーネントを使用して構成を変更できます。

- [Netサービス名ウィザードの使用](#)
- [ディレクトリ・サーバー移行ウィザードの使用](#)

親トピック: [Oracle Net Managerを使用したOracle Net Servicesの構成](#)

7.1.2.3.1 Netサービス名ウィザードの使用

Netサービス名ウィザードに従えば、ディレクトリ・サーバーやtnsnames.oraファイルに基本的なネットワーク・サービス名を作成できます。

次の手順では、Netサービス名ウィザードを起動してネットワーク・サービス名を作成する方法について説明します。

1. ナビゲータ・ペインで、「ディレクトリ」または「ローカル」→「サービス・ネーミング」を選択します。
2. 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。

関連項目:

Netサービス名ウィザードを使用したネットワーク・サービス名の作成方法の詳細は、Oracle Net Managerのオンライン・ヘルプを参照してください。

親トピック: [Oracle Net Managerのウィザードの使用](#)

7.1.2.3.2 ディレクトリ・サーバー移行ウィザードの使用

tnsnames.oraファイルがすでに存在する場合、そのネットワーク・サービス名は、ディレクトリ・サーバー移行ウィザードを使用してディレクトリ・サーバーへエクスポートできます。

次の手順では、ディレクトリ・サーバー移行ウィザードを使用する方法について説明します。

1. 「ツール」メニューから「ディレクトリ」を選択します。
2. 「Oracle Net Manager」メニューから「ネット・サービス名のエクスポート」を選択します。

関連項目:

[「ローカル・ネーミング・エントリのディレクトリ・ネーミング・サーバーへのエクスポート」](#)

親トピック: [Oracle Net Managerのウィザードの使用](#)

7.1.3 Oracle Enterprise Manager Cloud ControlおよびOracle Net Managerの使用時期の決定

以前はOracle Net Managerのみで利用可能だった機能の多くがOracle Enterprise Manager Cloud Controlに統合されました。Oracle Enterprise Manager Cloud Controlでは、複数のファイル・システムで展開される複数のOracleホームの構成を管理する機能が提供されています。Oracle Net Managerでは、ローカル・ホスト・コンピュータ上の1台のOracleホームの構成を管理することのみが可能です。ツール間の主な相違点は、次のとおりです。

Oracle Enterprise Manager Cloud Control

- 次の機能を構成します。
 - ローカル・ネーミング(tnsnames.oraファイル)
 - ディレクトリ・ネーミング
 - リスナー
- 複数ファイル・システムのOracleホームのサポートを提供します。
- ローカルおよびディレクトリ・ネーミングのエントリを検索し、ソートする機能を提供します。
- ディレクトリ・ネーミング・エントリをtnsnames.oraファイルにエクスポートします。
- 次の選択したリスナーの管理タスクを実行します。
 - 現在のステータスの表示

- ステータスの変更
- トレースレベル設定の変更
- ログ設定の変更
- 複数のリスナーがある場合の接続時フェイルオーバーおよびロード・バランシング方法の設定

Oracle Net Manager

- 次の機能を構成します。
 - ローカル・ネーミング(tnsnames.oraファイル)
 - ディレクトリ・ネーミング
 - リスナー
 - プロファイル
- 単一ホスト用にOracleホーム・サポートを提供します。
- 複数のリスナーがある場合の接続時フェイルオーバーおよびロード・バランシング方法の設定
- クライアントとサーバーに次のオプションを設定します。
 - トレース設定
 - ログ設定
 - セキュリティ、認証およびアクセス権
 - ルーティング

ノート:



自動診断リポジトリ(ADR)が有効な場合、Oracle Enterprise Manager Cloud Control を使用してトレースやログ設定に加えた変更はシステムによって無視されます。

親トピック: [ユーザー・インタフェース・ツールの使用](#)

7.1.4 Oracle Net Configuration Assistantを使用したネットワーク・コンポーネントの構成

[Oracle Net Configuration Assistant](#)は、次のような基本的なネットワーク・コンポーネントをインストール時に構成します。

- 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を接続記述子に解決するため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・サーバーの使用

Oracle Databaseのインストレーション・ガイドに示すとおり、ソフトウェアのインストール中に、Oracle Net Configuration Assistantは自動的に実行されます。また、スタンドアロン・モードでインストール後に実行して、ネーミング・メソッド、リスナー、tnsnames.oraファイルのネットワーク・サービス名およびディレクトリ・サーバーの使用を構成できます。

Oracle Net Configuration Assistantを起動する手順は、次のとおりです。

- LinuxおよびUNIXの場合、ORACLE_HOME/binディレクトリからnetcaを実行します。
- Microsoft Windowsの場合、「スタート」メニューから「プログラム」を選択し、「Oracle - HOME_NAME」を選択します。次に、「コンフィグレーションおよび移行ツール」を選択し、「Oracle Net Configuration Assistant」を選択します。

関連項目:

- Oracle Net Configuration Assistantのオンライン・ヘルプ
- サイレント・モードでのOracle Net Configuration Assistantの実行の詳細は、[『Oracle Grid Infrastructure インストール・ガイド』](#)を参照してください

Oracle Net Configuration Assistantの「ようこそ」ページに表示される構成オプションは、次のとおりです。

- リスナー構成 - リスナーを作成、変更、削除、または名前変更します。
- ネーミング・メソッド構成: 次に示すネーミング・メソッドの1つまたは複数を使用して、接続識別子が接続記述子に解決されるようにコンピュータを構成します。
 - ローカル・ネーミング
 - ディレクトリ・ネーミング
 - 簡易接続ネーミング
 - 外部ネーミング
- ローカル・ネット・サービス名構成
 - ローカルのtnsnames.oraファイルに格納されている接続記述子の接続を作成、変更、名前変更またはテストします
- ディレクトリ使用構成 - ディレクトリ対応の機能について、ディレクトリ・サーバーを構成します。

親トピック: [ユーザー・インタフェース・ツールの使用](#)

7.2 OracleNetAdminsグループについて

Oracle Net Managerを使用するには、OracleNetAdminsグループまたはOracleContextAdminsグループのメンバーである必要があります。Oracle Net Configuration Assistantは、Oracleコンテキストの作成時に、これらのグループに対するこれらのアクセス権を作成します。

- [OracleNetAdminsグループへのユーザーの追加](#)
- [OracleNetAdminsグループからのユーザーの削除](#)
- [OracleNetAdminsグループの所有権の変更](#)

親トピック: [Oracle Net Servicesの管理](#)

7.2.1 OracleNetAdminsグループへのユーザーの追加

次の手順では、ldapmodifyコマンドを使用して、ユーザーをOracleNetAdminsグループに追加する方法について説明します。

1. OracleNetAdminsグループにユーザーを追加することを指定する[LDIF](#)(Lightweight Directory Interchange

Format)ファイルを作成します。

次のサンプルのLDIFファイルを使用できます。cn=OracleNetAdminsに対する適切なDNおよび追加するユーザーを使用します。

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

2. コマンドラインから次のコマンドを入力して、ファイルを更新します。

```
$ ldapmodify -h directory_host -p port -D binddn -q -f ldif_file
```

前述のコマンドで、directory_hostはディレクトリ・サーバー・ホスト、portはディレクトリ・サーバーのリスニングTCP/IPポート、binddnはディレクトリ管理者またはユーザーDN、ldif_fileは入力ファイル名です。ポートが指定されていない場合、デフォルト・ポート389が使用されます。-qオプションを指定すると、単一のバインド・パスワードを求められます。

親トピック: [OracleNetAdminsグループについて](#)

7.2.2 OracleNetAdminsグループからのユーザーの削除

次の手順では、ldapmodifyコマンドを使用して、OracleNetAdminsグループからユーザーを削除する方法について説明します。

1. OracleNetAdminsグループからユーザーを削除することを指定するLDIFファイルを作成します。

次のサンプルのLDIFファイルを使用できます。cn=OracleNetAdminsに対する適切なDNおよび削除するユーザーを入力します。

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

2. 次のコマンドを入力して、ユーザーを削除します。

```
$ ldapmodify -h directory_host -p port -D binddn -q -f ldif_file
```

前述のコマンドで、directory_hostはディレクトリ・サーバー・ホスト、portはディレクトリ・サーバーのリスニングTCP/IPポート、binddnはディレクトリ管理者またはユーザーDN、ldif_fileは入力ファイル名です。ポートが指定されていない場合、デフォルト・ポート389が使用されます。-qオプションを指定すると、単一のバインド・パスワードを求められます。

親トピック: [OracleNetAdminsグループについて](#)

7.2.3 OracleNetAdminsグループの所有権の変更

次の手順では、OracleNetAdminsグループの所有者としてグループを追加する方法について説明します。

1. 次の手順に従って、LDIFファイルを作成します。
 - a. 所有者として追加するグループを指定します。

次のサンプルのLDIFファイルを使用できます。cn=OracleNetAdminsに対する適切なDN、および追加するグループのDNを入力します。

```
dn: cn=OracleNetAdmins,cn=OracleContext,...
```

```
changetype: modify
add: owner
owner: DN of group to add
```

たとえば、次のLDIF構文では、所有権がOracleNetAdminsグループからExampleSecurityAdminsという名前のグループに変更されます。このグループは、Oracleコンテキストの内部でも外部でも構いません。

```
dn: cn=OracleNetAdmins, cn=OracleContext, ...
changetype: modify
add: owner
owner: cn=ExampleSecurityAdmins
```

b. (オプション)所有者として削除するグループを指定します。

```
dn: cn=OracleNetAdmins, cn=OracleContext, ...
changetype: modify
delete: owner
owner: DN of group to delete
```

2. コマンドラインから次のコマンドを入力して、ファイルを更新します。

```
$ ldapmodify -h directory_host -p port -D binddn -q -f ldif_file
```

前述のコマンドで、directory_hostはディレクトリ・サーバー・ホスト、portはディレクトリ・サーバーのリスニングTCP/IPポート、binddnはディレクトリ管理者またはユーザーDN、ldif_fileは入力ファイル名です。ポートが指定されていない場合、デフォルト・ポート389が使用されます。-qオプションを指定すると、単一のバインド・パスワードの入力を求められます。

親トピック: [OracleNetAdminsグループについて](#)

7.3 リスナー制御ユーティリティを使用したリスナーの管理

Oracle Net Servicesには、各ネットワーク・コンポーネントを容易に起動、停止、構成および制御できるツールがあります。リスナー制御ユーティリティでは、リスナーを管理できます。このユーティリティは、Oracleインストールを所有するユーザーまたは指定したグループのメンバーによって、リスナーを実行している同じコンピュータ上で起動されます。このユーティリティの基本構文は次のとおりです。

```
lsnrctl command [listener_name]
```

たとえば、次のコマンドは、リスナー名lsnrを起動します。

```
lsnrctl START lsnr
```

リスナー制御ユーティリティ・コマンドは、LSNRCTL>プログラムのプロンプトでも発行できます。プロンプトを取得するには、オペレーティング・システムのコマンドラインで引数を指定せずにlsnrctlを入力します。lsnrctlを実行すると、ユーティリティが起動され、プログラム・プロンプトから必要なコマンドを入力できます。

たとえば:

```
lsnrctl
LSNRCTL> START lsnr
```

関連項目:

- リスナーの詳細は、[「Oracle Net Listener構成のカスタマイズ」](#)を参照してください
- リスナー制御ユーティリティの詳細は、[『Oracle Database Net Servicesリファレンス』](#)を参照してください。

親トピック: [Oracle Net Servicesの管理](#)

7.4 共通ネットワーク・タスクの実行

このマニュアルでは、ネットワークの構成と管理タスクについて説明しています。

Oracle Netで使用するディレクトリ・サーバーの構成

Oracle Netのディレクトリ・サーバーを構成するためのタスクは、次のとおりです。

- ディレクトリ・サーバーの使用の構成
このタスクの実行に使用するツール – Oracle Internet Directoryコンフィギュレーション・アシスタント

関連項目:

[『Oracle Fusion Middleware Oracle Internet Directory管理者ガイド』](#)

- OracleNetAdminsグループへのユーザーの追加
このタスクの実行に使用するツール – ldapmodify

関連項目:

[「ディレクトリ・サーバーのエントリを追加または変更できるユーザー」](#)

- ディレクトリでの認証
このタスクの実行に使用するツール – Oracle Enterprise Manager Cloud Control、Oracle Net Manager

関連項目:

- Oracle Enterprise Manager Cloud Controlのオンライン・ヘルプ
- [『Oracle Databaseエンタープライズ・ユーザー・セキュリティ管理者ガイド』](#)
- Oracleコンテキストの変更
このタスクの実行に使用するツール – Oracle Net Manager

関連項目:

Oracle Net Managerのオンライン・ヘルプ

ネーミング・メソッドの構成

ネーミング・メソッドを構成するためのタスクは、次のとおりです。

- ローカル・ネーミング・メソッドの構成

このタスクの実行に使用するツール – Oracle Enterprise Manager Cloud Control、Oracle Net Manager、Oracle Net Configuration Assistant

関連項目:

[「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

このタスクの実行に使用するツール – Oracle Enterprise Manager Cloud Control、Oracle Net Manager

関連項目:

[「ディレクトリ・ネーミング・メソッドの構成」](#)

- 簡易接続ネーミング・メソッドの構成

このタスクの実行に使用するツール – Oracle Net Manager

関連項目:

[「簡易接続ネーミング・メソッドの理解」](#)

- 外部ネーミング・メソッドの構成

このタスクの実行に使用するツール – Oracle Net Manager

関連項目:

[「外部ネーミング・メソッドの構成」](#)

ディレクトリ・ネーミングへの移行

ディレクトリ・ネーミングに移行するためのタスクは、次のとおりです。

tnsnames.oraファイルからのエクスポート

このタスクの実行に使用するツール – Oracle Enterprise Manager Cloud Control、Oracle Net Manager

関連項目:

[「ディレクトリ・ネーミング・エントリのtnsnames.oraファイルへのエクスポート」](#)

プロファイルの構成

プロファイルを構成するためのタスクは、次のとおりです。

- ネーミング・メソッドの優先順位の指定

このタスクの実行に使用するツール – Oracle Net Manager、Oracle Net Configuration Assistant

関連項目:

[「ネーミング・メソッドの優先順位」](#)

- 未修飾ネットワーク・サービス名に自動的に追加されるデフォルト・ドメインの構成

このタスクの実行に使用するツール – Oracle Net Manager、Oracle Net Configuration Assistant

関連項目:

[「クライアントのデフォルト・ドメインについて」](#)

- 接続リクエストのルーティング

このタスクの実行に使用するツール – Oracle Net Manager、Oracle Net Configuration Assistant

関連項目:

[「プロセスへの接続要求のルーティング」](#)

- アクセス制御の構成

このタスクの実行に使用するツール – Oracle Net Manager

関連項目:

[「データベース・アクセス制御の設定」](#)

- 認証方式の構成。

このタスクの実行に使用するツール – Oracle Net Manager

関連項目:

[「Oracleネットワーク・セキュリティの構成」](#)

- 接続要求タイムアウトの構成

このタスクの実行に使用するツール – 手動構成

関連項目:

[「権限のないユーザーによるリソース使用の制限」](#)

リスナーの構成

リスナーを構成するためのタスクは、次のとおりです。

- リスニング・プロトコル・アドレスの構成

このタスクの実行に使用するツール – Oracle Enterprise Manager Cloud Control、Oracle Net Manager、Oracle Net Configuration Assistant

関連項目:

[「リスニング用プロトコル・アドレスの構成」](#)

- 動的なサービス登録の構成

このタスクの実行に使用するツール – 自動構成

関連項目:

[「動的なサービス登録の構成」](#)

- 静的なサービス登録の構成

このタスクの実行に使用するツール – Oracle Enterprise Manager Cloud Control、Oracle Net Manager

関連項目:

[「静的なサービス登録の理解」](#)

- 接続要求タイムアウトの構成

このタスクの実行に使用するツール – 手動構成

関連項目:

[「権限のないユーザーによるリソース使用の制限」](#)

リスナーの管理

リスナーを管理するためのタスクは、次のとおりです。

- リスナーの起動と停止

このタスクの実行に使用するツール – リスナー制御ユーティリティ

関連項目:

[「リスナーの起動と停止」](#)

- 登録情報の表示

このタスクの実行に使用するツール – リスナー制御ユーティリティ

関連項目:

[「リスナーのサービスの監視」](#)

Oracle Connection Managerの構成

Oracle Connection Managerを構成するためのタスクは、次のとおりです。

- セッション多重化の構成

このタスクの実行に使用するツール – 手動構成

関連項目:

[「Oracle Connection Managerのセッションの多重化の有効化」](#)

- アクセス制御の構成

このタスクの実行に使用するツール – 手動構成

関連項目:

[アクセス制御の有効化](#)

親トピック: [Oracle Net Servicesの管理](#)

8 ネーミング・メソッドの構成

データベース・サーバーに接続するクライアントの接続情報を構成する方法について説明します。

- [簡易接続ネーミング・メソッドの構成](#)
簡易接続ネーミング・メソッドは、データベースに対する簡単なTCP/IP接続を提供します。これは、クライアントがデータベースのホスト名の他にオプションのポートとサービス名を使用してデータベースに接続できるようにすることで、ホスト・ネーミング・メソッドの機能を拡張します。
- [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](#)
ディレクトリ・ネーミング・メソッドでは、Oracle Internet DirectoryやMicrosoft Active DirectoryなどのLDAP準拠ディレクトリ・サーバーに含まれる接続記述子に接続識別子をマップします。
- [ディレクトリ・ネーミング・サーバーでの複数のデフォルト・コンテキストの作成](#)
- [ローカル・ネーミング・エントリのディレクトリ・ネーミング・サーバーへのエクスポート](#)
- [ディレクトリ・ネーミング・エントリのtnsnames.oraファイルへのエクスポート](#)
- [外部ネーミング・メソッドの構成](#)

関連項目:

ネーミング・メソッドの概要については、[「ネーミング・メソッドの理解」](#)を参照してください

親トピック: [Oracle Net Servicesの構成と管理](#)

8.1 簡易接続ネーミング・メソッドの構成

簡易接続ネーミング・メソッドは、データベースへの簡単なTCP/IP接続を提供します。これは、クライアントがデータベースのホスト名の他にオプションのポートとサービス名を使用してデータベースに接続できるようにすることで、ホスト・ネーミング・メソッドの機能を拡張します。

- [簡易接続ネーミング・メソッドについて](#)
簡易接続ネーミング・メソッドを使用すると、TCP/IP環境のtnsnames.oraファイルでサービス名を参照する必要がなくなります。さらに、このメソッドを使用するとネーミング・システムやディレクトリ・システムは必要ありません。
- [簡易接続プラスについて](#)
Oracle Database 19cリリースから、Oracle Databaseへの接続にアプリケーションで使用する簡易接続構文が機能強化されました。新しいバージョンは簡易接続プラスと呼ばれます。
- [簡易接続ネーミング・メソッドの例](#)
例では、簡易接続ネーミング構文を示し、各文字列がどのように接続記述子に変換されるかを示しています。
- [クライアントでの簡易接続ネーミングの構成](#)
クライアントが簡易接続ネーミング・メソッドを使用する前に確認する必要がある要件および構成タスクについて学習します。
- [DNS別名を使用するための簡易接続ネーミングの構成](#)
ホスト・ネーミング・メソッドで提供されているように、必要に応じてホスト名にDNS別名を構成できます。

親トピック: [ネーミング・メソッドの構成](#)

8.1.1 簡易接続ネーミング・メソッドについて

簡易接続ネーミング・メソッドを使用すると、TCP/IP環境で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。さらに、このメソッドを使用するとネーミング・システムやディレクトリ・システムは必要ありません。

概要

このネーミング・メソッドは、クライアントがデータベースのホスト名の他にオプションのポートとサービス名を使用してデータベースに接続できるようにすることで、ホスト・ネーミング・メソッドの機能を拡張します。

```
CONNECT username@[//]host[:port] [/[service_name][:server_type] [/instance_name]]
Enter password: password
```

この接続識別子は、次のような接続記述子に変換されます。

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp) (HOST=host) (PORT=port))
  (CONNECT_DATA=
    (SERVICE_NAME=service_name)
    (SERVER=server_type)
    (INSTANCE_NAME=instance_name))
)
```

Oracle Databaseサーバーのインストールを「標準」モードで実行した場合、Oracleインスタンスで使用されるデフォルトのサービス名はデータベース名で、次の簡易接続構文を使用してそのインスタンスに接続できます。

```
SQLPLUS /nolog
SQL> CONNECT username@host/db_name
SQL> Enter password: password
```

簡易接続構文の例

次の例の接続文字列は、データベース・サーバーsales-server上のリスニング・エンドポイント1521を使用して、クライアントをデータベース・サービスのsales.us.example.comに接続します。

```
CONNECT scott@sales-server:1521/sales.us.example.com
CONNECT scott@//sales-server/sales.us.example.com
CONNECT scott@//sales-server.us.example.com/sales.us.example.com
```

各接続文字列の後に、データベース・サービスに接続するためのパスワードを入力する必要があります。

これらの接続文字列は、次の接続記述子に変換されます。

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.example.com))
)
```

簡易接続構文の要素と説明

表8-1 簡易接続ネーミング・メソッドのための接続識別子

構文要素	説明
//	<p>//を使用して、URL または JDBC 接続を指定します。</p> <p>URL または JDBC 接続の場合、必須です。接続識別子の前にダブル・スラッシュ(//)を付ける必要があります。たとえば:</p> <pre>scott@//sales-server Enter password: password</pre> <p>SQL 接続の場合、オプションです。接続識別子の前にダブル・スラッシュ(//)を付けることができます。たとえば、次の接続文字列は意味的に同じです。</p> <pre>SQL> CONNECT scott@sales-server SQL> CONNECT scott@//sales-server</pre>
host	<p>必須。データベース・ホスト・コンピュータのホスト名または IP アドレスを指定します。</p> <p>ローカル・オペレーティング・システム構成でドメイン指定のときは、ホスト名は、ドメインで修飾されます。</p> <p>値として IPv4 または IPv6 アドレスを使用できます。IPv6 アドレスまたは IPv6 アドレスに解決されるホスト名は、[2001:0db8:0:0::200C:417A]や[salesdb]のように大括弧で囲む必要があります。</p>
port	<p>省略可能。リスニング・ポートを指定します。</p> <p>デフォルトは、1521 です。</p>
service_name	<p>省略可能。データベースのサービス名を指定します。</p> <p>ユーザーがサービス名を指定した場合、リスナーはその特定のデータベースにユーザーを接続します。サービス名を指定しない場合、リスナーは listener.ora ファイルの DEFAULT_SERVICE_listener_name パラメータで指定されたデータベースに接続します。リスナーについて DEFAULT_SERVICE_listener_name が設定されておらず、ユーザーがサービス名を簡易接続構文の一部として明示的に指定していない場合、リスナーはエラーを戻します。</p>
server_type	<p>省略可能。使用するデータベース・サーバーのタイプを指定します。</p> <p>このパラメータを指定すると、リスナーはクライアントを特定のタイプのサービス・ハンドラに接続します。</p> <p>server_type パラメータの値は、dedicated、shared および pooled です。簡易接続構文でサーバーが指定されていない場合、リスナーによってサーバーのタイプが選択さ</p>

構文要素	説明
	<p>れます(共有サーバーが構成済の場合は共有サーバー、それ以外の場合は専用サーバーが使用されます)。</p> <p>ノート: Oracle Call Interface のマニュアルでは、server は connect_type と呼ばれています。</p>
instance_name	<p>省略可能。アクセスするデータベース・インスタンスを識別します。</p> <p>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。</p>

関連項目

- [DEFAULT_SERVICE_listener_name](#)
- [INSTANCE_NAME](#)

親トピック: [簡易接続ネーミング・メソッドの構成](#)

8.1.2 簡易接続プラスについて

Oracle Database 19cリリースから、Oracle Databaseへの接続にアプリケーションで使用する簡易接続構文が機能強化されました。新しいバージョンは簡易接続プラスと呼ばれます。

簡易接続プラスでは、Oracle Databaseのアプリケーション構成と一般的なユースケースのデプロイメントが簡略化されます。簡易接続プラスでは、tnsnames.oraやsqlnet.oraなどのOracle Netパラメータ・ファイルを構成する必要がなくなりました。簡易接続プラスでは、TNS_ADMIN環境変数を設定する必要がなくなりました。

この新しい機能によって、Oracle Database Cloud Servicesへのクライアント接続がネットワーク・セキュリティにTLSを使用するため、クライアント構成が簡単になります。新しい構文は次のとおりです。

```
[[protocol: ]//]host1 {, host12} [:port1] {, host2:port2} [/ [service_name] [:server] [/instance_name]] [parameter_name=value [&parameter_name=value]]
```

疑問符(?)は名前/値ペアの始まりを示します。アンパサンド(&)は名前/値ペアの間のデリミタです。

プロトコルの指定のサポート: 簡易接続アダプタは、接続文字列の一部としてプロトコルの指定をサポートします。このプロトコルは、接続文字列内の各ホストに適用されます。

マルチホストまたはポート・サポート: 簡易接続アダプタは、接続文字列内の複数のホストまたはポートを受け入れることができるようになりました。これは、クライアント接続のロード・バランシングに役立ちます。

名前/値ペア: 簡易接続アダプタは、名前と値のペアのリストを受け入れることができます。名前/値の各ペアは、DESCRIPTIONレベルのパラメータとして追加されます。次の名前がサポートされています。

- ENABLE
- FAILOVER
- LOAD_BALANCE

- RECV_BUF_SIZE
- SEND_BUF_SIZE
- SDU
- SOURCE_ROUTE
- RETRY_COUNT
- RETRY_DELAY
- CONNECT_TIMEOUT
- TRANSPORT_CONNECT_TIMEOUT

たとえば、次の構文はセッション・データ・ユニット(SDU)を指定します

```
salesserver1:1521/sales.us.example.com?sdu=16384
```

次のような接続記述子に変換されます。

```
(DESCRIPTION=
(SDU=16384)
(ADDRESS=(PROTOCOL=tcp) (HOST=salesserver1) (PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))
```

同様に、次の構文は接続タイムアウト、トランスポート接続タイムアウト、および再試行回数の値を指定します

```
salesserver1:1521/sales.us.example.com?connect_timeout=60&transport_connect_timeout=30&retry_count=3
```

次のような接続記述子に変換されます。

```
(DESCRIPTION=
(retry_count=3)
(connect_timeout=60) (transport_connect_timeout=30)
(ADDRESS=(PROTOCOL=tcp) (HOST=salesserver1) (PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))
```

セキュリティ属性：次のSECURITY属性がTLSでサポートされています。

- SSL_SERVER_DN_MATCH=on/off
- SSL_SERVER_CERT_DN=longDN
- WALLET_LOCATION=Wallet location

親トピック: [簡易接続ネーミング・メソッドの構成](#)

8.1.3 簡易接続ネーミング・メソッドの例

例は、簡易接続ネーミングの構文と、各文字列がどのように接続記述子に変換されるかを示しています。

表8-2 簡易接続ネーミングの例

ネーミング・オプション	接続文字列	接続記述子
ホスト指定の簡易接続文字列。	sales-server	(DESCRIPTION= (CONNECT_DATA= (SERVICE_NAME=)) (ADDRESS= (PROTOCOL=TCP) (HOST=sales-server) (PORT=1521)))
ホスト名は sales-server です。		

ネーミング・オプション	接続文字列	接続記述子
<p>ホストおよびポート指定の簡易接続文字列。</p> <p>ホスト名は sales-server、ポートは 3456 です。</p>	sales-server:3456	(DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=))(ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=3456)))
<p>ホストおよびサービス名指定の簡易接続文字列。</p> <p>ホスト名は sales-server、サービス名は sales です。</p>	sales-server/sales	(DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=sales))(ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521)))
<p>IPv6 アドレス指定の簡易接続文字列。</p> <p>ホストの IPv6 アドレスは 2001:0db8:0:0::200c:417a、ポートは 80、サービス名は sales です。</p>	<p>[2001:0db8:0:0::200c:417a]:80/sales</p> <p>IPv6 ホスト名は大括弧で囲む必要があります。</p>	(DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=sales)(ADDRESS=(PROTOCOL=TCP)(HOST=2001:0db8:0:0::200c:417a)(PORT=80)))
<p>IPv6 ホスト・アドレス指定の簡易接続文字列。</p> <p>ホストは sales-server、ポートは 80、サービス名は sales です。</p>	sales-server:80/sales	(DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=sales)(ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=80)))
<p>ホスト、サービス名およびサーバー指定の簡易接続文字列。</p> <p>ホスト名は sales-server、サービス名は sales、サーバーは dedicated、インスタンス名は inst1 です。</p>	sales-server/sales:dedicated/inst1	(DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=sales)(INSTANCE_NAME=inst1)(SERVER=dedicated))(ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521)))
<p>ホストおよびインスタンス名指定の簡易接続。</p> <p>ホスト名は sales-server、インスタンス名は inst1 です。</p>	sales-server//inst1	(DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=)(INSTANCE_NAME=inst1))(ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521)))
<p>ノート: 簡易接続プラス機能で</p>	salesserver1:1521/sales?SDU=8128&re	(DESCRIPTION=

ネーミング・オプション	接続文字列	接続記述子
<p>は、このネーミング・オプションがサポートされています。</p> <p>名前/値ペアのリストがある簡易接続アダプタ。</p> <p>SDU、RETRY_COUNT、CONNECT_TIMEOUT</p> <p>ホストは salesserver、ポートは 1521、サービス名は sales です。</p>	try_count=3&connect_timeout=10	<pre>(SDU=8128) (retry_count=3) (connect_timeout=10) (AADDRESS=(PROTOCOL=tcp) (HOST=saleserver1) (PORT=1521)) (CONNECT_DATA=(SERVICE_NAME=sales)))</pre>
<p>ノート: 簡易接続プラス機能では、このネーミング・オプションがサポートされています。</p> <p>接続文字列に複数のホストまたはポートがある簡易接続アダプタ</p> <p>ホストは salesserver、ポートは 1521、サービス名は sales です。</p>	salesserver1:1521, salesserver2, salesserver3:1522/sales	<pre>((DESCRIPTION=(LOAD_BALANCE=ON) (AADDRESS=(PROTOCOL=tcp) (HOST=salesserver1) (PORT=1521)) (AADDRESS=(PROTOCOL=tcp) (HOST=salesserver2) (PORT=1522)) (AADDRESS=(PROTOCOL=tcp) (HOST=salesserver3) (PORT=1522))) (CONNECT_DATA=(SERVICE_NAME=sales)))</pre>
<p>ノート: 簡易接続プラス機能では、このネーミング・オプションがサポートされています。</p> <p>接続文字列の一部にプロトコルの指定がある簡易接続アダプタ。</p> <p>ホストは salesserver、ポートは 1521、サービス名は sales です。</p>	tcps://salesserver1:1521/sales	<pre>(DESCRIPTION= (AADDRESS=(PROTOCOL=tcps) (HOST=salesserver1) (PORT=1521)) (SEcurity=(SSL_SERVER_DN_MATCH=TRUE)) (CONNECT_DATA=(SERVICE_NAME=sales)))</pre>
<p>ノート: 簡易接続プラス機能では、このネーミング・オプションがサポートされています。</p> <p>次の SECURITY 属性が</p>	tcps://sales-server:1521/sales?ssl_server_cert_dn="cn=sales, cn=OracleContext, dc=us, dc=example, dc=com"&wallet_location="/tmp/oracle"	<pre>(DESCRIPTION= (AADDRESS=(PROTOCOL=tcps) (HOST=salesserver) (PORT=1521)) (CONNECT_DATA=(SERVICE_NAME=sales)) (SEcurity=(SSL_SERVER_DN_MATCH=TRUE) (SSL_SERVER_CERT_DN=cn=sales, cn=OracleContext, dc=us, dc=example, dc=com)))</pre>

ネーミング・オプション	接続文字列	接続記述子
) (WALLET_LOCATION=/tmp/oracle))
	ホストは sales-server、ポートは 1521、サービス名は sales です。	

関連項目

- [TCP/IPプロトコルについて](#)
- [リスニング用プロトコル・アドレスの構成](#)
- [IPv4およびIPv6のブリッジとしてのOracle Connection Managerの使用](#)

親トピック: [簡易接続ネーミング・メソッドの構成](#)

8.1.4 クライアントでの簡易接続ネーミングの構成

クライアントが簡易接続ネーミング・メソッドを使用する前に確認する必要がある要件および構成タスクについて学習します。

次の条件が満たされると、クライアントは、簡易接続ネーミングを使用して、Oracle Databaseに接続できます。

- Oracle Net Servicesソフトウェアがクライアントにインストールされている。
- Oracle TCP/IPプロトコルは、クライアントとデータベース・サーバーの両方でサポートされている。
- 拡張接続記述子を必要とする機能がない。

[外部プロシージャ・コール](#)、[異機種間サービス](#)などの追加の接続情報が必要な拡張機能を備えた大規模で複雑な環境には、簡易接続ネーミングは不向きです。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。

簡易接続ネーミングは、インストール時に自動的に構成されます。使用前には、EZCONNECTがsqlnet.oraファイルのNAMES.DIRECTORY_PATHパラメータで指定されていることを確認してください。このパラメータは、Oracle Netが接続識別子を接続記述子に解決する際に使用できるネーミング・メソッドの順序を指定します。

ノート:

Oracle Database 19c リリース 2 (19.2)以降、プロトコルが TCPS に設定されている接続については、簡易接続の使用時にデフォルトで SSL_SERVER_DN_MATCH が ON に設定されます。SSL_SERVER_CERT_DN が設定されていない場合は、次の順序で部分 DN 一致を実行することで、クライアントからサーバーへの接続が正常に確立されます。

1. 接続文字列のホスト名がサーバー証明書のホスト名と照合されます。
2. 接続文字列のサービス名がサーバー証明書内のサービス名と照合されます。

SSL_SERVER_CERT_DN が設定されている場合は、完全 DN 一致によってクライアントからサーバーへの接続が正常に確立されます。

次の手順では、簡易接続ネーミング・メソッドが構成されていることを確認する方法について説明します。

1. Oracle Net Managerを起動します。
2. ナビゲータ・ペインで、「ローカル」を展開し、次に「プロファイル」を選択します。
3. 右ペインのリストから、「ネーミング」を選択します。
4. 「メソッド」タブをクリックします。

「選択メソッド」リストにEZCONNECTがリストされていることを確認します。リストされていない場合は、ステップ5に進みます。リストされている場合は、ステップ7に進みます。

5. 「使用可能なメソッド」リストから「EZCONNECT」を選択し、右矢印ボタンをクリックします。
6. 「選択メソッド」リストで「EZCONNECT」を選択し、「上へ」ボタンを使用して選択済項目をリストの最上位に移動します。
7. 「ファイル」メニューから「ネットワーク構成の保存」を選択します。

sqlnet.oraファイルは、NAMES.DIRECTORY_PATHパラメータで更新され、次のようにhostnameが最初に表示されます。

```
NAMES.DIRECTORY_PATH=(ezconnect, tnsnames)
```

関連項目

- [Oracle Net Managerを使用したOracle Net Servicesの構成](#)

親トピック: [簡易接続ネーミング・メソッドの構成](#)

8.1.5 DNS別名を使用するための簡易接続ネーミングの構成

ホスト・ネーミング・メソッドで提供されているように、ホスト名にDNS別名をオプションとして構成できます。

ホスト・ネーミングでは、クライアントは、次のパターンを使用する接続文字列を使用します。

```
CONNECT username@DNS_alias
Enter password: password
```

次の手順では、DNS別名の構成方法について説明します。

1. リスナーにデータベース・サービスが登録されていることを確認します。

データベースがリスナーを検出できる場合は、サービス名を含むデータベース・サービスに関する情報は、[サービス登録中](#)に動的にリスナーに登録されます。リスナーは、次の条件が満たされる場合に検出されます。

- デフォルトの名前がLISTENERのリスナーが、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

データベースでリスナーが見つからない場合は、リスナーの静的登録を構成できます。

2. ホスト名解決環境を確立します。

DNS、NISまたは集中管理されているTCP/IPホスト・ファイルの/etc/hostsなどのメカニズムを構成できます。たとえば、あるデータベースのsales.us.example.comというサービス名が、sales-serverという名前のコンピュータ上に存在する場合、/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
192.0.2.35                 sales-server   sales.us.example.com
```

サービ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。

3. DNSの別名を使用してデータベースに接続します。

前のステップの例を使用すると、クライアントは、次のように接続文字列でsales.example.comを使用できます。

```
CONNECT username@sales.us.example.com
Enter password: password
```

クライアントとサーバーが同じus.example.comなどの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。

関連項目

- [リスナーの静的サービス情報の構成](#)

親トピック: [簡易接続ネーミング・メソッドの構成](#)

8.2 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・メソッドでは、ネットワーク・サービス名をtnsnames.oraファイルに追加します。各ネットワーク・サービス名は、接続記述子にマップされます。

[例8-1](#)は、DESCRIPTIONに含まれる接続記述子にマップされるネットワーク・サービス名salesを示しています。

DESCRIPTIONセクションにはプロトコル・アドレスが含まれており、接続先データベース・サービスを識別します。この例では、プロトコルはTCP/IP、ポートは1521です。

例8-1 ホスト名を含む接続記述子

```
sales=
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.example.com)))
```

[例8-2](#)は、IPv6アドレスおよびポート番号1522で識別されるホストに接続するための有効なtnsnames.oraエントリを示しています。

例8-2 IPv6アドレスを含む接続記述子

```
salesdb =
( DESCRIPTION =
  ( ADDRESS=(PROTOCOL=tcp) (HOST=2001:0db8:1:1::200c:417a) (PORT=1522) )
  ( CONNECT_DATA =
    (SERVICES_NAME=sales.example.com) )
)
```

次の項目で説明するように、ローカル・ネーミングは、インストール中または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](#)

関連項目:

親トピック: [ネーミング・メソッドの構成](#)

8.2.1 インストール中のtnsnames.oraファイル構成

[Oracle Net Configuration Assistant](#)を使用すると、クライアントに対するネットワーク・サービス名の構成が可能となります。

Oracle Universal Installerは、ソフトウェアのインストール後にOracle Net Configuration Assistantを起動します。

構成は、インストール・モードによって異なります。

- 管理者またはランタイム・インストール: Oracle Net Configuration Assistantでは、Oracle Database サービスへ接続するために、tnsnames.oraファイルにネットワーク・サービス名を構成するように求められます。
- カスタム・インストール: Oracle Net Configuration Assistantでは、使用するネーミング・メソッドを選択するように求められます。ローカル・ネーミングを選択した場合、Oracle Net Configuration Assistantでは、Oracle Databaseサービスへ接続するために、tnsnames.oraファイル中のネットワーク・サービス名を構成するように求められます。

親トピック: [ローカル・ネーミング・メソッドの構成](#)

8.2.2 インストール後のtnsnames.oraファイル構成

ネットワーク・サービス名は、インストール後、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。

- [タスク1「ネット・サービス名の構成」](#)
- [タスク2「第1番目のネーミング・メソッドとしてのローカル・ネーミングの構成」](#)
- [タスク3「他のクライアントへの構成のコピー」](#)
- [タスク4「リスナーの構成」](#)
- [タスク5「データベースへの接続」](#)

ノート:



Oracle Net との接続を構成する前に、基礎となるネットワーク接続が稼働している必要があります。

タスク 1 ネット・サービス名の構成

ネットワーク・サービス名を構成するには、次のいずれかの方法を使用します。

- [Oracle Enterprise Manager Cloud Control を使用したネット・サービス名の構成](#)
- [Oracle Net Manager を使用したネット・サービス名の構成](#)
- [Oracle Net Configuration Assistant を使用したネット・サービス名の構成](#)

それぞれの方法で同様の機能が提供されます。ただし、Oracle Net Manager には sqlnet.ora ファイルでの追加の構成オプションがあります。

- Oracle Enterprise Manager Cloud Control を使用したネット・サービス名の構成

次の手順では、Oracle Enterprise Manager Cloud Control を使用して tnsnames.ora ファイルのネットワーク・サービス名を構成する方法について説明します。

1. Oracle Enterprise Manager Cloud Control の「Net Services 管理」ページにアクセスします。

関連項目:

[「Net Services 管理」ページへのアクセス](#)

2. 「管理」リストから「ローカル・ネーミング」を選択し、構成ファイルの場所が含まれる Oracle ホームを選択します。
3. 「ローカル・ネーミング」ページが表示されます。データベース・サーバーへのログインを求められる場合があります。
4. 「類似作成」をクリックします。

「ネット・サービス名の作成」ページが表示されます。

5. 「ネット・サービス名」フィールドに名前を入力します。

ネットワーク・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 ファイルのパラメータ NAMES.DEFAULT_DOMAIN が設定されていると、ネットワーク・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:

[「クライアントのデフォルト・ドメインについて」](#)

6. 「データベース情報」セクションで次のようにサービス・サポートを設定します。
 - a. 宛先サービス名を入力します。

関連項目:

使用するサービス名文字列の詳細は、[「接続記述子について」](#)を参照

- b. データベース接続タイプを選択します。

接続タイプには「データベースのデフォルト」のデフォルト設定をお勧めします。[専用サーバー](#)が初期化パラメータ・ファイルで構成されている場合、「専用サーバー」を選択し、共有サーバー構成を使用せずに、リスナーに専用サーバーを起動させることができます。共有サーバーが初期化パラメータ・ファイルで構成され、接続時に常に共有サーバー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:

共有サーバーの構成の詳細は、「[共有サーバー・アーキテクチャの構成](#)」を参照してください。

7. アドレス・セクションで次のようにプロトコル・サポートを設定します。

- a. 「追加」をクリックします。

「アドレスの追加」ページが表示されます。

- b. 「プロトコル」リストから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。
- c. 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・パラメータの設定の詳細は、『[Oracle Database Net Services リファレンス](#)』を参照してください。

- d. (オプション)「拡張パラメータ」セクションを使用して、このセッションにおける送受信操作に必要な I/O バッファ・スペース制限を「合計送信バッファ・サイズ」および「合計受信バッファ・サイズ」フィールドに指定します。

関連項目:

バッファ・スペースの詳細は、[「I/O バッファ・スペースの構成」](#)を参照してください

- e. 「OK」をクリックします。

プロトコル・アドレスは、アドレス・セクションに追加されます。

8. 「OK」をクリックして、ネットワーク・サービス名を追加します。

ネットワーク・サービス名が「ローカル・ネーミング」ページに追加されます。

9. アドレスに対して接続時フェイルオーバーおよびクライアント・ロード・バランシング・オプションを選択します。

10. 「OK」をクリックします。

関連項目:

11. 複数のプロトコル・アドレスを構成する方法については、[「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

12. CONNECT_DATA オプションを追加して構成するには、[「拡張接続データ・パラメータについて」](#)を参照してください

- Oracle Net Manager を使用したネット・サービス名の構成

次の手順では、Oracle Net Manager を使用して tnsnames.ora ファイルのネットワーク・サービス名を構成する方法について説明します。

1. Oracle Net Manager を起動します。

関連項目:

[「Oracle Net Manager を使用した Oracle Net Services の構成」](#)

2. ナビゲータ・ペインで、「ローカル」メニューから「サービス・ネーミング」を選択します。

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 サービス名ウィザード」の「ようこそ」ページが表示されます。

4. 「ネット・サービス名」フィールドに名前を入力します。

ネットワーク・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 ファイルのパラメータ NAMES.DEFAULT_DOMAIN が設定されていると、ネットワーク・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:

[「クライアントのデフォルト・ドメインについて」](#)

5. 「次へ」をクリックします。

「プロトコル」ページが表示されます。

6. 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。

7. 「次へ」をクリックします。

「プロトコル設定」ページが表示されます。

8. 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・パラメータの設定の詳細は、[『Oracle Database Net Services リファレンス』](#)を参照してください。

9. 「次へ」をクリックします。

「サービス」ページが表示されます。

10. 宛先サービス名を入力し、オプションでデータベース接続タイプを選択します。

オラクル社では、接続タイプには「データベースのデフォルト」のデフォルト設定の使用をお薦

めします。専用サーバーが初期化パラメータ・ファイルで構成されている場合、「専用サーバー」を選択し、共有サーバー構成を使用せずに、リスナーに専用サーバーを起動させることができます。共有サーバーが初期化パラメータ・ファイルで構成され、接続時に常に共有サーバー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:

- a. 共有サーバーの構成の詳細は、[「共有サーバー・アーキテクチャの構成」](#)を参照してください
- b. 使用するサービス名文字列の詳細は、[「接続記述子について」](#)を参照

11. 「次へ」をクリックします。

「テスト」ページが表示されます。

12. ネットワーク・サービス名が機能するかどうかを確認するには、「テスト」をクリックし、Net サービス名ウィザードを終了するには、「終了」をクリックします。

「テスト」をクリックすると、Oracle Net は構成済の接続記述子情報を使用して、データベース・サーバーに接続します。このため、リスナーとデータベースを実行していなければテストを正常に実行できません。実行されていない場合は、テストの前に、[「Oracle Net Listener と Oracle Database サーバーの起動」](#)を参照してコンポーネントを起動してください。テスト中、ステータスとテスト結果を示す「接続テスト」ダイアログ・ボックスが表示されます。テストが成功すると、次のメッセージが表示されます。

```
The connection test was successful.
```

テストが成功した場合は、「閉じる」をクリックして「接続テスト」ダイアログ・ボックスを閉じてからステップ 13 に進みます。

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。

- a. 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、「テスト」をクリックします。
- b. 「ログインの変更」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、「テスト」をクリックします。

13. 「終了」をクリックして、「Net サービス名ウィザード」を終了します。

14. 「ファイル」メニューから「ネットワーク構成の保存」を選択します。

関連項目:

- a. 複数のプロトコル・アドレスを構成する方法については、[「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
- b. CONNECT_DATA オプションを追加して構成するには、[「拡張接続データ・パラメータについて」](#)を参照してください

● Oracle Net Configuration Assistant を使用したネット・サービス名の構成

次の手順では、Oracle Net Configuration Assistant を使用して tnsnames.ora ファイルのネットワーク・サービス名を構成する方法について説明します。

1. Oracle Net Configuration Assistant を起動します。

関連項目:

[「Oracle Net Configuration Assistant を使用したネットワーク・コンポーネントの構成」](#)

「ようこそ」ページが表示されます。

2. 「ローカル・ネット・サービス名構成」を選択してから、「次へ」をクリックします。

「ネット・サービス名の構成」ページが表示されます。

3. 「追加」をクリックし、「次へ」をクリックします。

「ネット・サービス名の構成」ページが表示されます。

4. 「サービス名」フィールドにネット・サービス名を入力します。

5. 「次へ」をクリックします。

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

タスク 2 第 1 番目のネーミング・メソッドとしてのローカル・ネーミングの構成

sqlnet.ora ファイルの NAMES.DIRECTORY_PATH パラメータで第 1 番目のネーミング・メソッドとして指定するローカル・ネーミングを構成します。このパラメータは、Oracle Net が接続識別子を接続記述子に解決する際に使用するネーミング・メソッドの順序を指定します。

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・メソッドを構成するには、次のいずれかの方法を使用します。

- [Oracle Enterprise Manager Cloud Control を使用したローカル・ネーミングの構成](#)
- [Oracle Net Manager を使用したローカル・ネーミングの構成](#)

それぞれの方法で同じ機能が提供されます。

Oracle Enterprise Manager Cloud Control を使用したローカル・ネーミングの構成

次の手順では、Oracle Enterprise Manager Cloud Control を使用して第 1 番目のネーミング・メソッドとしてローカル・ネーミングを指定する方法について説明します。

1. Oracle Enterprise Manager Cloud Control の「Net Services 管理」ページにアクセスします。

関連項目:

[「Net Services 管理」ページへのアクセス](#)

2. 「管理」リストから「ネットワーク・プロファイル」を選択します。
3. 「実行」をクリックします。
4. 「ネーミング・メソッド」を選択します。
5. 「使用可能なメソッド」リストから「TNSNAMES」を選択します。
6. 「移動」をクリックして、選択したメソッドを「選択したメソッド」リストに移動します。
7. 「昇格」ボタンを使用して TNSNAMES をリストの最上位に移動します。
8. 「OK」をクリックします。

Oracle Net Manager を使用したローカル・ネーミングの構成

次の手順では、Oracle Net Manager を使用して第 1 番目のネーミング・メソッドとしてローカル・ネーミングを指定する方法について説明します。

1. Oracle Net Manager を起動します。

関連項目:

[「Oracle Net Manager を使用した Oracle Net Services の構成」](#)

2. ナビゲータ・ペインで、「ローカル」メニューから「プロファイル」を選択します。
3. 右ペインのリストから、「ネーミング」を選択します。
4. 「メソッド」タブをクリックします。
5. 「使用可能なメソッド」リストから「TNSNAMES」を選択し、右矢印ボタンをクリックします。
6. 「選択メソッド」リストで「TNSNAMES」を選択し、「上へ」ボタンを使用して選択済項目をリストの最上位に移動します。
7. 「ファイル」メニューから「ネットワーク構成の保存」を選択します。

sqlnet.ora ファイルは、NAMES.DIRECTORY_PATH パラメータで更新され、次のように tnsnames が最初に表示されます。

```
NAMES.DIRECTORY_PATH=(tnsnames, EZCONNECT)
```

タスク 3 他のクライアントへの構成のコピー

クライアントを 1 台構成してから、tnsnames.ora と sqlnet.ora の構成ファイルを他のクライアントの同じ場所に単純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。その他に、Oracle Net Assistant をすべてのクライアント上で使用できます。

タスク 4 リスナーの構成

サーバー上のリスナーは、ネットワーク・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構成されています。

関連項目:

リスナー構成の詳細は、[「Oracle Net Listener の構成と管理」](#)を参照してください

タスク 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@net_service_name
```

親トピック: [ローカル・ネーミング・メソッドの構成](#)

8.3 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、Oracle Internet DirectoryやMicrosoft Active DirectoryなどのLDAP準拠ディレクトリ・サーバーに含まれる接続記述子に接続識別子をマップします。

ディレクトリでは、データベース・サービスおよびネットワーク・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・エントリはインストール時に作成されます。ネットワーク・サービス名のエントリと[ネットワーク・サービス別名](#)のエントリを作成および変更し、データベース・サービス・エントリを変更するには、Oracle Enterprise Manager Cloud ControlおよびOracle Net Managerを使用します。クライアントは、データベースへの接続にこれらのエントリを使用できます。

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。

- [タスク1「ディレクトリのバージョンの互換性の検証」](#)
- [タスク2「ネット・サービス名のディレクトリへの作成」](#)
- [タスク3「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」](#)
- [タスク4「ネット・サービス別名の作成」](#)
- [タスク5「クライアント検索の第1番目のネーミング・メソッドとしてのLDAPの構成」](#)
- [タスク6「リスナーの構成」](#)
- [タスク7「データベースへの接続」](#)

タスク 1 ディレクトリのバージョンの互換性の検証

ネットワーク・サービス名を作成しようとしているコンピュータから次の検証ステップを実行します。

1. コンピュータに Oracle Net Service ソフトウェアの最新リリースがインストールされていることを確認します。リリース情報は、「ヘルプ」メニューの「情報」オプションにあります。
2. Oracle Internet Directory コンフィギュレーション・アシスタントを実行して、ディレクトリ・サー

バー、Oracle コンテキスト、Oracle スキーマのリリースを確認します。

関連項目:

ディレクトリ・サーバーの使用の構成の詳細は、[『Oracle Fusion Middleware Oracle Internet Directory 管理者ガイド』](#)を参照してください。

タスク 2 ネット・サービス名のディレクトリへの作成

データベース・サービス・エントリではなくネットワーク・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。次の手順では、ネットワーク・サービス名の作成方法について説明します。

ノート:



- OracleNetAdmins グループまたは OracleContextAdmins グループのいずれか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ワーク・サービス名エントリを作成できます。OracleNetAdmins グループにユーザーを追加したり削除したりするには、[「ディレクトリ・サーバーのエントリを追加または変更できるユーザー」](#)を参照してください。
- 既存のネットワーク・サービス名は、tnsnames.ora ファイルからエクスポートできます。[「ローカル・ネーミング・エントリのディレクトリ・ネーミング・サーバーへのエクスポート」](#)を参照してください。

1. Oracle Enterprise Manager Cloud Control の「Net Services 管理」ページにアクセスします。

関連項目:

[「Net Services 管理」ページへのアクセス](#)

2. 「管理」リストから「ディレクトリ・ネーミング」を選択し、ディレクトリ・サーバーのある Oracle ホームを選択します。
3. 「実行」をクリックします。

「ディレクトリ・ネーミング」ページが表示されます。

4. 「ネット・サービス名」タブをクリックします。

5. 結果セクションの「作成」をクリックします。

「一般」タブのある「ネット・サービス名の作成」ページが表示されます。

6. 「ネット・サービス名」フィールドに名前を入力します。

7. データベース情報セクションで次のようにサービス・サポートを設定します。

a. 宛先サービス名を入力します。

宛先サービス名が Oracle9i データベース以降の場合は、「サービス名の使用」を選択し、「サービス名」フィールドにサービス名を入力します。

関連項目:

使用するサービス名文字列の詳細は、[「接続記述子について」](#)を参照

b. データベース接続タイプを選択します。オラクル社では、接続タイプには「データベースのデフォルト」の使用をお勧めします。初期化パラメータ・ファイルで共有サーバーが構成されている場合は、次のオプションを使用できます。

- 共有サーバー構成を使用せずに、リスナーに専用サーバーを起動させるには、「専用サーバー」を選択します。
- 接続時に常に共有サーバーが使用されるようにするには、「共有サーバー」を選択します。

関連項目:

共有サーバーの構成の詳細は、[「共有サーバー・アーキテクチャの構成」](#)を参照してください

8. アドレス・セクションで次のようにプロトコル・サポートを設定します。

a. 「追加」をクリックします。

「アドレスの追加」ページが表示されます。

b. 「プロトコル」リストから、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。

- c. 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・パラメータの設定の詳細は、[『Oracle Database Net Services リファレンス』](#)を参照してください。

- d. (オプション)「拡張パラメータ」セクションを使用して、このセッションにおける送受信操作に必要な I/O バッファ・スペース制限を「合計送信バッファ・サイズ」および「合計受信バッファ・サイズ」フィールドに指定します。

関連項目:

詳細は、[「I/O バッファ・スペースの構成」](#)を参照してください

- e. 「OK」をクリックします。

プロトコル・アドレスは、アドレス・セクションに追加されます。

9. 「OK」をクリックして、ネットワーク・サービス名を追加します。

ネットワーク・サービス名が「ネット・サービス名」タブの結果セクションに追加されます。

関連項目:

- a. 複数のプロトコル・アドレスを構成する方法については、[「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
- b. CONNECT_DATA オプションを追加して構成するには、[「拡張接続データ・パラメータについて」](#)を参照してください

タスク 3 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

ディレクトリ・ネーミングへのデータベース登録が完了すると、ディレクトリにデータベース・サービス・エントリが作成されます。デフォルトでは、このエントリにはプロトコル・アドレスで表されたリスナーの位置を含む、ネットワーク・ルート情報が含まれています。この情報を再作成したり、既存のネットワーク・ルート

情報を変更できます。

ノート:



OracleNetAdmins グループまたは OracleContextAdmins グループのメンバーであるユーザーのみが、ディレクトリ内のデータベース・サービスに関するネットワーク情報を変更できます。これらのグループにユーザーを追加したり削除したりするには、[「ディレクトリ・サーバーのエントリを追加または変更できるユーザー」](#)を参照してください。

次の手順では、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方法について説明します。

1. Oracle Enterprise Manager Cloud Control の「Net Services 管理」ページにアクセスします。

関連項目:

[「Net Services 管理」ページへのアクセス](#)

2. 「管理」リストから「ディレクトリ・ネーミング」を選択し、ディレクトリ・サーバーのある Oracle ホームを選択します。
3. 「実行」をクリックします。データベース・サーバーおよびディレクトリ・サーバーへのログインを求められる場合があります。

「ディレクトリ・ネーミング」ページが表示されます。

4. 「データベース・サービス」タブをクリックします。
5. 「簡易検索」セクションで「Oracle コンテキスト」と検索基準を選択し、Oracle コンテキストのネットワーク・サービス名を確認します。

データベース・サービス名が結果セクションに表示されます。

6. 結果セクションでデータベース・サービスを選択し、「編集」をクリックします。

タスク 4 ネット・サービス別名の作成

ディレクトリ・サーバーのネット・サービス別名によって、クライアントはデータベース・サービスまたはネット

ワーク・サービス名を代替名で参照できます。たとえば、sales というネットワーク・サービス名に対して salesalias というネットワーク・サービス別名を作成できます。salesalias を CONNECT scott@salesalias として使用してデータベースに接続すると、sales の接続記述子情報に解決されて使用されます。

ネットワーク・サービス別名の主な使用方法は、次の 2 つです。

- クライアントがデータベース・サービスまたはネットワーク・サービス名を別の名前でも参照する手段としてネットワーク・サービス別名を使用する方法。
- データベース・サービスの 1 つの Oracle コンテキストでネットワーク・サービス別名を使用し、別の Oracle コンテキストでネットワーク・サービス名を使用する方法。この方法によって、データベース・サービスまたはネットワーク・サービス名をディレクトリ・サーバーで一度定義すると、他の Oracle コンテキストを使用するクライアントでも参照できます。

関連項目:

ネットワーク・サービス別名の概要については、[「ネット・サービス別名のエントリの理解」](#)を参照してください

ノート:

- OracleNetAdmins グループまたは OracleContextAdmins グループのいずれか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ワーク・サービス別名エントリを作成または変更できます。OracleNetAdmins グループにユーザーを追加したり削除したりするには、[「ディレクトリ・サーバーのエントリを追加または変更できるユーザー」](#)を参照してください。
- ネットワーク・サービス別名を作成したり、アクセスするには、Oracle ホームが少なくともリリース 9.2.0.4 であることを確認してください。
- Microsoft Active Directory ではネット・サービス別名はサポートされていません。
- ネットワーク・サービス別名を使用する場合は、クライアントに対して NLS_LANG 環境変数が設定されていることを確認してください。

ネットワーク・サービス別名を作成するには、次のいずれかの方法を使用します。

- [Oracle Enterprise Manager Cloud Control を使用したネットワーク・サービス別名の構成](#)
- [Oracle Net Manager を使用したネットワーク・サービス別名の構成](#)

それぞれの方法で同様の機能が提供されます。

Oracle Enterprise Manager Cloud Control を使用したネットワーク・サービス別名の構成

次の手順では、Oracle Enterprise Manager Cloud Control を使用してネットワーク・サービス別名を構成する方法について説明します。

1. Oracle Enterprise Manager Cloud Control の「Net Services 管理」ページにアクセスします。

関連項目:

[「Net Services 管理」ページへのアクセス](#)

2. 「管理」リストから「ディレクトリ・ネーミング」を選択し、ディレクトリ・サーバーのある Oracle ホームを選択します。

3. 「実行」をクリックします。

「ディレクトリ・ネーミング」ページが表示されます。

4. 「ネット・サービス別名」タブをクリックします。

5. 結果セクションの「作成」をクリックします。

「ネット・サービス別名の作成」ページが表示されます。

6. 「ネット・サービス別名」フィールドに別名として名前を入力します。

7. 参照サービスの詳細セクションで、フィールドに次の情報を入力します。

- Oracle コンテキスト: データベース・サービスまたはネットワーク・サービス名の Oracle コンテキストを、リストから選択するかフィールドに入力します。
- 参照サービス名: データベース・サービスの DN またはネットワーク・サービス名を選択します。

8. 「OK」をクリックして、ネットワーク・サービス別名を追加します。

ネットワーク・サービス別名が「ディレクトリ・ネーミング」ページに追加されます。

Oracle Net Manager を使用したネットワーク・サービス別名の構成

次の手順では、Oracle Net Manager を使用してネットワーク・サービス別名を構成する方法について説明します。

1. Oracle Net Manager を起動します。

関連項目:

[「Oracle Net Manager を使用した Oracle Net Services の構成」](#)

2. ナビゲータ・ペインで、「ディレクトリ」メニューから「サービス・ネーミング」を選択します。
3. 「別名」を選択します。
4. 「編集」メニューから「作成」を選択します。
5. 「ネット・サービス別名」フィールドにネットワーク・サービス別名を入力します。
6. Oracle コンテキストと名前を選択します。
7. 「作成」をクリックします。
8. 「ファイル」メニューから「ネットワーク構成の保存」を選択します。

タスク 5 クライアント検索の第 1 番目のネーミング・メソッドとしての LDAP の構成

sqlnet.ora ファイルの NAMES.DIRECTORY_PATH パラメータで、使用する第 1 番目のネーミング・メソッドとしてディレクトリ・ネーミングを構成します。このパラメータは、Oracle Net が接続識別子を接続記述子に解決する際に使用するネーミング・メソッドの順序を指定します。LDAP を第 1 番目のネーミング・メソッドとして構成するには、次のいずれかの方法を使用できます。

- [Oracle Enterprise Manager Cloud Control を使用した LDAP の構成](#)
- [Oracle Net Manager を使用した LDAP の構成](#)

Oracle Enterprise Manager Cloud Control を使用した LDAP の構成

次の手順では、Oracle Enterprise Manager Cloud Control を使用して第 1 番目のネーミング・メソッドとしてディレクトリ・ネーミングを指定する方法について説明します。

1. Oracle Enterprise Manager Cloud Control の「Net Services 管理」ページにアクセスします。

関連項目:

[「\[Net Services 管理\]ページへのアクセス」](#)

2. 「管理」リストから「ネットワーク・プロファイル」を選択します。
3. 「実行」をクリックします。
4. 「ネーミング・メソッド」を選択します。
5. 「使用可能なメソッド」リストから「LDAP」を選択します。
6. 「移動」をクリックして、選択したメソッドを「選択したメソッド」リストに移動します。
7. 「昇格」ボタンを使用して LDAP をリストの最上位に移動します。
8. 「OK」をクリックします。

Oracle Net Manager を使用した LDAP の構成

次の手順では、Oracle Net Manager を使用して第 1 番目のネーミング・メソッドとしてディレクトリ・ネーミングを指定する方法について説明します。

1. Oracle Net Manager を起動します。

関連項目:

[「Oracle Net Manager を使用した Oracle Net Services の構成」](#)

2. ナビゲータ・ペインで、「ローカル」メニューから「プロファイル」を選択します。
3. 右ペインのリストから、「ネーミング」を選択します。
4. 「メソッド」タブをクリックします。
5. 「使用可能なメソッド」リストから「LDAP」を選択し、右矢印ボタンをクリックします。
6. 「選択メソッド」リストで「LDAP」を選択し、「上へ」ボタンを使用して選択済項目をリストの最上位に移動します。

7. 「ファイル」メニューから「ネットワーク構成の保存」を選択します。

sqlnet.ora ファイルは、NAMES.DIRECTORY_PATH パラメータで更新され、次のように ldap が最初に表示されます。

```
NAMES.DIRECTORY_PATH=(ldap, tnsnames, hostname)
```

タスク 6 リスナーの構成

サーバー上のリスナーは、ネットワーク・サービス名用に構成したものと同じプロトコル・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 プロトコルでリスニングするように構成されています。

関連項目:

リスナー構成の詳細は、[「Oracle Net Listener の構成と管理」](#)を参照してください

タスク 7 データベースへの接続

データベース・サービスやネットワーク・サービス名のディレクトリの場所と一致するディレクトリ・エントリがデフォルトに構成されているクライアントは、次の構文を使用してデータベースに接続できます。

```
CONNECT username@connect_identifier
```

エントリのディレクトリの場所と一致しないディレクトリ・エントリがデフォルトに構成されているクライアントは、エントリの識別名または完全修飾名を使用する必要があります。

関連項目:

完全修飾名の使用方法については、[「ディレクトリ情報ツリーの理解」](#)を参照してください

親トピック: [ネーミング・メソッドの構成](#)

8.4 ディレクトリ・ネーミング・サーバーでの複数のデフォルト・コンテキストの作成

クライアントに、複数の Oracle コンテキストが含まれるディレクトリでの検出機能を使用させる必要がある場合は、基本の admin

コンテキストに `orclCommonContextMap` 属性を定義できます。この属性は、`orclDefaultSubscriber` 属性をオーバーライドします。名前参照を実行すると、検出操作によって2つの属性値が戻され、クライアントはそれらの値に基づいて、使用する Oracle コンテキストを決定します。

`orclCommonContextMap` 属性が定義されていない場合、`orclDefaultSubscriber` がデフォルトとして使用されます。`orclCommonContextMap` が定義されている場合、クライアントは、DNSドメインに関連付けられているデフォルトの Oracle コンテキストを `orclCommonContextMap` から検索します。複数のデフォルト・コンテキストを有効にするには、デフォルトの `oracleContext` として使用するドメインと DN 間の関連付けリストを付けて、`orclCommonContextMap` を定義します。たとえば、LDIF ファイルのエントリは次のようになります。

```
$ ldapmodify -v -h sales-server -p 1389 -D cn=orcladmin -q
dn: cn=Common, cn=Products, cn=OracleContext
replace: orclCommonContextMap
orclCommonContextMap:
(contextMap=
(domain_map=(domain=us.example.com) (DN="dc=example, dc=com"))
(domain_map=(domain=uk.example.com) (DN="dc=sales, dc=com"))
)
```

`contextMap` エントリは、改行を含めずに入力する必要があります。

関連項目:

コンテキスト・マッピングのディレクトリを構成する方法の詳細は、[『Oracle Fusion Middleware Oracle Internet Directory 管理者ガイド』](#)を参照してください。

親トピック: [ネーミング・メソッドの構成](#)

8.5 ローカル・ネーミング・エントリのディレクトリ・ネーミング・サーバーへのエクスポート

`tnsnames.ora` ファイルがすでに存在する場合、そのネットワーク・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。エクスポート手順は、一度に1つのドメインに対して実行します。

この項では、`tnsnames.ora` ファイルに格納されたデータをディレクトリ・サーバーにエクスポートする方法を説明します。この項で説明するタスクは、次のとおりです。

- [タスク1「ディレクトリ・サーバーでの構造の作成」](#)
- [タスク2「Oracle コンテキストの作成」](#)
- [タスク3「ディレクトリ・サーバーの使用の構成」](#)
- [タスク4「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」](#)

ノート:



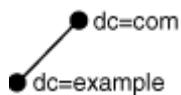
これらのタスクでは、ディレクトリ・サーバーがインストールされて実行中であると想定しています。

タスク 1 ディレクトリ・サーバーでの構造の作成

ディレクトリ・サーバーで、ネットワーク・サービス名をインポートする構造の[ディレクトリ情報ツリー\(DIT\)](#)を作成します。[Oracle コンテキスト](#)の先頭までの構造を作成します。

たとえば、tnsnames.ora ファイルがドメイン構造 example.com をサポートしていて、このドメインをディレクトリにレプリケートする場合は、[図 8-1](#)に示すように、このディレクトリに dc=com および dc=example のドメイン・コンポーネント・エントリを作成します。

図8-1 ディレクトリ・サーバー内のexample.com



現在 tnsnames.ora で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、完全に異なる構造を作成することもできます。完全に異なる構造を導入すると、クライアントが接続文字列にネットワーク・サービス名を入力する方法が変更されます。構造を変更する前に、[相対ネーミング](#)および[完全修飾ネーミング](#)を検討することをお勧めします。

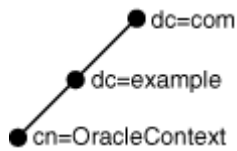
関連項目:

- ディレクトリ・エントリの構成方法については、ディレクトリ・サーバーのベンダーのマニュアル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クライアントの接続」](#)

タスク 2 Oracle コンテキストの作成

Oracle Internet Directory コンフィギュレーション・アシスタントを使用して、タスク 1 で作成した各 DIT の位置の下に Oracle コンテキストを作成します。Oracle コンテキストには、cn=0racleContext の[相対識別名\(RDN\)](#)があります。Oracle コンテキストには、その他の Oracle コンポーネントのエントリとともに、ネットワーク・オブジェクト・エントリが格納されます。[図 8-2](#)の cn=0racleContext は、dc=example, dc=com の下に作成されます。

図8-2 Oracleコンテキスト



関連項目:

- Oracle コンテキストの詳細は、[「ネットワーク・アドレス情報の管理」](#)を参照してください
- Oracle コンテキストの作成手順は、[『Oracle Fusion Middleware Oracle Internet Directory 管理者ガイド』](#)を参照してください。

タスク 3 ディレクトリ・サーバーの使用の構成

Oracle コンテキストを作成する際に作業の一部として構成が終了していない場合、Oracle ホームをディレクトリ・サーバーで使用できるように構成します。Oracle ホームは、エクスポートを実行できる構成にする必要があります。

関連項目:

ディレクトリ・サーバーの使用の構成の詳細は、[『Oracle Fusion Middleware Oracle Internet Directory 管理者ガイド』](#)を参照してください。

タスク 4 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

tnsnames.ora ファイルに含まれるネットワーク・サービス名をエクスポートするには、Oracle Enterprise Manager Cloud Control または Oracle Net Manager のいずれかを使用します。

- Oracle Enterprise Manager Cloud Control を使用したオブジェクトのエクスポート

次の手順では、Oracle Enterprise Manager Cloud Control を使用してオブジェクトをエクスポートする方法について説明します。

1. Oracle Enterprise Manager Cloud Control の「Net Services 管理」ページにアクセスします。

関連項目:

[「Net Services 管理」ページへのアクセス](#)

2. 「管理」リストから「ディレクトリ・ネーミング」を選択し、ディレクトリ・サーバーのある Oracle ホームを選択します。

3. 「実行」をクリックします。

「ディレクトリ・ネーミング」ページが表示されます。

4. 「ネット・サービス名」タブをクリックします。

5. 関連リンク・セクションで「ディレクトリ・サーバーへのネット・サービス名のインポート」をクリックします。

「ディレクトリ・サーバーへのネット・サービス名のインポート」ページが表示されます。

6. 「Oracle Internet Directory Server 接続先」セクションの「Oracle コンテキスト」リストから、選択したネットワーク・サービス名のエクスポート先である Oracle コンテキストを選択します。

7. 「インポートするネット・サービス名」セクションで、ネットワーク・サービス名を選択します。

8. 「追加」をクリックして、ネットワーク・サービス名をディレクトリに追加します。

ネットワーク・サービス名が「ディレクトリ・ネーミング」ページに追加されます。

● Oracle Net Manager を使用したオブジェクトのエクスポート

次の手順では、Oracle Net Manager を使用してオブジェクトをエクスポートする方法について説明します。

1. Oracle Net Manager を起動します。

関連項目:

[「Oracle Net Manager を使用した Oracle Net Services の構成」](#)

2. エクスポートする tnsnames. ora ファイルが Oracle Net Manager にロードされていない場合は、「ファイル」メニューから「ネットワーク構成を開く」を選択して、ディレクトリにエクスポートする tnsnames. ora ファイルを選択します。

3. 「ツール」メニューから「ディレクトリ」を選択し、「ネット・サービス名のエクスポート」を選択します。

ディレクトリ・サーバー移行ウィザードが起動します。

4. 「次へ」をクリックします。

tnsnames.ora ファイルで、複数ドメインを持つネットワーク・サービス名が検出された場合は、「ドメインの選択」ページが表示されます。ステップ 5 に進みます。

ネットワーク・サービス名がドメインで修飾されていない場合、「ネット・サービス名の選択」ページが表示されます。ステップ [6](#) に進みます。

5. エクスポート用のネットワーク・サービス名のネットワーク・ドメインを選択して、「次へ」をクリックします。

「ネット・サービス名の選択」ページが表示されます。

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

「接続先コンテキストの選択」ページが表示されます。

7. 「接続先コンテキストの選択」ページで、次を実行します。

- a. 「ディレクトリ・ネーミング・コンテキスト」リストから、Oracle コンテキストを含むディレクトリ・エントリを選択します。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・サブツリーの一部分です。

- b. 「Oracle コンテキスト」リストから、選択したネットワーク・サービス名のエクスポート先である Oracle コンテキストを選択します。

- c. 「次へ」をクリックします。

「ディレクトリ・サーバーの更新」ページがエクスポート操作のステータスで表示されます。

8. 「終了」をクリックして、ディレクトリ・サーバー移行ウィザードを終了します。

親トピック: [ネーミング・メソッドの構成](#)

8.6 ディレクトリ・ネーミング・エントリのtnsnames.oraファイルへのエクスポート

ディレクトリ・ネーミング・エントリの作成後、そのエントリをローカルのtnsnames.oraファイルにエクスポートし、さらにそのファイルをクライアントに配布します。ディレクトリ・サーバーが一時的に使用できないときは、クライアントはローカルに保存したファイルを使

用できます。

次の手順では、ローカルtnsnames.oraファイルにディレクトリ・ネーミングのエントリをエクスポートする方法について説明します。

1. Oracle Enterprise Manager Cloud Controlの「Net Services管理」ページにアクセスします。

関連項目:

[「Net Services管理」ページへのアクセス](#)

2. 「管理」リストから「ディレクトリ・ネーミング」を選択し、ディレクトリ・サーバーのあるOracleホームを選択します。
3. 「実行」をクリックします。
「ディレクトリ・ネーミング」ページが表示されます。
4. 「ネット・サービス名」タブをクリックします。
5. 「簡易検索」セクションで「Oracleコンテキスト」と検索基準を選択し、特定のOracleコンテキストのネットワーク・サービス名を確認します。
ネットワーク・サービス名が結果セクションに表示されます。
6. 結果セクションで、「tnsnames.oraに保存」をクリックします。
「処理中: tnsnames.oraファイルの作成」ページが表示され、作成中であることが通知されます。

親トピック: [ネーミング・メソッドの構成](#)

8.7 外部ネーミング・メソッドの構成

外部ネーミングとは、Network Information Service (NIS)のような、サード・パーティのネーミング・サービスに格納されるネットワーク・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。システム・インフラストラクチャの一部としてNISを使用している組織や企業のために、NIS外部ネーミングを使用してNIS内にネットワーク・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。

たとえば、ユーザーがネットワーク・サービス名(payroll)に次のようなコマンドを入力する場合があります。

```
SQLPLUS scott@payroll
```

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データベース・サーバー)を実行するノード上のNIS外部ネーミングは、ネットワーク上にあるNISサーバーに接続し、ネットワーク・サービス名をNISサーバーに渡します。NISサーバーはネットワーク・サービス名をOracle Netアドレスに解決し、このアドレスをクライアント・プログラムまたは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 Databaseに接続します。

NISサーバーは、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プログラムでは、[マップ](#)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 Databaseサービス名は、tnsnamesと呼ばれるマップに格納されます。

ユーザーが接続文字列を使用すると、NIS外部ネーミングはRPCコールを使用してypservプログラムに接続し、Oracleネットワーク・サービス名とマップの名前を渡します。ypservプログラムは、payrollなどの名前とネットワーク・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。

ノート:

一部のプラットフォームでは、NIS 外部ネーミング・メソッドを使用できません。adapters コマンドを使用し、システムで NIS 外部ネーミングを使用できるかどうかを確認してください。使用できる場合、次のように Oracle Net ネーミング・メソッドの下にリストされます。



```
$ adapters
Installed Oracle Net naming methods are:
Local Naming (tnsnames.ora)
Oracle Directory Naming
Oracle Host Naming
NIS Naming
```

詳細は、Oracle プラットフォーム固有のマニュアルを参照してください。

この項の内容は次のとおりです。

- [タスク1「NIS外部ネーミングをサポートするようにNISサーバーを構成」](#)
- [タスク2「クライアントの構成」](#)

タスク 1 NIS 外部ネーミングをサポートするように NIS サーバーを構成

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 Database ネットワーク・サービス名を解決する必要がある 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。このタスクを完了するには、tnsnames マップを既存の NIS マップに追加し、tnsnames マップが正しくインストールされていることを確認してください。

1. [「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。

ノート:



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、ネットワーク・サービス名を NIS マップにロードする必要があります。

2. 次のようなコマンドを使用して tns2nis プログラムを使用し、tnsnames.ora ファイルの内容を tnsnames マップに変換します。

```
tns2nis tnsnames.ora
```

tns2nis プログラムは、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ファイルの場所を指定します(/etc/tnsnames.ora や ORACLE_HOME/network/admin/tnsnames.ora など)。

これにより tnsnames マップが、現行の作業ディレクトリに書き込まれます。



ノート:

tns2nis プログラムは、NIS 外部ネーミングで提供されます。

3. tnsnames マップを 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。



ノート:

このステップは、NIS の管理担当者が実行する必要があります。

makedbm プログラムは、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、オペレーティング・システムによって異なります。

たとえば、Linux で tnsnames マップを生成してインストールするには、root ユーザー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
```

関連項目:

詳細は、Oracle のオペレーティング・システム固有のマニュアルを参照してください。

5. 次のコマンドを使用して、tnsnames が正しくインストールされていることを確認します。

```
ypmatch net_service_name tnsnames
```

たとえば、次のコマンドを入力できます。

```
ypmatch example.com tnsnames
```

次のようにアドレスの長さ(文字数)の付いたアドレスが戻されます。

```
99 (description=(address=(protocol=tcp) (host=sales) (port=1999)))  
    (connect_data=(service_name=dirprod)))
```

タスク 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。このパラメータは、Oracle Net が接続識別子を接続記述子に解決する際に使用できるネーミング・メソッドの順序を指定します。

1. Oracle Net Manager を起動します。

関連項目:

[「Oracle Net Manager を使用した Oracle Net Services の構成」](#)

2. ナビゲータ・ペインで、「ローカル」メニューから「プロファイル」を選択します。
3. 右ペインのリストから、「ネーミング」を選択します。
4. 「メソッド」タブをクリックします。
5. 「使用可能なメソッド」リストから「NIS」を選択し、右矢印ボタンをクリックします。
6. 「選択メソッド」リストで「NIS」を選択し、「上へ」ボタンを使用して選択済項目をリストの最上位に移動します。
7. 「ファイル」メニューから「ネットワーク構成の保存」を選択します。

sqlnet.ora ファイルは、NAMES.DIRECTORY_PATH パラメータで更新され、次のように nis が最初に表示されます。

```
NAMES.DIRECTORY_PATH=(nis, hostname, tnsnames)
```

親トピック: [ネーミング・メソッドの構成](#)

9 Oracle Net Listenerの構成と管理

Oracle Net Listenerは、データベース・サーバーで実行される独立したプロセスです。着信中のクライアント接続要求を受信し、データベース・サーバーに対する要求の通信量を管理します。クライアント接続を受け入れるようにリスナーを構成する方法について説明します。

ノート:



Oracle Database 12c リリース 2 (12.2)では、リスナーのパスワード機能はサポートされなくなりました。認証はローカル・オペレーティング・システムの認証機能によって実施されるため、セキュリティが失われることはありません。

- [Oracle Net Listenerの概要](#)
- [動的なサービス登録の構成](#)
- [インストール中のOracle Net Listenerの構成](#)
- [Oracle Net Listener構成のカスタマイズ](#)
- [リスナーの管理](#)
- [リスナーのリダイレクトの理解](#)

関連項目:

- 初期の接続要求時のリスナーの使用方法は、[「データベースの識別とアクセス」](#)を参照してください
- リスナーの概要は、[「通信レイヤーの理解」](#)を参照してください

親トピック: [Oracle Net Servicesの構成と管理](#)

9.1 Oracle Net Listenerの概要

ノート:



リスナーのリリースは、そのリスナーを介してサービスの提供を受けるすべての Oracle Database の最新のリリースと同等以上である必要があります。

リスナーは、1つ以上のリスニング・プロトコル・アドレス、サポートされるサービスについての情報および実行時の動作を制御するパラメータから構成されます。リスナーの構成は、listener.oraとネーミングされた構成ファイルに格納されます。

構成パラメータにはデフォルト値が設定されているため、構成前のリスナーを起動して使用できます。デフォルトのリスナーはLISTENERとネーミングされ、起動時はサービスがサポートされていないため次のTCP/IPプロトコル・アドレスのリスニングを行います。

```
(ADDRESS=(PROTOCOL=tcp) (HOST=host_name) (PORT=1521))
```

リスナーはクライアント要求をサポートされているサービスに転送します。これらのサービスは、リスナーに動的に登録されます。この

動的登録機能は、[サービス登録](#)と呼ばれます。登録は、リスナー登録(LREG)プロセスによって実行されます。動的サービス登録では、listener.oraファイルでの手動による構成は不要です。

サービス登録には次の利点があります。

- [接続時フェイルオーバー](#)
リスナーはインスタンスの状態を常時監視しているため、サービス登録では、あるインスタンスがダウンした場合に、別のインスタンスに対するクライアントの接続要求の自動フェイルオーバーが利用できます。
- [接続ロード・バランシング](#)
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インスタンスと[ディスパッチャ](#)、または[専用サーバー](#)に転送します。サービス登録によって、[サービス・ハンドラ](#)およびノード全体の負荷がバランスされます。
- Oracle Real Application ClustersおよびOracle Data Guardの高い可用性

関連項目:

- [「Oracle Netアーキテクチャの理解」](#)
- [「動的なサービス登録の構成」](#)
- [「アドレス・リスト・パラメータについて」](#)
- [「接続ロード・バランシングの理解」](#)

親トピック: [Oracle Net Listenerの構成と管理](#)

9.2 動的なサービス登録の構成

サービス登録によって、Oracle Databaseなどのプロセスはリスナーに対して使用できるサービスを識別できます。サービス登録は、これらのサービスのポート・マップとなります。リスナーは、サービス登録から受信したデータベースやインスタンスの動的なサービス情報を使用します。

動的なサービス登録は、データベース初期化ファイルで構成されています。listener.oraファイルでの構成は必要ありません。ただし、データベース初期化ファイルで指定したポートでリスニングするようにリスナー構成を設定する必要があります。COSTパラメータなどの自動登録を妨げるパラメータは設定しないでください。

この項で説明する、サービス登録に関連する構成に関する項目は、次のとおりです。

- [サービス登録の初期化パラメータの設定](#)
- [ローカル・リスナーへの情報の登録](#)
- [リモート・リスナーへの情報の登録](#)
- [ネットワーク内の全リスナーへの情報の登録](#)
- [ネーミング・メソッドの構成](#)

親トピック: [Oracle Net Listenerの構成と管理](#)

9.2.1 サービス登録の初期化パラメータの設定

サービス登録を正しく機能させるには、初期化パラメータ・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
- ローカル・リスナーのLOCAL_LISTENER
- リモート・リスナーのREMOTE_LISTENER(ある場合)
- フォワード・リスナーのFORWARD_LISTENER

たとえば:

```
SERVICE_NAMES=sales.us.example.com  
INSTANCE_NAME=sales
```

SERVICE_NAMESパラメータのデフォルト値には、初期化パラメータ・ファイルのDB_NAMEとDB_DOMAINパラメータからなる[グローバル・データベース名](#)が設定されています。INSTANCE_NAMEパラメータのデフォルト値は、Oracleシステム識別子(SID)です。

ノート:



Oracle Database 19c 以降、SERVICE_NAMES パラメータをお客様が使用することは非推奨になりました。サービスを管理するには、SRVCTL または GDSCTL コマンドライン・ユーティリティ、または DBMS_SERVICE パッケージを使用することをお勧めします。

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、[『Oracle Databaseリファレンス』](#)を参照してください。

親トピック: [動的なサービス登録の構成](#)

9.2.2 ローカル・リスナーへの情報の登録

デフォルトのLREGプロセスでは、TCP/IP、ポート1521のデフォルトのローカル・アドレス上のローカル・リスナーにサービス情報が登録されます。リスナー構成がデータベース構成と同期している場合、LREGでは、別のノード上の非デフォルトであるローカル・リスナーまたはリモート・リスナーにサービス情報を登録できます。同期が行われるのは、listener.oraファイルでリスナーのプロトコル・アドレスが指定され、初期化パラメータ・ファイルでリスナーの場所が指定されている場合です。

TCP/IPまたはポート1521を使用しないローカル・リスナーへLREGプロセス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。

共有サーバー環境では、初期化パラメータ・ファイルのDISPATCHERSパラメータのLISTENER属性を使用して、ディスパッチャを非デフォルトのローカル・リスナーに登録できます。LREGでは、ディスパッチャ情報のリスナーへの登録にLOCAL_LISTENERパラメータとLISTENER属性が使用できるため、リスナーの値が同じ場合は、パラメータと属性の両方の指定は不要です。

LOCAL_LISTENERは、カンマ区切りのリスト・パラメータです。文字列にカンマがある場合、文字列全体を二重引用符で囲む必要があります。LOCAL_LISTENERパラメータを次のように設定します。

```
ALTER SYSTEM SET LOCAL_LISTENER=[""]listener_address[""],...];
```

たとえば、リスナー・アドレス“ab, cd”が入力された場合、1つのリスナー・アドレスに解決されます。アドレスがab, cdと入力された場合、abとcdの2つのリスナー・アドレスに解決されます。

共有サーバー接続の場合は、LISTENER属性を次のように設定します。

```
ALTER SYSTEM SET DISPATCHERS="(PROTOCOL=tcp) (LISTENER=listener_address)";
```

前述のコマンドで、listener_addressは、データベース・サーバー上にあるtnsnames.oraファイルなどのネーミング・メソッドによってリスナー・プロトコル・アドレスに解決されます。

ノート:

- LOCAL_LISTENER パラメータの動的な更新には、次のように SQL 文の ALTER SYSTEM を使用します。

```
ALTER SYSTEM SET LOCAL_LISTENER=[""]listener_address[""],...];
```

次のような文でパラメータを NULL に設定すると、TCP/IP、ポート 1521 のデフォルトのローカル・アドレスが使用されます。

```
ALTER SYSTEM SET LOCAL_LISTENER='';
```

- LISTENER 属性は LOCAL_LISTENER パラメータに優先されます。その結果、SQL 文の ALTER SYSTEM SET LOCAL_LISTENER はこの属性の設定値に影響を与えません。

[例9-1](#)では、データベースがホストsales1-serverにあります。このホスト上のリスナーはlistener_sales1という名前で、ポート1521ではなくポート1421でリスニングするように構成されています。

例9-1 専用サーバー環境でのローカル・リスナーの登録

1. ローカル・リスナーのあるホストで、Oracle Net Managerを使用してリスナーのプロトコル・アドレスを備えた listener.oraファイルを構成します。
2. データベースで、データベース初期化パラメータ・ファイルのLOCAL_LISTENERパラメータをローカル・リスナーの別名に設定します。たとえば:

```
ALTER SYSTEM SET LOCAL_LISTENER=listener_sales1;
```

データベースが共有サーバー接続用に構成されている場合、LISTENER属性を次のように設定できます。

```
ALTER SYSTEM SET DISPATCHERS="(PROTOCOL=tcp) (LISTENER=listener_sales1)";
```

3. テキスト・エディタを使用して、データベース・ホストのtnsnames.oraファイルでLOCAL_LISTENER設定のリスナー名の別名を次のように解決します。

```
listener_sales1=  
(DESCRIPTION =  
(ADDRESS = (PROTOCOL=tcp) (HOST=sales-server) (PORT=1421)))
```

ノート:

- ローカル・リスナーを登録する場合で、Oracle Connection Managerを使用する場合は、(DESCRIPTION =や閉じかっこを含めないでください。
- ネットワーク・サービス名のエントリは、接続記述子のCONNECT_DATA セクションなしでプロトコル・アドレス用に作成できます。

関連項目:

- ALTER SYSTEM文の詳細は、『[Oracle Database SQL リファレンス](#)』を参照してください。
- [リスニング用プロトコル・アドレスの構成](#)
- [ネーミング・メソッドの構成](#)

親トピック: [動的なサービス登録の構成](#)

9.2.3 リモート・リスナーへの情報の登録

リモート・リスナーとは、あるコンピュータ上にあるリスナーのことで、別のコンピュータ上にあるデータベース・インスタンスに接続をリダイレクトします。リモート・リスナーは通常、Oracle Real Application Clusters(Oracle RAC)環境で使用されます。Oracle RACなどでは、リモート・リスナーへの登録は、共有サーバーまたは専用サーバーの環境用に構成できます。

専用サーバー環境では、LREGバックグラウンド・プロセスを有効にしてリモート・リスナーに登録する必要があります。このためには、初期化パラメータ・ファイルで、カンマ区切りリスト・パラメータであるREMOTE_LISTENERパラメータを構成します。REMOTE_LISTENERの構文は次のとおりです。

```
ALTER SYSTEM SET REMOTE_LISTENER=["]listener_address["],[...];
```

前述のコマンドで、listener_addressは、データベース・ホスト上にあるtnsnames.oraファイルなどのネーミング・メソッドによってリスナー・プロトコル・アドレスに解決されます。リスナー・アドレスにカンマがある場合、文字列全体を引用符で囲む必要があります。

共有サーバー環境では、専用サーバー環境に対するものと同じ登録テクニックを使用できます。あるいは、任意のリスナーにディスパッチャを登録するように初期化パラメータ・ファイルのDISPATCHERSパラメータのLISTENER属性を設定することもできます。LISTENER属性の構文は次のとおりです。

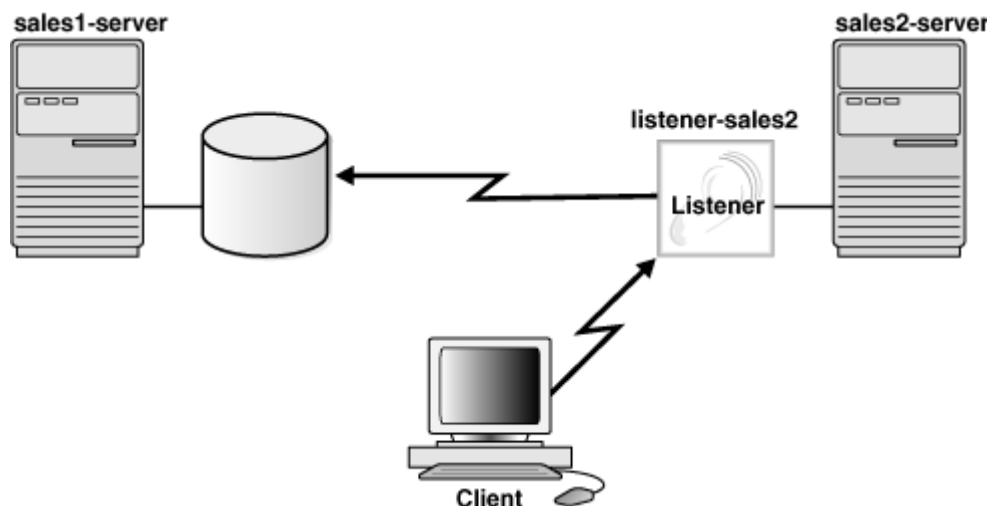
```
ALTER SYSTEM SET DISPATCHERS="(PROTOCOL=tcp) (LISTENER=listener_address)";
```

ノート:

LISTENER 属性は REMOTE_LISTENER 初期化パラメータに優先されます。LREG では、ディスパッチャ情報のリスナーへの登録に REMOTE_LISTENER 初期化パラメータと LISTENER 属性が使用できるため、リスナーの値が同じ場合は、パラメータと属性の両方を指定する必要がありません。

たとえば、listener-sales2 というリモート・リスナーがホスト sales2-server のポート 1521 でリスニングし、データベースがホスト sales1-server 上にあるとします。sales2-server 上のリスナーで、このデータベースに接続要求をリダイレクトするとします。[図 9-1](#) にこのシナリオを示します。

図9-1 リモート・リスナー



関連項目:

ALTER SYSTEM SET文の詳細は、[『Oracle Database SQLリファレンス』](#)を参照してください。

[例9-2](#)は、専用サーバー環境でリモート・リスナーを登録する方法を示しています。この例では、リモート・リスナーは sales2-server です。

例9-2 専用サーバー環境でのリモート・リスナーの登録

1. リモート・リスナーのあるホストで、Oracle Net Managerを使用してリモート・リスナーのプロトコル・アドレスを備えた listener.ora ファイルを構成します。
2. リクエストをリダイレクトさせたいデータベースで、データベース初期化パラメータ・ファイルの REMOTE_LISTENER パラメータを、たとえば次のようなリモート・リスナーの別名に設定します。

```
ALTER SYSTEM SET REMOTE_LISTENER=listener_sales2;
```

共有サーバー接続の場合は、ホスト sales1-server 上のデータベースの初期化ファイルの DISPATCHER パラメータを次のように設定します。

```
ALTER SYSTEM SET DISPATCHERS="(PROTOCOL=tcp)(LISTENER=listeners_sales2)";
```



ノート:

REMOTE_LISTENER 初期化パラメータを静的に更新するには、テキスト・エディタを使用して、以前

に情報を登録したリモート・リスナーから情報の登録を解除します。

3. データベース・ホストのtnsnames.oraファイルを介して、リモート・リスナーのリスナー名の別名を解決します。たとえば:

```
listener_sales2=  
(DESCRIPTION=  
(ADDRESS=(PROTOCOL=tcp)(HOST=sales2-server)(PORT=1521)))
```

関連項目:

- [「ネーミング・メソッドの構成」](#)
- [「リスニング用プロトコル・アドレスの構成」](#)
- REMOTE_LISTENER初期化パラメータについて学習するには、[『Oracle Databaseリファレンス』](#)を参照してください。
- Oracle RAC環境でリモート・リスナー(SCANリスナーとも呼ばれます)を構成する方法を学習するには、[『Oracle Real Application Clusters管理およびデプロイメント・ガイド』](#)を参照してください。

親トピック: [動的なサービス登録の構成](#)

9.2.4 ネットワーク内の全リスナーへの情報の登録

ネットワークには複数のローカル・リスナーとリモート・リスナーを含めることができます。デフォルトでは、すべてのリスナーが相互に登録されます。LISTENER_NETWORKS初期化パラメータに一連のリスナーを指定することで、リモート・リスナーのサブセットとともにローカル・リスナーのサブセットを指定できます。LISTENER_NETWORKSパラメータで指定されているリスナーを、LOCAL_LISTENERパラメータおよびREMOTE_LISTENERパラメータで指定しないでください。

LISTENER_NETWORKSパラメータの構文は次のとおりです。

```
LISTENER_NETWORKS = '( (NAME=network_name)  
                      (LOCAL_LISTENER=[""]listener_address[,...][])  
                      [(REMOTE_LISTENER=[""]listener_address[,...][])])'
```

前述の構文で、listener_addressは、LOCAL_LISTENERおよびREMOTE_LISTENERのルールに従って解決されます。

例9-3 サブネットでの2つのネットワークの使用

network1とnetwork2という2つの個別のネットワークがあるとします。network1には、local1というローカル・リスナーとremote1というリモート・リスナーが存在します。network2には、local2というローカル・リスナーとremote2というリモート・リスナーが存在します。次の構文により、リスナーが同じネットワーク上のリスナーにのみ接続をリダイレクトするように登録が設定されます。

```
LISTENER_NETWORKS =  
' ( (NAME=network1) (LOCAL_LISTENER=local1) (REMOTE_LISTENER=remote1))',  
' ( (NAME=network2) (LOCAL_LISTENER=local2) (REMOTE_LISTENER=remote2))'
```

前述の例で、local1はremote1にのみ登録され、remote1は接続をlocal1にのみリダイレクトします。local2はremote2にのみ登録され、remote2は接続をlocal2にのみリダイレクトします。

例9-4 複数のリスナーの構成

複数のリスナーがsales-networkというネットワークでリスニングしているとします。次の条件が真になります。

- 専用サーバーの接続用に構成されたデータベースがホストsales1-serverにあります。これはネットワーク上にある唯一のデータベースです。
- ローカル・リスナーがsales1-serverにあり、非デフォルトのポート1421でリスニングします。
- リモート・リスナーがホストsales2-serverにあり、ポート1521でリスニングします。
- 別のリモート・リスナーがホストsales3-serverにあり、ポート1521でリスニングします。

次の手順では、専用サーバー環境のすべてのリスナーに情報を登録する方法について説明します。

1. リモート・リスナーが存在するホスト(この例ではsales2-serverおよびsales3-server)で、リモート・リスナーのプロトコル・アドレスを使用してlistener.oraファイルを構成します。
2. リクエストをリダイレクトさせたいデータベースで、データベース初期化パラメータ・ファイルのREMOTE_LISTENERパラメータをリモート・リスナーの別名に設定し、LOCAL_LISTENERパラメータをローカル・リスナーの別名に設定します。

ホストsales1-server上のデータベースの初期化ファイルのパラメータを次のように設定します。

```
REMOTE_LISTENER="listener_sales2, listener_sales3"  
LOCAL_LISTENER=listener_sales1
```

3. データベース・ホストのtnsnames.oraファイルを介して、LOCAL_LISTENERおよびREMOTE_LISTENER設定のリスナー名の別名を解決します。

sales1-server上のtnsnames.oraで、ローカル・リスナー・エイリアスおよびリモート・リスナー・エイリアスlistener_sales1、listener_sales2およびlistener_sales3を次のように解決します。

```
listener_sales1=  
(DESCRIPTION=  
  (ADDRESS=(PROTOCOL=TCP) (HOST=sales1-server) (PORT=1421)))  
listener_sales2=  
(DESCRIPTION=  
  (ADDRESS=(PROTOCOL=TCP) (HOST=sales2-server) (PORT=1521)))  
listener_sales3=  
(DESCRIPTION=  
  (ADDRESS=(PROTOCOL=TCP) (HOST=sales3-server) (PORT=1521)))  
listener_sales_local=  
(DESCRIPTION=  
  (ADDRESS=(PROTOCOL=tcp) (HOST=sales1-server) (PORT=1421)))  
listener_sales_remote=  
(DESCRIPTION_LIST=  
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=sales2-server) (PORT=1521)))  
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=sales3-server) (PORT=1521))))
```

関連項目:

- [「ネーミング・メソッドの構成」](#)
- [「リスニング用プロトコル・アドレスの構成」](#)
- REMOTE_LISTENER初期化パラメータの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

親トピック: [動的なサービス登録の構成](#)

9.2.5 ネーミング・メソッドの構成

LOCAL_LISTENERまたはREMOTE_LISTENER初期化パラメータ、またはLISTENER属性に指定したリスナー名の別名は、tnsnames.oraファイルにより解決できます。たとえば、リスナーがinit.oraファイルで次のように定義されるとします。

```
LOCAL_LISTENER = (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1421))
```

リスナーの別名を使用するには、init.oraおよびtnsnames.oraファイルに次のように定義します。

- init.oraファイル:

```
LOCAL_LISTENER = listener_sales1
```

- tnsnames.oraファイル:

```
listener_sales1 = (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1421))
```

CONNECT_DATA情報は、tnsnames.oraファイルのネットワーク・サービス・エントリに含めません。ただし、CONNECT_DATA情報がないと、Oracle Enterprise Manager Cloud ControlおよびOracle Net Managerはtnsnames.oraファイルを構成できません。リスナー名の別名を使用するには、テキスト・エディタを使用してtnsnames.oraファイルを変更することをお勧めします。

ノート:

- 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・バランシング機能はサポートされていません。
- LOCAL_LISTENER パラメータで指定したリスナーの別名が無効であるか解決されない場合、LREG プロセスはデータベースの起動を許可しません。次のエラーが発生します。

```
ORA-00119: invalid specification for system parameter  
LOCAL_LISTENER  
ORA-00132: syntax error or unresolved network name '%s'
```

関連項目:

複数のアドレス構成の詳細は、[「Oracle Net Servicesの拡張機能の使用」](#)を参照してください

親トピック: [動的なサービス登録の構成](#)

9.3 インストール中のOracle Net Listenerの構成

Oracle Universal Installerは、インストール時に[Oracle Net Configuration Assistant](#)を起動します。Oracle Net Configuration Assistantを使用して、Oracle Databaseのリスニング用プロトコル・アドレスおよびサービス情報を構成します。

Enterprise EditionまたはStandard Editionをデータベース・サーバーへインストールする際、Oracle Net

Configuration Assistantは、自動的にリスナーを構成して、Oracle DatabaseのTCP/IPリスニング用プロトコル・アドレスを所有するLISTENERという名前を設定します。Customインストールでは、Oracle Net Configuration Assistantにより、リスナー名およびプロトコル・アドレスを入力するように求められます。

[外部プロシージャ](#)・コールのリスニングIPCプロトコル・アドレスは、インストール・タイプにかかわらず自動的に構成されます。Oracle Net Configuration Assistantでは、外部プロシージャのサービス情報もlistener.oraファイルに自動的に構成されます。

IPCプロトコルを使用している場合は、同時IPC接続要求の最大数を、必要な接続要求数と一致するように指定することで、パフォーマンスを向上できます。

親トピック: [Oracle Net Listenerの構成と管理](#)

9.4 Oracle Net Listener構成のカスタマイズ

デフォルトの構成またはインストールした構成が特定の環境に対して適切でない場合は、[Oracle Net Manager](#)を使用してlistener.oraの構成をカスタマイズできます。

- [リスニング用プロトコル・アドレスの構成](#)
- [多数の同時接続要求の処理](#)
- [Oracle Net Listenerのセキュリティの管理](#)

デフォルトでは、Oracle Net Listenerはセキュリティ上の理由からローカル管理のみを許可しています。ポリシーとして、リスナーは起動したユーザーのみが管理できます。これはローカル・オペレーティング・システム認証によって適用されます。

親トピック: [Oracle Net Listenerの構成と管理](#)

9.4.1 リスニング用プロトコル・アドレスの構成

Oracle Enterprise Manager Cloud ControlおよびOracle Net Managerは、リスナーのプロトコル・サポートの構成に使用できます。

[\[IPv6インタフェースとアドレス構成\]](#)で説明したように、Oracle Net Listenerのエンドポイント・アドレス構成は、IPv6アドレスとIPv6アドレスに解決されるホスト名の両方を受け入れます。これにより、IPv6クライアントを処理するリスニング・エンドポイントが作成されます。

- [Oracle Enterprise Manager Cloud Controlを使用したリスニング用プロトコル・アドレスの構成](#)
Oracle Enterprise Manager Cloud ControlのNet Services管理ページからリスナーのプロトコル・アドレスを構成します。
- [Oracle Net Managerを使用したリスニング用プロトコル・アドレスの構成](#)

親トピック: [Oracle Net Listener構成のカスタマイズ](#)

9.4.1.1 Oracle Enterprise Manager Cloud Controlを使用したリスニング用プロトコル・アドレスの構成

Oracle Enterprise Manager Cloud ControlのNet Services管理ページからリスナーのプロトコル・アドレスを構成します。

1. Oracle Enterprise Manager Cloud Controlの「Net Services管理」ページにアクセスします。
2. 「編集」をクリックします。データベース・サーバーへのログインを求められる場合があります。
リスナーの編集ページが表示されます。
3. アドレス・セクションで次のようにプロトコル・サポートを設定します。
 - a. 「追加」をクリックします。
「アドレスの追加」ページが表示されます。
 - b. 「プロトコル」リストから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。
TCP/IPの場合、使用しているコンピュータに複数のIPアドレスがあり、使用可能なすべてのIPアドレスでリスナーがリスニングするためには、「TCP/IP」またはTLS付きTCP/IPを選択し、「ホスト」フィールドにコンピュータのホスト名を入力します。
 - c. 「ポート」にポート番号を入力します。
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として1521を入力します。それ以外の場合は、初期化パラメータ・ファイルにLOCAL_LISTENERパラメータを構成し、任意のネーミング・メソッドで使用するためのデフォルト以外のポート番号を指定する必要があります。
 - d. 「ホスト」にホスト・アドレスを入力します。
 - e. (オプション)「拡張パラメータ」セクションを使用して、このセッションにおける送受信操作に必要なI/Oバッファ・スペース制限を「合計送信バッファ・サイズ」および「合計受信バッファ・サイズ」フィールドに指定します。
 - f. 「OK」をクリックします。
プロトコル・アドレスは、アドレス・セクションに追加されます。
4. その他のプロトコルについてもステップ3を繰り返します。

関連項目

- [「Net Services管理」ページへのアクセス](#)
- [I/Oバッファ・スペースの構成](#)
- [『Oracle Database Net Servicesリファレンス・ガイド』](#)

親トピック: [リスニング用プロトコル・アドレスの構成](#)

9.4.1.2 Oracle Net Managerを使用したリスニング用プロトコル・アドレスの構成

次の手順では、Oracle Net Managerを使用してリスナーのプロトコル・アドレスを構成する方法について説明します。

1. Oracle Net Managerを起動します。

関連項目:

[「Oracle Net Managerを使用したOracle Net Servicesの構成」](#)

2. ナビゲータ・ペインで、「ローカル」を展開し、次に「リスナー」を選択します。
3. リスナーを選択します。
4. 右ペインのリストから、リスナーの位置を選択します。

5. 「プロトコル」リストからプロトコルを選択します。
6. 「ホスト」フィールドにリスナーのホスト名を入力します。
7. 「ポート」フィールドにポート番号を入力します。
8. 送信および受信バッファのサイズを設定する場合は、「詳細の表示」をクリックし、適切なフィールドにサイズを入力します。
9. 「ファイル」メニューから「ネットワーク構成の保存」を選択して変更を保存します。

親トピック: [リスニング用プロトコル・アドレスの構成](#)

9.4.2 多数の同時接続要求の処理

リスナーによる同時接続要求の大量処理が予想される場合は、TCP/IPまたはIPCリスニング・エンドポイントに対してリスナー・キュー・サイズを指定できます。

リスナー・キュー・サイズを指定するには、次の手順に従います。

- プロトコル・アドレスの終端にあるQUEUESIZEパラメータに、予想される同時要求数を指定します。

次の例は、キュー・サイズを20に設定します。

```
LISTENER=
  (DESCRIPTION=
    (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521) (QUEUESIZE=20)))
```

ノート:

デフォルトの同時接続要求数は、オペレーティング・システムによって異なります。Linux オペレーティング・システムと Microsoft Windows の TCP/IP のデフォルト値は、次のとおりです。

- Linux オペレーティング・システム: 128
- Microsoft Windows XP Professional SP2: 10
- Microsoft Windows Server Enterprise Edition: 200

親トピック: [Oracle Net Listener構成のカスタマイズ](#)

9.4.3 Oracle Net Listenerのセキュリティの管理

デフォルトでは、Oracle Net Listenerではセキュリティ上の理由からローカル管理のみが可能です。ポリシーとして、リスナーは起動したユーザーのみが管理できます。これはローカル・オペレーティング・システム認証によって適用されます。

たとえば、user1がリスナーを起動した場合、user1のみがリスナーを管理できます。リスナーの管理を試みるその他のユーザーはエラーを受け取ります。例外となるのはスーパーユーザーのみです。

リスナーの管理はデフォルト・モード(ローカル・オペレーティング・システムの認証機能による保護)で実行し、リモート・ログインを

使用してシステムにリモート・アクセスすることをお勧めします。Oracle Enterprise Manager Cloud Controlは、リモート管理にも使用できます。

ファイアウォール機能が有効化されているIP (TCP、TCPS、およびSDP)ベースのエンドポイント上のリスナーに向かう接続は、サービスACLの検証を受けます。リスナーは、サービス名の受信後にACLリストで接続IPを検証します。

エンドポイントには、ファイアウォール機能を有効化するための新しい属性FIREWALLが追加されています。

```
(ADDRESS=(PROTOCOL=TCP) (HOST=) (PORT=1521) (FIREWALL=ON))
```

FIREWALLパラメータを次のように構成できます。

- エンドポイントで (FIREWALL=ON) が明示的に設定されている場合：このエンドポイントに到着したすべての接続に対して、厳密なACL検証が有効になります。あるサービスに対応するACLが構成されていない場合、そのサービスの接続はすべて拒否されます。
- エンドポイントでFIREWALLが設定されていない場合：緩い検証を意味します。あるサービスに対応するACLが構成されている場合、そのサービスに対応する検証が実行されます。ACLが存在しない場合、検証は実行されず、そのサービスの接続はすべて受け入れられます。
- エンドポイントで (FIREWALL=OFF) が設定されている場合：検証は実行されず、このエンドポイントからの接続はすべて受け入れられます。

サーバーもACLを強制適用します。

DBMS_SFW_ACL_ADMINパッケージは、アクセス制御ポリシーを管理するためのインタフェースを提供します。

- [有効なノードおよびサブネットの指定](#)

関連項目

- [アクセス・コントロール・リスト](#)
- [ファイアウォール](#)
- [LOCAL_REGISTRATION_ADDRESS_listener_name](#)
- [DBMS_SFW_ACL_ADMIN](#)

親トピック: [Oracle Net Listener構成のカスタマイズ](#)

9.4.3.1 有効なノードおよびサブネットの指定

リスナーの登録は、有効なノードおよびサブネットに制限するようにしてください。登録に指定できるのは有効なノードおよびサブネットですが、除外されたノードも登録に指定できます。デフォルトでは、リスナーでの登録のすべての着信接続でIPベースのフィルタリングが実行されます。接続が許可されるのは、発信元がローカル・マシンの場合のみです。もう一方のノードおよびサブネットが登録に指定されると、ローカル・マシンおよび指定されたマシンが許可されます。次のパラメータをlistener.oraファイルに指定して、有効なノードおよびサブネットと、制限されたノードおよびサブネットを指定できます。

- REGISTRATION_INVITED_NODES_listener_name: リスナーに登録可能なノードを指定します。リストは、ホスト名、またはIPv4およびIPv6アドレスのCIDR表記が可能です。リストにホスト名が存在する場合は、すべてのIPアドレスが招待の状態にマップされます。
- REGISTRATION_EXCLUDED_NODES_listener_name: リスナーに登録できないノードを指定します。リストに指定されていないノードでもリスナーに登録できます。

両方のパラメータが設定されている場合、REGISTRATION_EXCLUDED_NODES_listener_nameは無視されます。

デフォルトでは、SCANリスナー・エージェントはREMOTE_ADDRESS_REGISTRATION_listener_nameをプライベートIPエンドポイントに設定します。SCANリスナーは、プライベート・ネットワークからの登録要求のみを受け入れます。SCANリスナーのプライベート・ネットワークにアクセスできないリモート・ノードは、listener.oraファイルのregistration_invited_nodes_aliasパラメータを使用して、またはコマンドライン・インタフェースのSRVCTLを使用してSCANリスナーを変更して、有効なノードのリストに含める必要があります。

ノート:



Oracle Grid Infrastructure 12c以降、SCANリスナーについて、VALID_NODE_CHECKING_REGISTRATION_listener_nameおよびREGISTRATION_INVITED_NODES_listener_nameパラメータがlistener.oraファイルに設定されている場合、リスナー・エージェントはこれらのパラメータを上書きします。

SRVCTLユーティリティを使用してinvitednodes値とinvitedsubnets値を設定すると、リスナー・エージェントは自動的にVALID_NODE_CHECKING_REGISTRATION_listener_nameをSUBNETに設定し、REGISTRATION_INVITED_NODES_listener_nameをlistener.oraファイルで指定されたリストに設定します。

CRSによって管理されるその他のリスナーの場合、リスナー・エージェントは、listener.oraファイルでまだ設定されていない場合にのみ、listener.oraファイルでVALID_NODE_CHECKING_REGISTRATION_listener_nameをSUBNETに設定します。

SRVCTLユーティリティでは、SCAN以外のリスナーについてinvitednodes値とinvitedsubnets値の設定はサポートされていません。リスナー・エージェントは、SCAN以外のリスナーについてlistener.oraファイルのREGISTRATION_INVITED_NODES_listener_nameを更新しません。

関連項目:

VALID_NODE_CHECKING_REGISTRATION_listener_name、REGISTRATION_INVITED_NODES_listener_nameおよびREGISTRATION_EXCLUDED_NODES_listener_nameパラメータの詳細は、[Oracle Database Net Servicesリファレンス](#)を参照してください

親トピック: [Oracle Net Listenerのセキュリティの管理](#)

9.5 リスナーの管理

リスナーを構成後、リスナー制御ユーティリティ、Oracle Enterprise Manager Cloud Controlおよびサーバー制御ユーティリティ(SRVCTL)を使用して、リスナーを管理できます。この項では、リスナーの管理タスクの一部について説明します。次の項目が含まれます。

- [リスナーの起動と停止](#)
- [Oracle Restart構成のリスナーの管理](#)
- [リスナーの現在のステータスの確定](#)
- [リスナーのサービスの監視](#)

- [リスナー・ログ・ファイルの監視](#)

関連項目:

- リスナー制御ユーティリティのコマンドの完全なリストは、[『Oracle Database Net Servicesリファレンス』](#)を参照してください。
- Oracle Enterprise Manager Cloud Controlのオンライン・ヘルプ

親トピック: [Oracle Net Listenerの構成と管理](#)

9.5.1 リスナーの起動と停止

リスナーを停止または起動するには、次のいずれかの方法を使用します。

ノート:



リスナーは、コンピュータの実行や再起動に応じて自動的に起動するよう構成できます。詳細は、[「Oracle Restart 構成のリスナーの管理」](#)を参照してください。

- [リスナー制御ユーティリティを使用したリスナーの起動または停止](#)
- [Oracle Enterprise Manager Cloud Controlを使用したリスナーの起動または停止](#)

親トピック: [リスナーの管理](#)

9.5.1.1 リスナー制御ユーティリティを使用したリスナーの起動または停止

コマンドラインからリスナーを起動するには、次のように入力します。

```
lsnrctl START [listener_name]
```

前述のコマンドで、listener_nameは、listener.oraファイルに定義されているリスナーの名前です。デフォルトのLISTENERという名前のリスナーを使用している場合、リスナーを指定する必要はありません。

リスナー制御ユーティリティは、リスナーの起動に加えリスナーへの接続性を検証します。

コマンドラインからリスナーを停止するには、次のように入力します。

```
lsnrctl STOP [listener_name]
```

前述のコマンドで、listener_nameは、listener.oraファイルに定義されているリスナーの名前です。デフォルトのLISTENERという名前のリスナーを使用している場合、リスナーを指定する必要はありません。

ノート:



Oracle ホーム・ユーザーを使用する場合、Microsoft Windows システムでリスナー制御ユーティリティからパスワードを要求されます。このパスワードは、Oracle ホーム・ユーザーのオペレーティング・システム・パスワードです。パスワード・プロンプトが表示されるのは、リスナー・サービスが存在せず、リスナーの起動の一部として作成する必要が

ある場合のみです。

関連項目:

Oracleホーム・ユーザーの詳細は、[『Oracle Databaseプラットフォーム・ガイド for Microsoft Windows』](#)を参照してください。

親トピック: [リスナーの起動と停止](#)

9.5.1.2 Oracle Enterprise Manager Cloud Controlを使用したリスナーの起動または停止

次の手順では、Oracle Enterprise Manager Cloud Controlからリスナーを起動または停止する方法について説明します。

1. Oracle Enterprise Manager Cloud Controlの「Net Services管理」ページにアクセスします。

関連項目:

[『Oracle Enterprise Manager Cloud Controlを使用したOracle Net Servicesの構成』](#)

2. 「管理」リストから「リスナー」を選択し、構成ファイルの場所を含むOracleホームを選択します。
3. 「実行」をクリックします。
「リスナー」ページが表示されます。
4. リスナーを選択します。
5. 「アクション」リストから「開始/停止」を選択します。
6. 「実行」をクリックします。
「開始/停止」ページが表示されます。
7. 選択したリスナーの現行ステータスに応じて、「停止」または「起動」を選択し、「OK」をクリックします。

親トピック: [リスナーの起動と停止](#)

9.5.2 Oracle Restart構成のリスナーの管理

Oracle Restart機能は、シングル・インスタンス・データベース環境でプロセスおよびアプリケーションの可用性を強化します。Oracle Restartエージェントは、チェック操作を定期的に行い、必要に応じてコンポーネントを再起動することで、追加されたコンポーネントの状態を監視します。

リスナーをコンポーネントとしてOracle Restart構成に追加できます。リスナーは、障害が発生した場合、または動作していない場合にOracle Restartによって自動的に起動されます。たとえば、コンピュータを予定どおりに再起動した後にデータベース・インスタンスを再起動すると、Oracle Restartによりリスナーが再起動されます。サーバー制御(SRVCTL)は、Oracle Restart構成でリスナーを管理するために使用できるコマンドライン・インタフェースです。

- [SRVCTLユーティリティを使用した構成済のリスナーの表示](#)

- [SRVCTLユーティリティを使用したリスナーの追加または削除](#)
- [SRVCTLユーティリティを使用したリスナーの起動または停止](#)

親トピック: [リスナーの管理](#)

9.5.2.1 SRVCTLユーティリティを使用した構成済のリスナーの表示

構成されたすべてのリスナーを表示するには、次のコマンドを使用します。

```
% srvctl config listener
```

関連項目:

Oracle Restartの構成方法、およびSRVCTLの構文とセマンティクス方法を学習するには、『[Oracle Database管理者ガイド](#)』を参照してください。

親トピック: [Oracle Restart構成のリスナーの管理](#)

9.5.2.2 SRVCTLユーティリティを使用したリスナーの追加または削除

リスナーをエントリとしてグリッド・インフラストラクチャに追加すると、エージェントでリスナーを監視できます。同様に、リスナーをエントリとして削除できます。次のように、オペレーティング・システムのコマンドラインでsrvctlコマンドを使用します。

- リスナーを追加するには、srvctl add listenerと入力します。

次のコマンドは、listener_sales1のエントリをグリッド・インフラストラクチャに追加します。

```
% srvctl add listener -listener listener_sales1
```

- リスナーを削除するには、srvctl remove listenerと入力します。

次のコマンドは、listener_sales1のエントリをグリッド・インフラストラクチャから削除します。

```
% srvctl remove listener -listener listener_sales1
```

親トピック: [Oracle Restart構成のリスナーの管理](#)

9.5.2.3 SRVCTLユーティリティを使用したリスナーの起動または停止

SRVCTLユーティリティでは、リスナーを停止および起動できます。-listenerパラメータを指定しない場合は、SRVCTLユーティリティによってデフォルトのリスナーが起動および停止されます。

- リスナーを起動するには、srvctl start listenerと入力します。

次の例では、最初のコマンドではデフォルトのリスナーが起動され、2番目のコマンドではlistener1およびlistener2が起動されます。

```
% srvctl start listener  
% srvctl start listener -listener listener1,listener2
```

- リスナーを停止するには、srvctl stop listenerと入力します。

次の例では、最初のコマンドではデフォルトのリスナーが停止され、2番目のコマンドではlistener1およびlistener2

が停止されます。

```
% srvctl stop listener
% srvctl stop listener -listener listener1,listener2
```

親トピック: [Oracle Restart構成のリスナーの管理](#)

9.5.3 リスナーの現在のステータスの確定

リスナーの現在のステータスを表示するには、リスナー制御ユーティリティのSTATUSコマンドかOracle Enterprise Manager Cloud Controlを使用します。ステータス出力により、リスナーの基本的なステータス情報、リスナー構成の設定の概要、リスニング用プロトコル・アドレスおよびリスナーに登録されたサービスの概要が提供されます。

- [リスナー制御を使用したステータスの表示](#)
- [Oracle Enterprise Manager Cloud Controlを使用したステータスの表示](#)

親トピック: [リスナーの管理](#)

9.5.3.1 リスナー制御を使用したステータスの表示

コマンドラインからリスナーのステータスを表示するには、次のコマンドを入力します。

```
lsnrctl STATUS [listener_name]
```

前述のコマンドで、listener_nameは、listener.oraファイルに定義されているリスナーの名前です。デフォルトのLISTENERという名前のリスナーを使用している場合、リスナーを指定する必要はありません。

[例9-5](#)では、STATUSコマンドの出力例を示しています。

例9-5 リスナー制御ユーティリティのSTATUSコマンドの出力

```
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=net)))
STATUS of the LISTENER
-----
Alias                LISTENER
Version              TNSLSNR for Linux: Version 18.0.0.0.0
Start Date           15-MAY-2017 20:22:00
Uptime               65 days 10 hr. 5 min. 22 sec
Trace Level          support
Security             OFF
Listener Parameter File /oracle/admin/listener.ora
Listener Log File    /oracle/network/log/listener.log
Listener Trace File  /oracle/network/trace/listener.trc
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=net)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcps) (HOST=sales-server) (PORT=2484)))

Services Summary...
Service "sales.us.example.com" has 1 instance(s).
  Instance "sales", status READY, has 3 handler(s) for this service...
Service "hr.us.example.com" has 1 instance(s).
  Instance "hr", status READY, has 2 handler(s) for this service...
The command completed successfully
```

STATUSコマンドの出力には、[表9-1](#)で説明しているセクションが含まれます。

表9-1 リスナー制御ユーティリティのSTATUSコマンド

出力セクション	説明
リスナーのステータス	リスナーのステータスは次のとおりです。 <ul style="list-style-type: none">● リスナーの別名● リスナーのバージョン● 開始時刻と終了時刻● トレース・レベル● 使用している listener.ora ファイル● 構成の設定のロギングとトレース
Listening Endpoints Summary (リスニング・エンドポイントのサマリー)	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・アドレス。
Services Summary (サービスのサマリー)	リスナーに登録されたサービスおよび各サービスに割り当てられたサービス・ハンドラのサマリー。
Service (サービス)	登録されたサービス。
Instance (インスタンス)	サービスに関連付けられたインスタンスの名前。 ステータス・フィールドは、インスタンスが接続を受け入れられるかどうかを示します。 <ul style="list-style-type: none">● READY は、インスタンスが接続を受け入れられることを意味します。● BLOCKED は、インスタンスが接続を受け入れられないことを意味します。● READY/SECONDARY は、これが Oracle Real Application Clusters のプライマリ/セカンダリ構成のセカンダリ・インスタンスであり、接続を受け入れられることを意味しています。● RESTRICTED は、そのインスタンスが制約のあるモードであることを示しています。リスナーは、このインスタンスへの接続をすべて阻止します。● UNKNOWN は、インスタンスが、サービス登録を使用して動的に登録されているのではなく、listener.ora ファイルに静的に登録されていることを意味します。このた

め、ステータスは不明です。

親トピック: [リスナーの現在のステータスの確定](#)

9.5.3.2 Oracle Enterprise Manager Cloud Controlを使用したステータスの表示

次の手順では、Oracle Enterprise Manager Cloud Controlを使用してリスナーのステータスを表示する方法について説明します。

1. Oracle Enterprise Manager Cloud Controlの「Net Services管理」ページにアクセスします。

関連項目:

[「Oracle Enterprise Manager Cloud Controlを使用したOracle Net Servicesの構成」](#)

2. 「管理」リストから「リスナー」を選択し、構成ファイルの場所を含むOracleホームを選択します。
3. 「実行」をクリックします。データベース・サーバーへのログインを求められる場合があります。
「リスナー」ページが表示されます。
4. リスナーを選択します。
5. 「アクション」リストから「リスナー制御ステータスの表示」を選択します。
6. 「実行」をクリックします。
「リスナー制御ステータス」ページが表示されます。
7. 内容を確認した後、ページの一番上にあるリスナーのリンクをクリックします。

親トピック: [リスナーの現在のステータスの確定](#)

9.5.4 リスナーのサービスの監視

リスナー制御ユーティリティのSERVICESコマンドにより、リスナーに登録されたサービスやインスタンス、および各インスタンスに割り当てられたサービス・ハンドラの詳細情報が提供されます。コマンドラインからサービスやインスタンスについての情報を表示するには、次のように入力します。

```
lsnrctl SERVICES [listener_name]
```

[例9-6](#)では、SERVICESコマンドの出力例を示しています。

例9-6 リスナー制御ユーティリティのSERVICESコマンドの出力

```
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=net)))
Services Summary...
Service "sales.us.example.com" has 1 instance(s).
  Instance "sales", status READY, has 3 handler(s) for this service...
  Handler(s):
    "DEDICATED" established:0 refused:0 state:ready
    LOCAL SERVER
```

```

"D000" established:0 refused:0 current:0 max:10000 state:ready
DISPATCHER <machine: sales-server, pid: 1689>
  (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=52414))
"D001" established:0 refused:0 current:0 max:10000 state:ready
DISPATCHER <machine: sales-server, pid: 1691>
  (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=52415))
Service "hr.us.example.com" has 1 instance(s).
Instance "hr", status READY, has 2 handler(s) for this service...
Handler(s):
  "DEDICATED" established:0 refused:0 state:ready
  LOCAL SERVER
  "D000" established:0 refused:0 current:0 max:10000 state:ready
  DISPATCHER <machine: sales-server, pid: 11326>
  (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=58361))
The command completed successfully

```

この出力では、2つのデータベース・サービス、sales.us.example.comおよびhr.us.example.comがリスナーに登録されていることが示されています。

sales.us.example.comへのクライアントの接続要求は、D000およびD001とネーミングされた2つのディスパッチャと1つの専用サーバーにより処理されます。すべてのハンドラは、readyのステータスを所有していて、接続を受け取れることを示しています。

hr.us.example.comへのクライアントの接続要求は、D001とネーミングされた1つのディスパッチャと1つの専用サーバーにより処理されます。

SERVICESコマンドにより[表9-2](#)で説明している次の情報が出力されます。

表9-2 リスナー制御ユーティリティのSERVICESコマンド

出力セクション	説明
Services (サービス)	登録されたサービス。
Instance (インスタンス)	サービスに関連付けられたインスタンスの名前。 ステータス・フィールドは、インスタンスが接続を受け入れられるかどうかを示します。 <ul style="list-style-type: none"> ● READY は、インスタンスが接続を受け入れられることを意味します。 ● BLOCKED は、インスタンスが接続を受け入れられないことを意味します。 ● READY/SECONDARY は、これが Oracle Real Application Clusters のプライマリ/セカンダリ構成のセカンダリ・インスタンスであり、接続を受け入れられることを意味しています。 ● RESTRICTED は、そのインスタンスが制約のあるモードであることを示しています。リスナーは、このインスタンスへの接続をすべて阻止します。 ● UNKNOWN は、インスタンスが、サービス登録を使用して動的に登録されているのではなく、listener.ora ファイルに静的に登録されていることを意味します。このため、ステータスは

出力セクション	説明
	不明です。
Handlers (ハンドラ)	サービス・ハンドラの名前。ディスパッチャは、D000 から D999 までネーミングされます。専用サーバーの名前は、DEDICATED です。
	このセクションでは、サービス・ハンドラの次の項目も識別されます。
	<ul style="list-style-type: none"> ● established: このサービス・ハンドラが確立したクライアントの接続数。 ● refused: このサービス・ハンドラが拒否したクライアントの接続数。 ● current: 処理中のクライアントの接続数(すなわち、現在のロード)。 ● max: サービス・ハンドラが扱う最大接続数(すなわち、最大ロード)。 ● state: 次に示すハンドラの状態
	- READY は、サービス・ハンドラが新しい接続を受け入れられることを意味します。
	- BLOCKED は、サービス・ハンドラが新しい接続を受け入れられないことを意味します。

親トピック: [リスナーの管理](#)

9.5.5 リスナー・ログ・ファイルの監視

次の状況が認められた場合は、次のリスナー・ログ・ファイルでエラー情報を確認します。

- 長期接続の回数
- 接続障害と接続拒否
- サービス拒否攻撃を示唆するようなリスナーの予測外のシャットダウン

関連項目:

[「リスナー・ログ・ファイルの分析」](#)

親トピック: [リスナーの管理](#)

9.6 リスナーのリダイレクトの理解

Oracle Database 12cリリース2 (12.2)から、リスナー・リダイレクト機能により、プラグブル・データベース(PDB)がOracle Public Cloud内の新しい場所に移行された後でも、クライアントは未変更の接続記述子を使用してPDBに接続できるようになっています。

リスナー登録(LREG)プロセスが、新しいハンドラをPDBのローカル・リスナーまたは移行のプロセス内にあるサービスに登録します。このハンドラには、PDBまたはサービスが移行されたデータベースの新しいリスナー・アドレスが入っています。新しいリスナー・アドレスは、Oracle RACデータベースの場合は[単一クライアント・アクセス名\(SCAN\)](#)リスナーのアドレスに、また単一インスタンス・データベースの場合はローカル・リスナーのアドレスにできます。これで、リスナーがクライアントを新しいアドレスにリダイレクトできます。

ローカル・リスナーがOracle RAC構成内のSCANリスナーにリダイレクトした場合、このリスナーは、クライアントの接続要求をさらに別のクラスタ・ノードにリダイレクトする必要がある場合があります。このような複数リダイレクトは、Oracle Netリスナーではデフォルトでサポートされません。SCANリスナーがすでにリダイレクトされたクライアント接続要求を転送できるようにするには、ALLOW_MULTIPLE_REDIRECTS_listener_nameパラメータをそのlistener.oraファイルに追加します。パラメータをTRUEに設定します。このパラメータをノード・リスナー用に設定することはしないでください。特定のネットワーク構成内で無限リダイレクト・ループが生じる可能性があります。

親トピック: [Oracle Net Listenerの構成と管理](#)

10 Oracle Connection Managerの構成

Oracle Connection Managerは、データベースまたはプロキシ・サーバーに接続要求を転送する[プロキシ・サーバー](#)です。これはセッション・レベルで動作します。通常、データベース・サーバーおよびクライアント・コンピュータとは別のコンピュータに常駐しています。Oracle Connection Managerは、Oracle Database 19c Enterprise Editionをインストールすると使用できるようになります。これはクライアント・ディスク上のカスタム・インストール・オプションです。

Oracle Connection Managerの主要機能は次のとおりです。

- アクセス制御: ルールベースの構成を使用してユーザーが指定したクライアント要求をフィルタにかけ、フィルタを通過したものを受け入れます。
- セッションの多重化: [共有サーバー](#)の接続先へのネットワーク接続により、複数のクライアント・セッションを集中化します。

Oracle Connection Manager機能の構成方法を学習します。

- [cman.oraファイルについて](#)
- [Oracle Connection Managerの構成方法](#)
- [IPv4およびIPv6のブリッジとしてのOracle Connection Managerの使用](#)
- [Oracle Connection Managerの起動および停止](#)
Oracle Connection Managerを構成したら、Oracle Connection Manager制御(CMCTL)ユーティリティを使用してこれを起動および管理できます。
- [Oracle Connection Managerの機能拡張](#)
Oracle Connection Managerにより、Oracle Database Serverに対するリクエストをプロキシ化し選別します。

関連項目:

- Oracle Connection Managerの概要は、[「Oracle Net Servicesの概要」](#)を参照してください
- Oracle Connection Managerのアーキテクチャの概要は、[「通信レイヤーの理解」](#)を参照してください

親トピック: [Oracle Net Servicesの構成と管理](#)

10.1 cman.oraファイルについて

cman.oraファイルのパラメータを設定して、Oracle Connection Managerをホストするコンピュータを構成します。cman.oraファイルは、Oracle Connection Managerをホストするコンピュータ上のORACLE_HOME/network/adminディレクトリにあります。cman.oraファイルが存在しないと、Oracle Connection Managerは起動しません。このファイルには次のコンポーネントが含まれています。

- リスニング・エンドポイント
- アクセス制御ルール・リスト
- パラメータ・リスト

各Oracle Connection Manager構成は、1つの名前-値(NV)文字列内にカプセル化されており、その文字列は、前述の

コンポーネントで構成されています。

1台のコンピュータで任意の数のOracle Connection Managerをホスティングでき、そのそれぞれのエントリがcman.oraファイルに保持されます。このファイルに複数のOracle Connection Managerを定義する場合は、完全修飾ホスト名を1つのみ設定することによって、デフォルトとして割り当てることができます。

クライアントとOracle Connection Manager制御ユーティリティ(CMCTL)の2つの接続に対して複数のルールを指定できます。変更を行う場合、次のガイドラインが適用されます。

- クライアント接続とCMCTL接続に対して、1つ以上のルールを入力する必要があります。ルールを省略すると、省略されたルール・タイプの全接続が拒否されます。
- Oracle Connection Managerでは、ワイルドカードをIPアドレスの一部として使用できません。ワイルドカードを使用する場合は、完全なIPアドレスのかわりとして使用してください。たとえば、クライアントのIPアドレスの場合は、(SRV=*)と指定します。
- Oracle Connection Managerでは、サブネット・アドレスの表記として/nnのみを使用できます。[例10-1](#)では、最初のルールの/24が左端の24ビットで構成されるサブネット・マスクを表しています。クライアントIPアドレスの先頭の24ビットのみが、ルール内のIPアドレスと比較されます。

ノート:



Oracle Connection ManagerはIPv6 アドレッシングをサポートしています。[「IPv4 および IPv6 のブリッジとしての Oracle Connection Manager の使用」](#)を参照してください。

[例10-1](#)に、CMAN1というOracle Connection Managerの構成エントリが含まれるcman.oraファイルを示します。

例10-1 cman.oraファイルの例

```
CMAN1=
(CONFIGURATION=
  (ADDRESS=(PROTOCOL=tcp) (HOST=proxysvr) (PORT=1521))
  (RULE_LIST=
    (RULE=(SRC=192. 0. 2. 32/24) (DST=sales-server) (SRV=*) (ACT=accept)
      (ACTION_LIST=(AUT=on) (MCT=120) (MIT=30)))
    (RULE=(SRC=192. 0. 2. 32) (DST=proxysvr) (SRV=cmon) (ACT=accept)))
  (PARAMETER_LIST=
    (MAX_GATEWAY_PROCESSES=8)
    (MIN_GATEWAY_PROCESSES=3)))
```

[例10-1](#)は、次のルールを示しています。

- 例の最初のルールでは、次のパラメータを設定しています。
 - SRC=192. 0. 2. 32/24は、クライアント接続用です。クライアントのIPアドレスまたはソースを指定します。
 - DST=sales-serverは接続先のホスト名を示しています。ACTパラメータは、actionの略で、受入れ、拒否または削除を示しています。ACTION_LISTパラメータは、それが受け入れられた場合の接続の属性を設定し、接続に関するデフォルトのパラメータ設定を、接続ベースでオーバーライドできます。
- 2番目のルールでは、次のパラメータを設定しています。

- SRC=192. 0. 2. 32およびDST=proxysvrは同じサーバーを表し、Oracle Connection ManagerとCMCTLは同じコンピュータ上にある必要があることを示しています。

関連項目:

[「アクセス制御の有効化」](#)

[表10-1](#)は、cman. oraファイルのルール・レベル・パラメータを説明します。

表10-1 cman.oraファイルのルール・レベル・パラメータ

パラメータ	説明
SRC	ソースのホスト名またはクライアントの IP アドレス。IP アドレスはサブネットで、192. 0. 2. 62/24 など指定できます。
DST	接続先のホスト名またはデータベース・サーバーの IP アドレス。IP アドレスはサブネット で、192. 0. 2. 62/24 など指定できます。
SRV	初期化パラメータ・ファイル(init. ora)の SERVICE_NAMES パラメータから取得した Oracle データベースのサービス名。 サービス名は、リスナーへの接続時に、接続記述子の一部としてクライアントによって指定されます。このサービス名は、ルール・リストに指定されたサービス名と比較されます。
ACT	前述の 3 つのパラメータに基づいた、受信要求の受入れ、拒否または削除。

複数のルールをRULE_LISTに定義できます。最初に適合したRULEのアクション(ACT)が接続要求に適用されます。ルールが定義されていない場合はすべての接続が受け入れられます。

次の例では、クライアント・コンピュータclient1-pcは、サービスsales. us. example. comへのアクセスが拒否されますが、クライアント192. 0. 2. 45はサービスdb1へのアクセスが許可されます。

```
(RULE_LIST=
(RULE=(SRC=client1-pc) (DST=sales-server) (SRV=sales. us. example. com) (ACT=reject))
(RULE=(SRC=192. 0. 2. 45) (DST=192. 0. 2. 200) (SRV=db1) (ACT=accept)))
```

関連項目:

Oracle Connection Managerパラメータの詳細は、[『Oracle Database Net Servicesリファレンス』](#)を参照してください。

親トピック: [Oracle Connection Managerの構成](#)

10.2 Oracle Connection Managerの構成方法

Oracle Connection Managerを構成するには、プロキシ・サーバー、データベースおよびクライアントを構成する必要があります。

す。次の項目では、一般的な手順について説明します。

- [Oracle Connection Managerホスト用のcman.oraファイルの構成](#)
- [Oracle Connection Managerを使用する場合のクライアントの構成](#)
- [Oracle Connection Managerを使用する場合のOracle Databaseサーバーの構成](#)
- [Traffic DirectorモードのOracle Connection Managerの構成](#)

親トピック: [Oracle Connection Managerの構成](#)

10.2.1 Oracle Connection Managerホスト用のcman.oraファイルの構成

cman.oraファイルは手動で変更します。次の手順では、sqlnet.oraファイルにパラメータを設定する方法について説明します。

1. ORACLE_HOME/network/adminディレクトリのcman.oraファイルにナビゲートします。
2. テキスト・エディタを使用してcman.oraファイルを開きます。
3. リスニング・エンドポイント(ADDRESS)を構成します。

リスニング・エンドポイントでは、Oracle Connection Managerリスナーのプロトコル・アドレスを指定します。Oracle Connection Managerのモニタリング・プロセスCMONでは、このアドレスを使用してリスナーへのゲートウェイ・プロセスに関する情報を登録します。データベースはこのアドレスを使用して、Oracle Connection Managerノードでサービス情報を登録します。

Oracle Connection Managerのリスナーは、常にTCP/IPプロトコルでリスニングします。

```
(ADDRESS=(PROTOCOL=tcp) (HOST=proxysvr) (PORT=1521))
```

ノート:



Oracle Connection Manager は、TCP/IP(バージョン 4 およびバージョン 6)などのプロトコルを使用してデータベースに接続できます。Oracle Database12c リリース 2 (12.2)以降、TCPS プロトコルもサポートされています

4. アクセス制御ルール・リスト(RULE_LIST)を構成します。

アクセス制御ルール・リストではリスナーにより、受入れ、拒否または削除される接続を指定します。

5. パラメータ・リスト(PARAMETER_LIST)を構成します。

パラメータ・リストにより、Oracle Connection Managerの属性を設定します。パラメータの形式は次のとおりです。

- グローバル・パラメータの場合、ルール・レベルのパラメータでオーバーライドされないかぎり、すべてのOracle Connection Manager接続に適用されます。グローバル・パラメータのデフォルト設定を変更するには、そのパラメータをPARAMETER_LISTに許容値と一緒に入力します。
 - ルール・レベルのパラメータがRULE_LISTのACTION_LISTセクションで有効になっている場合、そのルールによって指定された接続にのみ適用されます。これはグローバルな対象にオーバーライドします。
- [Oracle Connection Manager \(CMAN\)でのTransport Layer Securityの構成](#)
この構成ステップでは、CMANウォレットが証明書を使用して作成されたことを確認し、TLS付きTCP/IPを使用するリスニング・エンドポイントを作成し、ウォレットの場所を指定します。

- [アクセス制御の有効化](#)

親トピック: [Oracle Connection Managerの構成方法](#)

10.2.1.1 Oracle Connection Manager (CMAN)でのTransport Layer Securityの構成

この構成ステップでは、CMANウォレットが証明書で作成されていることを確認し、TLS付きTCP/IPを使用するリスニング・エンドポイントを作成し、ウォレットの場所を指定します。

ステップ1: CMANウォレットが作成されていることと、証明書があることを確認してください。

次のステップに進む前に、ウォレットが作成されていることと、ウォレットに証明書があることを確認する必要があります。次のステップを実行します。

1. Oracle Wallet Managerを起動します。

- UNIXの場合: \$ORACLE_HOME/binから次のコマンドを入力します。

```
owm
```

- Windowsの場合: 「スタート」を選択して「プログラム」に移動し、「Oracle-HOME_NAME」をクリックし、「Integrated Management Tools」を選択してから「Wallet Manager」をクリックします。

2. 「ウォレット」メニューから「オープン」を選択します。

ウォレットにReadyステータスの証明書が含まれ、自動ログインがオンになっている必要があります。自動ログインがオンになっていない場合は、「ウォレット」メニューから選択し、ウォレットを再度保存します。自動ログインがオンになります。

ステップ2: TLS付きTCP/IP (TCPS)を使用するリスニング・エンドポイントを作成し、cman.oraのCMAN側でCMANウォレットの場所を指定します。

次の例は、TCPSプロトコルを使用した cman.oraを示しています。

```
CMAN_1=
  (CONFIGURATION=
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=tcps) (HOST=proxysvr) (PORT=1522))
      (ADDRESS=(PROTOCOL=tcp) (HOST=proxysvr) (PORT=1523))
    )
    (RULE_LIST=
      (RULE=(SRC=*) (DST=*) (SRV=*) (ACT=accept))
    )
    (PARAMETER_LIST=
      (MAX_GATEWAY_PROCESSES=8)
      (MIN_GATEWAY_PROCESSES=3)
    )
  )
wallet_location =
  (SOURCE=
    (METHOD=File)
    (METHOD_DATA=
      (DIRECTORY=wallet_location)
    )
  )
)
```

親トピック: [Oracle Connection Managerホストのcman.oraファイルの構成](#)

10.2.1.2 アクセス制御の有効化

TCP/IP環境で指定されたデータベース・サーバーへのクライアント・アクセスを制御するには、RULE_LISTパラメータを使用します。このパラメータでフィルタリング・ルールを入力し、データベース・サーバーへの特定のクライアント・アクセスを許可または制限できます。

次の手順では、アクセス制御の構成方法を説明します。

1. テキスト・エディタを使用してcman.oraファイルを開きます。
2. 次の形式を使用して、RULE_LISTパラメータを更新します。

```
(RULE_LIST=  
  (RULE=(SRC=source_host)  
        (DST=destination_host)  
        (SRV=service)  
        (ACT=accept | reject | drop)))
```

関連項目:

[表10-1](#)

親トピック: [Oracle Connection Managerホストのcman.oraファイルの構成](#)

10.2.2 Oracle Connection Managerを使用する場合のクライアントの構成

Oracle Connection Managerでクライアントをデータベース・サーバーに転送するには、Oracle Connection Managerのプロトコル・アドレスを指定する[接続記述子](#)を使用して、tnsnames.oraファイルを構成します。このアドレスを使用して、クライアントからOracle Connection Managerコンピュータに接続できます。接続記述子は次のとおりです。

```
sales=  
(DESCRIPTION=  
  (ADDRESS=  
    (PROTOCOL=tcp)  
    (HOST=cman-pc)  
    (PORT=1521))  
  (CONNECT_DATA=  
    (SERVICE_NAME=example.com)))
```

次の手順では、Oracle Connection Managerのプロトコル・アドレスを構成する方法について説明します。

1. Oracle Net Managerを起動します。

関連項目:

[「Oracle Net Managerを使用したOracle Net Servicesの構成」](#)

2. ナビゲータ・ペインで、「ディレクトリ」または「ローカル」メニューから「サービス・ネーミング」を選択します。
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サービス名ウィザード」の「ようこそ」ページが表示されます。

4. 「ネット・サービス名」フィールドに名前を入力します。
5. 「次へ」をクリックします。
「プロトコル」ページが表示されます。
6. Oracle Connection ManagerのTCP/IPプロトコルを選択します。
7. 「次へ」をクリックします。
「プロトコル設定」ページが表示されます。
8. Oracle Connection Managerのポートとプロトコルを指定します。Oracle Connection Managerのデフォルトのポート番号は1521、プロトコルはTCP/IPです。

関連項目:

プロトコル・パラメータの設定については、[『Oracle Database Net Servicesリファレンス』](#)を参照してください。

9. 「次へ」をクリックします。
「サービス」ページが表示されます。
10. 「サービス名」フィールドにサービス名を入力し、接続タイプを選択します。

関連項目:

サービス名文字列の設定の詳細は、[「接続記述子について」](#)を参照

11. 「次へ」をクリックします。



ノート:

この時点では接続のテストはできないため、「テスト」はクリックしないでください。

12. 「終了」をクリックして構成を保存し、Netサービス名ウィザードを閉じます。

新規のネットワーク・サービス名とOracle Connection Managerプロトコル・アドレスが「サービス・ネーミング」フォルダに追加されます。

親トピック: [Oracle Connection Managerの構成方法](#)

10.2.3 Oracle Connection Managerを使用する場合のOracle Databaseサーバーの構成

データベース・サーバーの構成には、リモートからOracle Connection Managerでデータベース情報を登録する方法と、オプションでサーバーの多重化を構成する方法があります。



ノート:

Oracle Connection Managerに登録するデータベースがリモート・ノードにある場合は、`cman.ora`のパラメー

タ VALID_NODE_CHECKING_REGISTRATION を構成してリモート登録を許可する必要があります。

- [Oracle Connection Managerで使用するためのサービス登録の構成](#)
- [Oracle Connection Managerのセッションの多重化の有効化](#)

関連項目

- [VALID_NODE_CHECKING_REGISTRATION](#)

親トピック: [Oracle Connection Managerの構成方法](#)

10.2.3.1 Oracle Connection Managerで使用するためのサービス登録の構成

データベース・サーバーがOracle Connection Managerと通信できるようにするには、tnsnames.oraファイルにサービス名のエントリが含まれ、初期化パラメータ・ファイル (init.ora) に、Oracle Connection Managerのリスニング・アドレスを指定する記述子が含まれている必要があります。次の手順では、サービス登録を構成する方法について説明します。

1. tnsnames.oraファイルのサービス名エントリへのOracle Connection Manager別名を次のように解決します。

```
cman_listener_address =
  (DESCRIPTION=
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=tcp) (HOST=proxy_server_name) (PORT=1521))))
```

たとえば、別名 listener_cman は、tnsnames.oraファイル内の次のエントリに解決されます。

```
listener_cman=
  (DESCRIPTION=
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=tcp) (HOST=proxyserver1) (PORT=1521))))
```

2. init.oraファイルのOracle Connection Managerの別名を次のように指定します。この別名は、ステップ1の tnsnames.oraファイルで指定した別名です。

```
REMOTE_LISTENER=cman_listener_address
```

このアドレスはTCP、ポート1521であり、データベース・サーバーのTCP、ポート1521のデフォルト・ローカル・リスニング・アドレスではないため、別名を指定する必要があります。

たとえば、ホストproxyserver1で動作しているOracle Connection Managerリスナーのステップ1で指定した別名は、init.oraファイル上で次のようになっている場合があります。

```
REMOTE_LISTENER=listener_cman
```

3. データベースがリモート・ノードに存在する場合、cman.oraのREGISTRATION_INVITED_NODESを構成します。たとえば:
(registration_invited_nodes=sales.us.example.com, 10.245.129.60)
4. 初期化パラメータ・ファイルがOracle Connection Managerの別名で構成されると、リスナー登録(LREG)プロセスは、Oracle Connection Managerリスナーとしてデータベース情報を登録できます。変更を登録するには、次のコマンドを使用します。

```
SQL> ALTER SYSTEM REGISTER
```

関連項目:

親トピック: [Oracle Connection Managerを使用する場合のOracle Databaseサーバーの構成](#)

10.2.3.2 Oracle Connection Managerのセッションの多重化の有効化

Connection Managerでセッションの多重化を利用できるようにするには、次に示すように初期化パラメータ・ファイル (init.ora)のDISPATCHERSパラメータに、属性PROTOCOLおよびMULTIPLEXを設定します。

```
DISPATCHERS="(PROTOCOL=tcp) (MULTIPLEX=on)"
```

[表10-2](#)では、多重化の異なるレベルを設定するパラメータを示しています。

表10-2 セッションを多重化するパラメータ

属性	説明
PROTOCOL	ディスパッチャがリスニング・エンドポイントを生成するときに使用するネットワーク・プロトコルです。
MULTIPLEX	このパラメータは、次のようにセッションの多重化に使用します。 <ul style="list-style-type: none">● 1、on、yes、true または both が指定された場合は、受信および送信の両方のネットワーク・セッションの多重化が有効となります。● in が指定された場合は、クライアントからの受信ネットワーク・セッションの多重化が有効となります。● out が指定された場合は、送信ネットワーク・セッションの多重化が有効となります。● 0(ゼロ)、off、no または false が指定された場合は、受信および送信両方のネットワーク・セッションの多重化が無効となります。

関連項目:

- 共有サーバーの構成の詳細は、[「共有サーバー・アーキテクチャの構成」](#)を参照してください
- パラメータの完全なリストおよびそのデフォルト値と許容値については、[『Oracle Database Net Servicesリファレンス』](#)を参照してください。

親トピック: [Oracle Connection Managerを使用する場合のOracle Databaseサーバーの構成](#)

10.2.4 Traffic DirectorモードのOracle Connection Managerの構成

Traffic DirectorモードのOracle Connection Managerは、データベース・クライアントとデータベース・インスタンスとの間に配置されたプロキシです。

- [Traffic DirectorモードのOracle Connection Managerの使用について](#)
Traffic DirectorモードのOracle Connection Managerは、サポートされているデータベース・クライアントとデータベース・インスタンスとの間に配置されたプロキシです。
- [Traffic DirectorモードのOracle Connection Manager用のcman.oraファイルの構成](#)
cman.oraファイルのTDM=YES設定を使用して、Traffic DirectorモードのOracle Connection Managerを設定できます。
- [Traffic DirectorモードのOracle Connection Managerのプロキシ認証用のウォレットの構成](#)
- [Traffic DirectorモードのOracle Connection Managerのプロキシ認証用のデータベースの構成](#)
- [Traffic DirectorモードのOracle Connection Managerでのサービス登録の構成](#)
- [Traffic DirectorモードのOracle Connection Managerのプロキシ常駐接続プールの構成](#)
- [計画外イベントのためのTraffic DirectorモードのOracle Connection Managerの構成](#)
- [計画済停止イベントのためのTraffic DirectorモードのOracle Connection Managerの構成](#)
- [サービス・アフィニティのTraffic DirectorモードのOracle Connection Managerの構成](#)
着信接続要求をルーティングするためのデフォルトの負荷分散メカニズムを変更するように、Traffic DirectorモードのOracle Connection Managerを構成します。
- [Traffic DirectorモードのOracle Connection ManagerでのTransport Layer Securityの構成](#)
この構成ステップでは、Traffic Directorモード・ウォレットのOracle Connection Managerが証明書を使用して作成されたことを確認し、TLS付きTCP/IPを使用するリスニング・エンドポイントを作成し、ウォレットの場所を指定します。
- [Traffic DirectorモードのOracle Connection Managerの制限事項](#)
これらの機能は、Traffic DirectorモードのOracle Connection Managerの制限事項では、すべてのドライバでサポートされていません。

親トピック: [Oracle Connection Managerの構成方法](#)

10.2.4.1 Traffic DirectorモードのOracle Connection Managerの使用について

Traffic DirectorモードのOracle Connection Managerは、サポートされているデータベース・クライアントとデータベース・インスタンスとの間に配置されたプロキシです。

現在のデータベースのOCIクライアントまたはサポートされている古いバージョンのOCIクライアント(Oracle Database 11gリリース2 (11.2)以降)は、Traffic DirectorモードのOracle Connection Managerに接続できます。Traffic DirectorモードのOracle Connection Managerでは、高可用性(HA)が改善され(計画済、計画外を問わず)、接続の多重化がサポートされ、ロード・バランシングが提供されます。また、この機能には、Oracle Connection Manager (CMAN)やサービスの停止イベントに関する計画済停止についてOCIクライアントに伝えるインバンド・クライアント通知メカニズムが用意されています。Traffic DirectorモードのOracle Connection Managerを構成するには、cman.ora構成ファイルで追加のCMANパラメータを指定する必要があります。

Traffic DirectorモードのOracle Connection Managerとして機能するCMANを構成するには、tdmやtdm_threading_modelなどの新しいパラメータをcman.ora構成ファイルに追加する必要があります。Oracle Connection Manager (CMAN)は、Oracle RACデータベースとRAC以外のデータベースの両方のための標準Oracle Netプロキシです。

Traffic DirectorモードのOracle Connection Managerが接続するデータベースには、アプリケーション・ユーザーとして接続できるよう、CONNECT THROUGH権限を付与されたtdmなどのユーザーが存在する必要があります。Traffic DirectorモードのOracle Connection Managerはプロキシ認証を使用し、このユーザーとして接続します。

関連項目:

Traffic DirectorモードのOracle Connection Managerをサポートするcman.oraファイル内の既存のパラメータについては、[Oracle Database Net Servicesリファレンス](#)を参照してください。

CONNECT THROUGHの詳細は、[Oracle Databaseセキュリティ・ガイド](#)を参照してください。

このリリースでは、Traffic DirectorモードのOracle Connection Managerは次のようなサポートを提供します。

- 透過的パフォーマンスの強化および接続の多重化
- アプリケーション停止時間ゼロ
 - 計画済のデータベース・メンテナンスまたはPDBの再配置
 - 大部分が読取りのワークロードでの計画外データベース停止
- 単一障害点のないTraffic DirectorモードのOracle Connection Managerの高可用性
- セキュリティと分離

操作のモード

- プールされた接続モード

これには、次のデータベース・クライアント・リリースを使用する任意のアプリケーションが含まれます。

- OCIおよびオープン・ソース・ドライバ(11.2.0.4以降)
- JDBC (12.1以降)
- ODP.NET (12.2以降)

また、アプリケーションでは、データベース常駐接続プール(DRCP)を認識する必要があります。これは、アプリケーションの接続文字列で指定します(SERVER=POOLED)。

- プールされていない接続(または専用)モード

これには、データベース・クライアント・リリース11.2.0.4以降を使用する任意のアプリケーションが含まれます。このモードでは、接続の多重化などの一部の機能は使用できません。

透過的パフォーマンスの強化およびセッションの多重化

- 文キャッシュ、行のプリフェッチおよび結果セット・キャッシュがすべての操作モードで自動的に有効になります。
- プロキシ常駐接続プール(PRCP、プロキシ・モードのDRCP)を使用したデータベース・セッションの多重化(プールされたモードのみ)。アプリケーションは、TDMとデータベースとの間における透過的な接続時ロード・バランシングおよび実行時ロード・バランシングを利用できます。
- Traffic DirectorモードのOracle Connection Managerの複数のインスタンス: クライアント側の接続時ロード・バランシングにより、またはロード・バランサ(BigIP、NGINXなど)により、アプリケーションのスケラビリティが向上します

アプリケーション停止時間ゼロ - 計画済のデータベース・メンテナンスまたはPDBの再配置

- プールされたモード

Traffic DirectorモードのOracle Connection Managerは、計画済停止のONSイベントに応答し、作業をリダイレクトします。要求が完了すると、Traffic DirectorモードのOracle Connection Managerのプールから接続がド

レインされます。

サービスの再配置は、データベース・リリース11.2.0.4以降でサポートされています。

PDBの再配置: PDBが再配置されると、ONSが構成されていなくても、Traffic DirectorモードのOracle Connection Managerはインバンド通知に応答します(18.1以降のサーバーの場合のみ)

- プールされていないモードまたは専用モード

クライアントからのリクエスト境界情報は受信しません。Traffic DirectorモードのOracle Connection Managerは、多くのアプリケーションの計画済停止をサポートしています(リクエストまたはトランザクションの境界を越えて単純なセッション状態やカーソル状態のみを保持する必要がある場合)。これには、次の操作が含まれます。

- トランザクション境界でサービスまたはPDBを停止するか、Oracle Database 18cの「継続的なアプリケーション可用性」を利用してリクエスト境界でサービスを停止します。
- Traffic DirectorモードのOracle Connection Managerは、「TAFのフェイルオーバーのリストア」を利用して、再接続して単純な状態をリストアします。

計画外のデータベース停止

プールされたモードとプールされていない(専用)モードの両方で、Traffic DirectorモードのOracle Connection Managerは、「TAFのフェイルオーバーのリストア」を利用して単純なセッション状態またはカーソル状態をリストアし、SELECT文と最初のDML文をリプレイすることで、大部分が読取りのアプリケーションの計画外停止をサポートします。

セキュリティと分離

Traffic DirectorモードのOracle Connection Managerには、次のような機能があります。

- TCP/TCPSとプロトコルの変換をサポートするデータベース・プロキシ
- IPアドレス、サービス名およびTLSウォレットに基づくファイアウォール
- マルチテナント環境でのテナントの分離
- サービス拒否攻撃とファジング攻撃からの保護
- オンプレミス・データベースとOracle Cloudの間のデータベース・トラフィックのセキュア・トンネリング

Traffic DirectorモードのOracle Connection Managerの高可用性

これには次が含まれます。

- ロード・バランサやクライアント側ロード・バランシング、接続文字列でのフェイルオーバーを使用した、Traffic DirectorモードのOracle Connection Managerの複数のインスタンス
- Traffic DirectorモードのOracle Connection Managerのインスタンスのローリング・アップグレード
- 計画済停止におけるクライアントからTraffic DirectorモードのOracle Connection Managerへの既存の接続のクローズ
- Oracle Database!リリース18c以降のクライアントへのインバンド通知。以前のリリースのクライアントの場合、現在のリクエストのレスポンスとともに通知が送信されます。

親トピック: [Traffic DirectorモードのOracle Connection Managerの構成](#)

10.2.4.2 Traffic DirectorモードのOracle Connection Manager用のcman.oraファイルの構成

Traffic DirectorモードのOracle Connection Managerは、cman.oraファイルのTDM=YES設定を使用して設定できます。

Traffic DirectorモードのOracle Connection Managerのサンプルcman.ora構成

```
CMAN_1=
  (CONFIGURATION=
    (ADDRESS_LIST=
      (ADDRESS= (PROTOCOL=tcp) (HOST=TDMHOST) (PORT=1522))
    )
    (RULE_LIST=
      (RULE= (SRC=*) (DST=*) (SRV=*) (ACT=accept))
    )
    (PARAMETER_LIST=
      (TDM=YES)
      (TDM_THREADING_MODE=DEDICATED)
      (MAX_GATEWAY_PROCESSES=8)
      (MIN_GATEWAY_PROCESSES=3)
    )
  )
)
wallet_location =
  (SOURCE=
    (METHOD=File)
    (METHOD_DATA= (DIRECTORY=wallet_location)
  )
)
)
SQLNET.WALLET_OVERRIDE = TRUE
```

親トピック: [Traffic DirectorモードのOracle Connection Managerの構成](#)

10.2.4.3 Traffic DirectorモードのOracle Connection Managerのプロキシ認証用のウォレットの構成

Traffic DirectorモードのOracle Connection Managerは、cman.oraファイルで構成する必要があるウォレットを使用してデータベースに接続します。このウォレットには、Traffic DirectorモードのOracle Connection Managerでプロキシ認証に使用されるtdmユーザーのユーザー名およびパスワード情報が含まれています。このtdmユーザーは、Traffic DirectorモードのOracle Connection Managerが接続するすべてのデータベースに存在する必要があります。

関連項目:

[Oracle Databaseエンタープライズ・ユーザー・セキュリティ管理者ガイド](#)

次の設定では、Traffic DirectorモードのOracle Connection Managerからのすべてのアウトバウンド接続で、プロキシ認証用に指定された場所のウォレットの資格証明を使用します。

```
WALLET_LOCATION =
      (SOURCE =
        (METHOD = FILE)
        (METHOD_DATA =
          (DIRECTORY = wallet_location)
```

```
SQLNET.WALLET_OVERRIDE = TRUE
```

ウォレットは、各サービスに構成されている必要があります。新しいサービスが追加された場合、同じウォレットを使用して新しいサービス用の資格証明を指定する必要があります。

ノート:



新しいサービスが追加され、この新しいサービス用の資格証明がウォレットに追加されるたびに、変更を有効にするために Oracle Cloud Traffic Manager を再起動する必要があります。

- [外部パスワード・ストアを使用するためのTraffic DirectorモードのOracle Connection Managerの有効化](#)
構成ステップでは、ウォレットを作成し、そのウォレットに各データベース・サービスごとのデータベース接続資格証明を作成します。

関連項目

- [Traffic DirectorモードのOracle Connection Managerのプロキシ認証用のデータベースの構成](#)

親トピック: [Traffic DirectorモードのOracle Connection Managerの構成](#)

10.2.4.3.1 Traffic DirectorモードのOracle Connection Managerでの外部パスワード・ストアの使用

構成ステップでは、ウォレットを作成し、そのウォレットに各データベース・サービスごとのデータベース接続資格証明を作成します。

ステップ1: コマンドラインで次の構文を使用して、Traffic DirectorモードのOracle Connection Managerでウォレットを作成します。

```
mkstore -wrl wallet_location -create
```

wallet_locationは、ウォレットを作成して格納するディレクトリのパスです。このコマンドにより、指定した場所にOracleウォレットが作成され、自動ログイン機能が使用可能になります。自動ログイン機能を使用すると、パスワードを指定せずにTraffic DirectorモードのOracle Connection Managerからウォレットの内容にアクセスできます。

ステップ2: コマンドラインで次の構文を使用して、ウォレットにデータベース接続の資格証明を作成します。

```
mkstore -wrl wallet_location -createCredential db_service_name username password
```

wallet_locationは、ステップ1でウォレットを作成したディレクトリのパスです。db_service_nameは、アプリケーションがOracle Connection Managerに接続するときに接続文字列で使用するサービス名です。usernameとpasswordは、tdmのユーザー名とパスワードです。

Traffic DirectorモードのOracle Connection Managerを使用してアクセスする必要があるデータベース・サービスごとに、このステップを繰り返します。

ノート:



- 指定されたデータベースのすべてのサービス間で同じ tdm ユーザーを使用できます。ただし、必要に応じ

て、各サービスに異なる tdm ユーザーを関連付けることもできます。

- プラガブル・データベース(PDB)のサービスの場合、tdm ユーザーの設定には次の 2 つの選択肢があります。

共通 tdm ユーザー: tdm ユーザーを共通ユーザーにすることができます。この場合、Traffic Director モードの Oracle Connection Manager は、マルチテナント・コンテナ・データベース(CDB)内の異なる PDB からのユーザーをプロキシ認証するために、資格証明のセットを 1 つ使用します。

PDB ごとの tdm ユーザー: tdm ユーザーを PDB 固有のユーザーにすることができます。この場合、Traffic Director モードの Oracle Connection Manager は、特定の PDB のユーザーをプロキシ認証するために、PDB 固有のプロキシ・ユーザーを使用します。

- TLS (TCPS) 付き TCP/IP 構成の場合、Traffic Director モードの Oracle Connection Manager ウォレットがすでに作成されています。したがって、ステップ 1 を省略し、mkstore で wallet_location を指定する際、TCPS 構成に使用したのと同じ場所を使用します。

親トピック: [Traffic DirectorモードのOracle Connection Managerのプロキシ認証用のウォレットの構成](#)

10.2.4.4 Traffic DirectorモードのOracle Connection Managerのプロキシ認証用のデータベースの構成

アプリケーションがTraffic DirectorモードのOracle Connection Managerを介して接続するすべてのデータベースには、tdmなどのユーザーが必要です。Traffic DirectorモードのOracle Connection Managerはプロキシ認証を使用し、tdm ユーザーとしてデータベースに接続します。Traffic DirectorモードのOracle Connection Managerを経由して接続する必要があるすべてのユーザーには、次のようにCONNECT THROUGH tdm権限を付与する必要があります。

```
ALTER user SCOTT GRANT CONNECT THROUGH tdm
```

親トピック: [Traffic DirectorモードのOracle Connection Managerの構成](#)

10.2.4.5 Traffic DirectorモードのOracle Connection Managerでのサービス登録の構成

[Oracle Connection Managerで使用するためのサービス登録の構成](#)を参照してください。

親トピック: [Traffic DirectorモードのOracle Connection Managerの構成](#)

10.2.4.6 Traffic DirectorモードのOracle Connection Managerのプロキシ常駐接続プールの構成

Traffic DirectorモードのOracle Connection Manager 18.1は、プロキシ常駐接続プーリング(PRCP)をサポートしています。PRCPは、ターゲット・データベースへの少数のサーバー接続で構成される接続プールを使用して、ルーティングされる多数のクライアント接続に対して接続サービスを提供します。PRCPは、データベース層での接続負荷(接続時のメモリー使用量)を軽減し、実行時ロード・バランシング(RLB)を実行してパフォーマンスを高めます。これにより、データベースに対する接続負荷を軽減できます。

プロキシ常駐接続プールは、oraaccess.xml 構成ファイルで次のパラメータのセットを指定することで構成されます。

- <session_pool> – セッション・プール・パラメータを設定します。
- <enable> – trueまたはfalseを指定して、セッション・プールの構成を有効にします。これは必須のパラメータです。つまり、<session_pool>が構成されている場合、<enable>の構成も必要です。
- <min_size> – プール内の最小接続数。デフォルト値は0です。Traffic DirectorモードのOracle Connection Managerは異機種間プールのシナリオであるため、他のすべての値は無視されます。
- <max_size> – プール内の最大接続数。これは必須のパラメータであるため、デフォルト・パラメータはありません。つまり、<session_pool>が構成されている場合、<max_size>の構成も必要です。
- <increment> – プール拡張時のプール内の接続数の増加数。デフォルト値は1です。
- <inactivity_timeout> – プール内でアイドル状態の接続が切断されるまでの最大時間(秒単位)。デフォルト値は0です。これは制限されていないことを意味します。
- <max_use_session> – 接続を取得し、プールに解放できる最大回数。デフォルト値は0です。これは制限されていないことを意味します。
- <max_life_time_session> – プール内の接続の有効期限(秒単位)。デフォルト値は0です。これは制限されていないことを意味します。

oraaccess.xmlファイルを使用して、必要な接続サービスごとにプロキシ常駐接続プールを構成できます。次の例では、2つのグループの接続パラメータが示され、それぞれがsales_configとhr_configというconfig別名に関連付けられています。アプリケーションが使用する各接続文字列はそれぞれのconfig別名にマップされているため、2つのプロキシ常駐接続プールが示されています。

```
<oraaccess xmlns="http://xmlns.oracle.com/oci/oraaccess"
  xmlns:oci="http://xmlns.oracle.com/oci/oraaccess"
  schemaLocation="http://xmlns.oracle.com/oci/oraaccess
  http://xmlns.oracle.com/oci/oraaccess.xsd">
  <default_parameters>
</default_parameters>
  <!--
    Create configuration descriptions, which are
    groups of connection parameters associated with
    a config_alias.
  -->
  <config_descriptions>
    <config_description>
      <config_alias> sales_config </config_alias>
      <parameters>
        <session_pool>
          <enable>true</enable>
          <min_size> 10 </min_size>
          <max_size> 100 </max_size>
          <increment> 5 </increment>
        </session_pool>
      </parameters>
    </config_description>
    <config_description>
      <config_alias> hr_config </config_alias>
      <parameters>
        <session_pool>
          <enable>true</enable>
          <max_size> 10 </max_size>
        </session_pool>
      </parameters>
    </config_description>
  </config_descriptions>
</oraaccess>
```

```

    </parameters>
  </config_description>
</config_descriptions>
<!--
    Now map the connection string used by the application
    with a config_alias.
-->
<connection_configs>
  <connection_config>
    <connection_string>sales.us.example.com</connection_string>
    <config_alias>sales_config</config_alias>
  </connection_config>
  <connection_config>
    <connection_string>hr.us.example.com</connection_string>
    <config_alias>hr_config</config_alias>
  </connection_config>
</connection_configs>
</oraaccess>

```

関連項目:

Oracle CMAN構成ファイルの概要については、[Oracle Database Net Servicesリファレンス](#)を参照してください

親トピック: [Traffic DirectorモードのOracle Connection Managerの構成](#)

10.2.4.7 計画外イベントのためのTraffic DirectorモードのOracle Connection Managerの構成

Traffic DirectorモードのOracle Connection Managerは、高速アプリケーション通知(FAN)イベントに暗黙的にサブスクライブします。このためには、oraaccess.xmlでeventsを有効にする必要があります。

DBMS_SERVICEまたはSRVCTL (Oracle RACデータベースの場合)を使用して、特定のサービスのCOMMIT_OUTCOMEを指定します。

関連項目:

- DBMS_SERVICEの詳細は、[Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)を参照してください
- SRVCTLに関する詳細は、[Oracle Clusterware管理およびデプロイメント・ガイド](#)を参照してください

親トピック: [Traffic DirectorモードのOracle Connection Managerの構成](#)

10.2.4.8 計画済停止イベントのためのTraffic DirectorモードのOracle Connection Managerの構成

Traffic DirectorモードのOracle Connection Managerは、高速アプリケーション通知(FAN)イベントに暗黙的にサブスクライブします。このためには、oraaccess.xmlでeventsを有効にする必要があります。

計画済停止イベントの場合、DBMS_SERVICEまたはSRVCTLを使用してサービスを構成し、failover_modeをselectに、

commit_outcomeをTRUEに、failover_restoreをLEVEL1に設定します。

関連項目:

- DBMS_SERVICEの詳細は、[Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)を参照してください
- SRVCTLに関する詳細は、[Oracle Clusterware管理およびデプロイメント・ガイド](#)を参照してください

親トピック: [Traffic DirectorモードのOracle Connection Managerの構成](#)

10.2.4.9 サービス・アフィニティのためのTraffic DirectorモードのOracle Connection Managerの構成

Oracle Connection ManagerをTraffic Directorモードで構成し、着信接続要求をルーティングするためのデフォルトの負荷分散メカニズムを変更します。

デフォルトでは、Traffic DirectorモードのOracle Connection Managerはサービス・アフィニティを使用して、着信接続要求をルーティングするゲートウェイを選択します。新しい接続要求はすべて、データベース・サービスに関連付けられたゲートウェイにルーティングされます。

cman.oraパラメータSERVICE_AFFINITYを使用して、デフォルトの動作を変更し、パラメータをONまたはOFFに設定します。

関連項目

- [『Oracle Database Net Servicesリファレンス・ガイド』](#)
- [Oracle Multitenant管理者ガイド](#)

親トピック: [Traffic DirectorモードのOracle Connection Managerの構成](#)

10.2.4.10 Traffic DirectorモードのOracle Connection ManagerでのTransport Layer Securityの構成

この構成ステップでは、Traffic Directorモード・ウォレットのOracle Connection Managerが証明書を使用して作成されたことを確認し、TLS付きTCP/IPを使用するリスニング・エンドポイントを作成し、ウォレットの場所を指定します。

ステップ1: Traffic DirectorモードのOracle Connection Managerのウォレットが作成されていることと、証明書があることを確認してください。

次のステップに進む前に、ウォレットが作成されていることと、ウォレットに証明書があることを確認する必要があります。次のステップを実行します。

1. Oracle Wallet Managerを起動します。
 - UNIXの場合: \$ORACLE_HOME/binから次のコマンドを入力します。
owm
 - Windowsの場合: 「スタート」を選択して「プログラム」に移動し、「Oracle-HOME_NAME」をクリックし、「Integrated Management Tools」を選択してから「Wallet Manager」をクリックします。
2. 「ウォレット」メニューから「オープン」を選択します。

ウォレットにReadyステータスの証明書が含まれ、自動ログインがオンになっている必要があります。自動ログインがオンになっていない場合は、「ウォレット」メニューから選択し、ウォレットを再度保存します。自動ログインがオンになります。

ステップ2: TLS付きTCP/IP (TCPS)を使用するリスニング・エンドポイントを作成し、cman.oraのCMAN側でTraffic DirectorモードのOracle Connection Managerのウォレットの場所を指定します。

次の例は、TCPSプロトコルを使用したcman.oraを示しています。

```
CMAN_1=
  (CONFIGURATION=
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=tcps) (HOST=proxysvr) (PORT=1522))
      (ADDRESS=(PROTOCOL=tcp) (HOST=proxysvr) (PORT=1523))
    )
    (RULE_LIST=
      (RULE=(SRC=*) (DST=*) (SRV=*) (ACT=accept))
    )
    (PARAMETER_LIST=
      (MAX_GATEWAY_PROCESSES=8)
      (MIN_GATEWAY_PROCESSES=3)
    )
  )
wallet_location =
  (SOURCE=
    (METHOD=File)
    (METHOD_DATA=
      (DIRECTORY=wallet_location)
    )
  )
)
SQLNET.WALLET_OVERRIDE = TRUE
```

親トピック: [Traffic DirectorモードのOracle Connection Managerの構成](#)

10.2.4.11 Traffic DirectorモードのOracle Connection Managerの制限事項

これらの機能は、Traffic DirectorモードのOracle Connection Managerの制限事項では、すべてのドライバでサポートされていません。

- 分散トランザクション
- アドバンスド・キューイング(AQ)
- データベースの起動または停止のコール
- シャーディング
- XML
- SQL翻訳
- プロキシ認証およびTLS外部認証(DNなど)
- オブジェクトREF
- セッションの切替え
- ナビゲーション・アクセスのためのOCIObject*コール

- OCIPickerImage*コール
- OCIAnyData*コール
- OCIDirPath*コール
- OCISubscription*コール
- OCILCR*、OCIXStream*コール
- OCIStmtExecute:
 - スクロール可能カーソル
 - 暗黙的結果
- 変更通知(CQN)
- クライアント結果キャッシュ
- OCI_MIGRATEモード
- DRCP: マルチプロパティのタグとPL/SQLコールバック
- アプリケーション・コンティニューイティ
- OCIConnectionPool
- SYSDBA、SYSOPERなどの認証
- Real Application Security (Javaの場合のみ)
- データ型
 - PL/SQL索引付き表のバインド
- 一括コピー(ODP.Netのみ)
- 自己チューニング(ODP.Netのみ)
- ASO暗号化およびサポートされているアルゴリズム(ASOのみ)

親トピック: [Traffic DirectorモードのOracle Connection Managerの構成](#)

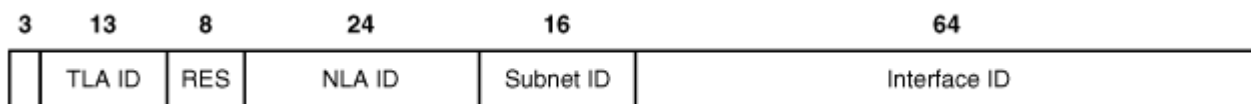
10.3 IPv4およびIPv6のブリッジとしてのOracle Connection Managerの使用

一部のデータベース接続環境では、クライアントとデータベースが異なるバージョンのIPプロトコルを使用できるため、完全な接続性が存在しません。この場合、接続に少なくとも2つのホップが異なるバージョンのIPプロトコルを使用しています。たとえば、要求は、IPv4ソースからIPv6宛先、IPv6ソースからIPv4宛先、またはIPv4ネットワークを通じてIPv6からIPv6に渡されます。

[IPv4](#)と[IPv6](#)の間のネットワーク・ブリッジとしてOracle Connection Managerを使用できます。Oracle Connection Managerがブリッジとして機能するには、少なくとも1つのIPv4インタフェースと少なくとも1つのIPv6インタフェースで構成されたデュアル・スタック・ホスト上で実行する必要があります。

IPv6アドレスに基づいてフィルタにかけるには、Oracle Connection Managerのフィルタリング機能を使用します。ルールは、完全なIPアドレスまたは部分的なIPアドレスに基づいて設定できます。[図10-1](#)は、IPv6アドレスの形式を示しています。

図10-1 IPv6アドレスの形式



図の上部の数字は、アドレス内のビット数を示しています。IPv6アドレスの各16進文字は、4ビットを表します。ビット4から16はアドレスの最上位レベルのアグリゲータ識別子(TLA ID)部分です。ビット25から49は次のレベルのアグリゲータ識別子(NLA ID)部分です。

たとえば、アドレス2001:0db8::203:BAFF:FE0F:C74Bでは、最初の4つの16進文字(2001)のバイナリ表記は次のようになります。

```
0010000000000001
```

したがって、アドレス内の先頭の3ビットは001になります。アドレスのTLA ID部分は00000000000001になります。

次の手順では、IPv6アドレス用のルール・フィルタを作成する方法について説明します。

1. ORACLE_HOME/network/adminディレクトリにあるcman.oraファイルにナビゲートします。
2. テキスト・エディタを使用してcman.oraファイルを開きます。
3. IPv6アドレス形式に基づいて、RULE_LISTでRULEを作成します。

たとえば、ソース・ホストがアドレス2001:0db8::203:BAFF:FE0F:C74BのIPv6専用ホストで、宛先がSALES1593という名前のIPv4専用ホストであるとします。次のいずれかのルールを作成して、Oracle Connection ManagerをIPv6とIPv4の間のブリッジとして構成します。

ルールのタイプ	説明	例
サブネット ID に基づいたフィルタ	フィルタリングはサブネット ID 以下の 64 ビットに基づいています。	(RULE = (SRC = 2001:0db8::203:BAFF:FE0F:C74B/64) (DST = SALES1593) (SRV = SALES) (ACT = ACCEPT) (ACTION_LIST = (AUT=ON) (MOCT=10) (MIT=30) (CONN_STATE=YES)))
NLA ID に基づいたフィルタ	フィルタリングは NLA ID 以下の 48 ビットに基づいています。	(RULE = (SRC = 2001:0db8::203:BAFF:FE0F:C74B/48) (DST = SALES1593) (SRV = SALES) (ACT = ACCEPT) (ACTION_LIST = (AUT=ON) (MOCT=10) (MIT=30) (CONN_STATE=YES)))
TLA ID に基づいたフィルタ	フィルタリングは TLA ID 以下の 16 ビットに基づいています。	(RULE = (SRC = 2001:0db8::203:BAFF:FE0F:C74B/16) (DST = SALES1593) (SRV = SALES) (ACT = ACCEPT) (ACTION_LIST = (AUT=ON) (MOCT=10) (MIT=30) (CONN_STATE=YES)))

ルールのタイプ	説明	例
ビット数に基づいたフィルタ	フィルタリングはアドレスの先頭の60ビットに基づいています。	(RULE = (SRC = 2001:0db8::203:BAFF:FE0F:C74B/60) (DST = SALES1593) (SRV = SALES) (ACT = ACCEPT) (ACTION_LIST = (AUT=ON) (MOCT=10) (MIT=30) (CONN_STATE=YES)))

関連項目:

- IPv6についてさらに学習するには、[「TCP/IPプロトコルについて」](#)を参照してください
- [「接続記述子のIPv6アドレスについて」](#)
- [「リスニング用プロトコル・アドレスの構成」](#)

親トピック: [Oracle Connection Managerの構成](#)

10.4 Oracle Connection Managerの起動および停止

Oracle Connection Managerを構成したら、Oracle Connection Manager制御(CMCTL)ユーティリティを使用してこれを起動および管理できます。

オペレーティング・システムのコマンドラインで、このユーティリティの基本的な構文は次のとおりです。

```
cmctl [command] [argument1 . . . argumentN] [-c instance_name]
```

-cパラメータは、管理するOracle Connection Managerインスタンスを指定します。インスタンスはcman.oraファイルに定義されます。

ノート:



インストール時または以前のOracle Connection Managerセッション時に指定した場合、パスワードの入力を求められます。ただし、これにより画面上にパスワードが表示されるため、潜在的なセキュリティ上のリスクがあります。Oracleでは、コマンドラインでパスワード・オプション(-p)を使用しないことをお勧めします。

Oracle Connection Manager制御ユーティリティを使用してOracle Connection Managerを起動および停止するには:

1. cman.oraファイルを作成します。
サンプル・ファイルは、Oracle Connection Managerをインストールした後のORACLE_HOME/network/admin/samplesディレクトリにあります。
2. 次の方法のいずれかを使用して、Oracle Connection Managerを起動します。
 - コマンド行で、次のコマンドを実行します。

STARTUPコマンドを実行します。たとえば:

```
cmctl STARTUP -c [cman_example_instance]
```

次のコマンドは、リスナー、CMADMIN (Connection Manager Administration)および `cman_example_instance` という名前のインスタンスのゲートウェイ・プロセスを起動します。

- CMCTLプロンプトで次の操作を行います。

コマンドラインで、引数なしで `cmctl` と入力してCMCTLプロンプトを取得し、ADMINISTERおよびSTARTUPコマンドを実行します。たとえば:

```
cmctl
CMCTL> ADMINISTER [cman_example_instance]
CMCTL> STARTUP
```

3. 次のいずれかの方法を使用して、Oracle Connection Managerの実行中のインスタンスを停止します。

- コマンド行で、次のコマンドを実行します。

SHUTDOWNコマンドを実行します。たとえば:

```
cmctl SHUTDOWN -c [cman_example_instance]
```

- CMCTLプロンプトで次の操作を行います。

コマンドラインで、引数なしで `cmctl` と入力してCMCTLプロンプトを取得し、ADMINISTERおよびSHUTDOWNコマンドを実行します。たとえば:

```
cmctl
CMCTL> ADMINISTER [cman_example_instance]
CMCTL> SHUTDOWN
```

関連項目

- [Oracle Connection Managerのアーキテクチャの理解](#)
- [『Oracle Database Net Servicesリファレンス・ガイド』](#)

親トピック: [Oracle Connection Managerの構成](#)

10.5 Oracle Connection Managerの機能強化

Oracle Connection Managerは、Oracle Databaseサーバーへのリクエストをプロキシおよび選別します。

- Oracle Connection Managerは、Transport Layer Security (TLS)プロトコルをサポートすることにより、サーバーへのより安全なアクセスを提供します。このサポートにより、データベース・クライアントは、Oracle Connection Managerを経由したTCPSプロトコルによるサーバーとの通信が可能になります。Oracle Connection Managerを構成して、一方の側にTLS接続を、もう一方の側に非TLS接続を設定することもできます。これは、外部クライアント向けの内部データベース・サーバーをインターネットから隠すための安全策としても機能します。

多重化: データベース・クライアントがリクエストしたエンドポイントに、すでにゲートウェイがTCPS接続している場合、新しい接続はゲートウェイによって同じ接続で多重化されます。

- 複数のプロトコル・アドレス上でのリスニングが可能になっています。複数のリスニング・エンドポイントにより、Oracle Connection ManagerはTCPとTCPSの両方を同時にサポートできるようになりました。従来の単一プロトコル・アドレスの構成も引き続きサポートされます。たとえば:

```
CMAN_ALIAS=
(configuration=
(address_list=
```

```
(address=(protocol=TCP) (host=a. b. c. d) (port=12522))
(address=(protocol=TCPS) (host=a. b. c. d) (port=12523))
)
..
)
```

- ネットワーク・データの圧縮機能も追加されています。これにより、ネットワーク・スループットが向上し、Oracle DatabaseクライアントとOracle Databaseサーバーとの間のデータ転送が高速化されます。これは、データベース・クライアント、データベース・サーバー、およびネクスト・ホップの圧縮機能に応じて、様々な方法で実現されます。圧縮は、あらゆる2つのノード間でネゴシエートされたときに有効化されます。

連続した3つ以上のノードが圧縮をサポートしていてネゴシエートした場合、このケースは、中間ノードが圧縮解除を実行することなく単に圧縮されたデータの中継するという方法で処理されます。

圧縮は、Oracle DatabaseクライアントがOracle Database 12cリリース以前で圧縮をサポートできない場合でも、Oracle Connection Managerとサーバーとの間でサポートされます。

- 最大2 MBのSDUがサポートされているため、Oracle DatabaseクライアントとOracle Databaseサーバーは、Oracle Connection Managerによって接続を確立するときに、より高いSDUをネゴシエートできます。
- このリリースから、登録に対する有効ノード・チェックがOracle Connection Managerのデフォルトで有効化されます。デフォルトでは、Oracle Connection Managerに登録できるデータベースは、ローカル・ノード上にあるデータベースにかぎられます。リモート・ノード上のデータベースは、パラメータのREGISTRATION_INVITED_NODESとREGISTRATION_EXCLUDED_NODESを使用することで、Oracle Connection Managerへの登録を許可または拒否できます。

関連項目

- [『Oracle Database Net Servicesリファレンス・ガイド』](#)

親トピック: [Oracle Connection Managerの構成](#)

11 共有サーバー・アーキテクチャの構成

クライアントの負荷によりメモリーやその他のシステム・リソースに負荷がかかりすぎる場合、データベース管理者は、共有サーバー・リソースを起動してこの負荷の問題を解消できます。[共有サーバー・アーキテクチャ](#)によって、データベース・サーバーでは、多数のクライアント・プロセスで非常に少数のサーバー・プロセスを共有できるため、サポート可能なユーザー数が増大します。共有サーバー・アーキテクチャにより、多数のクライアント・プロセスが[ディスパッチャ](#)に接続されます。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ります。サーバー・プロセスの共有プールの中のアイドル状態の共有サーバー・プロセスは、共通キューから要求を取り出します。つまり、サーバー・プロセスの小規模プールによる多数のクライアントの処理が可能になります。これは、システムが過負荷の場合やシステムのメモリーが限られている場合に役に立ちます。

- [ディスパッチャについて](#)
- [セッションの多重化の有効化](#)
- [共有サーバーと専用サーバーの両方を使用する環境でのクライアントの構成](#)

関連項目:

共有サーバー構成の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。

親トピック: [Oracle Net Servicesの構成と管理](#)

11.1 ディスパッチャについて

ディスパッチャ、バーチャル・サーキットおよび共有サーバーの共有メモリー・リソースをあらかじめ設定することによって、実行時に共有サーバーを使用可能にします。データベース管理者は、SQL ALTER SYSTEM文を使用することによって、ディスパッチャと共有サーバーを起動でき、インスタンスを再起動する必要がありません。共有サーバー・モードをオンにすると、DISPATCHERSパラメータが設定されていない場合でも、ディスパッチャはTCP/IPプロトコルで自動的に起動されます。共有サーバーの使用方法は、データベース初期化パラメータ・ファイルの次のDISPATCHERSパラメータの設定と同じです。

```
DISPATCHERS="(PROTOCOL=tcp)"
```

次の状態の場合に、DISPATCHERSパラメータを直接構成します。

- TCP/IP以外のプロトコルを構成する必要がある場合。
- 多重化など、1つ以上のオプションのディスパッチャ属性を構成する必要がある場合

DISPATCHERSパラメータの次の属性を指定できます。PROTOCOL属性は必須であり、その他はオプションです。ADDRESS属性は、ファイアウォールを使用するときなど、ポート番号を指定する場合に使用されます。

- ADDRESS
- CONNECTIONS
- DESCRIPTION
- DISPATCHERS
- LISTENER
- MULTIPLEX

- PROTOCOL(必須)
- SERVICE
- SESSIONS

ディスパッチャの構成は、SQL文ALTER SYSTEMを使用して変更します。パラメータの設定後に、インスタンスを再起動する必要はありません。

- [ディスパッチャ単位のグループ化](#)
- [ディスパッチャの監視](#)

関連項目:

ディスパッチャの構成の詳細は、『[Oracle Database管理者ガイド](#)』を参照してください。

親トピック: [共有サーバー・アーキテクチャの構成](#)

11.1.1 ディスパッチャ単位のグループ化

Oracle Databaseは複数のサービス名で表すことができます。このため、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・ファイルには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.example.comをリクエストするクライアントのリクエストを処理しますもう一方のディスパッチャは、adminsales.us.example.comをリクエストしているクライアントのリクエストのみを処理します。

```
SERVICE_NAMES=sales.us.example.com
INSTANCE_NAME=sales
DISPATCHERS="(PROTOCOL=tcp)"
DISPATCHERS="(PROTOCOL=tcp) (SERVICE=adminsales.us.example.com)"
```

ノート:



プラグブル・データベースのサービスは、手動で開始する必要があります。

親トピック: [ディスパッチャについて](#)

11.1.2 ディスパッチャの監視

構成を確認し、ディスパッチャを監視するには、次のビューを使用します。

- V\$QUEUE: 共有サーバー・メッセージ・キューの情報を示します。このビューを利用できるのは、SYSユーザー、およびSYSTEMなどのSELECT ANY TABLEシステム権限を持つユーザーのみです。
- V\$DISPATCHER: 名前、ネットワーク・アドレス、ステータス、各種利用状況統計および索引番号など、ディスパッチャ・プロセスに関する情報を提供します。
- V\$DISPATCHER_CONFIG: ディスパッチャに関する構成情報を提供します。

- V\$DISPATCHER_RATE: ディスパッチャ・プロセスの速度統計を提供します。

関連項目:

- これらのビューの詳細は、[『Oracle Databaseパフォーマンス・チューニング・ガイド』](#)および[『Oracle Databaseリファレンス』](#)を参照してください。
- 共有サーバー構成の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。
- DISPATCHERSパラメータ構成およびサポートされる属性の詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。
- ALTER SYSTEM文の詳細は、[『Oracle Database SQLリファレンス』](#)を参照してください。

親トピック: [ディスパッチャについて](#)

11.2 セッション多重化の有効化

Oracle Connection Managerで使用可能なセッションの多重化によって、複数のクライアント・セッションを単一のプロトコル接続を経由して集中化できます。たとえば、Oracle Connection Managerを経由した1つの接続で、複数のクライアント・プロセスを1つのディスパッチャに接続できます。

Oracle Connection Managerでは、共有接続によってユーザーによるディスパッチャへの通信を可能にします。ある特定の時間には、Connection Managerプロセス経由でディスパッチャにリンクしている他のクライアント・プロセスがアイドル状態の間に、ユーザーは接続を必要とすることがあります。ディスパッチャ・プロセスが最大限に使用されるため、セッションの多重化は有効です。

また、セッションの多重化はディスパッチャ間のデータベース・リンク・セッションにも役立ちます。各ディスパッチャのセッション数の制限は、オペレーティング・システムによって異なります。

セッションの多重化を有効にするには、DISPATCHERSパラメータのMULTIPLEX属性をonまたはそれと同等の値に設定します。

```
DISPATCHERS="(PROTOCOL=tcp) (MULTIPLEX=on)"
```

関連項目:

構成の詳細は、[『Oracle Connection Managerのセッションの多重化の有効化』](#)を参照してください

親トピック: [共有サーバー・アーキテクチャの構成](#)

11.3 共有サーバーと専用サーバーの両方を使用する環境でのクライアントの構成

共有サーバーがサーバー側で構成されていて、ディスパッチャが登録されていない場合にクライアント接続要求が到着すると、その要求は専用サーバー・プロセスによって処理されます。特定のクライアントが常にディスパッチャを使用する必要がある場合は、接続記述子のCONNECT_DATAセクションに(SERVER=shared)を構成します。たとえば:

```
sales=
(DESCRIPTION=
 (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521))
 (CONNECT_DATA=
 (SERVICE_NAME=sales.us.example.com)
 (SERVER=shared)))
```

(SERVER=shared) 属性が構成されているときにディスパッチャを利用できない場合、クライアントの接続要求は拒否され、クライアントにメッセージが送信されます。

データベースが共有サーバー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、そのクライアントが専用サーバーを使用するように構成できます。それには次の方法があります。

- CONNECT_DATAセクションの接続記述子に (SERVER=dedicated) と設定することによって、ネットワーク・サービス名を構成できます。たとえば：

```
sales=
(DESCRIPTION=
 (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521))
 (CONNECT_DATA=
 (SERVICE_NAME=sales.us.example.com)
 (SERVER=dedicated)))
```

- クライアント・プロファイル・ファイル sqlnet.ora 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。



ノート：

USE_DEDICATED_SERVER が ON に設定されると、接続記述子の既存の (SERVER=value) エントリが (SERVER=dedicated) で上書きされます。

関連項目：

- SERVERパラメータの設定方法については、[「拡張接続データ・パラメータについて」](#)を参照してください
- USE_DEDICATED_SERVERパラメータの設定方法については、[「プロセスへの接続要求のルーティング」](#)を参照してください
- [『Oracle Call Interfaceプログラマーズ・ガイド』](#)および[『Oracle Database管理者ガイド』](#)を参照してください。

親トピック： [共有サーバー・アーキテクチャの構成](#)

12 プロファイルの構成

プロファイルのクライアント構成パラメータおよびサーバー構成パラメータの構成方法を学習します。プロファイルとは、クライアントまたはデータベース・サーバー上でのOracle Net機能の有効化および構成の優先順位を指定するパラメータの集合です。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います。

- [プロファイル構成の概要](#)
- [インストール中のプロファイルの構成](#)
- [名前解決のクライアント属性の理解](#)
- [データベース・アクセス制御の設定](#)
- [プロファイルの詳細情報について](#)
- [外部ネーミング・メソッドの構成](#)
- [Oracleネットワーク・セキュリティの構成](#)

Oracleネットワーク・セキュリティ機能を使用すると、データの暗号化、整合性チェック、強化された認証およびシングル・サインオンが可能になります。この機能によって、LDAP準拠ディレクトリ・サーバーにおける一元化されたユーザー管理および認証ベースのシングル・サインオンも可能になります。この機能は、Transport Layer Security (TLS)プロトコルに依存しています。

親トピック: [Oracle Net Servicesの構成と管理](#)

12.1 プロファイル構成の概要

プロファイルを使用すると、次の操作を実行できます。

- クライアントのドメインを指定して未修飾名に追加
- ネーミング・メソッドの優先順位の指定
- ログイン機能およびトレース機能の有効化
- 特定のプロセスを通じてのルート接続
- 外部プロシージャのパラメータの構成
- Oracle Advanced Securityの構成
- プロトコル固有パラメータを使用したデータベースへのアクセスの制限

親トピック: [プロファイルの構成](#)

12.2 インストール中のプロファイルの構成

クライアントおよびサーバーにソフトウェアがインストールされると、Oracle Universal Installerは[Oracle Net Configuration Assistant](#)を起動します。Oracle Net Configuration Assistantは、コンピュータが[接続記述子](#)に対する[接続識別子](#)の解決に使用するネーミング・メソッドの順序を構成します。

インストール時にOracle Net Configuration Assistantを使用して構成を行うと、sqlnet.oraファイルに次のようなエントリが作成されます。

```
NAMES.DIRECTORY_PATH=(ezconnect, tnsnames)
```

NAMES.DIRECTORY_PATHパラメータで、接続識別子の解決に使用するネーミング・メソッドの優先順位を指定します。インストールされた構成が適切でない場合、[Oracle Net Manager](#)を使用してsqlnet.oraの構成を変更します。

親トピック: [プロファイルの構成](#)

12.3 名前解決のクライアント属性の理解

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・ドメインについて](#)
- [ネーミング・メソッドの優先順位](#)
- [プロセスへの接続要求のルーティング](#)

親トピック: [プロファイルの構成](#)

12.3.1 クライアントのデフォルト・ドメインについて

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。このパラメータは、ローカル・ネーミング・メソッドと外部ネーミング・メソッドで使用できます。

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ワーク・サービス名に自動的に追加され、tnsnames.oraファイルに格納されているネットワーク・サービス名と比較されます。

たとえば、クライアントのtnsnames.oraファイルがsales.us.example.comのネットワーク・サービス名を含み、デフォルト・ドメインがus.example.comの場合、ユーザーは次の接続文字列を入力できます。

```
CONNECT scott@sales
Enter password: password
```

前の例では、salesはsales.us.example.comとして検索されます。

CONNECT scott@sales.us.example.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ファイル内のネットワーク・サービス名がドメインで修飾されていないと、NAMES.DEFAULT_DOMAINパラメータが設定されている場合は、ネットワーク・サービス名の最後にピリオド(.)を付けて入力します。たとえば、ドメインがus.example.comに設定され、クライアントtnsnames.oraファイルにsales2のネットワーク・サービス名がある場合は、次の接続文字列を入力します。

```
CONNECT scott@sales2.
Enter password: password
```

前の例では、クライアントはsales2.us.example.comではなくsales2に接続します。

- [デフォルト・ドメインの指定](#)

親トピック: [名前解決のクライアント属性の理解](#)

12.3.1.1 デフォルト・ドメインの指定

次の手順では、デフォルト・ドメインの指定方法を説明します。

1. Oracle Net Managerを起動します。

関連項目:

[「Oracle Net Managerを使用したOracle Net Servicesの構成」](#)

2. ナビゲータ・ペインで、「ローカル」メニューから「プロファイル」を選択します。
3. 右ペインのリストから、「ネーミング」を選択します。
4. 「メソッド」タブをクリックします。
5. 「デフォルト・ドメイン」フィールドにドメインを入力します。
6. 「ファイル」メニューから「ネットワーク構成の保存」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=us.example.com
```

親トピック: [クライアントのデフォルト・ドメインについて](#)

12.3.2 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、[「ネーミング・メソッドの構成」](#)の説明に従って、その優先順位付けを行う必要があります。接続識別子を解決するためのネーミング・メソッドは、リストに出現する順序で試みられます。リストの先頭のネーミング・メソッドが接続識別子を解決できなかった場合、リストの2番目のメソッドが使用されます。以降、同様です。

次の手順では、ネーミング・メソッドの順序の指定方法を説明します。

1. Oracle Net Managerを起動します。

関連項目:

[「Oracle Net Managerを使用したOracle Net Servicesの構成」](#)

2. ナビゲータ・ペインで、「ローカル」メニューから「プロファイル」を選択します。
3. 右ペインのリストから、「ネーミング」を選択します。
4. 「メソッド」タブをクリックします。

[表12-1](#)では、「メソッド」タブにリストされているネーミング・メソッドの値を説明します。

表12-1 ネーミング・メソッドの値

ネーミング・メソッドの値	説明
TNSNAMES	ネットワーク・サービス名 を、クライアントの tnsnames.ora ファイルで解決します。

ネーミング・メソッドの値	説明
	関連項目: 「ローカル・ネーミング・メソッドの構成」
LDAP	データベース・サービス名、ネットワーク・サービス名または ネットワーク・サービス別名 を、 ディレクトリ・サーバー で解決します。 関連項目: 「ディレクトリ・ネーミング・メソッドの構成」
EZCONNECT	クライアントが、ホスト名、ポート名(オプション)およびサービス名(オプション)で構成されるTCP/IP 接続識別子を使用できるようにする場合、または既存の名前解決サービスや集中管理された一連の/etc/hosts ファイルを使用してホスト名の別名を解決できるようにします。 関連項目: 「簡易接続ネーミング・メソッドの理解」
NIS	サービス情報を、既存の Network Information Service(NIS)で解決します。

5. 「使用可能なメソッド」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。

選択されたネーミング・メソッドが「選択メソッド」リストに移動します。

6. Oracle Netによるネットワーク・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。「選択メソッド」リスト内でネーミング・メソッドを選択し、「上へ」または「下へ」をクリックして選択項目をリスト内で移動します。
7. 「ファイル」メニューから「ネットワーク構成の保存」を選択します。

sqlnet.oraファイルは、次のようなNAMES.DIRECTORY_PATHパラメータで更新されます。

```
NAMES.DIRECTORY_PATH=(ldap, tnsnames)
```

親トピック: [名前解決のクライアント属性の理解](#)

12.3.3 プロセスへの接続要求のルーティング

クライアントおよびサーバーは、接続要求が特定のプロセスに送られるように構成できます。次の手順では、接続要求をプロセスにルーティングする方法を説明します。

1. Oracle Net Managerを起動します。

関連項目:

[「Oracle Net Managerを使用したOracle Net Servicesの構成」](#)

2. ナビゲータ・ペインで、「ローカル」メニューから「プロファイル」を選択します。
3. 右ペインのリストから、「一般」を選択します。

4. 「ルーティング」タブをクリックします。
5. 接続をルーティングするいずれかのオプションを選択します。

ノート:

すべての接続が特定のサーバーを使用するように構成するには、Oracle Net Manager で「常に専用サーバーを使用」オプションを選択します。これにより、sqlnet.ora ファイルの USE_DEDICATED_SERVER パラメータが設定され、リスナーは、クライアントからのすべてのネットワークセッションに専用サーバーを起動するようになります。共有サーバーが構成されている場合でも、結果的に専用サーバーの接続が使用されます。

6. 「ファイル」メニューから「ネットワーク構成の保存」を選択します。

関連項目:

フィールドとオプションの説明は、[表12-3](#)を参照してください

親トピック: [名前解決のクライアント属性の理解](#)

12.4 データベース・アクセス制御の設定

あるクライアントへのアクセスは許可し、他のクライアントへのアクセスは制限するようにsqlnet.oraファイルを構成できます。[表12-2](#)では、利用可能な設定を説明します。

表12-2 sqlnet.oraのアクセス制御の設定

Oracle Net Managerのフィールド / オプション		
	sqlnet.oraファイルのパラメータ	説明
TCP/IP クライアント・アクセス権のチェック	TCP.VALIDNODE_CHECKING	データベースへのアクセスを選別するかどうかを指定します。 このフィールドが選択された場合、Oracle Net Manager は、パラメータ TCP.EXCLUDED_NODES および TCP.INVITED_NODES をチェックして、データベースへのアクセスを許可するクライアントを特定します。このフィールドが選択解除された場合、Oracle Net Manager は、クライアントを選別しません。
アクセスを許可されないクライアント	TCP.EXCLUDED_NODES	TCP/IP プロトコルを使用するデータベースへのアクセスで、許可を与えないクライアントを指定します。
アクセスを許可されたクライアント	TCP.INVITED_NODES	TCP/IP プロトコルを使用するデータベースへのアクセスで、許

Oracle Net

Managerのフィールド

/オプション	sqlnet.oraファイルのパラメータ	説明
--------	----------------------	----

クライアント		可を与えるクライアントを指定します。
--------	--	--------------------

TCP.INVITED_NODESパラメータにリスナー・ノードが含まれないと、リスナー制御ユーティリティはリスナーに接続できません。その場合、開始、停止および管理コマンドをリスナーで実行できなくなります。

TCP.INVITED_NODESパラメータまたはTCP.EXCLUDED_NODESパラメータに無効なホスト名またはIPアドレスがリストされている場合、リスナー制御ユーティリティはリスナーに接続できません。

- [データベース・アクセス制御の構成](#)

親トピック: [プロファイルの構成](#)

12.4.1 データベース・アクセス制御の構成

次の手順では、データベース・アクセス制御の構成方法を説明します。

1. Oracle Net Managerを起動します。

関連項目:

[「Oracle Net Managerを使用したOracle Net Servicesの構成」](#)

2. ナビゲータ・ペインで、「ローカル」メニューから「プロファイル」を選択します。
3. 右ペインのリストから、「一般」を選択します。
4. 「アクセス権」タブをクリックします。
5. 「TCP/IPクライアント・アクセス権のチェック」オプションを選択します。
6. 「アクセスを許可されるクライアント」フィールドおよび「アクセスを許可されないクライアント」フィールドで、ホスト名またはIPアドレス(許可に含めるクライアント、あるいは含めないクライアント)を入力します(入力ではカンマを使用して、同一行に配置されたエントリを区切ります)。

親トピック: [データベース・アクセス制御の設定](#)

12.5 プロファイルの詳細情報について

[表12-3](#)では、sqlnet.oraファイルの詳細設定を説明します。

表12-3 sqlnet.oraの詳細設定

Oracle Net

Managerのフィールド

/オプション	sqlnet.oraファイルのパラメータ	説明
--------	----------------------	----

Oracle Net**Managerのフィールド sqlnet.oraファイルのパラ****/オプション****メータ****説明**

送信操作タイムアウト

SQLNET.SEND_TIMEO
UT

データベース・クライアントまたはサーバーが接続の確立後にピアからのデータを待機する時間をミリ秒単位、秒単位または分単位で指定します。

このパラメータをクライアントに設定すると、サーバー・プロセスの異常終了やサーバーのビジー状態が原因で、受信操作が無期限または長時間待機したまま放置されることがなくなります。指定した時間内にクライアントが応答データを受信しない場合、ORA-12535: TNS: 操作はタイムアウトしました。および ORA-12609: TNS: 受信タイムアウトが発生しましたというメッセージが sqlnet. log ファイルにロギングされます。

また、このパラメータをサーバー側で設定すると、接続の確立後にサーバーがクライアント・データを待機する時間をミリ秒単位、秒単位または分単位で指定できます。指定した時間内にクライアントがデータを何も送信しない場合は、ORA-12535:ORA-12535: TNS: 操作はタイムアウトしました。および ORA-12609: TNS: 受信タイムアウトが発生しましたというメッセージがデータベース・サーバーによって sqlnet. log ファイルにロギングされます。このパラメータを使用しない場合、データベース・サーバーは、ダウンしているか、または障害が発生している可能性のあるクライアントから、データを待ち続けることがあります。低い値に設定すると、サーバーは多くの場合クライアントからの入力をブロックして、このようなタイムアウトを頻繁に発生させます。

受信操作タイムアウト

SQLNET.RECV_TIMEO
UT

接続が確立された後、データベース・サーバーがクライアントからのデータを待機する時間(秒単位)を指定します。クライアントは指定された時間内にデータを送信する必要があります。

クライアントが稀にシャットダウンしたり、異常終了する環境では、このパラメータを設定することをお勧めします。指定した時間内にクライアントがデータを何も送信しない場合、データベース・サーバーは、ORA-12535: TNS:operation timed out および ORA-12609: TNS: Receive timeout occurredというエラーを sqlnet. log ファイルにロギングします。

このパラメータを使用しない場合、データベース・サーバーは、ダウンしているか、または障害が発生している可能性のあるクライアントから、データを待ち続けます。

Oracle Net**Managerのフィールド sqlnet.oraファイルのパラ****/オプション****メータ****説明**

また、この設定をクライアント側で使用し、接続の確立後にクライアントがデータベース・サーバーからの応答データを待機する秒数を指定することもできます。このパラメータを指定しない場合、クライアントは要求で飽和状態になっているデータベース・サーバーからの応答を長時間待機することがあります。

接続タイムアウト

SQLNET.INBOUND_CONNECT_TIMEOUT

クライアントがデータベース・サーバーに接続して必要な認証情報を提供する時間を、秒単位で指定します。

合計送信バッファ・サイズ

SEND_BUF_SIZE

セッションの送信操作に適用するバッファ・スペース制限を指定します。

合計受信バッファ・サイズ

RECV_BUF_SIZE

セッションの受信操作に適用するバッファ・スペース制限を指定します。

TNS タイムアウト値

SQLNET.EXPIRE_TIME

クライアント/サーバーの接続がアクティブであることを確認するプローブの送信時間間隔(分単位)を指定します。0 より大きい値を設定すると、クライアントが異常終了したときに接続が無期限に開かれたままにならないようにします。終了済接続や使用されなくなった接続を検出したプローブは、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。このオプションは、通常は一度に複数の接続を処理するデータベース・サーバーに対して設定します。

終了済接続の検出機能を使用する上での制約は、次のとおりです。

- この機能は Bequeathed 接続では機能しません。
- 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発生させます。
- 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があります。

Oracle Net

Managerのフィールド sqlnet.oraファイルのパラ

/オプション

メータ

説明

クライアント登録 ID	SQLNET.CLIENT_REGISTRATION	クライアントに一意の識別子を指定します。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。
-------------	----------------------------	-----------------------------------------------------------------------------

ログオン認証プロトコルのバージョン	SQLNET.ALLOWED_LOGON_VERSION_CLIENT SQLNET.ALLOWED_LOGON_VERSION_SERVER	Oracle Database インスタンスに接続する際に使用可能な最小の認証プロトコルを定義します。パラメータ名の VERSION は、Oracle Database のリリースではなく、認証プロトコルのバージョンを指します。クライアントまたはサーバーがパートナーにより指定された最小リリースを満たしていない場合、認証に失敗して ORA-28040 エラーが戻されます。
-------------------	--------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

サポートされる値は次のとおりです。

- 12a: Oracle Database 12c の認証プロトコルの場合 (最も強力な保護)
- 12: クリティカル・パッチ更新 CPUOct2012、およびより新しい Oracle Database 11g の認証プロトコルの場合 (推奨)
- 11: Oracle Database 11g の認証プロトコルの場合
10: Oracle Database 10g の認証プロトコルの場合
- 9: Oracle9i Database の認証プロトコルの場合
- 8: Oracle8i Database の認証プロトコルの場合

デフォルト値は、12 または 12a です。設定値 11 または 12 は、次の意味を持つことに注意してください。

- SEC_CASE_SENSITIVE_LOGON=FALSE の設定は使用しないでください。FALSE に設定されている場合、ユーザー・アカウントおよびセキュア・ロールは使用できなくなります。SEC_CASE_SENSITIVE_LOGON 初期化パラメータでは、パスワードの大/小文字の区別を有効化または無効化します。
 - Oracle Database 11g から導入されたパスワード保護を利用する場合、ユーザーはパスワードを変更する必要があります。
-

Oracle Net

Managerのフィールド sqlnet.oraファイルのパラ

/オプション

メータ

説明

- Oracle Database 10g より前の OCI クライアントの各リリースおよびすべてのバージョンの JDBC Thin クライアントでは、パスワードベース認証を使用して Oracle Database に対して認証することができません。

Oracle Database 12c サーバーに、前のリリースの Oracle Database を実行しているサーバーへのデータベース・リンクが含まれている場合は、前のリリースを実行しているサーバーの SQLNET.ALLOWED_LOGON_VERSION 設定と一致するように、Oracle Database 12c サーバーの SQLNET.ALLOWED_LOGON_VERSION_CLIENT パラメータの値を設定します。

UNIX シグナル処理をオフにする BEQUEATH_DETACH

UNIX シグナル処理をオンまたはオフにします。

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、それは終了済プロセスになります。シグナル・ハンドラは、これらの終了済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 初期化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。

アウト・オブ・バンド・ブ
レークの使用禁止 DISABLE_OOB

バンド外ブレイクをオンまたはオフにします。

選択を解除するか off にすると、Oracle Net は基礎となるプロトコルによって提供される緊急データ要求を使用してブレイク・メッセージを送受信できます。これは、一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。

選択するか on にすると、基礎となるプロトコルの緊急データ要求によるブレイク・メッセージの送受信は使用できなくなります。

アウト・オブ・バンド自動
ブレイクの使用禁止オプ
ション DISABLE_OOB_AUTO

接続時にクライアントのサーバー・パスに対する TCP OOB (アウト・オブ・バンド)のサポートをチェックします。サーバー・パスが OOB をサポートしている場合にのみ、クライアントは OOB を有効にします。

- [Oracle Net Servicesを使用したsqlnet.oraファイルでの拡張機能の設定](#)

関連項目:

- SQLNET.INBOUND_CONNECT_TIMEOUT設定の構成方法の詳細は、[「権限のないユーザーによるリソース使用の制限」](#)を参照してください
- SEND_BUF_SIZEおよびRECV_BUF_SIZE設定の構成方法の詳細は、[「I/Oバッファ・スペースの構成」](#)を参照してください
- SQLNET.ALLOWED_LOGON_VERSION_CLIENTおよびSQLNET.ALLOWED_LOGON_VERSION_SERVERの設定の詳細は、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください。
- SQLNET.ALLOWED_LOGON_VERSION_CLIENTおよびSQLNET.ALLOWED_LOGON_VERSION_SERVERの設定の詳細は、[『Oracle Database Net Servicesリファレンス』](#)を参照してください。
- プロトコルが緊急データ要求をサポートするかどうかを確認するには、Oracleのオペレーティング・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。

親トピック: [プロファイルの構成](#)

12.5.1 Oracle Net Servicesを使用したsqlnet.oraファイルでの拡張機能の設定

次の手順では、sqlnet.oraファイルで拡張機能を設定する方法について説明します。

1. Oracle Net Managerを起動します。

関連項目:

[「Oracle Net Managerを使用したOracle Net Servicesの構成」](#)

2. ナビゲータ・ペインで、「ローカル」メニューから「プロファイル」を選択します。
3. 右ペインのリストから、「一般」を選択します。
4. 「詳細」タブをクリックします。
5. 設定するフィールドまたはオプションに値を入力します。
6. 「ファイル」メニューから「ネットワーク構成の保存」を選択します。

親トピック: [プロファイルの詳細情報について](#)

12.6 外部ネーミング・メソッドの構成

Network Information Service(NIS)の外部ネーミングに必要なクライアント・パラメータを構成するには、sqlnet.oraファイルを使用します。次の手順では、sqlnet.oraファイルでNISパラメータを構成する方法について説明します。

1. Oracle Net Managerを起動します。

関連項目:

[「Oracle Net Managerを使用したOracle Net Servicesの構成」](#)

2. ナビゲータ・ペインで、「ファイル」メニューから「プロファイル」を選択します。
3. 右ペインのリストから、「ネーミング」を選択します。
4. 「外部」タブをクリックします。
5. 「メタ・マップ」フィールドに、NAMES.NIS.META_MAPと入力します。
6. 「ファイル」メニューから「ネットワーク構成の保存」を選択します。

親トピック: [プロファイルの構成](#)

12.7 Oracleネットワーク・セキュリティの構成

Oracleネットワーク・セキュリティ機能を使用すると、データの暗号化、整合性チェック、強化された認証およびシングル・サインオンが可能になります。この機能によって、LDAP準拠ディレクトリ・サーバーにおける一元化されたユーザー管理および認証ベースのシングル・サインオンも可能になります。この機能は、Transport Layer Security (TLS)プロトコルに依存しています。

次の手順では、Oracleネットワーク・セキュリティの機能を使用するようにクライアントまたはサーバーを構成する方法について説明します。

1. Oracle Net Managerを起動します。
2. ナビゲータ・ペインで、「ローカル」メニューから「プロファイル」を選択します。
3. 右ペインのリストから、「ネットワーク・セキュリティ」を選択します。

「ネットワーク・セキュリティ」タブの各ページで、パラメータ・セットを個別に構成できます。タブ・ページは次のとおりです。

- 認証: KERBEROS5、RADIUSなどの使用可能な認証方法の構成用。
- その他のパラメータ: 認証サービスの構成用。
- 整合性: 整合性のタイプ、チェックサム・レベルおよび使用可能な方法の構成用。
- 暗号化: 暗号化のタイプと方法の構成用。
- TLS: TLSの使用の設定用

4. パラメータを適切に選択または編集します。
5. 「ファイル」メニューから「ネットワーク構成の保存」を選択します。

ノート:



詳細は、タブ・ページの「ヘルプ」ボタンまたは Oracle Net Manager オンライン・ヘルプのネットワーク・セキュリティのトピックを参照してください。ネットワーク・セキュリティ・トピックにアクセスするには、「ネットワーク・セキュリティ」を選択してから、「方法」オプションを選択します。

関連項目

- [Oracleネットワーク・セキュリティの構成](#)
- 『[Oracle Databaseセキュリティ・ガイド](#)』

親トピック: [プロフィールの構成](#)

13 Oracle Net Servicesの拡張機能の使用

拡張接続データ・パラメータ、ロード・バランシング、フェイルオーバーおよび非データベース・サービスへの接続など、Oracle Net Servicesの拡張機能を構成する方法について説明します。

- [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](#)
データベース・サービスには、複数のルートおよびプロトコル・アドレスからアクセスできます。プロトコル・アドレスのリストを設定して、使用するルートを構成します。オーダー・アドレスは、アドレス・パラメータを指定して構成します。
- [接続ロード・バランシングの理解](#)
接続ロード・バランシング機能を使用すると、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、接続のパフォーマンスが向上します。
- [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)は、データベース・インスタンスの障害発生時に、障害が発生した接続を別のリスナーに透過的に再接続するようOracle Netに指示します。これにより、元の接続に障害が発生したことを意識せずに、新しい接続を使用して作業を続行できます。
- [プライマリおよびセカンダリ・インスタンス構成のインスタンス・ロールの指定](#)
INSTANCE_ROLEパラメータは、接続記述子のCONNECT_DATAセクション用のオプションのパラメータです。これを使用すると、Oracle RAC構成のプライマリまたはセカンダリ・インスタンスへの接続を指定できます。
- [静的サービス登録の構成](#)
リスナーは、listener.oraファイルの静的な構成情報を使用する前に、データベースやインスタンスの動的なサービス情報を使用します。
- [サード・パーティのデータベース・サービスへの接続の構成](#)

親トピック: [Oracle Net Servicesの構成と管理](#)

13.1 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

データベース・サービスは、複数のルートおよびプロトコル・アドレスからアクセスできます。プロトコル・アドレスのリストを設定して、使用するルートを構成します。オーダー・アドレスは、アドレス・パラメータを指定して構成します。

- [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・パラメータについて](#)
- [拡張接続データ・パラメータについて](#)

親トピック: [Oracle Net Servicesの拡張機能の使用](#)

13.1.1 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・ルートまたはプロトコル・アドレスからアクセスできます。次の例では、sales.us.example.comで、sales1-serverまたはsales2-serverのいずれかのリスナーを使用して、sales.us.example.comに接続できます。

```
sales.us.example.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.example.com))
```

既存のネットワーク・サービス名やデータベース・サービスにネットワーク・プロトコル・アドレスを追加するには、次のいずれかの手順を使用します。

- [Oracle Enterprise Manager Cloud Controlを使用したネットワーク・プロトコルの追加](#)
- [Oracle Net Managerを使用したネットワーク・プロトコルの追加](#)

Oracle Enterprise Manager Cloud Controlを使用したネットワーク・プロトコルの追加

次の手順では、Oracle Enterprise Manager Cloud Controlを使用して既存のネットワーク・サービス名またはデータベース・サービスにネットワーク・プロトコルを追加する方法について説明します。

1. Oracle Enterprise Manager Cloud Controlの「ディレクトリ・ネーミング」または「ローカル・ネーミング」ページにアクセスします。
 - a. Oracle Enterprise Manager Cloud Controlの「Net Services管理」ページにアクセスします。

関連項目:

[「Oracle Enterprise Manager Cloud Controlを使用したOracle Net Servicesの構成」](#)

- b. 「管理」リストから「ローカル・ネーミング」または「ディレクトリ・ネーミング」を選択し、次にディレクトリ・サーバーまたはローカル構成ファイルの場所を選択します。
 - c. 「実行」をクリックします。

「ローカル・ネーミング」ページまたは「ディレクトリ・ネーミング」ページが表示されます。
2. ディレクトリ・サービス名またはネットワーク・サービス名を選択します。
 - a. 「ローカル・ネーミング」の場合、リストからネットワーク・サービスを選択し、「編集」をクリックします。
 - b. 「ディレクトリ・ネーミング」の場合、「簡易検索」セクションでネットワーク・サービス名の検索を実行し、「結果」リストからネットワーク・サービスまたはデータベース・サービスを選択して、「編集」をクリックします。
 3. アドレス・セクションで、「追加」をクリックします。

「アドレスの追加」ページが表示されます。
 4. 「プロトコル」リストから、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。
 5. 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、[『Oracle Database Net Servicesリファレンス』](#)を参照してください。

6. (オプション)「拡張パラメータ」セクションを使用して、このセッションにおける送受信操作に必要なI/Oバッファ・スペース制限を「合計送信バッファ・サイズ」および「合計受信バッファ・サイズ」フィールドに指定します。

関連項目:

バッファ・スペースの詳細は、[「I/Oバッファ・スペースの構成」](#)を参照してください

7. 「OK」をクリックします。

プロトコル・アドレスは、アドレス・セクションに追加されます。

8. アドレス情報を更新するには、「OK」をクリックします。

Oracle Net Managerを使用したネットワーク・プロトコルの追加

次の手順では、Oracle Net Managerを使用して既存のネットワーク・サービス名またはデータベース・サービスにネットワーク・プロトコルを追加する方法について説明します。

1. Oracle Net Managerを起動します。

関連項目:

[「Oracle Net Managerを使用したOracle Net Servicesの構成」](#)

2. ナビゲータ・ペインで、「ディレクトリ」または「ローカル」メニューから「サービス・ネーミング」を選択します。

3. ネットワーク・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. 「アドレスの構成」ボックスで、プラス(+)をクリックして、新規アドレスを追加します。

新規の「アドレス」タブが表示されます。

a. 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、[『Oracle Database Net Servicesリファレンス』](#)を参照してください。

b. (オプション)「アドレス」タブ内の「詳細」をクリックし、このセッションにおける送受信操作に必要なI/Oバッファ・スペース制限を「合計送信バッファ・サイズ」および「合計受信バッファ・サイズ」フィールドに指定します。

関連項目:

バッファ・スペースの詳細は、[「I/Oバッファ・スペースの構成」](#)を参照してください

c. 左矢印ボタンと右矢印ボタンを使用して、プロトコル・アドレスの順序を指定します。これにより、プロトコル・アドレス・リスト内のアドレスの順序が並べ替えられます。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。

5. 「ファイル」メニューから「ネットワーク構成の保存」を選択します。

親トピック: [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](#)

13.1.2 アドレス・リスト・パラメータについて

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、ランダムに選択したり、順番に試行するなど、アドレスの使用順序を指定します。[表13-1](#)は、複数のプロトコル・アドレスで使用されるパラメータを示しています。

表13-1 tnsnames.oraファイルのアドレス・リスト・パラメータ

パラメータ	説明
FAILOVER	<p>on に設定した場合、このパラメータは、接続時に最初のリスナーに障害が発生したら、別のリスナーにフェイルオーバーするよう Oracle Net に指示します。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 に設定した場合は、接続を試行するアドレスは 1 つのみです。</p> <p>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 on になっています。</p>
LOAD_BALANCE	<p>on に設定すると、このパラメータは、プロトコル・アドレスのリストから順不同に選択して、複数のリスナーに対する負荷を均衡化するよう Oracle Net に指示します。off に設定した場合、Oracle Net は接続が成功するまでアドレスを順番に試行します。</p> <p>複数の接続記述子 (DESCRIPTION_LIST) がある場合、クライアント・ロード・バランシングはデフォルトで on になっています。</p>
SOURCE_ROUTE	<p>on に設定すると、このパラメータは、各アドレスを提供された順序で、接続先に到達するまで使用するよう Oracle Net に指示します。特定のルート、すなわち特定のコンピュータを使用して接続先に到達するには、このパラメータが必要です。このパラメータは、Oracle Connection Manager への接続を使用可能にするために使用されます。</p>

ノート:

ソース・ルーティング (SOURCE_ROUTE) を接続時フェイルオーバー (FAILOVER) またはクライアント・ロード・バランシング (LOAD_BALANCE) と同じレベルで設定することはできません。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が、接続時フェイルオーバーとクライアント・ロード・バランシングではリストから 1 つのアドレスを選択します。

tnsnames.ora ファイル内の接続記述子に Oracle Connection Manager ホップのプロトコル・アドレスが少なくとも 2 つ含まれている場合、ホップ内の接続時フェイルオーバーとロード・バランシングのパラメータをファイル内に含めることができます。

[表13-2](#)では、アドレス・リスト・オプションについて説明しています。

表13-2 「アドレス・リスト・オプション」ダイアログ・ボックス

オプション	パラメータ設定
接続に成功するまで各アドレスを順番に試行	FAILOVER=on
接続に成功するまで各アドレスをランダムに試行	LOAD_BALANCE=on FAILOVER=on
ランダムに選択した 1 つのアドレスを試行	LOAD_BALANCE=on
各アドレスを、接続先に達するまで順番に使用します。	SOURCE_ROUTE=on
最初のアドレスのみを使用します	LOAD_BALANCE=off FAILOVER=off SOURCE_ROUTE=off

次の例は、クライアント・ロード・バランシング用に構成されたtnsnames.oraファイルです。

```
sales.us.example.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.example.com)))
```

次の例は、接続時フェイルオーバー用に構成されたtnsnames.oraファイルです。

```
sales.us.example.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.example.com)))
```

次の例は、Oracle Connection Managerおよびロード・バランシング用に構成されたtnsnames.oraファイルです。

```
sales.us.example.com=
  (DESCRIPTION=
    (SOURCE_ROUTE=ON)
    (ADDRESS=(PROTOCOL=tcp)(HOST=cman-pc1)(PORT=1630))
    (ADDRESS=
      (LOAD_BALANCE=ON)
      (ADDRESS=(PROTOCOL=tcp)(HOST=cman-pc2)(PORT=1521))
      (ADDRESS=(PROTOCOL=tcp)(HOST=cman-pc3)(PORT=1521)))
    (CONNECT_DATA=
```

```
(SERVICE_NAME=sales.us.example.com))
```

- [アドレス・リスト・パラメータの構成](#)

関連項目:

ソース・ルーティングに関するクライアントの構成については、[「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

親トピック: [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](#)

13.1.2.1 アドレス・リスト・パラメータの構成

次の手順では、アドレス・リスト・パラメータを構成する方法について説明します。

1. [「リスナー・プロトコル・アドレスのリスト作成」](#)の手順を実行してください。
2. Oracle Enterprise Manager Cloud ControlまたはOracle Net Managerを使用したアドレス・リスト・オプションの構成
 - Oracle Enterprise Manager Cloud Controlでは、「接続時フェイルオーバーとクライアント・ロード・バランシング」セクションの適切なオプションを選択します。
 - Oracle Net Managerで、「アドレスの構成」ボックスの「詳細」をクリックします。「アドレス・リスト・オプション」ダイアログ・ボックスが表示されます。適切なオプションを選択します。

親トピック: [アドレス・リスト・パラメータについて](#)

13.1.3 拡張接続データ・パラメータについて

Oracle Database 12c リリース2 (12.2)からは、sqlnet.oraファイルでデータ圧縮を設定できます。圧縮を設定するパラメータは、SQLNET.COMPRESSIONおよびSQLNET.COMPRESSION_LEVELSです。これらのパラメータをsqlnet.oraファイルに設定すると、sqlnet.oraファイルを使用するすべての接続に影響します(REDOおよびSecureFiles LOB (ラージ・オブジェクト)をストリーミングするOracle Data Guardを除く)。次の例に、圧縮を設定する方法を示します。

```
SQLNET.COMPRESSION = on
SQLNET.COMPRESSION_LEVELS = (low, high)
```

tnsnames.oraファイルの接続記述子のCONNECT_DATAセクションは、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.example.comというサービスを定義します。

```
sales.us.example.com=
  (DESCRIPTION=
    (ADDRESS= (PROTOCOL=tc) (HOST=sales-server) (PORT=1521))
    (CONNECT_DATA=
      (SERVICE_NAME=sales.us.example.com)))
```

サービス名の他にも、オプションで、[表13-3](#)に示されるパラメータを使用して、接続データ情報を構成できます。

表13-3 tnsnames.oraファイルの拡張接続データ設定

Oracle Enterprise Manager Cloud Control/Oracle

Net Managerのオ プション

tnsnames.oraファイル のパラメータ

説明

インスタンス名	INSTANCE_NAME	アクセスするデータベース・インスタンス。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。
セッション・データ・ユニット・サイズ(バイト)	SDU	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。 セッション・データ・ユニット(SDU) サイズを指定して、ネットワークで送信されるパケット関連のパフォーマンス特性を変更します。SDU サイズの上限は 2MB です。
異機種間サービスに使用	HS	Oracle Database サーバーで 異機種間サービス を使用してサード・パーティ・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。
Oracle Rdb データベース	RDB_DATABASE	Oracle Rdb データベースのファイル名。
サービスのタイプ	TYPE_OF_SERVICE	Oracle Rdb データベースで使用するサービスの種類。
グローバル・データベース名	GLOBAL_NAME	Oracle Rdb データベース識別子。

次の例では、データ・パケットの転送レートを設定しています。

```
sales.us.example.com=  
(DESCRIPTION=  
  (ADDRESS= (PROTOCOL=tcp) (HOST=sales-server) (PORT=1521))  
  (CONNECT_DATA=  
    (SERVICE_NAME=sales.us.example.com)  
    (SDU=8192)))
```

Oracle Enterprise Manager Cloud ControlまたはOracle Net Managerを使用して、ネットワーク・サービス名またはデータベース・サービスの拡張CONNECT_DATAパラメータを構成します。

Oracle Enterprise Manager Cloud Controlを使用した拡張接続記述子パラメータの構成

次の手順では、Oracle Enterprise Manager Cloud Controlを使用して拡張接続記述子パラメータを構成する方法について説明します。

1. 次の手順に従って、Oracle Enterprise Manager Cloud Controlの「ディレクトリ・ネーミング」または「ローカル・ネーミング」ページにアクセスします。
 - a. Oracle Enterprise Manager Cloud Controlの「Net Services管理」ページにアクセスします。

関連項目:

[「Oracle Enterprise Manager Cloud Controlを使用したOracle Net Servicesの構成」](#)

- b. 「管理」リストから「ローカル・ネーミング」または「ディレクトリ・ネーミング」を選択し、次にディレクトリ・サーバーまたはローカル構成ファイルの場所を選択します。
 - c. 「実行」をクリックします。
「ディレクトリ・ネーミング」または「ローカル・ネーミング」ページが表示されます。
2. ディレクトリ・サービス名またはネットワーク・サービス名を選択します。
 - a. 「ローカル・ネーミング」の場合、リストからネットワーク・サービスを選択し、「編集」をクリックします。
 - b. 「ディレクトリ・ネーミング」の場合、「簡易検索」セクションでネットワーク・サービス名を検索し、「結果」リストからネットワーク・サービスまたはデータベース・サービスを選択して、「編集」をクリックします。
 3. 「詳細」タブをクリックします。
 4. 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。
 5. 接続データ情報を更新するには、「OK」をクリックします。

Oracle Net Managerを使用した拡張接続記述子パラメータの構成

次の手順では、Oracle Net Managerを使用して拡張接続記述子パラメータを構成する方法について説明します。

1. Oracle Net Managerを起動します。

関連項目:

[「Oracle Net Managerを使用したOracle Net Servicesの構成」](#)

2. ナビゲータ・ペインで、「ディレクトリ」または「ローカル」メニューから「サービス・ネーミング」を選択します。
3. ネットワーク・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. 「サービスの識別」ボックスの「詳細」ボタンをクリックします。
「詳細サービス・オプション」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. 「ローカル」フォルダに対してこれらの変更を行う場合は、「ファイル」メニューから「ネットワーク構成の保存」を選択します。
ディレクトリ・フォルダへの変更は自動的に保存されます。

親トピック: [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](#)

13.2 接続ロード・バランシングの理解

接続ロード・バランシング機能を利用すると、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 Real Application Clusters (Oracle RAC)環境では、接続ロード・バランシングによって、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ことも可能です。

リスナー登録(LREG)プロセスはリモート・リスナーを登録できるため、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、それらのある場所にかかわらず常に認識できます。リスナーは、着信したクライアント要求の送信先となるインスタンスを、また共有サーバーが構成されている場合は送信先となるディスパッチャを、ロード情報に応じて判別します。

共有サーバー構成では、リスナーは次の順番でディスパッチャを選択します。

1. ロード量が最小のノード。
2. ロード量が最小のインスタンス。
3. そのインスタンスのロード量が最小のディスパッチャ。

専用サーバー構成では、リスナーは次の順番でインスタンスを選択します。

1. ロード量が最小のノード。
2. ロード量が最小のインスタンス。

このリリースで、HTTPプレゼンテーション用の複数ノードへのロード・バランシングが導入されました。リモート・リスナーが、HTTPリダイレクトを使用して、HTTPプレゼンテーション用の異なる複数のノード上にある複数のインスタンスに対してロード・バランシングを行えます。

Oracle RAC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も相互登録される必要があります。これを行うには、DISPATCHERSパラメータのLISTENER属性を使用します。

ノート:



接続ロード・バランシングで最適な結果を得るには、同じデータベース・サービスに属するインスタンスを、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。

- [共有サーバーの構成のための接続ロード・バランシングの例](#)
- [専用サーバーの構成のための接続ロード・バランシングの例](#)
- [クライアント接続のCOLOCATION_TAG](#)

COLOCATION_TAGパラメータは、TNS接続文字列のCONNECT_DATAパラメータで使用できる英数字の文字列です。

関連項目:

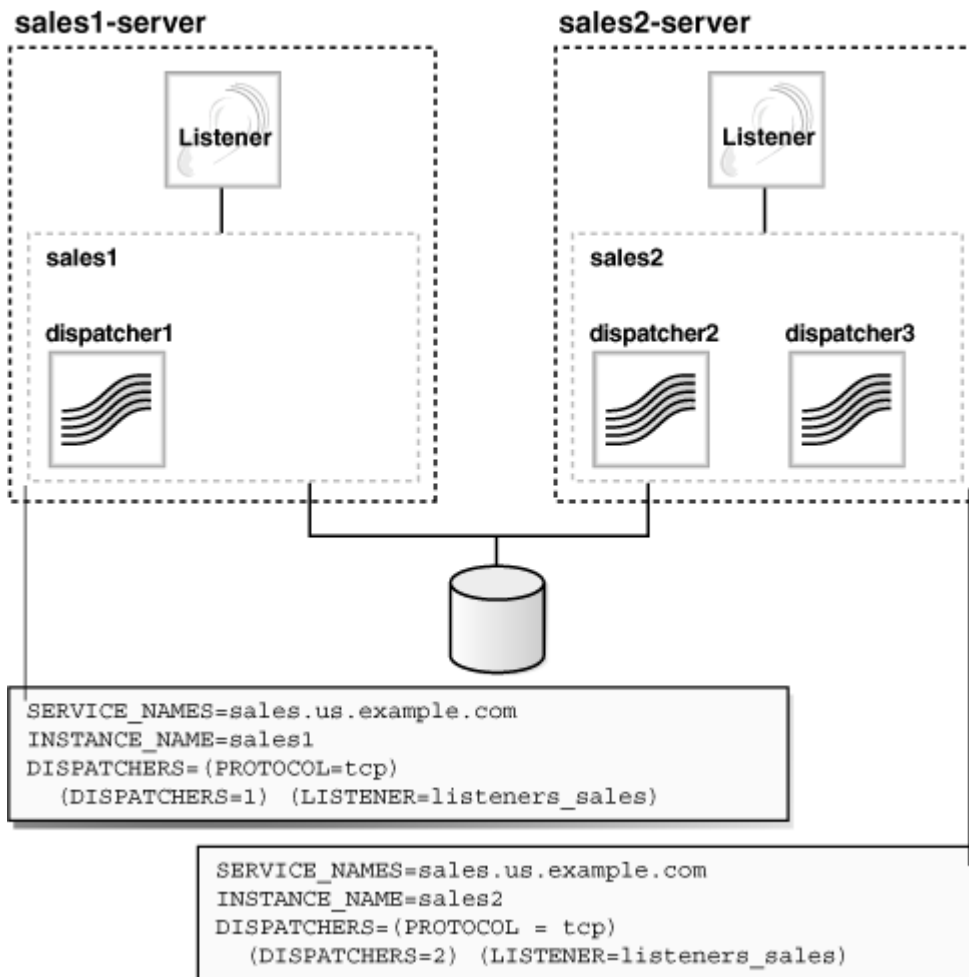
- 相互登録の詳細は、[「リモート・リスナーへの情報の登録」](#)を参照してください
- SERVICE_NAMESパラメータおよびINSTANCE_NAMEパラメータの詳細は、[『Oracle Database!リファレンス』](#)を参照してください。
- LISTENER属性の詳細は、[「共有サーバー・アーキテクチャの構成」](#)を参照してください
- グローバル・サービス管理の詳細は、[Oracle Database Global Data Services概要および管理ガイド](#)を参照してください。

親トピック: [Oracle Net Servicesの拡張機能の使用](#)

13.2.1 共有サーバーの構成のための接続ロード・バランシングの例

図13-1は、同じサービスsales.us.example.comの2つのインスタンスsales1およびsales2を持つ、Oracle RACの共有サーバー・データベースを示しています。インスタンスsales1およびsales2は、それぞれコンピュータsales1-serverおよびsales2-serverに常駐します。インスタンスsales1は1つのディスパッチャ、インスタンスsales2は2つのディスパッチャを持ちます。listenerという名前のリスナーは、それぞれノード1および2上で稼働しています。DISPATCHERSパラメータのlistener属性が構成され、両方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13-1 共有サーバーの構成のためのロード・バランシング環境



この例では、sales2-serverがロード量が最小のノード、sales2がロード量が最小のインスタンス、dispatcher2がロード量が最小のディスパッチャです。次のロード情報が登録されます。

- 各インスタンスの1分当たりのロード量平均は、sales1の場合600、sales2の場合400になります。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。
- 各インスタンスへの接続数は、sales1の場合200、sales2の場合300になります。
- 各インスタンスへのディスパッチャ接続数は、dispatcher1の場合200、dispatcher2の場合100、dispatcher3の場合200になります。
- sales1への接続数(200)は、その唯一のディスパッチャであるdispatcher1への接続数と同じです。
- 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。

(LISTENER=listeners_sales)のlisteners_salesの値は、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、次のように決定します。

```
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、[図13-2](#)に示す矢印の番号に対応しています。

1. LREGプロセスは、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時に動的に更新されます。
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プロトコル・アドレスが1つ成功するまで、各プロトコル・アドレスがランダムに試行されます。

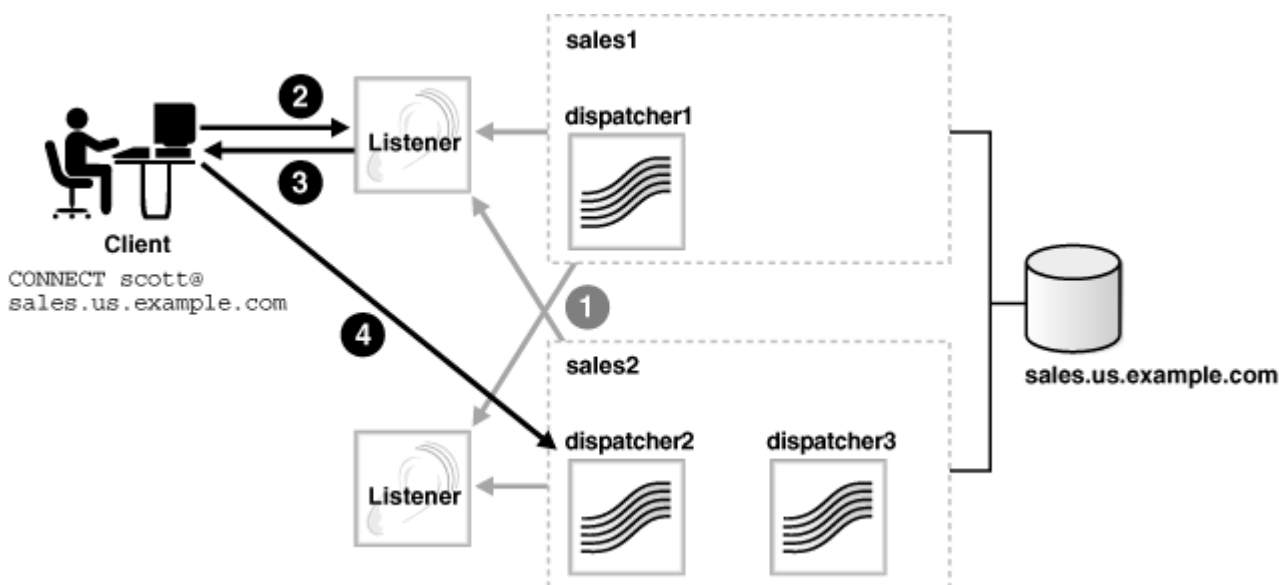
```
sales.us.example.com=  
(DESCRIPTION=  
  (LOAD_BALANCE=on)  
  (FAILOVER=on)  
  (ADDRESS=(PROTOCOL=tcp) (HOST=sales1-server) (PORT=1521))  
  (ADDRESS=(PROTOCOL=tcp) (HOST=sales2-server) (PORT=1521))  
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))
```

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13-2 共有サーバーの構成のためのロード・バランシングの例

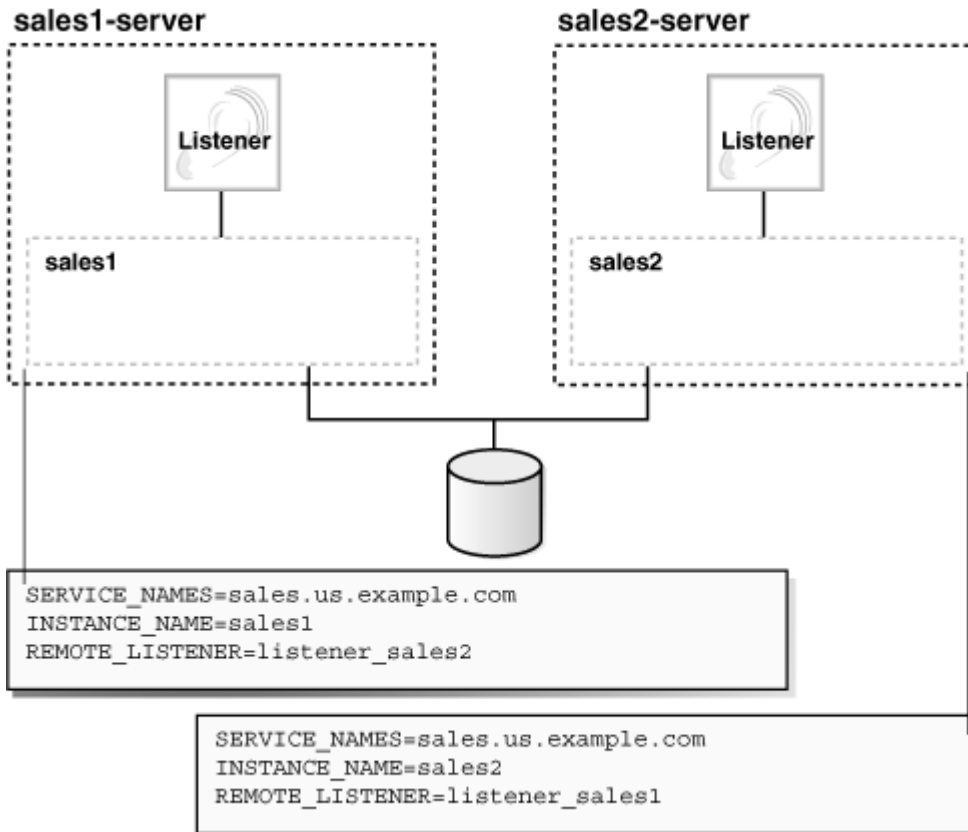


親トピック: [接続ロード・バランシングの理解](#)

13.2.2 専用サーバーの構成のための接続ロード・バランシングの例

図13-3は、同じサービスsales.us.example.comの2つのインスタンスsales1およびsales2を持つ、Oracle RACの専用サーバー・データベースを示しています。インスタンスsales1およびsales2は、それぞれコンピュータsales1-serverおよびsales2-serverに常駐します。listenerという名前のリスナーは、それぞれノード1および2上で稼働しています。REMOTE_LISTENER初期化パラメータが構成され、両方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13-3 専用サーバーの構成のためのロード・バランシング環境



この例では、次のロード情報が登録されます。

- sales1-serverの1分当たりのノードのロード量平均は450です。
- sales2-serverの1分当たりのノードのロード量平均は200です。
- sales1の接続数は200です。
- sales2の接続数は150です。

(REMOTE_LISTENER=listener_sales1)のlistener_sales1の値は、sales2-server上に存在するローカルのtnsnames.oraファイルによって、次のように決定します。

```
listener_sales1=
(DESCRIPTION=
 (ADDRESS= (PROTOCOL=tcp) (HOST=sales1-server) (PORT=1521)))
```

(REMOTE_LISTENER=listener_sales2)のlistener_sales2の値は、sales1-server上に存在するローカルのtnsnames.oraファイルによって、次のように決定します。

```
listener_sales2=
(DESCRIPTION=
```

```
(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、[図13-4](#)に示す矢印の番号に対応しています。

1. LREGプロセスは、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、インスタンスのロード時に動的に更新されます。

前述の情報では、sales2-serverがロード量が最小のノード、sales2がロード量が最小のインスタンスです。

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プロトコル・アドレスが1つ成功するまで、各プロトコル・アドレスがランダムに試行されます。

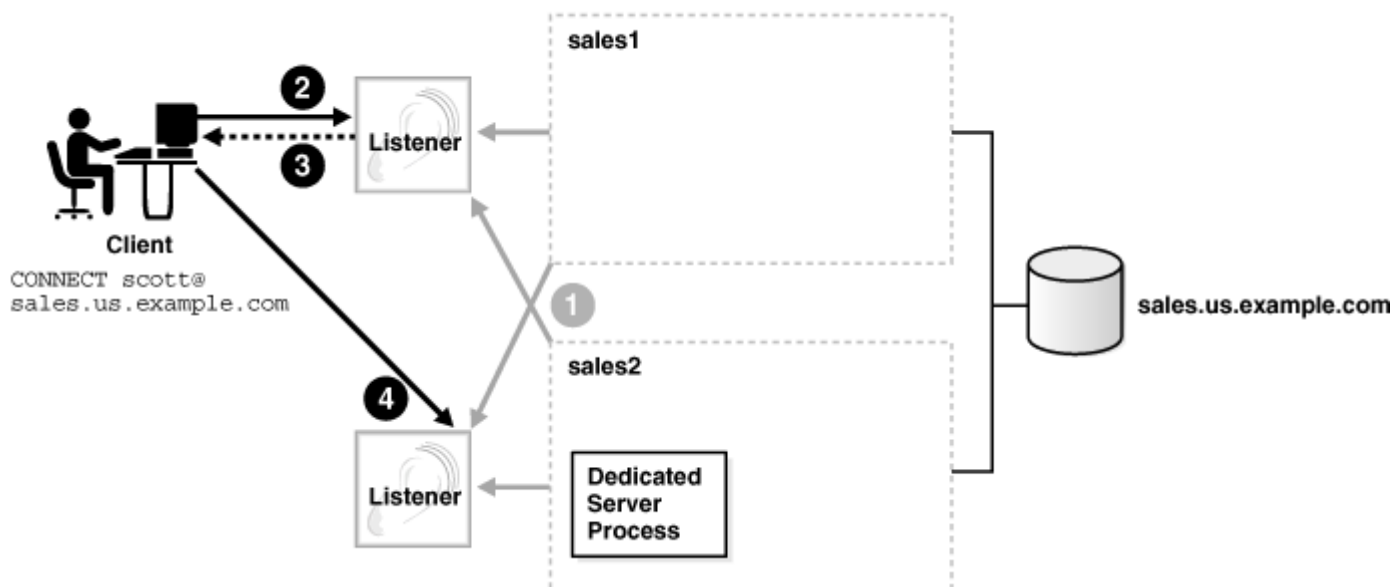
```
sales.us.example.com=  
(DESCRIPTION=  
  (ADDRESS_LIST=  
    (LOAD_BALANCE=on)  
    (FAILOVER=on)  
    (ADDRESS=(PROTOCOL=tcp) (HOST=sales1-server) (PORT=1521))  
    (ADDRESS=(PROTOCOL=tcp) (HOST=sales2-server) (PORT=1521))  
  )  
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))  
)
```

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. sales1-server上のリスナーが、クライアント接続要求をsales2-serverのリスナーにリダイレクトします。
4. クライアントがsales2-server上のリスナーに接続します。リスナーが専用サーバー・プロセスを開始し、専用サーバー・プロセスがリスナーから接続要求を継承します。

図13-4 専用サーバーの構成のためのロード・バランシングの例



親トピック: [接続ロード・バランシングの理解](#)

13.2.3 クライアント接続のCOLOCATION_TAG

COLOCATION_TAGパラメータは、TNS接続文字列のCONNECT_DATAパラメータで使用できる英数字の文字列です。

CONNECT_DATAパラメータでcolocation_tagを設定すると、ロード・バランシングが無視されます。リスナーは、同じcolocation_tagを持つすべての接続を同じデータベース・インスタンスに送信するように試行します。インスタンス選択アルゴリズムは、colocation_tagと、指定したサービスで使用可能なインスタンスのリストに基づきます。

たとえば、リスナーは、接続記述子でCOLOCATION_TAGがinteractiveに設定されている同じデータベース・インスタンスに、すべてのクライアントをルーティングしようとします。

```
sales.us.example.com=  
(DESCRIPTION=  
  (ADDRESS=(PROTOCOL=tcp) (HOST=sales-scan) (PORT=1521))  
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com) (COLOCATION_TAG=interactive)))
```

ノート:



インスタンスの最大ロードに達した、またはサービスに対して新しいインスタンスが追加または削除されたなど特定の
場合、同じ colocation_tag を持つクライアント接続の同じデータベース・インスタンスに対するコロケーションが一
貫していない可能性があります。

関連項目

- [COLOCATION_TAG](#)

親トピック: [接続ロード・バランシングの理解](#)

13.3 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)は、データベース・インスタンスの障害発生時に、障害が発生した接続を別のリスナーに透過的に再接続するようOracle Netに指示します。これにより、元の接続に障害が発生したことを意識せずに、新しい接続を使用して作業を続行できます。

TAFを使用する場合は、ネットワーク・サービス名の構成において接続記述子のCONNECT_DATAセクションにFAILOVER_MODEパラメータを手動で設定する必要があります。

ノート:



TAF およびアプリケーション・コンティニューイティのために、接続されたサービスでフェイルオーバーを構成することをお薦めします。これがクライアント側の設定を上書きします。

TAF機能と、接続文字列で構成する方法について学習します。

- [透過的アプリケーション・フェイルオーバーについて](#)

透過的アプリケーション・フェイルオーバーを使用すると、接続先の現在のインスタンスに障害が発生した場合、クライアントがデータベース・インスタンスに自動的に再接続できます。サーバーは通知を使用して、クライアント側でTAFコール

バックをトリガーします。

- [透過的アプリケーション・フェイルオーバーのリストア](#)

TAFは、アクティブなデータベース接続に関連するこれらの要素の一部またはすべてを自動的にリストアします。TAFによって接続をリカバリするには、他の要素をアプリケーション・コードに埋め込むことが必要な場合もあります。

- [FAILOVER_MODEパラメータについて](#)

FAILOVER_MODEパラメータは、これらの追加パラメータをサポートします。これらは、接続記述子のCONNECT_DATAセクションで指定できます。

- [透過的アプリケーション・フェイルオーバーの実装](#)

FAILOVER_MODEパラメータに応じて、いくつかの方法でTAFを実装できます。次の方法をお勧めします。

- [透過的アプリケーション・フェイルオーバーの検証](#)

「V\$SESSION」ビューのFAILOVER_TYPE、FAILOVER_METHODおよびFAILED_OVER列を問い合せて、TAFが正しく構成されていることを確認できます。

関連項目

- [接続ロード・バランシング](#)
- [アプリケーション・コンティニューイティの確保](#)に関する項

親トピック: [Oracle Net Servicesの拡張機能の使用](#)

13.3.1 透過的アプリケーション・フェイルオーバーについて

透過的アプリケーション・フェイルオーバーを使用すると、クライアントは、接続先の現在のインスタンスの障害発生時に、データベース・インスタンスに自動的に再接続できます。サーバーは通知を使用して、クライアント側でTAFコールバックをトリガーします。

サーバー側のサービス属性を使用してTAFを構成できます。この方法を使用することをお勧めします。または、アプリケーション接続文字列でTAFを構成できます。サーバー側のサービス属性は、接続文字列で指定された値より優先されます。

TAFは、次のいずれかのモードで動作します。

- セッション・フェイルオーバー：失われた接続およびセッションを再作成します。
- 選択フェイルオーバー：進行中の問合せを再実行します。

障害が発生すると、Oracle Call Interface(OCI)コールバックを使用してクライアント側でコールバック機能が開始されます。これは、標準のOCI接続、および接続プール接続とセッション・プール接続を使用します。

TAFはOracle Data Guardで動作して、自動フェイルオーバーを提供します。TAFは次のデータベース構成を使用して、データベース障害を効果的に隠ぺいします。

- Oracle Real Application Clusters
- レプリケート・システム
- スタンバイ・データベース
- シングル・インスタンスのOracle Database

関連項目

- [Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)
- [Oracle Call Interfaceプログラマーズ・ガイド](#)

13.3.2 透過的アプリケーション・フェイルオーバーのリストア対象

TAFは、アクティブなデータベース接続に関連するこれらの要素の一部またはすべてを、自動的にリストアします。TAFによって接続をリカバリするには、他の要素をアプリケーション・コードに埋め込むことが必要な場合があります。

- クライアント/サーバー・データベース接続: TAFは、同じ接続文字列、またはフェイルオーバーを構成するときに指定する代替接続文字列を使用して、自動的に接続を再確立します。
- ユーザーのデータベース・セッション: TAFは、障害発生前と同じユーザーIDで自動的にユーザーをログインします。複数のユーザーが接続を使用していて、これらのユーザーがデータベース・コマンドを処理しようとする、TAFはこれらのユーザーを自動的にログインします。残念ながら、TAFは他のセッション・プロパティを自動的にリストアできません。これらのプロパティは、コールバック関数をコールすることによりリストアできます。
- 完了したコマンド: 接続失敗の時点でコマンドが完全に実行され、データベースの状態が変更された場合は、TAFはこのコマンドを再送しません。データベースを変更した可能性があるコマンドに応答してTAFが再接続した場合は、TAFはアプリケーションにエラー・メッセージを発行します。
- フェッチに使用されるオープン・カーソル: TAFでは、フェイルオーバー前にカーソルから行のフェッチを開始していたアプリケーションが、フェイルオーバー後も行のフェッチを続行できます。これは選択フェイルオーバーと呼ばれています。これは、同じスナップショットを使用してSELECT文を再実行し、すでにフェッチした行を廃棄し、最初にフェッチしなかった行を取得することによって実現されます。TAFでは、廃棄した行が最初に戻された行であることを確認するか、またはエラー・メッセージを返します。
- アクティブ・トランザクション: TAFはフェイルオーバー後にアクティブ・トランザクションを保つことができないため、障害発生時にはアクティブ・トランザクションはすべてロールバックされます。このため、アプリケーションは、ROLLBACKコマンドが発行されるまでエラー・メッセージを受け取ることになります。
- サーバー側プログラム変数: PL/SQLパッケージの状態などのサーバー側プログラム変数は、障害発生時に消失します。TAFはこれらをリカバリできません。これらは、フェイルオーバー・コールバックからコールすることにより初期化できます。

関連項目

- [Oracle Call Interfaceプログラマーズ・ガイド](#)

親トピック: [透過的アプリケーション・フェイルオーバーの構成について](#)

13.3.3 FAILOVER_MODEパラメータについて

FAILOVER_MODEパラメータは、これらの追加パラメータをサポートします。これらは、接続記述子のCONNECT_DATAセクションで指定できます。

表13-4 FAILOVER_MODEパラメータの追加パラメータ

パラメータ	説明
BACKUP	バックアップ接続として使用する別のネットワーク・サービス名。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。

パラメータ	説明
DELAY	次に接続を試みるまでの待機時間(秒数)。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。 コールバック関数が登録されていると、このパラメータは無視されます。
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の設定。 <ul style="list-style-type: none"> ● 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: 接続を事前に確立します。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できる必要があります。
RETRIES	フェイルオーバー後に接続を行う回数。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。 コールバック関数が登録されていると、このパラメータは無視されます。
TYPE	フェイルオーバーの種類。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 Oracle Net フェイルオーバー機能を利用できます。 <ul style="list-style-type: none"> ● session: フェイルオーバー・セッションに設定します。ユーザーの接続が消失した場合、バックアップ上にそのユーザーの新規セッションが自動的に作成されます。このフェイルオーバーでは、検索操作はリカバリされません。 ● select: 障害発生後もユーザーはカーソルを継続してオープンし、フェッチできるようにします。ただしこのモードでは、通常の検索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: これがデフォルトです。フェイルオーバー機能は使用されません。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。

ノート:



Oracle Net Manager 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、手動で設定する必要があります。

親トピック: [透過的アプリケーション・フェイルオーバーの構成について](#)

13.3.4 透過的アプリケーション・フェイルオーバーの実装

TAFは、FAILOVER_MODEパラメータの設定次第で複数の方法で実装できます。次の方法をお勧めします。

重要:



listener.ora ファイルの SID_LIST_listener_name セクションの GLOBAL_DBNAME パラメータは設定しないでください。静的に構成されたグローバル・データベース名は、TAF を使用禁止にします。

- [接続時フェイルオーバーおよびクライアント・ロード・バランシングのTAF](#)
複数のアドレスに対して接続時フェイルオーバーおよびクライアント・ロード・バランシングのTAFを実装します。
- [接続再試行のTAF](#)
TAFを使用すると、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できます。
- [接続事前確立のTAF](#)
TAFを使用すると、バックアップ接続を事前確立できます。初期接続とバックアップ接続を明示的に指定する必要があります。

親トピック: [透過的アプリケーション・フェイルオーバーの構成について](#)

13.3.4.1 接続時フェイルオーバーおよびクライアント・ロード・バランシングのTAF

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・バランシングを行うようにTAFを実装します。

次の例では、Oracle Netはsales1-serverまたはsales2-serverのいずれかのプロトコル・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、TAFアプリケーションが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.example.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.example.com)
      (FAILOVER_MODE=
        (TYPE=select)
        (METHOD=basic))))
```

前述の例には複数のアドレスが含まれていますが、オプションのADDRESS_LISTパラメータは使用されていません。

親トピック: [透過的アプリケーション・フェイルオーバーの実装](#)

13.3.4.2 接続再試行のTAF

TAFを使用すると、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できます。

次の例では、Oracle Netはsales1-serverのリスナーへの再接続を試行します。このフェイルオーバー接続に失敗した場合は、15秒待機してから再び再接続を試行します。Oracle Netは最大20回まで再接続を試行します。

```
sales.us.example.com=
(DESCRIPTION=
 (ADDRESS=
  (PROTOCOL=tcp)
  (HOST=sales1-server)
  (PORT=1521))
 (CONNECT_DATA=
  (SERVICE_NAME=sales.us.example.com)
  (FAILOVER_MODE=
   (TYPE=select)
   (METHOD=basic)
   (RETRIES=20)
   (DELAY=15))))
```

親トピック: [透過的アプリケーション・フェイルオーバーの実装](#)

13.3.4.3 接続事前確立のTAF

TAFを使用すると、バックアップ接続を事前に確立できます。初期接続とバックアップ接続を明示的に指定する必要があります。

次の例では、ネットワーク・サービス名sales1.us.example.comを使用してsales1-serverのリスナーに接続するクライアントが、sales2-serverにも事前接続されます。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。同様に、sales2-serverのリスナーに、sales2.us.example.comを使用して接続するクライアントのために、Oracle Netはsales1-serverに事前に接続します。

```
sales1.us.example.com=
(DESCRIPTION=
 (ADDRESS=
  (PROTOCOL=tcp)
  (HOST=sales1-server)
  (PORT=1521))
 (CONNECT_DATA=
  (SERVICE_NAME=sales.us.example.com)
  (INSTANCE_NAME=sales1)
  (FAILOVER_MODE=
   (BACKUP=sales2.us.example.com)
   (TYPE=select)
   (METHOD=preconnect))))
sales2.us.example.com=
(DESCRIPTION=
 (ADDRESS=
  (PROTOCOL=tcp)
  (HOST=sales2-server)
  (PORT=1521))
 (CONNECT_DATA=
  (SERVICE_NAME=sales.us.example.com)
  (INSTANCE_NAME=sales2)
```

```
(FAILOVER_MODE=  
  (BACKUP=sales1.us.example.com)  
  (TYPE=select)  
  (METHOD=preconnect)))
```

親トピック: [透過的アプリケーション・フェイルオーバーの実装](#)

13.3.5 透過的アプリケーション・フェイルオーバーの確認

V\$SESSIONビューでFAILOVER_TYPE、FAILOVER_METHODおよびFAILED_OVER列を問い合わせて、TAFが正しく構成されているかを確認できます。

問合せを使用して列を表示する例を次に示します。

```
SELECT MACHINE, FAILOVER_TYPE, FAILOVER_METHOD, FAILED_OVER, COUNT(*)  
FROM V$SESSION  
GROUP BY MACHINE, FAILOVER_TYPE, FAILOVER_METHOD, FAILED_OVER;
```

フェイルオーバー前の出力は次のようになります。

MACHINE	FAILOVER_TYPE	FAILOVER_METHOD	FAILED_OVER	COUNT (*)
sales1	NONE	NONE	NO	11
sales2	SELECT	PRECONNECT	NO	1

フェイルオーバー後の出力は次のようになります。

MACHINE	FAILOVER_TYPE	FAILOVER_METHOD	FAILED_OVER	COUNT (*)
sales2	NONE	NONE	NO	10
sales2	SELECT	PRECONNECT	YES	1

ノート:

TAF の各ステップは、適切に構成された OCI TAF CALLBACK ファンクションを使用して監視できます。

関連項目

- [Oracle Call Interfaceプログラマーズ・ガイド](#)
- [Oracle Databaseリファレンス](#)

親トピック: [透過的アプリケーション・フェイルオーバーの構成について](#)

13.4 プライマリおよびセカンダリ・インスタンス構成のインスタンス・ロールの指定

INSTANCE_ROLEパラメータは、接続記述子のCONNECT_DATAセクションのオプションのパラメータです。これを使用すると、Oracle RAC構成のプライマリまたはセカンダリ・インスタンスへの接続を指定できます。

このパラメータは、次の場合に役立ちます。

- プライマリまたはセカンダリ・インスタンスに明示的に接続する場合。デフォルトはプライマリ・インスタンスです。
- TAFを使用してセカンダリ・インスタンスに事前に接続する場合。

表13-5に、INSTANCE_ROLEパラメータを説明します。

表13-5 INSTANCE_ROLEパラメータ

INSTANCE_ROLEパラメータ	説明
PRIMARY	プライマリ・インスタンスへの接続を指定します。
SECONDARY	セカンダリ・インスタンスへの接続を指定します。
ANY	プライマリ・インスタンス・ロールかセカンダリ・インスタンス・ロールかにかかわらず、ロード量が最小のインスタンスへの接続を指定します。

インスタンス・ロール・タイプへの接続

次のtnsnames.oraファイルの例では、ネットワーク・サービス名sales_primaryがプライマリ・インスタンスへの接続を使用可能にし、ネットワーク・サービス名sales_secondaryがセカンダリ・インスタンスへの接続を使用可能にします。

```
sales_primary=
(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.example.com)
    (INSTANCE_ROLE=primary)))
sales_secondary=
(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.example.com)
    (INSTANCE_ROLE=secondary)))
```

特定のインスタンスへの接続

Oracle Enterprise Manager Cloud Controlおよびその他のシステム管理製品が、管理タスクを実行するロールにかかわ

らず、特定のインスタンスに接続する必要がある場合があります。このような接続の場合は、(INSTANCE_NAME=instance_name)および(INSTANCE_ROLE=any)を構成して、ロールにかかわらずインスタンスに接続します。次の例では、ネットワーク・サービス名sales1によってsales1-serverのインスタンスへの接続が可能になり、sales2によってsales2-serverのインスタンスへの接続が可能になります。(SERVER=dedicated)は、専用サーバー接続を強制するために指定します。

```
sales1=
(DESCRIPTION=
 (ADDRESS=
  (PROTOCOL=tcp)
  (HOST=sales1-server)
  (PORT=1521))
 (CONNECT_DATA=
  (SERVICE_NAME=sales.us.example.com)
  (INSTANCE_ROLE=any)
  (INSTANCE_NAME=sales1)
  (SERVER=dedicated)))
sales2=
(DESCRIPTION=
 (ADDRESS=
  (PROTOCOL=tcp)
  (HOST=sales2-server)
  (PORT=1521))
 (CONNECT_DATA=
  (SERVICE_NAME=sales.us.example.com)
  (INSTANCE_ROLE=any)
  (INSTANCE_NAME=sales2)
  (SERVER=dedicated)))
```

ノート:



フェイルオーバーは前の設定と互換性がありません。

接続事前確立のTAF

TAFが構成された場合、セカンダリ・インスタンスにバックアップ接続を事前に確立できます。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、Oracle Netがsales1-serverのリスナーに接続し、セカンダリ・インスタンスのsales2-serverに事前に接続します。接続後にsales1-serverに障害が発生した場合は、TAFアプリケーションがセカンダリ・インスタンスの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。

```
sales1.example.com=
(DESCRIPTION=
 (ADDRESS=
  (PROTOCOL=tcp)
  (HOST=sales1-server)
  (PORT=1521))
 (CONNECT_DATA=
  (SERVICE_NAME=sales.us.example.com)
  (INSTANCE_ROLE=primary)
  (FAILOVER_MODE=
  (BACKUP=sales2.example.com)
```

```
(TYPE=select)
(METHOD=preconnect)))
sales2.example.com=
(DESCRIPTION=
(AADDRESS=
(PROTOCOL=tcp)
(HOST=sales2-server)
(PORT=1521))
(CONNECT_DATA=
(SERVICE_NAME=sales.us.example.com)
(INSTANCE_ROLE=secondary)))
```

親トピック: [Oracle Net Servicesの拡張機能の使用](#)

13.5 静的サービス登録の構成

リスナーはlistener.oraファイルの静的な構成情報を使用する前に、データベースやインスタンスの動的なサービス情報を使用します。

静的なサービス情報の構成は次の場合に必要です。

- 外部プロシージャ・コールを使用する場合
- Oracle異機種間サービスを使用する場合
- Oracle Data Guardを使用する場合
- Oracle Enterprise Manager Cloud Control以外のツールからデータベースをリモートで起動する場合
- Oracle8iリリース2 (8.1)より前のOracleデータベースに接続する場合
- [リスナーの静的サービス情報の構成](#)
Oracle Enterprise Manager Cloud Controlを使用してリスナーのデータベース・サービス情報を静的に構成する方法について学習します。

親トピック: [Oracle Net Servicesの拡張機能の使用](#)

13.5.1 リスナーの静的サービス情報の構成

Oracle Enterprise Manager Cloud Controlを使用してリスナーのデータベース・サービス情報を静的に構成する方法について学習します。

ノート:



Oracle Real Application Clusters 環境内など、接続時フェイルオーバーまたは TAF を使用している場合は、GLOBAL_DBNAME パラメータを設定しないでください。

1. Oracle Enterprise Manager Cloud Controlの「Net Services管理」ページにアクセスします。
2. 「管理」リストから「リスナー」を選択し、構成ファイルを含むOracleホームを選択します。
3. 「実行」をクリックします。データベース・サーバーへのログインを求められる場合があります。

- 「リスナー」ページが表示されます。
- リスナーを選択し、「編集」をクリックします。
リスナーの編集ページが表示されます。
 - 「静的データベース登録」タブをクリックし、続いて「追加」をクリックします。
「データベース・サービスを追加」ページが表示されます。必要な情報をフィールドに入力します。
 - 「OK」をクリックします。



ノート:

Oracle Net Manager を使用すると、静的なサービス情報の設定も行うことができます。詳細は、オンライン・ヘルプの「Statically Configure Database Service Information」を参照してください。

関連項目

- [Oracle Enterprise Manager Cloud Controlを使用したOracle Net Servicesの構成](#)
- [動的なサービス登録の構成](#)

親トピック: [静的なサービス登録の構成](#)

13.6 サード・パーティのデータベース・サービスへの接続の構成

次の項目では、サード・パーティ・データベース・サービスへの接続の構成方法を説明します。

- [外部プロシージャのデフォルト構成](#)
- [Oracle異機種間サービスのOracle Net Servicesについて](#)
- [Oracle Rdbデータベースを使用する場合のOracle Net Servicesの構成](#)

親トピック: [Oracle Net Servicesの拡張機能の使用](#)

13.6.1 外部プロシージャのデフォルト構成

[外部プロシージャ](#)は、別のプログラムからコールされる、異なる言語で記述されたプロシージャです。特定目的の処理実行に必要な1つ以上のCルーチンをコールするPL/SQLプログラムは、その一例です。

アプリケーションが外部プロシージャをコールすると、Oracle Databaseはextprocという名前の外部プロシージャ・エージェントを開始します。Oracle Databaseによって確立されたネットワーク接続を使用して、アプリケーションがエージェントに次の情報を渡します。

- DLLまたは共有ライブラリの名前
- 外部プロシージャ名
- 任意のパラメータ

次に、エージェントはDLLや共有ライブラリをロードして外部プロシージャを実行し、外部プロシージャから戻された値をアプリケーションに戻します。エージェントは、外部プロシージャをコールするアプリケーションと同じコンピュータに常駐している必要があります。

外部プロシージャのデフォルト構成を使用する場合、extprocエージェントはOracle Databaseによって直接生成されます。

listener.oraファイル、tnsnames.oraファイルのいずれも構成の変更は必要ありません。ただし、ORACLE_HOME/hs/adminディレクトリにあるextproc.oraファイルの外部プロシージャで使用される環境変数を定義する必要があります。外部プロシージャのデフォルトの構成を使用しない場合は、[表13-6](#)に示すパラメータを設定する必要があります。

表13-6 listener.oraファイルの外部プロシージャ設定

Oracle Enterprise Manager Cloud Controlのフィールド		
listener.oraの	パラメータ	説明
プログラム名	PROGRAM	<p>実行可能な外部プロシージャ・エージェントの名前。</p> <p>ノート: Microsoft Windows の場合、実行可能ファイルが、ORACLE_HOME¥bin ディレクトリに存在している必要があります。</p>
環境変数	ENVS	<p>ORACLE_HOME/hs/admin ディレクトリにある extproc.ora ファイルの外部プロシージャで使用される環境変数。</p> <p>ノート: extproc.ora が使用されている場合、listener.ora の同じ環境変数 ENVS より優先されます。</p> <p>構文: SET name=value</p> <p>例: SET EXTPROC_DLLS=ANY</p> <p>EXTPROC_DLLS 環境変数を指定して、extproc エージェントがロードできる DLL を制限します。EXTPROC_DLLS 環境変数の指定がない場合、extproc エージェントは、UNIX オペレーティング・システムの場合は ORACLE_HOME/lib ディレクトリから、Microsoft Windows の場合は ORACLE_HOME¥bin ディレクトリから DLL をロードします。</p> <p>EXTPROC_DLLS を次のいずれかの値に設定します。</p> <ul style="list-style-type: none"> ● DLL のコロン区切りリスト 脚注 1 <p>構文: "DLL: DLL"</p> <p>この値に設定すると、extproc エージェントは指定された DLL をロードしたり、または UNIX オペレーティング・システムの場合は ORACLE_HOME/lib ディレクトリから、Microsoft Windows の場合は ORACLE_HOME¥bin ディレクトリから DLL をロードできます。DLL の完全なディレクトリ・パスとファイル名を入力する必要があります。</p> ● ONLY (セキュリティを最大にする場合に推奨) 脚注 1

構文: "ONLY: DLL: DLL"

この値に設定すると、extproc エージェントは指定された DLL のみロードできます。DLL の完全なディレクトリ・パスとファイル名を入力する必要があります。

- ANY

構文: "ANY"

説明: この値に設定すると、extproc エージェントは任意の DLL をロードできます。ANY に設定すると、DLL チェックは無効になります。

例:

```
"EXTPROC_DLLS=/home/xyz/mylib.so:/home/abc/urllib.so,
LD_LIBRARY_PATH=/private/xpm/lib:/private/mylibs,
MYPATH=/usr/ucb:/usr/local/packages, APL_ENV_FILE=/apl/conf/
env.txt"
```

```
"EXTPROC_DLLS=ONLY:/home/xyz/mylib.so:/home/abc/urllib.so,
LD_LIBRARY_PATH=/private/xpm/lib:/private/mylibs,
MYPATH=/usr/ucb:/usr/local/packages, APL_ENV_FILE=/apl/conf/
env.txt"
```

```
"EXTPROC_DLLS=ANY, LD_LIBRARY_PATH=/private/xpm/lib:/private/
mylibs,
MYPATH=/usr/ucb:/usr/local/packages, APL_ENV_FILE=/apl/conf/
env.txt"
```

ノート:

有効なユーザーと実際のユーザー、または有効なグループと実際のグループが異なる場合、LD_LIBRARY_PATH 環境設定は無視されます。

Oracle ホーム・ディレクトリ	ORACLE_HOME	エージェントの Oracle ホームの場所。
-------------------	-------------	------------------------

Oracle システム識別子 (SID)	SID_NAME	外部プロシージャ・エージェントのシステム識別子(任意の名前)。 Oracle リスナーによって生成された外部プロシージャ・エージェントの使用方の詳細は、 外部プロシージャを使用する場合の Oracle Net Services の構成 の例を参照してください。
----------------------	----------	-----------------------------------------------------------------------------------------------------------------------------------------------------------

DLLは、Microsoft Windowsではセミコロン(;)で区切ります。

ノート:

外部プロシージャのデフォルト構成では、ネットワーク・リスナーが Oracle Database および extproc エージェントを使用する必要はありません。extproc エージェントは Oracle Database によって直接生成されるため、Oracle リスナーによって予期せず extproc エージェントが生成される恐れはありません。セキュリティを最大限にするため、このデフォルト構成をお勧めします。

外部プロシージャのデフォルト構成を変更し、Oracle リスナーによって extproc エージェントを生成できます。このためには、追加のネットワーク構成ステップを実行する必要があります。

次のものを使用する場合、Oracle リスナーによって extproc エージェントを生成する必要があります。

- マルチスレッド・エージェント
 - MTS モードの Oracle Database(Microsoft Windows)
 - 外部プロシージャを別の extproc エージェントにリダイレクトできるような、LIBRARY 指定の AGENT 句または PROCEDURE 指定の AGENT IN 句。
- [外部プロシージャを使用する場合のOracle Net Servicesの構成](#)

関連項目:

外部プロシージャの保護の詳細は、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください。

親トピック: [サード・パーティのデータベース・サービスへの接続の構成](#)

13.6.1.1 外部プロシージャを使用する場合のOracle Net Servicesの構成

外部プロシージャのデフォルト構成を変更し、旧リリースのOracle Databaseに類似したリスナーによってextprocエージェントを生成できます。プロセスは次のとおりです。

1. 既存のリスナーを構成するか、新規リスナーを作成して、外部プロシージャを処理します。

[例13-1](#)では、listener.oraファイルの構成例を示します。

例13-1 外部プロシージャを設定したlistener.oraファイル

```
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.example.com)
```

```
(ORACLE_HOME=/oracle)
(SID_NAME=sales))
(SID_DESC=
(SID_NAME=plsextproc)
(ORACLE_HOME=/oracle)
(PROGRAM=extproc))
```

2. tnsnames.oraに新しいエントリを追加します。

[例13-2](#)では、tnsnames.oraファイルの構成例を示します。

例13-2 外部プロシージャを設定したtnsnames.oraファイル

```
EXTPROC_CONNECTION_DATA_1=
(DESCRIPTION=
(AADDRESS=(PROTOCOL=ipc) (KEY=extproc))
(CONNECT_DATA=
(SID=plsextproc)))
```

3. 外部プロシージャを別のextprocエージェント(Oracleリスナーによって生成されたextprocなど)にリダイレクトできるような、LIBRARY指定のAGENT句またはPROCEDURE指定のAGENT IN句を使用します。

```
$ cat test.c
#include <stdlib.h>
int negative(char* db, int n)
{
    return -1*n;
}
char* mygetenv(const char* env)
{
    return getenv(env);
}
$ gcc -shared -fPIC -o test.so test.c
$ cp test.so $ORACLE_HOME/lib
```

SQL*PLUSで、次を実行します。

```
DROP DATABASE LINK extproclink;
CREATE DATABASE LINK extproclink USING 'extproc_connection_data_1';
CREATE OR REPLACE LIBRARY test1 AS '$ORACLE_HOME/lib/test.so';
/
--
-- Use 'AGENT' clause in LIBRARY SPEC
--
CREATE OR REPLACE LIBRARY test2 AS '$ORACLE_HOME/lib/test.so' AGENT 'extproclink';
/
--
-- Use 'AGENT IN' clause in FUNCTION
--
CREATE OR REPLACE FUNCTION ftest1(x VARCHAR2, y BINARY_INTEGER)
RETURN BINARY_INTEGER
AS LANGUAGE C
LIBRARY test1
NAME "negative"
PARAMETERS(x STRING, y INT)
AGENT IN ( x );
```

```

/
CREATE OR REPLACE FUNCTION ftest2(x VARCHAR2)
RETURN VARCHAR2
AS LANGUAGE C
LIBRARY test2
NAME "mygetenv";
/
$ select ftest1(' extproclink', 123) from dual;
$ select ftest2(' LD_LIBRARY_PATH') from dual;

```

外部プロシージャのリスナーには、oracleユーザーが所有するファイルへの一般的なアクセス権のないユーザー・アカウントを含めないようにしてください。具体的には、このユーザーは、データベース・ファイルまたはOracleサーバーのアドレス空間に対する読取り権限または書込み権限を持つことはできません。また、このユーザーは、listener.oraファイルへの読取りアクセス権を持つ必要がありますが、書込みアクセス権を持つことはできません。

低い権限でリスナーを実行することによって、リスナー制御のSETコマンドを使用せずに、listener.oraファイルにあるリスナーの構成を変更できます。このため、オラクル社では、リスナーの実行前にlistener.oraファイルの構成を完了することをお勧めします。

- [外部プロシージャのデフォルト構成の変更](#)
- [外部プロシージャを実行するための新規リスナーの作成](#)

親トピック: [外部プロシージャのデフォルト構成](#)

13.6.1.1.1 外部プロシージャのデフォルト構成の変更

外部プロシージャのデフォルト構成を変更するには、外部プロシージャを処理するための別のリスナーまたは既存のリスナーを構成および実行します。次の手順では、デフォルトの構成を変更する方法について説明します。

1. 次のようにOracle Net Configuration Assistantを使用して、外部プロシージャを処理するための既存のリスナーを構成します。ほとんどのインストールの場合、このリスナーの名前はLISTENERです。
 - a. Oracle Enterprise Manager Cloud Controlの「Net Services管理」ページにアクセスします。
 - b. 「管理」リストから「リスナー」を選択し、構成ファイルの場所を含むOracleホームを選択します。
 - c. 「実行」をクリックします。
「リスナー」ページが表示されます。
 - d. Oracle Net Configuration Assistantによって生成された既存のリスナーを選択し、「編集」をクリックします。
リスナーの編集ページが表示されます。
 - e. アドレス・セクションで外部プロシージャのプロトコル・アドレスを選択し、「追加」をクリックします。
 - f. 「その他のサービス」タブをクリックします。
 - g. 外部プロシージャのサービス情報を表す行を選択し、「追加」をクリックします。
2. [表13-6](#)に示されるパラメータを含めたextprocに関するサービス情報を、listener.oraファイルに追加します。

親トピック: [外部プロシージャを使用する場合のOracle Net Servicesの構成](#)

13.6.1.1.2 外部プロシージャを実行するための新規リスナーの作成

外部プロシージャを処理するための別のリスナーを構成および実行するには、Oracle Net Configuration Assistantを使用し、別のリスナーの外部プロシージャ・エントリを作成します。次の手順では、新しいリスナーを作成する方法について説明します。

1. 次のように、外部プロシージャを排他的に処理するリスナーを作成します。
 - a. 「リスナー」ページにナビゲートします。
 - b. 「作成」をクリックします。
「リスナーを作成」ページが表示されます。
 - c. 一意のリスナー名を「リスナー名」フィールドに入力します（「リスナー名」フィールドにLISTENEREXTPROCなど）。
2. アドレス・セクションに、次のようにIPCプロトコル・アドレスを構成します。
 - a. 「追加」をクリックします。
「アドレスの追加」ページが表示されます。
 - b. 「プロトコル」リストからIPCを選択します。
 - c. 「キー」フィールドに、extprocエージェントのキー値を入力します。
 - d. 「OK」をクリックします。

関連項目:

リスナー・プロトコル・アドレスの構成の詳細は、[「リスニング用プロトコル・アドレスの構成」](#)を参照してください。

3. [表13-6](#)に示されるパラメータを含めたextprocに関するサービス情報を、次のようにlistener.oraファイルに追加します。
 - a. 「その他のサービス」タブをクリックします。
 - b. 「追加」をクリックします。
「その他のサービスの作成」ページが表示されます。
 - c. 次の値をフィールドに入力します。
 - extprocを「プログラム名」フィールドに。
 - extproc実行ファイルが存在するOracleホームを「Oracleホーム・ディレクトリ」フィールドに。
 - extprocなどのシステム識別子を「SID」フィールドに。
 - d. 環境変数セクションで、「行の追加」をクリックします。
 - e. EXTPROC_DLLS環境変数を「名前」フィールドに、DLLのディレクトリ・パスとファイル名を「値」フィールドに入力します。
 - f. 「OK」をクリックします。
「リスナーを作成」ページが表示されます。
 - g. 「OK」をクリックしてリスナーを追加します。

リスナーが「リスナー」ページに追加されます。

listener.oraファイルの外部プロシージャに関する情報は、次の出力のように更新されます。

```
LISTENEREXTPROC=  
(DESCRIPTION=  
(ADDRESS=  
(PROTOCOL=ipc) (KEY=extproc)))
```

4. oracleユーザーより低い権限のユーザー・アカウントから、外部プロシージャのリスナーを起動します。

関連項目:

- リスナー制御ユーティリティのSTARTコマンドを使用してリスナーを起動する方法については、[「Oracle Net ListenerとOracle Databaseサーバーの起動」](#)を参照してください
- 外部プロシージャ・コールを使用可能にする方法については、[『Oracle Database開発ガイド』](#)を参照してください。

親トピック: [外部プロシージャを使用する場合のOracle Net Servicesの構成](#)

13.6.2 Oracle異機種間サービスのOracle Net Servicesについて

異機種間サービスは、Oracle Databaseサーバー内に統合されたコンポーネントで、Oracle Databaseサーバーからサード・パーティ・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。異機種間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 Databaseサーバーに存在するデータと同じように、サード・パーティ・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分散環境からサード・パーティのシステム、サービスまたはApplication Programming Interface (API)に透過的にアクセスすること

異機種間サービスはOracle Databaseサーバーで汎用的なテクノロジーを提供しますが、特定のサード・パーティ・システムにアクセスするには異機種間サービス・エージェントが必要です。

- [エージェントに接続するためのOracle Databaseの構成](#)

親トピック: [サード・パーティのデータベース・サービスへの接続の構成](#)

13.6.2.1 エージェントに接続するためのOracle Databaseの構成

サード・パーティ・システムへの接続を開始する際、Oracle Database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。次の手順では、Oracle Databaseサーバーがエージェントに接続できるように設定する方法について説明します。

1. ゲートウェイ上のリスナーが、Oracle Databaseサーバーから受信した要求をリスニングし、異機種間サービス・エージェントを起動するように構成します。このために、次のパラメータをlistener.oraファイルに構成します。
 - PROGRAM: 実行可能エージェントの名前
 - ORACLE_HOME: 実行可能エージェントのOracleホーム
 - SID_NAME: Oracleシステム識別子(SID)

2. Oracle Enterprise Manager Cloud Controlで、PROGRAM、ORACLE_HOMEおよびSIDパラメータを構成します。

- Oracle Enterprise Manager Cloud Controlの「Net Services管理」ページにアクセスします。

関連項目:

[「Oracle Enterprise Manager Cloud Controlを使用したOracle Net Servicesの構成」](#)

- 「管理」リストから「リスナー」を選択し、構成ファイルの場所を含むOracleホームを選択します。
- 「実行」をクリックします。
「リスナー」ページが表示されます。
- Oracle Net Configuration Assistantによって生成されたリスナーを選択し、「編集」をクリックします。
リスナーの編集ページが表示されます。
- 「その他のサービス」タブをクリックします。
- 「追加」をクリックします。
「その他のサービスの作成」ページが表示されます。
- ゲートウェイの作成で実行するプログラム名を「プログラム名」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ホーム・ディレクトリ」フィールドに、サード・パーティ・システムのOracleシステム識別子(SID)またはサービス名を「SID」フィールドに入力します。
- 「OK」をクリックします。
リスナーの編集ページが表示されます。
- 「OK」をクリックしてリスナーを変更します。
「リスナー」ページが表示されます。

listener.oraファイルの異機種間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=sybasegw)  
      (ORACLE_HOME=/oracle12c)  
      (PROGRAM=tg4sybs)))
```

3. Oracle Databaseが存在するコンピュータで、ゲートウェイ上のリスナーに接続するネットワーク・サービス名を設定します。接続で異機種間サービスを利用できるように、次のように接続記述子にHS=ok句を設定する必要があります。

- Oracle Databaseサーバーからサード・パーティ・システムへの接続に使用するネットワーク・サービス名を作成します。

関連項目:

ローカル・ネーミングの方法については、[タスク1「ネット・サービス名の構成」](#)、ディレクトリ・ネーミングの方法については、[タスク2「ネット・サービス名のディレクトリへの作成」](#)を参照してください。

- Oracle Enterprise Manager Cloud ControlまたはOracle Net Managerを使用して、HS=okを構成します。
 - Oracle Enterprise Manager Cloud Controlの場合は、「Net Services管理」ページにアクセスし、リスナーに対して「ローカル・ネーミング」を選択して、「ネット・サービス名の作成」ページで「詳細」タブをクリックします。次に、「異機種間サービスに使用」をクリックします。
 - Oracle Net Managerでは、「サービスの識別」ボックスの「詳細」をクリックします。「詳細サービス・オプション」ダイアログ・ボックスが表示されます。「異機種間サービスを使用」をクリックします。
- 「OK」をクリックして変更を確認します。

tnsnames.oraファイルでは、異機種間サービス用に構成された新規ネットワーク・サービス名が次のように更新されます。

```

sybase_gtw=
  (DESCRIPTION=
    (ADDRESS= (PROTOCOL=tcp) (HOST=gate-server) (PORT=1521))
    (CONNECT_DATA=
      (SERVICE_NAME=sybasegw)
    )
    (HS=ok)))
  )

```

関連項目:

[『Oracle Database Heterogeneous Connectivity管理者ガイド』](#)

親トピック: [Oracle異機種間サービスのOracle Net Servicesについて](#)

13.6.3 Oracle Rdbデータベースを使用する場合のOracle Net Servicesの構成

Oracle Rdbは、Digitalの64ビット・オペレーティング・システムで使用されるデータベースです。Oracle Rdbには独自のリスナーがあるため、クライアントはOracle Databaseと通信する場合と同じようにOracle Rdbと通信できます。

Oracle Rdbとの接続を開始するには、[表13-7](#)に示される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ワーク・サービス名を設定します。

表13-7 接続記述子でのOracle RDBデータベース設定

Oracle Enterprise Manager Cloud Controlのフィールド	tnsnames.oraのパラメータ	説明
RDB データベース	RDB_DATABASE	Oracle Rdb データベースのファイル名。
サービスのタイプ	TYPE_OF_SERVICE	Oracle RDB データベースで使用するサービスのタイプ。これは、Rdb インタフェース・ツールで使用されます。アプリケーションがOracle Rdb サービスと Oracle Database サービスの両方をサ

Oracle Enterprise

Manager Cloud

tnsnames.oraのパラメータ

Controlのフィールド

タ

説明

ポートし、アプリケーションがこの両方のサービス間でロード・バランシングを実行する場合にのみ、この機能を使用する必要があります。

グローバル・データベース名 GLOBAL_NAME

Oracle Rdb データベース。省略可能。

次の手順では、Oracle Rdbデータベースのクライアントを構成する方法について説明します。

1. Oracleサーバーからサード・パーティ・システムへの接続に使用するネットワーク・サービス名を作成します。

関連項目:

ローカル・ネーミングの方法については、[タスク1「ネット・サービス名の構成」](#)、ディレクトリ・ネーミングの方法については、[タスク2「ネット・サービス名のディレクトリへの作成」](#)を参照してください。

2. Oracle Enterprise Manager Cloud ControlまたはOracle Net Managerを使用して、Oracle Rdbパラメータを設定します。
 - Oracle Enterprise Manager Cloud Controlの場合は、「Net Services管理」ページにアクセスし、リスナーに対して「ローカル・ネーミング」を選択して、「ネット・サービス名の作成」ページで「詳細」タブをクリックします。
 - Oracle Net Managerでは、「サービスの識別」セクションの「詳細」をクリックします。「詳細サービス・オプション」ダイアログ・ボックスが表示されます。
3. 「Rdbデータベース」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。
4. (オプション)必要に応じて「サービスのタイプ」フィールドにサービスの種類を指定し、「グローバル・データベース名」フィールドにグローバル・データベース名を入力して、「OK」をクリックします。

tnsnames.oraファイルでは、Oracle Rdbデータベース用に構成された新規ネットワーク・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)))
```

次の例のTYPE_OF_SERVICEパラメータは、Oracle Rdbデータベース・サービスとOracleデータベース・サービス間のロード・バランシングに使用されます。

```
alpha5=
  (DESCRIPTION_LIST=
    (DESCRIPTION=
      (ADDRESS=...)
      (CONNECT_DATA=
```

```
(SERVICE_NAME=generic)
(RDB_DATABASE=[.mf]mf_personnel.rdb)
(GLOBAL_NAME=alpha5))
(DESCRIPTION=
(AADDRESS=...)
(CONNECT_DATA=
(SERVICE_NAME=sales.us.example.com)
(TYPE_OF_SERVICE=oracle_database))
```

親トピック: [サード・パーティのデータベース・サービスへの接続の構成](#)

14 パフォーマンスの最適化

接続パフォーマンスの最適化方法を学習します。

- [ネットワーク・データの圧縮化の利点の理解](#)
- [セッション・データ・ユニットの構成](#)
- [帯域幅遅延積の求め方](#)
- [I/Oバッファ領域の構成](#)

TCP/IPなどの信頼性の高いネットワーク・プロトコルでは、下位および上位のレイヤー・プロトコルとの間でデータの送受信中に、送受信バッファにデータをバッファリングします。これらのバッファのサイズは、フロー制御の決定に影響することで、ネットワーク・パフォーマンスに影響を与えます。

- [InfiniBand接続をサポートするSDPの構成](#)
- [InfiniBand接続をサポートするExadirectの構成](#)

Oracle Net Servicesでは、InfiniBand高速ネットワークのためのExadirectをサポートします。インフィニバンド環境では、Remote Direct Memory Access (RDMA)の活用によって待機時間とスループットが向上する新しいトランスポートを使用してください。

- [権限のないユーザーによるリソース使用の制限](#)
- [クライアント・サーバー接続のためのキーベースのルーティングの構成](#)

キーベース(または直接)ルーティングを使用して、シャード・データベースでクライアントとサーバー接続を確立する方法について学習します。シャードへのキーベース・ルーティングでは、必要なトランザクションに関連するデータが含まれている単一の関連するシャードに対して、シャーディング・キーを使用して接続が確立されます。

親トピック: [Oracle Net Servicesの構成と管理](#)

14.1 ネットワーク・データの圧縮化の利点の理解

ネットワーク・データを圧縮することにより、データ接続を介して送信される[セッション・データ・ユニット\(SDU\)](#)のサイズが少なくなります。データのサイズが小さくなると、ネットワークを介してSQL問合せおよび結果を送信するのに必要な時間が短縮します。また、圧縮されたデータでは、使用する帯域幅が少なく済むため、大量のデータを短時間で送信できます。データ圧縮プロセスは、アプリケーション・レイヤーに対して透過的です。

データ圧縮を使用する利点は、次のとおりです。

- ネットワーク・スループットが向上するため、制約された帯域幅の環境でも圧縮を利用することで、問合せの応答時間を短縮できます。
- 帯域幅の使用が減るため、他のアプリケーションで帯域幅を使用できます。
- サイト間で送信されるデータ量が減ります。

親トピック: [パフォーマンスの最適化](#)

14.2 セッション・データ・ユニットの構成

通常のデータベースの構成では、Oracle Netはデータをネットワークへ送出する前にカプセル化して、SDUのサイズでバッファに格納します。Oracle Netがそれぞれのバッファを送信するのは、バッファがいっぱいになったとき、フラッシュされたとき、またはアプリ

セッションがデータの読取りを試行したときです。任意のある時点で送信するためにOracle Netに送られるデータ量を基準にSDUバッファのサイズを調整すると、パフォーマンス、ネットワーク使用率およびメモリー消費を改善できます。大量のデータを転送するときは、SDUのサイズを大きくすることによって、パフォーマンスおよびネットワークのスループットを改善できます。特定のデプロイメントで高いスループットを達成できるようにSDUを調整できます。

任意のある時点で送信するためにOracle Netに送られるデータの量は、メッセージ・サイズとも呼ばれます。

設定できるSDUサイズの範囲は、512-22MBです。サイズの範囲が広いため、ネットワーク管理者は、特定のデプロイメントで最適なネットワーク・パフォーマンスを発揮できるように、SDUのサイズをチューニングできます。SDUサイズの値を大きくすると、多くのメモリーが必要になります。クライアントおよび専用サーバーのデフォルトのSDUのサイズは8192バイトです。共有サーバーのデフォルトのSDUのサイズは65535バイトになります。

使用される実際のSDUサイズは、接続時にクライアントとサーバー間でネゴシエートされ、クライアントとサーバーの各SDU値より小さい値になります。デフォルト以外のSDUサイズを構成する場合は、共有サーバーを使用している場合を除き、クライアントとサーバーの両方のコンピュータでSDUを構成する必要があります。

メッセージの大半が8192バイトより小さい場合または大きい場合は、SDUサイズを変更することを検討する必要があります。SDUサイズは、最も多いメッセージのサイズより70バイトのみ大きい値にします。70バイトを加えると最大SDUサイズを超える場合は、メッセージ・サイズを最小限の数に等分割し、各分割がSDUサイズより70バイト少なくなるようにSDUを設定します。デフォルトを変更するには、sqlnet.oraファイルでDEFAULT_SDU_SIZEパラメータを変更します。

たとえば、アプリケーションで送受信される大半のメッセージが8KB未満の場合は、70バイトをオーバーヘッドと考慮して、SDUを8KBに設定すれば問題ありません。利用可能なメモリーが十分にある場合は、SDUの最大値を使用すると、システム・コール数やOracle Net Servicesのオーバーヘッドを最小限に抑えることができます。

ノート:



Oracle Database 11g で、Oracle Net Services のバルク・データ転送が Oracle SecureFiles LOB や Oracle Data Guard REDO トランスポート・サービスなどのコンポーネント用に最適化されました。ネットワーク・パラメータ・ファイルで指定されている SDU サイズ制限は、これらのバルク・データ転送には適用されません。バルク・データ転送最適化は、ASO オプションが有効な場合や TLS トランスポートが使用されている場合には適用されません。

- [データベースのSDUサイズの設定](#)
- [クライアントのSDUサイズの設定](#)

関連項目:

- [「セッション・データ・ユニットのサイズによるデータ転送の最適化」](#)
- [「統計に関する出力例」](#)

親トピック: [パフォーマンスの最適化](#)

14.2.1 データベースのSDUサイズの設定

データベース・サーバーのSDUサイズを構成するには、次のファイルを構成します。

- sqlnet.ora

次のように、sqlnet.oraファイルでDEFAULT_SDU_SIZEパラメータを構成します。

```
DEFAULT_SDU_SIZE=32767
```

- 初期化パラメータ・ファイル

共有サーバー・プロセスを使用している場合、SDUのサイズを初期化パラメータ・ファイルのDISPATCHERSパラメータで次のように設定します。

```
DISPATCHERS="(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp))(SDU=8192))"
```

- listener.ora

listener.oraファイルのターゲットのリストを使用してリスナーを構成した場合は、専用サーバー・プロセスを使用する際に、sqlnet.oraファイルの現在の設定がSID_LISTパラメータのSDUの値で上書きされます。

親トピック: [セッション・データ・ユニットの構成](#)

14.2.2 クライアントのSDUサイズの設定

クライアントのSDUサイズを構成するには、次のファイルを構成します。

- sqlnet.ora

クライアント側のグローバル構成用に、次のように、sqlnet.oraファイルでDEFAULT_SDU_SIZEパラメータを構成します。

```
DEFAULT_SDU_SIZE=32767
```

- tnsnames.ora

特定の接続記述子では、DESCRIPTIONパラメータでSDUパラメータを指定できます。

```
sales.us.example.com=(DESCRIPTION=(SDU=11280)(ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))
```

SDUサイズは、特定の接続記述子のすべてのOracle Netプロトコルに適用されます。

親トピック: [セッション・データ・ユニットの構成](#)

14.3 帯域幅遅延積の求め方

帯域幅遅延積は、ネットワーク帯域幅とネットワークを通過するデータの往復時間の積です。往復時間を簡単に求めるには、ホストから相手のコンピュータにpingなどのコマンドを発行し、pingによって戻された応答時間を使用します。

たとえば、ネットワークの帯域幅が100Mbpsで、往復時間が5msの場合、送受信バッファは少なくとも $(100 \times 10^6) \times$

(5/10³)ビット、すなわち約62.5KBになります。

次の式は、使用する単位と係数の関係を示したものです。

$$\frac{100,000,000 \text{ bits}}{1 \text{ second}} \times \frac{1 \text{ byte}}{8 \text{ bits}} \times \frac{5 \text{ seconds}}{1000} = 62,500 \text{ bytes}$$

SEND_BUF_SIZEとRECV_BUF_SIZEパラメータを帯域幅遅延積以上の値に設定すると、ネットワーク帯域幅を最適に利用できるように、大量のデータが送信されます。

前の式を基にすると、このネットワーク・リンクの帯域幅遅延積は、約64KBになります。プライマリ・データベースとスタンバイ・データベースの間のREDOデータの転送に使用される最大のメッセージが1MBの場合、SEND_BUF_SIZEおよびRECV_BUF_SIZEパラメータの値は1MBになります。しかし、平均のメッセージがこれより小さい場合は、利用可能な帯域幅を最適に使用するには、64KBに設定すれば十分です。

ほとんどのネットワーク・プロトコルにおいて、一方のネットワーク接続(通常クライアント)のRECV_BUF_SIZEパラメータが、もう一方のネットワーク接続(通常サーバー)のSEND_BUF_SIZEパラメータの値と同一である必要があります。

関連項目:

メッセージ・サイズの決定の詳細は、[「統計に関する出力例」](#)を参照してください

親トピック: [パフォーマンスの最適化](#)

14.4 I/Oバッファ・スペースの構成

TCP/IPのような信頼性の高いネットワーク・プロトコルでは、下位または上位のレイヤー・プロトコル間でデータの送受信をしながら、送受信バッファにデータをバッファリングします。これらのバッファのサイズは、フロー制御の決定に影響することで、ネットワーク・パフォーマンスに影響を与えます。

RECV_BUF_SIZEおよびSEND_BUF_SIZEパラメータでは、Oracle Net接続に関連付けられるソケット・バッファのサイズを指定します。連続的なデータの流れを確保し、ネットワーク帯域幅の利用率を改善するには、RECV_BUF_SIZEおよびSEND_BUF_SIZEパラメータで、セッションの送受信操作に使用するI/Oバッファ・スペースの制限値を指定します。RECV_BUF_SIZEおよびSEND_BUF_SIZEパラメータの値は一致する必要はありませんが、環境に従って設定する必要があります。

最適なパフォーマンスを確保するには、ネットワーク接続で同時に送信されるすべてのデータを格納するのに十分な送受信バッファ・サイズを設定する必要があります。ネットワーク・パフォーマンスを最適にするには、これらのバッファを少なくとも帯域遅延積に設定してください。

これらの値は、ネットワークおよびシステムの性能に大きな影響を与えるので、慎重に使用してください。これらのパラメータのデフォルト値は、オペレーティング・システムによって異なります。Linuxオペレーティング・システムのデフォルトのデフォルト値は、次のとおりです。

- SEND_BUF_SIZE: 131,072バイト(128k)

- RECV_BUF_SIZE: 174,700バイト

これらのパラメータは、TCP、TLS付きTCP/IP、およびSDPでサポートされています。オペレーティング・システムによっては、これ以外のプロトコルがこれらのパラメータをサポートしていることがあります。これらのパラメータの推奨値は、インストール・ガイドで指定されています。詳細は、オペレーティング・システムに関するドキュメントを参照してください。

ノート:



- SEND_BUF_SIZE および RECV_BUF_SIZE パラメータの実際の値は、ホストのオペレーティング・システム上の制限やメモリの制約によって、設定した値より小さくなる場合があります。
- システムでサポートする必要がある同時接続の総数と、利用可能なメモリ・リソースを検討することが重要です。これらの接続によって消費される総メモリ容量は、同時接続数と各接続のバッファ・サイズに依存します。

- [サーバー上のI/Oバッファ・サイズの構成](#)
- [クライアント上のI/Oバッファ・スペースの構成](#)

関連項目

- [Oracle Call Interfaceプログラマーズ・ガイド](#)

親トピック: [パフォーマンスの最適化](#)

14.4.1 サーバー上のI/Oバッファ・サイズの構成

データベース・サーバーはデータをクライアントに書き込むため、通常はサーバー側でSEND_BUF_SIZEパラメータを設定してください。データベース・サーバーが大量の要求を受信する場合、RECV_BUF_SIZEパラメータも設定します。データベース・サーバーを構成するには、listener.oraおよびsqlnet.oraファイルでバッファ・スペースのサイズを設定します。

listener.oraファイルには、特定のプロトコル・アドレスまたは記述子のバッファ・スペース・パラメータを指定します。設定の例を次に示します。

```
LISTENER=
(DESCRIPTION=
(AADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521)
(SEND_BUF_SIZE=11784)
(RECV_BUF_SIZE=11784))
(AADDRESS=(PROTOCOL=ipc) (KEY=extproc)
(SEND_BUF_SIZE=11784)
(RECV_BUF_SIZE=11784)))
LISTENER2=
(DESCRIPTION=
(SEND_BUF_SIZE=8192)
(RECV_BUF_SIZE=16384)
(AADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521)))
```

前述の例には複数のアドレスが含まれていますが、ADDRESS_LISTパラメータは使用されていません。これは、ADDRESS_LISTパラメータは必須ではないためです。

sqlnet.oraファイルでの設定の例を次に示します。


```
RECV_BUF_SIZE=65536
SEND_BUF_SIZE=65536
```

- [共有サーバー・プロセスのバッファ・サイズ・パラメータの設定](#)

親トピック: [I/Oバッファ・スペースの構成](#)

14.4.1.1 共有サーバー・プロセスのバッファ・サイズ・パラメータの設定

共有サーバー・プロセスを使用する場合は、DISPATCHERS初期化パラメータに次のようにバッファ・スペース・パラメータを設定することで、サーバーのsqlnet.oraファイルから取得した現在の設定を上書きできます。

```
DISPATCHERS="(ADDRESS=(PROTOCOL=tcp) (SEND_BUF_SIZE=65536))"
```

親トピック: [サーバー上のI/Oバッファ・サイズの構成](#)

14.4.2 クライアント上のI/Oバッファ・スペースの構成

クライアントを構成するには、指定されているファイル内の次の場所にバッファ・スペース・サイズを設定してください。

- 通常は、RECV_BUF_SIZEパラメータを設定すれば十分機能します。しかし、クライアントが大きなサイズの要求を送信する場合は、SEND_BUF_SIZEパラメータも設定します。これらのパラメータは、クライアントのsqlnet.oraファイルで設定します。
- 特定の接続記述子では、クライアントのsqlnet.oraファイルの現在の設定値を上書きできます。tnsnames.oraファイルでは、次のようにして、特定のプロトコル・アドレスまたは記述子のバッファ・スペース・パラメータを指定できます。

```
sales.us.example.com=
  (DESCRIPTION=
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=tcp) (HOST=sales1-server) (PORT=1521)
        (SEND_BUF_SIZE=11784)
        (RECV_BUF_SIZE=11784))
      (ADDRESS=(PROTOCOL=tcp) (HOST=sales2-server) (PORT=1521)
        (SEND_BUF_SIZE=11784)
        (RECV_BUF_SIZE=11784))
    (CONNECT_DATA=
      (SERVICE_NAME=sales.us.example.com)))
hr.us.example.com=
  (DESCRIPTION=
    (SEND_BUF_SIZE=8192)
    (RECV_BUF_SIZE=8192)
    (ADDRESS=(PROTOCOL=tcp) (HOST=hr1-server) (PORT=1521))
    (CONNECT_DATA=
      (SERVICE_NAME=hr.us.example.com)))
```

親トピック: [I/Oバッファ・スペースの構成](#)

14.5 InfiniBand接続をサポートするSDPの構成

Oracle Net Servicesでは、InfiniBand高速ネットワークのSockets Direct Protocol([SDP](#))をサポートしています。

SDPは、クラスタ化されたサーバー環境における標準的な通信プロトコルです。SDPは、ネットワーク・インタフェース・カードとアプリケーション間のインタフェースの役割を果たします。SDPを使用することにより、アプリケーションではメッセージ交換の負荷の大部

分をネットワーク・インタフェース・カードに割り当て、CPUを他のタスクに解放することが可能になります。その結果、SDPによってネットワーク待機時間およびCPUの利用率を軽減できます。

SDPは特にSAN(System Area Network)向けに設計されています。SANは、複数のサーバー・システム間(Oracle WebLogic Serverやその他のサード・パーティの1つのスイッチでクラスタ化された中間層クライアントとデータベース・サーバーなど)の短距離の高性能通信という点に特徴があります。

ノート:

Oracle Database 12c とのバージョン互換性については、各ベンダーにお問い合わせください。

SDP サポートの詳細は、次の場所で Oracle Technology Network を参照してください。

<http://www.oracle.com/technetwork/index.html>

次の項では、中間層とデータベース・サーバー通信のためのSDPをサポートするOracle Netの設定方法について説明します。次の項目が含まれます。

- [SDP利用の前提](#)
- [サーバー上のSDPの構成](#)
- [クライアント上のSDPの構成](#)

関連項目:

サポート対象となっているデプロイメントの概要は、[「パフォーマンスの理解」](#)を参照してください

親トピック: [パフォーマンスの最適化](#)

14.5.1 SDP利用の前提

SDPのサポートを構成する前に、アプリケーションWebサーバーおよびデータベース・サーバーの両方に必要なハードウェアをインストールし、所定のベンダーのOpenFabrics Enterprise Distribution (OFED) 1.4または1.5と互換性のあるInfiniBand対応ハードウェアおよびソフトウェアを設定します。

InfiniBandソフトウェアのインストール時に、システムのSDPまたはアドレス・ファミリを定義する定数を識別します。この定数は、使用しているオペレーティング・システムまたはOFEDのマニュアルから取得できます。

関連項目:

インストールの詳細は、ベンダーのドキュメントを参照してください。

親トピック: [InfiniBand接続をサポートするSDPの構成](#)

14.5.2 サーバー上のSDPの構成

データベース・サーバーを構成するには、データベース・サーバー上の listener.ora ファイルで SDP アドレスを構成します。

ノート:



SDP または アドレス・プロトコル・ファミリの定数が、Oracle Net Services のデフォルト値である 27 ではない場合、sqlnet.ora ファイルの SDP.PF_INET_SDP パラメータに定数を定義してください。

次の例は、コンピュータ sales-server にポート番号 1521 を使用する SDP エンドポイントを示したものです。

```
LISTENER=
  (DESCRIPTION=
    (ADDRESS_LIST=
      (ADDRESS= (PROTOCOL=sdp) (HOST=sales-server) (PORT=1521))
      (ADDRESS= (PROTOCOL=tcp) (HOST=sales-server) (PORT=1521))
      (ADDRESS= (PROTOCOL=ipc) (KEY=extproc))))
```

関連項目:

[「リスナー・プロトコル・アドレスのリスト作成」](#)

親トピック: [InfiniBand 接続をサポートする SDP の構成](#)

14.5.3 クライアント上のSDPの構成

ノート:



SDP または アドレス・プロトコル・ファミリの定数が、Oracle Net Services のデフォルト値である 27 ではない場合、sqlnet.ora ファイルの SDP.PF_INET_SDP パラメータに定数を定義してください。

次の手順では、Oracle WebLogic Server サーバーまたは サード・パーティ 中間層クライアントを構成する方法について説明します。

1. サード・パーティ 中間層クライアントを構成する場合は、次に示すように、Oracle Database 12c クライアント・ソフトウェアを使用するようにクライアントをアップグレードします。
 - a. Oracle Universal Installer を実行します。
 - b. 「使用可能な製品」ページで、「Oracle Database 12c Client」を選択します。
2. Oracle WebLogic Server サーバーと サード・パーティ 中間層クライアントの両方に、次のようにネットワーク・サービス名を作成し、データベース・サーバーに接続します。
 - a. Oracle WebLogic Server サーバー向けに tnsnames.ora ファイルで設定した同じ TCP/IP プロトコル・アドレスを使用するネットワーク・サービス名を指定します。たとえば:

```
sales=
```

```
(DESCRIPTION=
  (ADDRESS= (PROTOCOL=tcp) (HOST=sales-server)))
(CONNECT_DATA=
  (SERVICE_NAME=sales.us.example.com)))
```

- b. サード・パーティ中間層クライアント向けにtnsnames.oraファイルで設定した同じSDPアドレスを使用するネットワーク・サービス名を指定します。

たとえば:

```
sales=
  (DESCRIPTION=
    (ADDRESS= (PROTOCOL=sdp) (HOST=sales-server)))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.example.com)))
```

関連項目:

接続記述子作成の詳細は、[「ネーミング・メソッドの構成」](#)を参照してください

親トピック: [InfiniBand接続をサポートするSDPの構成](#)

14.6 InfiniBand接続をサポートするExadirectの構成

Oracle Net Servicesでは、InfiniBand高速ネットワークのためのExadirectをサポートします。インフィニバンド環境では、Remote Direct Memory Access (RDMA)の活用によって待機時間とスループットが向上する新しいトランスポートを使用してください。

Exadirectプロトコルは、通信の制御にTCPを使用し、データにIB RCトランスポートを使用します。このリリースでは、Exadirectプロトコル・アダプタはOracle Linuxでのみサポートされています。

ノート:



Oracle Database 12c リリース 2 (12.2)では、Exadirect は単一 Exadirect エンドポイント構成をサポートします。DRCP はサポートしていません。

次の各項では、中間層とデータベース・サーバー通信のためのExadirectをサポートするOracle Netの設定方法について説明します。

- [Exadirectを使用するための前提条件](#)
- [サーバー上のExadirectの構成](#)
データベース・サーバーを構成するには、データベース・サーバー上のlistener.oraファイルでExadirectアドレスを構成します。
- [クライアント上のExadirectの構成](#)
tnsnames.oraファイルで構成されたものと同じExadirectアドレスを使用するネットワーク・サービス名を指定します。

親トピック: [パフォーマンスの最適化](#)

14.6.1 Exadirectを使用するための前提条件

Exadirectは、Exadataノード間のトランスポート・プロトコルとして使用できます。Exadirectが動作するには、Exadataのバージョンが12.1.2.3.3以上である必要があります。

親トピック: [InfiniBand接続をサポートするExadirectの構成](#)

14.6.2 サーバー上のExadirectの構成

データベース・サーバーを構成するには、データベース・サーバー上のlistener.oraファイルでExadirectアドレスを構成します。

listener.oraファイルでenable_exadirect_listener_name=onパラメータを構成します。

次の例は、192.168.10.1のIBインタフェースを装備したコンピュータ上のポート番号1522を使用するExadirectエンドポイントを示しています。

```
LISTENER
=
(DESCRIPTION=
(AADDRESS_LIST=
(AADDRESS=(PROTOCOL=exadirect) (HOST=192.168.10.1) (PORT=1522))
(AADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521))
(AADDRESS=(PROTOCOL=ipc) (KEY=extproc))))
ENABLE_EXADIRECT_LISTENER=on
```

ノート:



Exadirect フロー制御を有効にするには、sqlnet.ora ファイルで exadirect_flow_control=on を設定します。これは、ネゴシエート・パラメータであり、サーバーとクライアントの両方の sqlnet.ora ファイルで設定されている必要があります。EXADIRECT_RECVPOLL パラメータは、受け取り側が着信データをポーリングする時間を指定します。

Exadirect アダプタは HugePages を使用します。

親トピック: [InfiniBand接続をサポートするExadirectの構成](#)

14.6.3 クライアント上のExadirectの構成

tnsnames.oraファイルで構成されたものと同じExadirectアドレスを使用するネットワーク・サービス名を指定します。

Exadirectフロー制御は、sqlnet.oraファイルのexadirect_flow_control=onを設定することで有効にします。

たとえば:

```
(DESCRIPTION=
(AADDRESS=(PROTOCOL=exadirect) (HOST=192.168.10.1) (port=1522))
(CONNECT_DATA=
(SERVICE_NAME=sales.us.example.com)))
```

親トピック: [InfiniBand接続をサポートするExadirectの構成](#)

14.7 権限のないユーザーによるリソース使用の制限

リスナーまたはデータベース・サーバーへの権限のないアクセスにより、サービス拒否攻撃を受ける可能性があります。この攻撃は、

権限のあるユーザーがシステムにアクセスして使用する機能を、権限のないクライアントがブロックしようとします。悪意のあるクライアントが、接続、プロセスまたはスレッドなどのリソースを単に消費する目的で、リスナーやデータベース・サーバーに過剰な接続要求を送信する場合があります。この種の攻撃を軽減するには、認証の前にリソースを保持できる時間の制限を構成します。構成された制限を超えようとするクライアントは接続が終了し、クライアントのIPアドレスを含む監査証跡がロギングされます。

権限のないユーザーによるリソース使用を制限して監査証跡を使用可能にするには、[表14-1](#)で説明するパラメータに制限時間の値を設定します。これらのパラメータにはデフォルト値はありません。

表14-1 接続タイムアウト・パラメータ

パラメータ	ファイル	説明
INBOUND_CONNECT_TIMEOUT_listener_name	listener.ora	<p>ネットワーク接続が確立した後、クライアントがリスナーへの接続要求を完了する時間を秒単位で指定します。</p> <p>リスナーが指定の時間内にクライアント・リクエストを受信しない場合、接続は終了します。また、リスナーでは、クライアントの IP アドレスおよび「ORA-12525: TNS: リスナーは、クライアントのリクエストを許容時間内に受信しませんでした」のエラー・メッセージが listener.log ファイルに記録されます。</p>
SQLNET.INBOUND_CONNECT_TIMEOUT	データベース・サーバー上の sqlnet.ora	<p>クライアントがデータベース・サーバーに接続して必要な認証情報を提供する時間を、秒単位で指定します。</p> <p>クライアントが指定した時間内での接続の確立と認証の完了に失敗した場合、データベース・サーバーは接続を終了します。また、データベース・サーバーでは、クライアントの IP アドレスおよび「ORA-12170: TNS: 接続タイムアウトが発生しました」のエラー・メッセージが sqlnet.log ファイルに記録されます。クライアントは、「ORA-12547: TNS: 接続を失いました」または「ORA-12637: パケット受信に失敗しました」エラー・メッセージのいずれかを受信します。</p>

これらのパラメータの値を指定する場合、次の推奨事項を考慮してください。

- 両方のパラメータの初期値を低く設定してください。
- INBOUND_CONNECT_TIMEOUT_listener_nameパラメータの値を、SQLNET.INBOUND_CONNECT_TIMEOUTパラメータの値より低く設定してください。

たとえば、INBOUND_CONNECT_TIMEOUT_listener_nameパラメータを2秒に設定し、SQLNET.INBOUND_CONNECT_TIMEOUTパラメータを3秒に設定します。特定の環境におけるシステムあるいはネットワークの通常の遅延により、クライアントが指定の時間内に接続を完了できない場合は、必要なだけ時間を増やします。

関連項目:

- エラー・メッセージの対処については、[「Oracle Net Servicesの最も一般的なエラー・メッセージの解決」](#)を参照してください。

ださい

- listener.logファイル内のエントリの詳細は、[「リスナー・ログ・ファイルの分析」](#)を参照してください
- これらのパラメータの設定方法の詳細は、[「プロファイルの詳細情報について」](#)を参照してください

親トピック: [パフォーマンスの最適化](#)

14.8 クライアント/サーバー接続のためのキーベース・ルーティングの構成

キーベース(または直接)ルーティングを使用して、シャード・データベースにクライアント/サーバー接続を確立する方法を学習します。シャードへのキーベース・ルーティングでは、必要なトランザクションに関連するデータが含まれている単一の関連するシャードに対して、シャード・キーを使用して接続が確立されます。

- [キーベースのルーティングについて](#)
キーベースまたは直接ルーティングでは、接続文字列で指定されたシャード・キーの値に従って、クライアントからシャードにデータベース接続が直接確立されます。
- [接続文字列でのシャード・キーの指定](#)
接続文字列のCONNECT_DATAセクションにシャード・キーを指定することで、シャードに直接接続できます。シャード・デプロイメントのシャード・ディレクタ(エンドポイントとして)、グローバル・サービス名およびリージョン名を指定することもできます。

親トピック: [パフォーマンスの最適化](#)

14.8.1 キーベースのルーティングについて

Oracle Shardingを使用すると、複数の独立したOracleデータベースにデータを水平にパーティション化できます。このような構成内の各物理データベースをシャードと呼びます。シャード表パーティションはシャード・キーに基づいて表領域レベルでシャード間に分散されます。キーの例として顧客ID、アカウント番号、国IDなどがあります。キーベースまたは直接ルーティングでは、接続文字列で指定されたシャード・キーの値に従って、クライアントからシャードにデータベース接続が直接確立されます。

シャード・ディレクタは、グローバル・サービス・マネージャ(GSM)の特殊な実装であり、シャード・データベースに接続するクライアントのリージョナル・リスナーとして機能します。シャード・キーが接続要求で渡されると、シャード・ディレクタは、次のように適切なシャードへの接続を確立します。

- シャード・ディレクタは、サービスとキー値の検索を実行し、クライアントをデータの格納されている共有場所へとリダイレクトします。
- クライアントは、SQL文を実行して、結果をそのシャードから直接受け取ります。

直接ルーティングでは、1つの接続内で発行されたすべてのデータベース要求が、指定されたキー値と関連付けられているデータと関連があることが前提とされます。クライアントが別のキー値を持つデータにアクセスする必要がある場合は、新しいデータベース接続を確立する必要があります。

接続文字列は、Oracle Databaseが一貫性のあるハッシュ、リストまたは範囲によってシャードされる場合、単一のシャード・キーを含みます。Oracle Databaseでは、2レベルのシャード・キーを可能にするコンポジット・シャード・キーもサポートしています。まずデータが範囲またはリストによってシャードされ、次に一貫したハッシュによってさらにシャードされます。コンポジット・シャード・キー方法では、接続文字列にはシャード・キー(一貫性のあるハッシュによるシャード用)とスーパー・シャード・キー(リストまたは範囲によるシャード用)の両方が含まれます。

関連項目

- [Oracle Shardingの使用](#)
- [Oracle Database Global Data Services概要および管理ガイド](#)

親トピック: [クライアント/サーバー接続のためのキー・ベースのルーティングの構成](#)

14.8.2 接続文字列でのシャーディング・キーの指定

接続文字列のCONNECT_DATAセクションにシャーディング・キーを指定することで、シャードに直接接続できます。シャーディング・デプロイメントのシャード・ディレクタ(エンドポイントとして)、グローバル・サービス名およびリージョン名を指定することもできます。

次のいずれかのパラメータを使用して、シャード(パーティション)キー値を指定します。

- SHARDING_KEY: 特定のシャードのシャーディング・キーを簡易テキスト形式で指定します。
- SUPER_SHARDING_KEY: シャードのコレクションのシャード領域キーを簡易テキスト形式で指定します。
- SHARDING_KEY_B64: 特定のシャードのシャーディング・キーをbase64でエンコードされたバイナリ形式で指定します。
- SUPER_SHARDING_KEY_B64: base64でエンコードされたバイナリ形式でシャードのコレクションにシャード領域キーを指定します。

接続文字列は、グローバル・サービスへの接続で要求される形式になっている必要があります。

シャードの接続文字列の例を次に示します。ここでは、SHARDING_KEYおよびSUPER_SHARDING_KEYパラメータ値が簡易テキスト形式で指定されています。

```
(DESCRIPTION=
  (FAILOVER=on)
  (ADDRESS_LIST=
    (LOAD_BALANCE=ON)
    (ADDRESS=(host=sales-east1) (port=1522))
    (ADDRESS=(host=sales-east2) (port=1522))
    (ADDRESS=(host=sales-east3) (port=1522)))
  (ADDRESS_LIST=
    (LOAD_BALANCE=ON)
    (ADDRESS=(host=sales-west1) (port=1522))
    (ADDRESS=(host=sales-west2) (port=1522))
    (ADDRESS=(host=sales-west3) (port=1522)))
  (CONNECT_DATA=
    (SERVICE_NAME=sales)
    (SHARDING_KEY=40598230) (SUPER_SHARDING_KEY=gold)
    (REGION=east)))
```

関連項目

- [Oracle Shardingの使用](#)
- [SHARDING_KEY](#)
- [SUPER_SHARDING_KEY](#)

親トピック: [クライアント/サーバー接続のためのキー・ベースのルーティングの構成](#)

第III部 Oracle Net Servicesのテストおよびトラブルシューティング

第III部では、Oracle Net Servicesでの接続の確立方法および問題の識別と診断の方法について説明します。

- [接続のテスト](#)
- [Oracle Net Servicesのトラブルシューティング](#)

15 接続のテスト

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。Oracle Net Servicesには、リスナー、データベースおよびOracle Connection Managerを容易にテストできるツールがあります。

- [ネットワークのテスト](#)
- [TNSPINGユーティリティを使用したクライアントからの接続テスト](#)
- [TRCROUTEユーティリティを使用したクライアントからの接続テスト](#)

親トピック: [Oracle Net Servicesのテストおよびトラブルシューティング](#)

15.1 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. リスナーを個別に起動し、テストします。リスナーを起動するには、[「Oracle Net ListenerとOracle Databaseサーバーの起動」](#)で説明されている手順を実行します。

リスナーをテストするには、クライアントからそのリスナーによって制御される任意の有効なデータベースまでの接続を開始します。

2. Oracle Connection Managerがネットワークに含まれている場合は、各Oracle Connection Managerを起動し、テストします。Oracle Connection Managerを起動するには、[「Oracle Connection Managerの起動」](#)で説明されている手順を実行します。

Oracle Connection Managerをテストするには、クライアントから接続を開始して、Oracle Connection Managerに登録した任意の有効なデータベースまで接続を行います。

3. ループバック・テストまたはOracle Net Managerでサーバーをテストします。

[ループバック・テスト](#)では、Oracle Netを使用してデータベース・サーバーからInterprocess Communication(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、Oracle Netがデータベース・サーバーで機能していることが確認できます。次の手順では、Oracle Net Managerを使用してループバック・テストを実行する方法について説明します。

- a. Oracle Net Managerを起動します。

関連項目:

[「Oracle Net Managerを使用したOracle Net Servicesの構成」](#)

- b. ナビゲータ・ペインで、「ディレクトリ」または「ローカル」を展開し、「サービス・ネーミング」を選択します。
- c. ネットワーク・サービス名またはデータベース・サービスを選択します。
- d. 「ツール」を選択し、ネット・サービスのテストを選択します。

このテストでは、リスナーとデータベース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、[「Oracle Net ListenerとOracle Databaseサーバーの起動」](#)を参照してコンポーネントを起動してください。

テスト中、ステータスとテスト結果を示す「接続テスト」ダイアログ・ボックスが表示されます。テストが成功すると、次のメッセージが表示されます。

```
The connection test was successful.
```

テストが成功した場合は、ステップ3.eに進んでください。

テストが正常に終了しなかった場合は、エラー・メッセージを使用して対処方法を決定します。たとえば、エラー・メッセージが次の場合の対処方法について説明します。

```
Attempting to connect using userid: scott
The test did not succeed.
ORA-28000: the account is locked
```

```
There may be an error in the fields entered,
or the server may not be ready for a connection.
```

ロック解除されていることがわかっているアカウントにユーザー名を変更します。ユーザー名を変更するには、「ログインの変更」をクリックします。パスワードの入力を求められます。

e. 「閉じる」をクリックして「接続テスト」ダイアログ・ボックスを閉じます。

4. クライアントで接続できるかどうかテストします。

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、次のコマンドを使用して各クライアントからデータベース・サーバーへの接続を開始します。

```
CONNECT username@connect_identifier
```

親トピック: [接続のテスト](#)

15.2 TNSPINGユーティリティを使用したクライアントからの接続テスト

TNSPINGユーティリティでは、Oracle Netネットワーク上のサービスのリスナーに正常に到達できるかどうかを判断します。

TNSPINGユーティリティを使用して、クライアントからサーバー(またはサーバーから別のサーバー)に正常に接続できる場合、サーバーに対応するOracleリスナーへのSQLNET接続を確立するのにかかる推定ラウンドトリップ時間(ミリ秒)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。

接続をテストするには、次のコマンドを使用します。

```
tnsping net_service_name count
```

前述のコマンドでは、次の引数が使用されます。

- `net_service_name`: NISなど、`tnsnames.ora`ファイルまたは使用中のネーム・サービスに存在する名前にする必要があります。
- `count`: 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。この引数はオプションです。

指定されたネットワーク・サービス名がデータベース名の場合、TNSPINGでは、対応するリスナーへの接続を試みます。実際は、データベースが実行中かどうかは判別されません。SQL*Plusを使用して、データベースへの接続を試みます。

次にTNSPINGの例を示します。

ノート:



プラットフォームによってインター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 を呼び出すと、正しいインターフェース要件が表示されます。

[例15-1](#)は、TNSPINGコマンドを使用してsalesのネットワーク・サービス名で、データベースに対してリスナーを確認している例を示しています。

例15-1 TNSPINGによるリスナーの確認

```
TNSPING sales
```

これによって、次のメッセージが出力されます。

```
TNS Ping Utility for Linux: Version 18.0.0.0.0 - Production on 21-JAN-2018
Copyright (c) 1997, 2016 Oracle Corporation. All rights reserved.
Used parameter files:
Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP) (HOST = sales-server) (PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
sales.us.example.com)))
OK (10 msec)
```

salesデータベースに対してリスナーを使用できるかどうかを判別し、TNSPINGを8回試行してから終了するように指定するには、次の構文を使用します。

```
tnsping sales 8
```

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 Linux: Version 18.0.0.0.0 - Production on 21-JAN-2018
Copyright (c) 1997, 2016 Oracle Corporation. All rights reserved.
Used parameter files:
Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP) (HOST = sales-server) (PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
sales.us.example.com)))
OK (10 msec)
OK (0 msec)
OK (10 msec)
OK (0 msec)
OK (10 msec)
OK (10 msec)
OK (10 msec)
OK (10 msec)
OK (0 msec)
```

[例15-2](#)は、無効なネットワーク・サービス名の使用を確認するTNSPINGの例を示しています。

例15-2 TNSPINGを使用した無効なネット・サービス名の確認

```
tnsping badname
```

この試行では、次のメッセージが出力されます。

```
TNS Ping Utility for Linux: Version 18.0.0.0.0 - Production on 21-MAY-2018
Copyright (c) 1997, 2018 Oracle Corporation. All rights reserved.
Used parameter files:
TNS-03505: Failed to resolve name
```

[例15-3](#)は、TNSPINGを使用して名前を確認し、名前は有効であるが、解決されたアドレスに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など)場合の出力の例を示しています。

例15-3 TNSPINGを使用したリスナーなしの有効なネット・サービス名の確認

```
tnsping sales
TNS Ping Utility for Linux: Version 18.0.0.0.0 - Production on 21-JAN-2018
Copyright (c) 1997, 2016 Oracle Corporation. All rights reserved.
Used parameter files:
Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP) (HOST = sales-server) (PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
sales.us.example.com)))
TNS-12541: TNS:no listener
```

親トピック: [接続のテスト](#)

15.3 TRCROUTEユーティリティを使用したクライアントからの接続テスト

LinuxおよびUNIX環境では、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検出できます。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。TRCROUTE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できます。

TRCROUTEは最小限のリソースを使用します。接続パケットの接続データ内の情報を収集します。通常の接続パケットは影響を受けません。

TRCROUTEは、サーバーに作用しません。リスナーがTRCROUTE接続パケットを受信し、処理します。リスナーは、拒絶パケットに情報を入れることによって、クライアントに情報を返します。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ユーティリティを使用するには、次のコマンドを入力します。

```
trcroute net_service_name
```

[例15-4](#)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15-4 正常なTrace Route

```
trcroute sales
```

```
Trace Route Utility for Linux: Version 18.0.0.0.0 - Production on 21-JAN-2018
Copyright (c) 1999, 2018 Oracle Corporation. All rights reserved.
Route of TrcRoute:
-----
```

```
Node: Client          Time and address of entry into node:
-----
09-NOV-2012 21:48:48 ADDRESS= PROTOCOL=TCP  HOST=10.150.21.136  PORT=14001

Node: Server          Time and address of entry into node:
-----
09-NOV-2012 21:48:05 ADDRESS= PROTOCOL=TCP  HOST=10.150.21.136  PORT=14001
```

[例15-5](#)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15-5 エラーの発生したTrace Route

```
trcroute sales
Trace Route Utility for Linux: Version 18.0.0.0.0 - Production on 21-JAN-2018
Copyright (c) 1999, 2018 Oracle Corporation. All rights reserved.
Route of TrcRoute:
-----
Node: Client          Time and address of entry into node:
-----
25-FEB-2012 14:43:05 ADDRESS= PROTOCOL=TCP  HOST=sales-server  PORT=1521
TNS-12543: TNS:unable to connect to destination
TNS-12541: TNS:no listener
TNS-12560: TNS:protocol adapter error
TNS-03601: Failed in route information collection
```

親トピック: [接続のテスト](#)

16 Oracle Net Servicesのトラブルシューティング

Oracle Net Servicesでは、ネットワークの問題を理解し、テストして、解決するための手段が提供されています。Oracle Databaseには、ネットワークの接続と問題をテストおよび診断するための、ユーティリティおよびログ・ファイルとトレース・ファイルが含まれます。TNSPINGおよびTRCROUTEユーティリティは接続をテストします。ログ・ファイルとトレース・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、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できます。

一般的なテスト手順とネットワーク・エラー、および問題を解決する手順の概要について説明します。また、より複雑なネットワークの問題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も学習します。

- [自動診断リポジトリの理解](#)
- [Oracle Net Servicesの診断](#)
- [Oracle Net Servicesの最も一般的なエラー・メッセージの解決](#)
- [Oracle Net Servicesのトラブルシューティングのヒント](#)
- [TNS-12154エラーのトラブルシューティングの例](#)
- [Oracle Net Servicesのエラー情報のロギング](#)
- [Oracle Net Servicesのエラー情報のトレース](#)
- [Oracleサポート・サービスへの連絡](#)

親トピック: [Oracle Net Servicesのテストおよびトラブルシューティング](#)

16.1 自動診断リポジトリの理解

[自動診断リポジトリ\(ADR\)](#)は、システム全体のトレースおよびロギング用中央リポジトリです。このリポジトリは、ネットワーク・トレースおよびロギング情報などの診断情報を蓄積するための、ファイルベースの階層データ・ストアです。

ADRホームは、Oracle製品のインスタンスに割り当てられるADRディレクトリのユニットです。各データベース・インスタンスに独自のADRホームが割り当てられます。同様に、各リスナー、Oracle Connection Managerおよびクライアント・インスタンスにも独自のADRホームが割り当てられます。

プロセス障害の場合は、インシデントが生成されます。インシデント・ダンプ・ファイルは、ADR_BASE/ADR_HOME/incident/ディレクトリにあり、ORACLE_BASE変数が設定されている場合、ADR_BASEはデフォルトでORACLE_BASEになります。変数が設定されていない場合、ADR_BASEはORACLE_HOME/logになります。ADR_BASEは任意の場所に設定できます。

インシデント・ダンプ・ファイルの場所は、プロセス・トレース・ファイル内でわかります。

ADRホームの場所は、ADRの基本ディレクトリから始まる次のパスで指定されます。

```
diag/product_type/product_id/instance_id
```

[表16-1](#)では、Oracle Net Listenerインスタンスのパス・コンポーネントの値を示します。

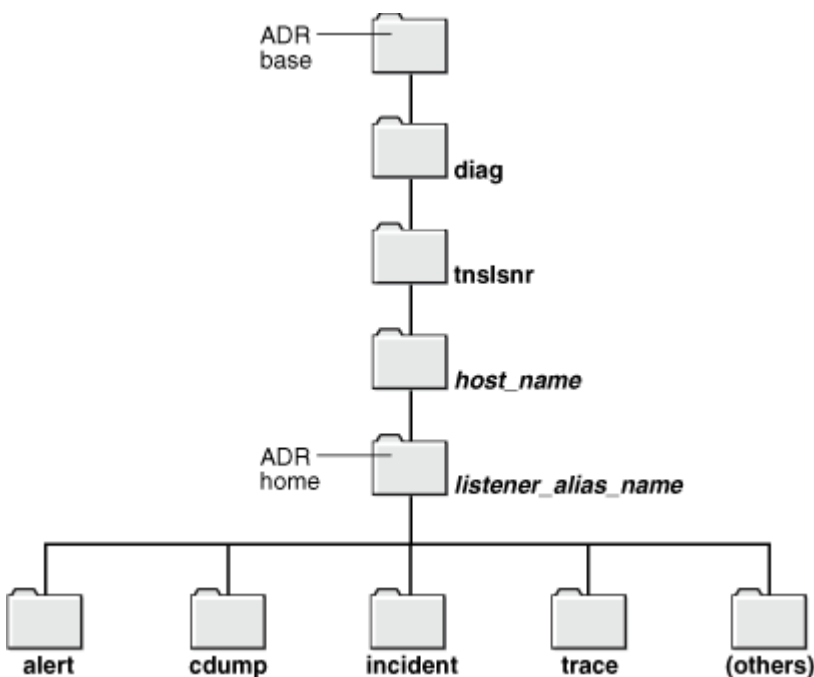
表16-1 Oracle Net ListenerインスタンスのADRホームのパス・コンポーネント

パス・コンポーネント	Oracle Net Listenerの値
------------	-----------------------

パス・コンポーネント	Oracle Net Listenerの値
product_type	tnslsnr
product_id	ホスト名
instance_id	リスナーの別名

[図16-1](#)では、Oracle Net ListenerインスタンスのADRのディレクトリ階層を示しています。その他のOracle製品またはコンポーネント(Oracle Automatic Storage Management(Oracle ASM)またはOracle Databaseなど)に対するその他のADRホームが、この階層内の同じADRベースの下に存在する場合があります。

図16-1 Oracle Net Listenerインスタンスのディレクトリ構造



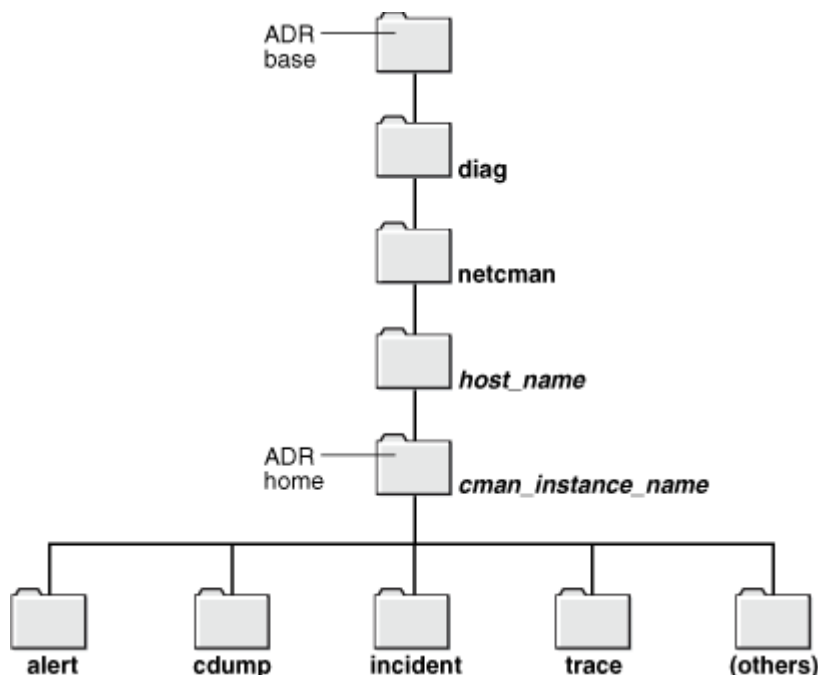
[表16-2](#)では、Oracle Connection Managerインスタンスのパス・コンポーネントの値を示します。

表16-2 Oracle Connection ManagerインスタンスのADRホームのパス・コンポーネント

パス・コンポーネント	Oracle Connection Managerの値
product_type	netcman
product_id	ホスト名
instance_id	Oracle Connection Manager インスタンス名

[図16-2](#)では、Oracle Connection ManagerインスタンスのADRのディレクトリ階層を示しています。その他のOracle製品またはコンポーネント(Oracle ASMまたはOracle Databaseなど)に対するその他のADRホームが、この階層内の同じADRベースの下に存在する場合があります。

図16-2 Oracle Connection Managerインスタンスのディレクトリ構造



ADRホーム・ディレクトリ内のサブディレクトリでは、各インスタンス(データベース、リスナー、Oracle Connection Managerまたはクライアント)により診断データが格納されます。[表16-3](#)は、前の図で示したすべてのサブディレクトリと、その内容のリストです。

表16-3 ADRホームのサブディレクトリ

サブディレクトリ名	目次
alert	log.xml という名前の XML 形式のログ。
cdump	コア・ファイル。
incident	複数のサブディレクトリがあり、各サブディレクトリには特定のインシデントの名前が付けられ、そのインシデントのみに関するダンプが含まれます。
trace	バックグラウンド・プロセスとサーバー・プロセスのトレース・ファイル、SQLトレース・ファイル、および alert ディレクトリの log.xml ファイルのテキスト・バージョン
(others)	ADR ホームのその他のサブディレクトリには、インシデント・パッケージ、状態モニター・レポートなどの情報が格納されます。

ADR_BASEディレクトリは、1つ以上のADRホームが置かれる物理的な場所です。概念的には、これはADRのルート・ディレクトリです。

非ADR (つまりDIAG_ADR_ENABLEDパラメータがOFFに設定されている)の診断およびトレースの方法が現在も主流であり適用可能ですが、ADRが有効である場合、このパラメータは無視されます。ADRはデフォルトで有効になります。

診断パラメータは次の構成ファイルにあります。

- sqlnet.ora (クライアント用)
- listener.ora (リスナー用)
- cman.ora (Oracle Connection Manager用)

表16-4では、ADRベースの診断と非ADRベースの診断の両方で使用されるsqlnet.oraファイル内の、診断パラメータの使用について比較しています。

表16-4 sqlnet.oraファイルの診断パラメータの比較

パラメータ	DIAG_ADR_ENABLE D=ON	DIAG_ADR_ENABLE D=OFF
ADR_BASE	有効	無効
TRACE_LEVEL_CLIENT	有効	有効
TRACE_LEVEL_SERVER	有効	有効
TRACE_DIRECTORY_CLIENT	無効	有効
TRACE_FILE_CLIENT	無効	有効
TRACE_UNIQUE_CLIENT	無効	有効
LOG_DIRECTORY_CLIENT	無効	有効
LOG_FILE_CLIENT	無効	有効
LOG_DIRECTORY_SERVER	無効	有効
TRACE_DIRECTORY_SERVER	無効	有効
TRACE_FILE_SERVER	無効	有効

表16-5では、ADRベースの診断と非ADRベースの診断の両方で使用されるlistener.oraファイル内の、診断パラメータの使用について比較しています。

表16-5 listener.oraファイルの診断パラメータの比較

パラメータ	DIAG_ADR_ENABLE D=ON	DIAG_ADR_ENABLE D=OFF
ADR_BASE_listener_name	有効	無効

パラメータ	DIAG_ADR_ENABLE D=ON	DIAG_ADR_ENABLE D=OFF
LOGGING_listener_name	有効	有効
TRACE_LEVEL_listener_name	有効	有効
TRACE_TIMESTAMP_listener_name	有効	有効
LOG_DIRECTORY_CLIENT_listener_name	無効	有効
LOG_FILE_CLIENT_listener_name	無効	有効
TRACE_DIRECTORY_CLIENT_listener_name	無効	有効
TRACE_FILELEN_listener_name	無効	有効
TRACE_FILENO_listener_name	無効	有効

[表16-6](#)では、ADRベースの診断と非ADRベースの診断の両方で使用されるcman.oraファイル内の、診断パラメータの使用について比較しています。

表16-6 cman.oraファイルの診断パラメータの比較

パラメータ	DIAG_ADR_ENABLE D=ON	DIAG_ADR_ENABLE D=OFF
ADR_BASE	有効	無効
LOG_LEVEL	有効	有効
TRACE_LEVEL	有効	有効
TRACE_TIMESTAMP	有効	有効
LOG_DIRECTORY	無効	有効
TRACE_DIRECTORY	無効	有効
TRACE_FILELEN	無効	有効

パラメータ	DIAG_ADR_ENABLE D=ON	DIAG_ADR_ENABLE D=OFF
TRACE_FILENO	無効	有効

- [ADRCI: ADRコマンド・インタプリタ](#)

関連項目:

- クライアントADRホームの場所の詳細は、『[Oracle Call Interfaceプログラマーズ・ガイド](#)』を参照してください。
- ADRの詳細は、『[Oracle Database管理者ガイド](#)』を参照してください。
- 診断パラメータの詳細は、『[Oracle Database Net Servicesリファレンス](#)』を参照してください。

親トピック: [Oracle Net Servicesのトラブルシューティング](#)

16.1.1 ADRCI: ADRコマンド・インタプリタ

ADRCIは、故障診断機能インフラストラクチャに使用されるコマンドライン・ツールです。ADRCIでは、次のことができます。

- ADR内で診断データを表示
- インシデントおよび問題に関する情報をZIPファイルにパッケージ化し、Oracleサポート・サービスに転送

診断データには、インシデントおよび問題に関する説明、トレース・ファイル、ダンプ、状態監視レポート、アラート・ログ・エントリなどが含まれます。

ADRCIには豊富なコマンド・セットがあり、対話型モードまたはスクリプト内で使用できます。さらに、ADRCIは、SQL*PlusがSQLコマンドおよびPL/SQLコマンドを使用してスクリプトを実行するのと同じ方法で、ADRCIコマンドのスクリプトを実行できます。

ADRCIを使用してトレース・ファイルを表示するには、コマンドラインでADRCIと入力します。次に示すのは、クライアントの確認に使用される共通のADRCIコマンドです。

クライアント側

```
adrci> SHOW ALERT
adrci> SHOW BASE -product client
adrci> SET BASE -product client
adrci> SHOW TRACEFILE
adrci> SHOW TRACE trace_file.trc
adrci> SHOW SPOOL
```

前述のコマンドで、SHOW ALERTコマンドはlog.xmlファイルをVIなどのテキスト・エディタで表示します。SHOW BASE -product clientコマンドは、クライアントのADR_BASEディレクトリの値を表示します。表示された値は、SET BASEコマンドのclientに使用します。

次に示すのは、サーバーの確認に使用される共通のADRCIコマンドです。

サーバー側

```
adrci> SHOW BASE
adrci> SHOW TRACEFILE
adrci> SHOW TRACE trace_file.trc
```

さらに対象を絞ったOracle Netトレース・ファイル分析を行うには、その他のADRCIコマンド・オプションを使用できます。ヘルプ・ドキュメントにアクセスするには、ADRCIプロンプトでHELPと入力します。

関連項目:

ADRCIの詳細は、『[Oracle Databaseユーティリティ](#)』を参照してください。

親トピック: [自動診断リポジトリの理解](#)

16.2 Oracle Net Servicesの診断

すべての基礎となるエラーは、重大かどうかに関係なく、エラー番号またはエラー・メッセージでOracle Net Servicesによって報告されます。エラー番号およびエラー・メッセージは問題の診断に役立つ情報を提供しますが、常に実際の問題を指しているわけではありません。この項のタスクは、Oracle Net Servicesが正しく機能してい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、エラーが次のどのカテゴリに属しているかを判別する際にも役立ちます。

- 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、各種ネットワーク層を順にテストすることで問題を発見できます。

- [サーバーの問題の診断](#)
- [クライアントの問題の診断](#)

親トピック: [Oracle Net Servicesのトラブルシューティング](#)

16.2.1 サーバーの問題の診断

サーバーの問題の診断を開始するには、次の項目をチェックします。

- 他のシステム(ワークステーション、サーバーなど)はOracle Netを使用してサーバーに接続できる。
- 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ない。

前述の項目に1つでも当てはまる場合は、『[クライアントの問題の診断](#)』に進みます。

不明な場合、またはいずれにも当てはまらない場合、この項のタスクを使用して問題を診断します。サーバー上のOracle Net Servicesを診断するには次の作業を行います。

- [タスク1、データベースの実行状態の検証](#)
- [タスク2、ループバック・テストの実行](#)

タスク 1 データベースの実行状態の検証

データベースが起動していることを確認するため、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および接続します。たとえば：

```
SQLPLUS system
Enter password: password
```

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。

タスク 2 ループバック・テストの実行

ループバック・テストでは、Oracle Net を使用してデータベース・サーバーから Interprocess Communication(IPC)プロトコルを利用せずにサーバー自体に戻ります。多数のネットワーク・プロトコルが、ネットワーク接続をテストする方法を用意しています。PING ユーティリティは、TCP/IP ネットワークで使用できます。ループバックが正常に終了すると、Oracle Net がデータベース・サーバーで機能していることが確認できます。

次の手順では、サーバーからデータベースに[ループバック・テスト](#)を実行する方法について説明します。

1. listener.ora、tnsnames.ora および sqlnet.ora の各ファイルが、[「ローカライズされた管理の使用」](#)で説明しているとおり、正しい場所にあることを確認します。
2. Oracle Net Manager を起動します。

関連項目:

[「Oracle Net Manager を使用した Oracle Net Services の構成」](#)

3. ナビゲータで、「ディレクトリ」または「ローカル」オプションを展開します。
4. 「サービス・ネーミング」を展開し、使用可能なネットワーク・サービスとデータベースの名前を表示します。
5. ネットワーク・サービス名またはデータベース・サービスを選択します。
6. 「ツール」を選択し、ネット・サービスのテストを選択します。

このテストでは、リスナーとデータベース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、[「Oracle Net ListenerとOracle Database サーバーの起動」](#)を参照してコンポーネントを起動してください。

テスト中、ステータスとテスト結果を示す「接続テスト」ダイアログ・ボックスが表示されます。テストが成功すると、次のメッセージが表示されます。

```
The connection test was successful.
```

テストが成功した場合は、ステップ [7](#) に進んでください。

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。

- a. 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、「テスト」をクリックします。
 - b. 「ログインの変更」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、「テスト」をクリックします。
 - c. ループバック・テストが正常に終了する場合は、[「クライアントの問題の診断」](#)に進みます。
 - d. ループバック・テストでエラーが発生する場合、Oracle サポート・サービスに問い合わせてください。
7. 「閉じる」をクリックして「接続テスト」ダイアログ・ボックスを閉じます。

親トピック: [Oracle Net Servicesの診断](#)

16.2.2 クライアントの問題の診断

次の文の少なくとも1つを検証します。これはクライアントの問題かどうかを判定するのに役立ちます。

- 接続が機能していることを示すデータベース・サーバーでのループバック・テストが正常終了した。
- 他のコンピュータがOracle Net Servicesを使用して同じデータベースに接続している。
- このコンピュータに変更を加える(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。

次の手順では、クライアントでの診断の実行方法を説明します。

1. データベース・サーバーにインストールされているのと同じプロトコル・サポートがインストールされていることを確認します。
UNIXおよびUNIXプラットフォームの場合は、ADAPTERSユーティリティを使用してプロトコル・サポートを確認できます。
データベース・サーバーで、ORACLE_HOME/binディレクトリから次のコマンドを実行して、oracle実行可能プログラムにリンクするプロトコル・サポート、ネーミング・メソッドおよびセキュリティ・オプションを表示します。

```
adapters ./oracle
```

次に、adaptersユーティリティによる出力例を示します。

Installed Oracle Net transport protocols are:

- IPC
- BEQ
- TCP/IP
- SSL
- RAW
- SDP/IB

Installed Oracle Net naming methods are:

- Local Naming (tnsnames.ora)
- Oracle Directory Naming
- Oracle Host Naming
- NIS Naming

Installed Oracle Advanced Security options are:

- RC4 40-bit encryption
- RC4 56-bit encryption
- RC4 128-bit encryption
- RC4 256-bit encryption
- DES40 40-bit encryption
- DES 56-bit encryption
- 3DES 112-bit encryption
- 3DES 168-bit encryption
- AES 128-bit encryption
- AES 192-bit encryption
- AES 256-bit encryption
- MD5 crypto-checksumming
- SHA crypto-checksumming (for FIPS)
- SHA-1 crypto-checksumming
- Kerberos v5 authentication
- RADIUS authentication

クライアントで、ORACLE_HOME/binディレクトリからadaptersコマンドを実行して、構成されたOracle protocol support、ネーミング・メソッドおよびセキュリティ・オプションを表示します。次に、ADAPTERSユーティリティによる出力例を示します。

Installed Oracle Net transport protocols are:

- IPC
- BEQ
- TCP/IP
- SSL
- RAW
- SDP/IB
- Exadirect

Installed Oracle Net naming methods are:

- Local Naming (tnsnames.ora)
- Oracle Directory Naming
- Oracle Host Naming

Installed Oracle Advanced Security options are:

- RC4 40-bit encryption
- RC4 56-bit encryption
- RC4 128-bit encryption
- RC4 256-bit encryption
- DES40 40-bit encryption
- DES 56-bit encryption
- 3DES 112-bit encryption
- 3DES 168-bit encryption
- AES 128-bit encryption
- AES 192-bit encryption

AES 256-bit encryption
MD5 crypto-checksumming
SHA-1 crypto-checksumming
Kerberos v5 authentication
RADIUS authentication

ノート:

- DES、DES40、3DES 112、3DES 168、RC4 40、RC4 56、RC4 128、RC4 256 および MD5 アルゴリズムは、このリリースでは非推奨です。



より強力なアルゴリズムを使用するように Oracle Database 環境を移行するには、My Oracle Support ノート [2118136.2](#) で説明されているパッチをダウンロードしてインストールします。

- RAW は Oracle Net で使用される内部プロトコルです。

関連項目:

adaptersユーティリティの詳細は、[『Oracle Database管理者リファレンス』](#)を参照してください。

2. 基底のネットワーク・トランスポートの基本的な接続性をチェックします。Oracle Netテクノロジーは、正常な接続のために基底のネットワークに依存しています。

表16-7 ネットワーク・トランスポートの基本的な接続性の確認

プロトコル	確認方法
TCP/IP	クライアントからデータベース・サーバーに端末エミュレーションまたはファイル転送ユーティリティ(PING、FTP、TELNET)を使用する。
名前付きパイプ	<ul style="list-style-type: none">● 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。● 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。

3. Oracle Net Foundationレイヤーおよび適切なOracle protocol supportが存在することを確認するため、すべてのクライアント用Oracle Net Services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・コンピュータの正しい場所に、tnsnames.oraファイルとsqlnet.oraファイルが存在していることを確認します。

関連項目:

[「ローカル管理の使用」](#)

選択したOracle Databaseに接続している他の稼働中のクライアント・コンピュータがある場合は、既存のファイルの

バックアップをとって、稼働中のクライアント・コンピュータの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. Oracle Net Foundationレイヤーをテストします。次のコマンドを使用してSQL*Plusに接続してテストできます。

```
SQLPLUS user/password@connect_string
```

ノート:



TNSPING ユーティリティは使用しないでください。TNSPING ユーティリティは TCP/IP ping ユーティリティと同様に機能します。これはソケットの作成やオープンは行わず、リスナーとは接続しません。表示されるのは、リスナーがデータベース・サーバー上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。
 - a. [「Oracle Net Servicesのエラー情報のトレース」](#)の項の説明に従って、トレースを使用する。
 - b. Oracleサポート・サービスのWebサイトで、発生したエラーに関する診断情報を確認する。
 - c. Oracleサポート・サービスに問い合わせてください。

親トピック: [Oracle Net Servicesの診断](#)

16.3 Oracle Net Servicesの最も一般的なエラー・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、Oracle Net Servicesからのネットワーク・サービスに依存しているSQL*Plusなどのアプリケーションは、通常、エラー・メッセージを生成します。

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。

特定のエラー・メッセージの情報に関しては、Oracleエラー・ツールのoerrを使用し、任意のコマンドラインで次のコマンドを入力します。

```
oerr code error_number
```

前述のコマンドで、codeはORAやTNSなどのメッセージのタイプ、error_numberはエラー・メッセージに関連付けられている番号です。

- [ORA-03113: 通信チャネルでend-of-fileが検出されました](#)
- [ORA-12154: TNS: 指定された接続識別子を解決できませんでした](#)
- [ORA-12170: TNS: 接続タイムアウトが発生しました](#)
- [TNS-12500またはORA-12500: TNS: リスナーが専用サーバー・プロセスの起動に失敗しました](#)
- [ORA-12514: TNS: リスナーは現在、接続記述子でリクエストされているサービスを認識していません](#)
- [ORA-12520: TNS: リスナーは、リクエストしたサーバー・タイプに使用可能なハンドラを検出できませんでした](#)
- [ORA-12521: TNS: リスナーは接続記述子でリクエストされたインスタンスを現在認識していません](#)
- [ORA-12525: TNS: リスナーは、クライアントのリクエストを許容時間内に受信しませんでした](#)
- [ORA-12533: TNS: ADDRESSパラメータが不正です](#)
- [TNS-12540またはORA-12540: TNS:内部制限を超えましたおよびTNS-00510: TNS:内部制限を超えまし](#)

た

- [TNS-12541またはORA-12541: TNS: リスナーがありません](#)
- [TNS-12549またはORA-12549: TNS:オペレーティング・システム・リソース割当て制限を超えましたおよびTNS-00519: TNS:オペレーティング・システム・リソース割当て制限を超えました](#)
- [TNS-12560またはORA-12560: TNS: プロトコル・アダプタ・エラーが発生しました](#)
- [ディレクトリ・ネーミング・エラー](#)

関連項目:

エラー・メッセージの完全なリストは、『[Oracle Databaseエラー・メッセージ](#)』を参照してください。

親トピック: [Oracle Net Servicesのトラブルシューティング](#)

16.3.1 ORA-03113: 通信チャネルでend-of-fileが検出されました

このメッセージは、データベース・サーバーでエラーが発生したことを示します。

サーバー上のalert_sid.logファイルをチェックします。通信チャネルで予期しないファイルの終了処理が行われました。通信リンクが少なくとも一時的に停止していることを示しているか、サーバーが停止したことを示している可能性があります。再送カウントを変更する必要があります。

親トピック: [Oracle Net Servicesの最も一般的なエラー・メッセージの解決](#)

16.3.2 ORA-12154: TNS: 指定された接続識別子を解決できませんでした

このメッセージは、データベースまたは他のサービスへの接続が接続識別子を使用して要求されたが、指定された接続識別子は、構成されているどのネーミング・メソッドを使用しても接続記述子に解決できなかったことを示します。たとえば、使用された接続識別子のタイプがネットワーク・サービス名の場合、そのネットワーク・サービス名がネーミング・メソッド・リポジトリで見つからないか、またはリポジトリが見つからないか、アクセスできません。

次のステップを実行します。

1. sqlnet.oraファイルのNAMES.DIRECTORY_PATHパラメータにリストされているネーミング・アダプタのタイプを調べます。構成されていない場合は、adaptersコマンドを使用して使用中のアダプタを特定します。adaptersの例を次に示します。

```
$ adapters
...
Installed Oracle Net naming methods are:

Local Naming (tnsnames.ora)
Oracle Directory Naming
Oracle Host Naming
NIS Naming
```

接続文字列で指定されているネットワーク・サービス名が、少なくとも1つのネーミング・メソッドに定義されている必要があります。

2. 各アダプタの解決パスで問題の可能性を調べます。たとえば、接続文字列で指定されている名前が正しく完全である

ことを確認し、必要であればネットワーク・サービスの完全な名前を使用します。

前述のステップに加えて、ネーミング・メソッドに応じて各ステップを実行します。

- [ローカル・ネーミング・メソッドの使用時のORA-12154エラー](#)
- [ディレクトリ・ネーミング・メソッドの使用時のORA-12154エラー](#)
- [簡易接続ネーミング・メソッドの使用時のORA-12154エラー](#)
- [外部ネーミング・メソッドの使用時のORA-12154エラー](#)

親トピック: [Oracle Net Servicesの最も一般的なエラー・メッセージの解決](#)

16.3.2.1 ローカル・ネーミング・メソッドの使用時のORA-12154エラー

ローカル・ネーミング・メソッドを使用する場合は、次のステップを実行します。

1. tnsnames.oraファイルが正しい場所に存在することを確認します。場所は、ORACLE_HOME/network/adminディレクトリまたはTNS_ADMIN環境変数で指定されているディレクトリです。
2. tnsnames.oraファイルに接続文字列で指定されている名前のエントリがあることを確認します。名前が単純でsqlnet.oraファイルにNAMES_DEFAULT_DOMAINがない場合、またはネットワーク・サービス名が完全修飾名である場合は、このネットワーク・サービス名がtnsnames.oraファイルの名前と厳密に一致している必要があります。接続文字列のネットワーク・サービス名が単純な場合は、sqlnet.oraファイルのNAMES_DEFAULT_DOMAINパラメータを調べます。その値が、接続文字列で指定されているネットワーク・サービス名に追加されます。この完全修飾名が、tnsnames.oraファイル内のエントリである必要があります。
3. ログイン・ダイアログ・ボックスから接続する場合は、接続するネットワーク・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。
4. クライアントのトレースをアクティブにして、操作を繰り返します。

親トピック: [ORA-12154: 指定された接続識別子を解決できませんでした](#)

16.3.2.2 ディレクトリ・ネーミング・メソッドの使用時のORA-12154エラー

ディレクトリ・ネーミング・メソッドを使用する場合は、次のステップを実行します。

1. ldap.oraファイルが正しい場所に存在することを確認します。次のディレクトリで、この順番に、ldap.oraファイルが検索されます。見つかったldap.oraが使用されます。
 - 環境変数TNS_ADMINで指定されているディレクトリ。
 - ORACLE_HOME/network/adminディレクトリ。
 - 環境変数LDAP_ADMINで指定されているディレクトリ。
 - ORACLE_HOME/ldap/adminディレクトリ。
2. 次のように、ldap.oraファイルで定義されているパラメータが正しいことを確認します。
 - DIRECTORY_SERVERSパラメータで、1つ以上の有効なLDAPサーバーの正しいホストとポートが定義されている。
 - DEFAULT_ADMIN_CONTEXTパラメータで、ネットワーク・サービス・エントリを含んでいる必要のある、このディレクトリ内のOracleコンテキストの場所が定義されている。

ldap.oraファイルが存在しない場合、これらのパラメータは自動検出を使用して解決されます。

3. LDAPサーバーのホストとポートがDNSで定義されていることを確認します。
4. ディレクトリでデフォルトのOracleコンテキストが定義されていることを確認します。
5. ldapsearchユーティリティまたはディレクトリ管理ツールを使用して、DEFAULT_ADMIN_CONTEXTパラメータの値で指定されている場所のOracleコンテキスト内にネットワーク・サービス・オブジェクトが存在することを確認します。

親トピック: [ORA-12154: 指定された接続識別子を解決できませんでした](#)

16.3.2.3 簡易接続ネーミング・メソッドの使用時のORA-12154エラー

指定されているホスト名が正しいこと、およびローカル・ホスト・ファイルやDNSなどのローカル・ホスト名解決サービスでホスト名が定義されていることを確認します。

親トピック: [ORA-12154: 指定された接続識別子を解決できませんでした](#)

16.3.2.4 外部ネーミング・メソッドの使用時のORA-12154エラー

外部ネーミング・メソッドを使用する場合は、次のステップを実行します。

1. tnsnamesマップ用のNISファイルが正しく設定されていることを確認します。
2. [「ローカル・ネーミング・メソッドの使用時のORA-12154エラー」](#)で説明されているように、ネットワーク・サービス名が tnsnamesのエントリと一致することを確認します。

関連項目:

- エラーのトラブルシューティングの詳細は、[「TNS-12154エラーのトラブルシューティングの例」](#)を参照してください
- 構成ファイルの場所に関する情報は、[「ローカル管理の使用」](#)を参照してください
- ネーミングの情報については、[「ネーミング・メソッドの構成」](#)を参照してください

親トピック: [ORA-12154: 指定された接続識別子を解決できませんでした](#)

16.3.3 ORA-12170: TNS: 接続タイムアウトが発生しました

このメッセージは、クライアントがsqlnet.oraファイルのSQLNET.INBOUND_CONNECT_TIMEOUTパラメータで指定した時間内で接続の確立と認証の完了に失敗したことを示します。このエラーは、ネットワークまたはシステムの遅延が原因の可能性がありますが、または、悪質なクライアントがデータベース・サーバーでサービス拒否攻撃しようとしていることを示す可能性もあります。

- 特定の環境におけるシステムあるいはネットワークの通常の遅延によるエラーである場合は、次のステップを実行します。
 1. トレースをオンにして、タイムアウトしたクライアントを特定します。

関連項目:

[「Oracle Net Servicesのエラー情報のトレース」](#)

2. sqlnet.oraで、SQLNET.INBOUND_CONNECT_TIMEOUT、SQLNET.SEND_TIMEOUT、SQLNET.RECV_TIMEOUTのいずれかのパラメータをより大きい値に再構成します。

- 不当なクライアントが疑われる場合は、次のステップを実行します。
 1. そのクライアントへのアクセス制限を行います。sqlnet.oraファイルでアクセス権のパラメータを構成できます。
 2. データベース・サーバー上のsqlnet.logファイルでクライアントのIPアドレスを探索して、ソースを識別します。IPアドレスは偽装できることに注意してください。

たとえば、次のsqlnet.logの例は、クライアントのIPアドレス192.0.2.35を示しています。

```
Fatal NI connect error 12170.
VERSION INFORMATION:
TNS for Linux: Version 18.0.0.0.0
Oracle Bequeath NT Protocol Adapter for Linux: Version 18.0.0.0.0
TCP/IP NT Protocol Adapter for Linux: Version 18.0.0.0.0
Time: 03-JAN-2018 13:51:12
Tracing to file: /ora/trace/svr_13279.trc
Tns error struct:
  nr err code: 0
  ns main err code: 12637
  TNS-12637: Packet receive failed
  ns secondary err code: 12604
  nt main err code: 0
  nt secondary err code: 0
  nt OS err code: 0
Client address: (ADDRESS=(PROTOCOL=tcp) (HOST=192.0.2.35) (PORT=52996))
```

データベース・サーバーでIPアドレスを取り出す前にタイム・アウトが発生した場合は、リスナー・トレースを有効にして、要求を送信したクライアントを特定します。

関連項目:

- [SQLNET.INBOUND_CONNECT_TIMEOUTパラメータの設定の詳細は、「権限のないユーザーによるリソース使用の制限」を参照してください](#)
- [「データベース・アクセス制御の設定」](#)
- [「Oracle Net Servicesのエラー情報のトレース」](#)

親トピック: [Oracle Net Servicesの最も一般的なエラー・メッセージの解決](#)

16.3.4 TNS-12500またはORA-12500: TNS: リスナーが専用サーバー・プロセスの起動に失敗しました

これらのメッセージは、リスナーがOracleプログラムの起動に失敗したことを示します。考えられる原因は次のとおりです。

- 1ユーザーに許可されるプロセスの最大数を超過した
- リスナーがOracleプログラムの実行権限を持っていない
- 関連付けられたMicrosoft Windowsサービスが起動されていない

状況によっては、これらのエラーは次のエラーと同じ条件下で発生する可能性があります。

- TNS-12549またはORA-12549
- TNS-00519

- TNS-12540またはORA-12540
- TNS-00510
- TNS-12560またはORA-12560

適切な処置を実行します。

- データベース初期化ファイルのPROCESSESパラメータを現在より大きな値に設定して、プロセス数を増やします。
- listener.logファイルで詳しいエラー・スタック情報をチェックします。

親トピック: [Oracle Net Servicesの最も一般的なエラー・メッセージの解決](#)

16.3.5 ORA-12514: TNS: リスナーは現在、接続記述子でリクエストされているサービスを認識していません

このメッセージは、リスナーが、データベースまたは他のサービスとの接続を確立するための要求を受信したことを示します。リスナーが受信した接続記述子には、リスナーに動的に登録されなかったかまたはリスナーに静的に構成されていないサービス(通常はデータベース・サービス)の名前が指定されていました。これはリスナーが起動した後、データベース・インスタンスがリスナーに登録されるまでの間に発生した場合など、一時的な状況と考えられます。

次のステップを実行します。

1. しばらく待ってから、もう一度接続します。
2. リスナー制御ユーティリティのSTATUSまたはSERVICESコマンドを実行して、リスナーが現在認識しているサービスを確認します。
3. 接続記述子のSERVICE_NAMEパラメータに、リスナーが認識しているサービス名が指定されていることを確認します。
4. listener.logファイル内のイベントを確認します。

関連項目:

- [「リスナーの現在のステータスの確定」](#)および[「リスナーのサービスの監視」](#)
- [「リスナー・ログ・ファイルの分析」](#)

親トピック: [Oracle Net Servicesの最も一般的なエラー・メッセージの解決](#)

16.3.6 ORA-12520: TNS: リスナーは、リクエストしたサーバー・タイプに使用可能なハンドラを検出できませんでした

このメッセージは、クライアントに要求されたサービス・ハンドラのタイプが不適切であるか、要求されたSERVICE_NAME/INSTANCE_NAMEパラメータに登録されていないか、データベース・インスタンスがリスナーに登録されていないことを示します。

問題がサービス・ハンドラのタイプの不正にあると思われる場合は、次のステップを実行します。

1. 接続記述子に(server=value)が設定されている場合は、データベースに適したタイプのサービス・ハンドラに値が設定されていることを確認します。適切な値は、専用サーバーの場合はdedicated、ディスクパッチャの場合はsharedです。リ

リスナー制御ユーティリティのSERVICESコマンドを使用すると、現在リスナーに登録されているサービス・ハンドラを確認できます。

2. sqlnet.oraファイルでUSE_DEDICATED_SERVERパラメータがONに設定されている場合は、データベースが専用サーバーを使用するように構成されていることを確認します。そのように構成されていない場合は、このパラメータをOFFに設定します。
3. データベース・インスタンスが稼働中であることを確認します。インスタンスが稼働していない場合は、リスナーに登録できるようにインスタンスを起動します。

関連項目:

サービス・ハンドラの詳細は、[「リスナーのサービスの監視」](#)を参照してください

親トピック: [Oracle Net Servicesの最も一般的なエラー・メッセージの解決](#)

16.3.7 ORA-12521: TNS: リスナーは接続記述子でリクエストされたインスタンスを現在認識していません

このメッセージは、接続記述子のインスタンス名が不適切であるか、データベース・インスタンスがリスナーに登録されていないことを示します。

次のステップを実行します。

1. 接続記述子で指定されたサービス名が正しいことを確認します。
2. データベース・インスタンスが稼働中であることを確認します。インスタンスが稼働していない場合は、リスナーに登録できるようにインスタンスを起動します。リスナー制御ユーティリティのSERVICESコマンドを使用すると、現在リスナーに登録されているインスタンスを確認できます。

関連項目:

SERVICESコマンドの詳細は、[「リスナーのサービスの監視」](#)を参照してください

親トピック: [Oracle Net Servicesの最も一般的なエラー・メッセージの解決](#)

16.3.8 ORA-12525: TNS: リスナーは、クライアントのリクエストを許容時間内に受信しませんでした

このメッセージは、クライアントがlistener.oraファイルのINBOUND_CONNECT_TIMEOUT_listener_nameパラメータで指定された時間内に接続要求を完了するのに失敗したことを示します。このエラーは、ネットワークまたはシステムの遅延が原因の可能性があります。または、悪質なクライアントがリスナーでサービス拒否攻撃しようとしていることを示す可能性もあります。

- 特定の環境におけるシステムあるいはネットワークの通常の遅延によるエラーである場合は、listener.oraでINBOUND_CONNECT_TIMEOUT_listener_nameパラメータをより大きい値に再構成します。
- 不当なクライアントが疑われる場合は、次のステップを実行します。

1. ソースを識別するために、listener.logファイルでクライアントのIPアドレスを探索します。IPアドレスは偽装

できることに注意してください。

たとえば、次の listener.log ファイルの例は、クライアントの IP アドレス 192.0.2.35 を示しています。

```
03-MAY-2012 16:42:35 * <unknown connect data> *
(ADDRESS=(PROTOCOL=tcp) (HOST=192.0.2.35) (PORT=53208)) * establish *
<unknown sid> * 12525
TNS-12525: TNS:listener has not received client's request in time
allowed
TNS-12604: TNS: Application timeout occurred
```

2. そのクライアントへのアクセス制限を行います。sqlnet.ora ファイルでアクセス権のパラメータを構成できます。

関連項目:

- INBOUND_CONNECT_TIMEOUT_listener_name パラメータの設定の詳細は、[「権限のないユーザーによるリソース使用の制限」](#)を参照してください
- [「データベース・アクセス制御の設定」](#)

親トピック: [Oracle Net Servicesの最も一般的なエラー・メッセージの解決](#)

16.3.9 ORA-12533: TNS: ADDRESSパラメータが不正です

このメッセージは、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ないことを示します。

プロトコル・アドレスを訂正します。

ノート:



このエラーは、tnsnames.ora ファイルを手作業で編集したために発生することがよくあります。tnsnames.ora ファイルは、Oracle Enterprise Manager Cloud Control または Oracle Net Manager を使用してのみ編集してください。

関連項目:

正しいプロトコル構文は、[『Oracle Database Net Servicesリファレンス』](#)を参照してください。

親トピック: [Oracle Net Servicesの最も一般的なエラー・メッセージの解決](#)

16.3.10 TNS-12540またはORA-12540: TNS:内部制限を超えましたおよび TNS-00510: TNS:内部制限を超えました

これらのメッセージは内部制限を超えたことを示します。次の制限が考えられます。

- Oracle Netが同時に処理できるオープン接続の数
- 同時に使用できるメモリー・バッファの数

- 特定のデータベース・インスタンスで許可されるプロセスの数

最初の2つは、ハード上の制限を示しています。3つ目の制限は、データベース初期化ファイルのPROCESSESパラメータを大きな値に設定することで増やすことができる例です。この場合は、TNS-12500またはORA-12500エラーも戻されます。状況によっては、これらのエラーはTNS-12549またはORA-12549、およびTNS-00519のエラーと同じ条件下で発生する可能性があります。

オープン接続がクローズするのを待ち、再試行します。それでもエラーが発生する場合は、sqlnet.logファイルまたはlistener.logファイルで詳しいエラー・スタック情報をチェックします。

親トピック: [Oracle Net Servicesの最も一般的なエラー・メッセージの解決](#)

16.3.11 TNS-12541またはORA-12541: TNS: リスナーがありません

これらのメッセージは、リスナーが実行されていないため、接続要求が完了できなかったことを示します。

次の手順を実行します。

- 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。
- 要求で指定されているアドレスでリスナーが実行していることを確認します。
- 要求で指定されているホストとポートでリスナーがリスニングしていることを確認します。
- クライアントがリスナーを指していることを確認します。

親トピック: [Oracle Net Servicesの最も一般的なエラー・メッセージの解決](#)

16.3.12 TNS-12549またはORA-12549: TNS:オペレーティング・システム・リソース割当て制限を超えましたおよびTNS-00519: TNS:オペレーティング・システム・リソース割当て制限を超えました

これらのメッセージは、オペレーティング・システムで設定されている割当てまたはハード・リミットを超えたことを示します。

考えられる制限には次のものがあります。

- 1ユーザーに許可されるプロセスの最大数
- ページング・スペースに起因するオペレーティング・システムの性能低下

適切な処置を実行します。

- データベース初期化ファイルのPROCESSESパラメータを現在より大きな値に設定して、プロセス数を増やします。
- sqlnet.logファイルまたはlistener.logファイルで、オペレーティング・システムのエラー・コードなどの詳しいエラー・スタック情報を調べ、超過している割当て制限を特定します。

親トピック: [Oracle Net Servicesの最も一般的なエラー・メッセージの解決](#)

16.3.13 TNS-12560またはORA-12560: TNS: プロトコル・アダプタ・エラーが発生しました

これらのメッセージは、特定のプロトコルを使用した場合にエラーが発生したことを示します。このエラーは、ADDRESSパラメータ

の構成が不適切な場合、または基礎となるプロトコルやオペレーティング・システム・インタフェースからエラーが戻された場合に発生します。

状況によっては、これらのエラーはTNS-00510、TNS-00519、TNS-12540、ORA-12540、TNS-12549またはORA-12549のエラーと同じ条件下で発生する可能性があります。

これらのエラーは、Microsoft Windowsシステムでのみ発生します。次の手順を実行します。

1. Microsoft Windowsの「スタート」メニューから、「ファイル名を指定して実行」を選択します。
2. 「名前」フィールドにMSCONFIGと入力します。
3. 「サービス」タブに移動します。
4. 無効になっている場合は、OracleServicesidを有効にします。
5. コンピュータを再起動します。
6. Oracleサービスが開始したことを確認します。

親トピック: [Oracle Net Servicesの最も一般的なエラー・メッセージの解決](#)

16.3.14 ディレクトリ・ネーミング・エラー

ディレクトリ・サーバーのデータベース・サービス名またはネットワーク・サービス名のエントリに対する接続性エラーに関連するディレクトリ・ネーミングの問題の解決には、データの分析が必要です。ディレクトリ・サーバーに含まれるデータの分析には、ldifwriteコマンドライン・ツールを使用します。ldifwriteツールはOracle Internet Directoryツールです。

ldifwriteツールを使用して、ディレクトリ・サーバーにある情報のすべてまたは一部をLDIFに変換できます。ldifwriteツールでは、指定した識別名(DN)の下のすべてのエントリを対象にサブツリー検索を実行します(DN自体も対象です)。

ldifwriteツールの構文は次のとおりです。

```
ldifwrite -c net_service_name/database_service -b base_DN -f ldif_file
```

[表16-8](#)では、ldifwriteツールの引数とその説明を示しています。

表16-8 ldifwriteの引数

引数	説明
-c net_service_name/database_service	ディレクトリ・サーバーに接続するためのネットワーク・サービス名またはデータベース・サービス名。
-b base_DN	LDIF フォーマットで書き出されるサブツリーのベース。
-f ldif_file	出力ファイル名。

次の例は、dc=us, dc=example, dc=comの下のすべてのディレクトリ・ネーミング・エントリをoutput1.ldiファイルに書き込みます。

```
ldifwrite -c ldap -b "dc=us, dc=example, dc=com" -f output.ldi
```

ノート:



ldap.ora ファイルをチェックして、base_DN 値を確認します。これは、ldap.ora ファイルの DEFAULT_ADMIN_CONTEXT エントリと同じです。

親トピック: [Oracle Net Servicesの最も一般的なエラー・メッセージの解決](#)

16.4 Oracle Net Services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を診断するときは、次のことが役に立ちます。

- 構成時に、サーバー・コンピュータの名前ではなくノードまたはネットワーク・アドレスを使用する。これにより内部ルックアップ問題がなくなり、接続が多少早くなります。
- TCP/IPアドレスを使用している場合は、ホスト名ではなくIPアドレスを使用する。たとえば、tnsnames.oraファイルの (HOST=server_name)の行を(HOST=192. 0. 2. 5)のようにIPアドレスに変更します。
- [\[タスク2、ループバック・テストの実行\]](#)の説明に従って、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。
- クライアントとサーバーの間のシステムをチェックする。Wide Area Network(WAN)の場合は、正しく作動していない可能性がある中間システムを識別します。すべてのコンピュータが正常である場合、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あるかどうか検証する。タイミングの問題は、クライアント・ログ・ファイルのORA-12535のエラーに関係しています。

タイミング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INBOUND_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。

- 障害が発生しているOracleアプリケーションを判断する。SQL*Plusは作動するが、CASEツールは作動しない場合があります。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。
- [Oracle Net Servicesのトラブルシューティングのためのチェック項目](#)

親トピック: [Oracle Net Servicesのトラブルシューティング](#)

16.4.1 Oracle Net Servicesのトラブルシューティングのためのチェック項目

問題を診断する際に役立つチェック項目を次に示します。

- すべてのコンピュータに問題があるのか、1台のコンピュータのみか
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、機能するコンピュータと機能しないコンピュータがある場合、可能であれば、ネットワーク・ケーブルを取り替えて2番目のクライアントで問題が発生するかどうかを確認します。問題が発生した場合、問題はクライアント/サーバー接続に関するもので、クライアント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接続の種類(X.25、ISDNまたは専用回線など)は何か

SnifferやLANアナライザは、断続的な接続エラーを特定し、タイムアウトおよび再送パケットの検出を行うのに便利です。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。

親トピック: [Oracle Net Servicesのトラブルシューティングのヒント](#)

16.5 TNS-12154エラーのトラブルシューティングの例

この項では、TNS-12154エラーのソリューションを提供します。TNS-12154エラーは、SQL*Netがtnsnames.oraファイルまたはその他のネーミング・アダプタの接続に指定された接続識別子を検出できない場合に発生します。

問題を解決する前に、tnsnames.oraファイルとsqlnet.oraファイルを印刷または表示しておくことが便利です。両方を参照するため、これらのファイルを同時に参照できるようにすると便利です。

この例では、tnsnames.oraファイルおよびsqlnet.oraファイルは、クライアント・システムのデフォルトのネットワーク管理ディレクトリにあります。

tnsnames.oraファイルおよびsqlnet.oraファイルが次の例のようになっていることを確認します。

[例16-1](#)では、tnsnames.oraファイルの例を示します。

例16-1 tnsnames.oraの例

```
DEV1.WORLD =
  (DESCRIPTION =
    (ADDRESS =
      (PROTOCOL = TCP)
      (HOST = 192.0.2.56)
      (PORT = 1521)
    )
    (CONNECT_DATA =
      (SERVICE_NAME = sales.example.com)
    )
  )
```

[例16-2](#)では、sqlnet.oraファイルの例を示します。

例16-2 sqlnet.oraの例

```
TRACE_LEVEL_CLIENT = OFF
SQLNET.AUTHENTICATION_SERVICES = (NONE)
NAMES.DIRECTORY_PATH = (TNSNAMES)
AUTOMATIC_IPC = OFF
```

[例16-1](#)の別名はDEV1.WORLDです。ただし、[例16-2](#)にはNAMES.DEFAULT_DOMAIN=WORLDパラメータが存在しません。この問題を解決するには、sqlnet.oraファイルの任意の場所にNAMES.DEFAULT_DOMAIN = WORLDパラメータを追加します。ファイルを保存し、接続を再度試行します。

親トピック: [Oracle Net Servicesのトラブルシューティング](#)

16.6 Oracle Net Servicesのエラー情報のロギング

Oracle Net Services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・メッセージについて、管理者用の追加情報を提供します。ログ・ファイルのエラー・スタックは、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情報と大半のリスナー制御コマンドが含まれています。

- [Oracle Netエラー・スタック](#)
- [Oracle Net Servicesのログ・ファイル名](#)
- [Oracle Networkログ・ファイルのセグメンテーション](#)
Oracle Net Listener、Oracle Connection Manager、グローバル・サービス・マネージャなどのOracleネットワーク・コンポーネントには、テキスト・ログ・ファイルの最大サイズと数を構成できます。
- [ロギング・パラメータについて](#)
- [構成ファイルのロギング・パラメータの設定](#)
- [制御ユーティリティ実行時のロギング設定](#)
- [ログ・ファイルの使用](#)
- [リスナー・ログの分析](#)
リスナー・ログ・ファイルには、監査証跡、サービス登録関連イベント、ダイレクト・ハンドオフ・イベント、Oracle Notification Service (ONS)ノードダウン・イベントのサブスクリプションおよびOracle Clusterware通知に関する情報が記録されます。
- [Oracle Connection Managerログの分析](#)
Oracle Connection Manager (CMAN)は、指定されたログ・ディレクトリにcman_alias.logファイルを生成します。このログ・ファイルには、CMANリスナー、ゲートウェイ、CMADMINプロセスおよびアラートに関連するメッセージが記録されます。

親トピック: [Oracle Net Servicesのトラブルシューティング](#)

16.6.1 Oracle Net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。

[表16-9](#)に、エラー・スタック・コンポーネントの説明を示します。

表16-9 エラー・スタック・コンポーネント

エラー・スタック・コンポーネント	説明
NI	ネットワーク・インタフェース。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 Oracle Net 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。
NS	ネットワーク・セッション(1 次および 2 次層)。これらのレイヤーは NI から要求を受け取り、次のようなコンピュータ・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。

エラー・スタック・コンポーネント	説明
	<ul style="list-style-type: none"> ● 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)。
NA	ネットワーク認証。このレイヤーは認証要求および暗号化要求のネゴシエーションを行います。
NT	ネットワーク・トランスポート(1 次、2 次およびオペレーティング・システム層)。これらのレイヤーは、Oracle Net Foundation レイヤー機能を業界標準プロトコルにマップします。

- [エラー・スタック・メッセージの理解](#)

親トピック: [Oracle Net Servicesのエラー情報のロギング](#)

16.6.1.1 エラー・スタック・メッセージの理解

クライアント・アプリケーションのユーザーがOracle Net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、次のコマンドを入力します。

```
SQLPLUS scott@example.com
Enter password: password
```

コマンドを入力すると、次のエラーが表示されます。

```
ORA-12543: TNS:Unable to connect to destination
```

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、[例16-3](#)にあるように、sqlnet.logファイルにORA-12543のエラーに対応するエラー・スタックが含まれています。

例16-3 sqlnet.logファイル

```
Fatal OSN connect error 12543, connecting to:
  (DESCRIPTION=(CONNECT_DATA=(SID=trace) (CID=(PROGRAM=
    (HOST=lala) (USER=scott))) (ADDRESS_LIST=(ADDRESS=
    (PROTOCOL=ipc) (KEY=trace)) (ADDRESS=(PROTOCOL=tcp)
    (HOST=lala) (PORT=1521))))))
VERSION INFORMATION:
TNS for Linux:
Oracle Bequeath NT Protocol Adapter for Linux:
```

```

Unix Domain Socket IPC NT Protocol Adaptor for Linux:
TCP/IP NT Protocol Adapter for Linux:
  Tracing to file: /home/db_tracefiles/trace_admin.trc
  Tns error struct:
    TNS-12543: TNS:unable to connect to destination
    ns main err code: 12541
    TNS-12541: TNS:no listener
    ns secondary err code: 12560
    nt main err code: 511
    TNS-00511: No listener
    nt secondary err code: 61
    nt OS err code: 0

```

親トピック: [Oracle Netエラー・スタック](#)

16.6.2 Oracle Net Servicesのログ・ファイル名

Oracle Net Servicesの各コンポーネントは、それぞれ独自のログ・ファイルを生成します。デフォルトのADRを使用するときは、ログ・ファイル名は適切なalertディレクトリにあるlog.xmlです。[表16-10](#)は、デフォルトのログ・ファイル名と、ADR/diag/instance_name/traceディレクトリのログ・ファイルを生成するコンポーネントのリストです。

16-10 ADR使用時のログ・ファイル名

コンポーネント	ログ・ファイル
リスナー	listener.log
クライアントまたはデータベース・サーバー	sqlnet.log
Oracle Connection Manager リスナー	instance-name_pid.log
Oracle Connection Manager CMGW (Oracle Connection Manager Gateway) プロセス	instance-name_cmgw_pid.log
Oracle Connection Manager CMADMIN(Oracle Connection Manager Administration) プロセス	instance-name_cmadmin_pid.log
Oracle Connection Manager アラート・ログ	instance-name_alert.log

親トピック: [Oracle Net Servicesのエラー情報のロギング](#)

16.6.3 Oracle Networkログ・ファイルのセグメンテーション

Oracle Net Listener、Oracle Connection Manager、グローバル・サービス・マネージャなどのOracleネットワーク・コンポーネントには、テキスト・ログ・ファイルの最大サイズと数を構成できます。

この機能により、特にクラウド環境でログ・ファイルの管理が向上します。これはADRの専用機能で、テキスト・ログ・ファイルと

XMLログ・ファイルの両方に適用できます。

親トピック: [Oracle Net Servicesのエラー情報のロギング](#)

16.6.4 ロギング・パラメータについて

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、[表16-11](#)に示される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。

表16-11 ログ・パラメータの位置

ネットワーク・コンポーネント	構成ファイル
Oracle Connection Manager プロセス	cman.ora
リスナー	listener.ora
クライアント	sqlnet.ora
データベース・サーバー	sqlnet.ora

ノート:



ADR_ENABLED を ON に設定すると、すべてのロギング・パラメータは ADR によって設定されます。Oracle Net Manager を使用してパラメータを変更することはできません。

この項では、次の項目について説明します。

- [sqlnet.oraのログ・パラメータ](#)
- [listener.oraのログ・パラメータ](#)
- [cman.oraのログ・パラメータ](#)

関連項目:

これらのパラメータの詳細は、『[Oracle Database Net Servicesリファレンス](#)』を参照してください。

親トピック: [Oracle Net Servicesのエラー情報のロギング](#)

16.6.4.1 sqlnet.oraのログ・パラメータ

[表16-12](#)では、sqlnet.oraファイルで設定できるログ・パラメータの設定を示します。

表16-12 sqlnet.oraのログ・パラメータ

Oracle Net

sqlnet.oraのパラメータ Managerのフィールド 説明

ADR_BASE	このパラメータは手動で設定する必要があります。	ADR_BASE パラメータは、インシデントのトレースおよびロギングが格納される基本ディレクトリを指定します。 DIAG_ADR_ENABLED が ON に設定されている場合、このパラメータを使用します。
DIAG_ADR_ENABLED	このパラメータは手動で設定する必要があります。	DIAG_ADR_ENABLED パラメータは、ADR トレースが有効かどうかを示します。 DIAG_ADR_ENABLED パラメータが OFF に設定されている場合は、非 ADR ファイルのトレースが使用されます。
LOG_DIRECTORY_CLIENT	「クライアント情報: ログ・ディレクトリ」	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。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。このパラメータは、ADR_ENABLED が ON に設定されている場合は無効になります。
LOG_DIRECTORY_SERVER	「サーバー情報: ログ・ディレクトリ」	データベース・サーバー・ログ・ファイルの作成先となるディレクトリ。デフォルトのサーバー・ディレクトリは ORACLE_HOME/network/log です。このパラメータは、ADR_ENABLED が ON に設定されている場合は無効になります。
LOG_FILE_CLIENT	「クライアント情報: ログ・ファイル」	クライアントのログ・ファイルの名前。デフォルトのログ・ファイル名は sqlnet.log です。
LOG_FILE_SERVER	このパラメータは手動で設定する必要があります。	データベース・サーバーのログ・ファイルの名前。デフォルトのログ・ファイル名は sqlnet.log です。

親トピック: [ロギング・パラメータについて](#)

16.6.4.2 listener.oraのログ・パラメータ

[表16-13](#)では、listener.oraファイルで設定できるログ・パラメータの設定を示します。

表16-13 listener.oraファイルのログ・パラメータ

	Oracle Net Managerのフィールド 説明	
listener.oraのパラメータ		
ADR_BASE_listener_name	このパラメータは手動で設定する必要があります。	ADR_BASE_listener_name パラメータは、インシデントのトレースおよびロギングが格納される基本ディレクトリを

listener.oraのパラメータ	Oracle Net Managerのフィールド	説明
	す。	指定します。 DIAG_ADR_ENABLED_listener_name が ON に設定されている場合、このパラメータを使用します。
DIAG_ADR_ENABLED_listener_name	このパラメータは手動で設定する必要があります。	DIAG_ADR_ENABLED_listener_name パラメータは、ADR トレースが有効かどうかを示します。 DIAG_ADR_ENABLED_listener_name が OFF に設定されている場合、非 ADR ファイルのトレースが使用されます。
LOG_DIRECTORY_listener_name	ログ・ファイル	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。デフォルトのディレクトリは ORACLE_HOME/network/log、デフォルトのファイル名は listener.log です。これらのパラメータは、ADR_ENABLED が ON に設定されている場合は無効になります。
LOG_FILE_listener_name		

親トピック: [ロギング・パラメータについて](#)

16.6.4.3 cman.oraのログ・パラメータ

[表16-14](#)では、cman.oraファイルで設定できるログ・パラメータの設定を示します。

表16-14 cman.oraファイルのログ・パラメータ

cman.oraのパラメータ	説明
ADR_BASE	ADR_BASE パラメータは、インシデントのトレースおよびロギングが格納される基本ディレクトリを指定します。 DIAG_ADR_ENABLED が ON に設定されている場合、このパラメータを使用します。
DIAG_ADR_ENABLED	DIAG_ADR_ENABLED パラメータは、ADR トレースが有効かどうかを示します。 DIAG_ADR_ENABLED パラメータが OFF に設定されている場合は、非 ADR ファイルのトレースが使用されます。
EVENT_GROUP	ロギングされるイベント・グループ。複数のイベントを指定するには、カンマ区切りリストを

cman.oraのパラメータ	説明
	<p>使用します。このパラメータには次の値を使用できます。</p> <ul style="list-style-type: none"> ● INIT_AND_TERM: 初期化と終了 ● MEMORY_OPS: メモリー操作 ● CONN_HDLG: 接続処理 ● PROC_MGMT: プロセス管理 ● REG_AND_LOAD: 登録とロード更新 ● WAKE_UP: CMADMIN 起動キュー関連のイベント ● TIMER: ゲートウェイのタイムアウト ● CMD_PROC: コマンド処理 ● RELAY: 接続制御ブロックに関連付けられたイベント
LOG_DIRECTORY	<p>ログ・ファイルの作成先となるディレクトリ。</p> <p>デフォルトのディレクトリは ORACLE_HOME/network/log です。</p> <p>このパラメータは、ADR_ENABLED が ON に設定されている場合は無効になります。</p>
LOG_LEVEL	<p>ロギングのレベル。次の 4 つのレベルがサポートされています。</p> <ul style="list-style-type: none"> ● off(デフォルト): ロギングなし ● user: ユーザー・ログ情報 ● admin: 管理ログ情報 ● support: Oracle サポート・サービス情報

親トピック: [ロギング・パラメータについて](#)

16.6.5 構成ファイルのロギング・パラメータの設定

sqlnet.oraファイルのロギング・パラメータはOracle Net Managerで設定し、listener.oraファイルのロギング・パラメータはOracle Enterprise Manager Cloud ControlまたはOracle Net Managerで設定します。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。

- [Oracle Net Managerを使用したsqlnet.oraファイルのパラメータの設定](#)
- [Oracle Enterprise Manager Cloud Controlを使用したlistener.oraファイルのパラメータの設定](#)
- [Oracle Net Managerを使用したlistener.oraファイルのパラメータの設定](#)

親トピック: [Oracle Net Servicesのエラー情報のロギング](#)

16.6.5.1 Oracle Net Managerを使用したsqlnet.oraファイルのパラメータの設定

次の手順では、sqlnet.oraファイルでロギング・パラメータを設定する方法について説明します。

1. Oracle Net Managerを起動します。

関連項目:

[「Oracle Net Managerを使用したOracle Net Servicesの構成」](#)

2. ナビゲータ・ペインで、「ローカル」見出しから「プロファイル」を展開します。
3. 右ペインのリストから、「一般」を選択します。
4. 「ロギング」タブをクリックします。
5. 設定を指定します。
6. 「ファイル」メニューから「ネットワーク構成の保存」を選択します。

ログ・ファイルの名前はsqlnet.logです。

親トピック: [構成ファイルのロギング・パラメータの設定](#)

16.6.5.2 Oracle Enterprise Manager Cloud Controlを使用したlistener.oraファイルのパラメータの設定

次の手順では、Oracle Enterprise Manager Cloud Controlを使用してlistener.oraファイルでロギング・パラメータを設定する方法について説明します。

1. Oracle Enterprise Manager Cloud Controlの「Net Services管理」ページにアクセスします。

関連項目:

[「Oracle Enterprise Manager Cloud Controlを使用したOracle Net Servicesの構成」](#)

2. 「管理」リストから「リスナー」を選択し、構成ファイルの場所を含むOracleホームを選択します。
3. 「実行」をクリックし、「リスナー」ページを表示します。
4. リスナーを選択し、「編集」をクリックし、「リスナーの編集」ページを表示します。
5. 「ロギングとトレース」タブをクリックします。
6. 設定を指定します。
7. 「OK」をクリックします。

ログ・ファイルの名前はlistener.logです。

親トピック: [構成ファイルのロギング・パラメータの設定](#)

16.6.5.3 Oracle Net Managerを使用したlistener.oraファイルのパラメータの設定

次の手順では、Oracle Net Managerを使用してlistener.oraファイルでロギング・パラメータを設定する方法について説明します。

1. Oracle Net Managerを起動します。

関連項目:

[「Oracle Net Managerを使用したOracle Net Servicesの構成」](#)

2. ナビゲータ・ペインで、「ローカル」見出しから「リスナー」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、「一般」を選択します。
5. 「ロギングとトレース」タブをクリックします。
6. 設定を指定します。
7. 「ファイル」メニューから「ネットワーク構成の保存」を選択します。

ログ・ファイルの名前はlistener.logです。

関連項目:

[『Oracle Database Net Servicesリファレンス』](#)

親トピック: [構成ファイルのロギング・パラメータの設定](#)

16.6.6 制御ユーティリティ実行時のロギング設定

ロギングは、制御ユーティリティの実行時に設定できます。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ず、設定は制御ユーティリティのセッションでのみ有効です。

次の設定を制御ユーティリティに対して設定できます。

- リスナーの場合は、リスナー制御ユーティリティでSET LOG_FILEコマンドおよびSET LOG_DIRECTORYコマンドを使用します。
- Oracle Connection Managerの場合は、Oracle Connection Manager制御ユーティリティのSET LOG_DIRECTORY、SET LOG_LEVELおよびSET EVENTコマンドを使用します。

ノート:

ADR_ENABLED を ON に設定すると、すべてのロギング・パラメータは ADR によって設定されます。Oracle

Connection Manager を使用してパラメータを変更することはできません。

関連項目:

[『Oracle Database Net Servicesリファレンス』](#)

親トピック: [Oracle Net Servicesのエラー情報のロギング](#)

16.6.7 ログ・ファイルの使用

次の手順では、ログ・ファイルを使用してネットワーク・エラーを診断する方法について説明します。

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。通常、これがログ・ファイルの最後のエントリです。
2. ファイルの最も下から順にたどり、エラー・レポートで0(ゼロ)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ログ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コマンドを繰り返します。

親トピック: [Oracle Net Servicesのエラー情報のロギング](#)

16.6.8 リスナー・ログの分析

リスナー・ログ・ファイルには、監査証跡、サービス登録関連イベント、ダイレクト・ハンドオフ・イベント、Oracle Notification Service (ONS)ノードダウン・イベントのサブスクリプションおよびOracle Clusterware通知に関する情報が記録されます。

- [リスナー・ログ監査証跡](#)
リスナー・ログ・ファイル内の監査証跡情報を使用すると、ネットワーク使用状況統計、クライアント接続リクエスト、リスナー制御ユーティリティによって発行されたコマンドなどを分析できます。
- [リスナー・サービス登録イベント](#)
リスナー・ログ・ファイル内のサービス登録イベント情報を使用すると、インスタンスのサービス名、インスタンス名、サービス・ハンドラ、ロード情報、動的リスニング・エンドポイントなどの登録関連の統計を分析できます。
- [リスナー・ダイレクト・ハンドオフ情報](#)
リスナー・ログ・ファイル内のダイレクト・ハンドオフ情報を使用すると、ディスパッチャへのダイレクト・ハンドオフ・イベントを分析できます。
- [ONSノード停止イベント情報のリスナー・サブスクリプション](#)
ONS構成ファイルが使用可能な場合、リスナーは起動時にOracle Notification Service (ONS)のノード停止イベントをサブスクライブします。リスナー・ログ・ファイル内のONSノード停止イベント情報を使用すると、これらのメッセージを分析できます。
- [リスナーOracle Clusterware通知情報](#)
必要なOracle Clusterwareライブラリがインストールされていて、ホストでOracle Clusterwareが起動されている場合、リスナーは起動プロセスおよび停止プロセスでOracle Clusterwareのステータスを通知します。リスナー・ログ・ファイル内のOracle Clusterware通知情報を使用すると、これらのメッセージを分析できます。

親トピック: [Oracle Net Servicesのエラー情報のロギング](#)

16.6.8.1 リスナー・ログ監査証跡

リスナー・ログ・ファイル内の監査証跡情報を使用すると、ネットワーク使用状況統計、クライアント接続リクエスト、リスナー制御ユーティリティによって発行されたコマンドなどを分析できます。

- [リスナー・ログ監査証跡情報](#)

監査証跡情報は、ネットワーク使用状況、クライアント接続リクエストおよびリスナー制御ユーティリティによって発行されるコマンド(RELOAD、START、STOP、STATUS、SERVICESなど)に関する統計で構成されます。

- [リスナー・ログ監査証跡の書式](#)

これは、リスナー・ログ・ファイルに取得される監査証跡テキスト・フィールドの書式です。

親トピック: [リスナー・ログの分析](#)

16.6.8.1.1 リスナー・ログ監査証跡情報

監査証跡情報は、ネットワーク使用状況、クライアント接続リクエストおよびリスナー制御ユーティリティによって発行されるコマンド(RELOAD、START、STOP、STATUS、SERVICESなど)に関する統計で構成されます。

監査証跡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・アクティビティを確認できます。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。

親トピック: [リスナー・ログ監査証跡](#)

16.6.8.1.2 リスナーのログ監査証跡の書式

これは、リスナー・ログ・ファイルに取得される監査証跡テキスト・フィールドの書式です。

```
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、およびサービス名またはSID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(ゼロ)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。

[例16-4](#)に、RELOADコマンド要求のログ・ファイルの出力例を示します。

例16-4 正常なRELOAD要求に対するリスナー・ログ・イベント

```
14-MAY-2012 00:29:54 *
(connect_data=(cid=(program=) (host=sales-server) (user=jdoe)) (command=reload)
(arguments=64) (service=listener) (version=135290880))
* reload * 0
```

[例16-5](#)に、正常な接続要求のログ・ファイルの出力例を示します。

例16-5 正常な接続要求に対するリスナー・ログ・イベント

```
14-MAY-2012 15:28:58 *
(connect_data=(service_name=sales.us.example.com) (cid=(program=) (host=sales-server)
```

```
(user=jdoe))
* (address=(protocol=tcp) (host=192.0.2.35) (port=41349)) * establish
* sales.us.example.com * 0
```

[例16-6](#)は、ホストsales-serverによるSTATUSコマンドの正常な実行を抜粋したログ・ファイルを示しています。この後、IPアドレス192.0.2.35のクライアントによる接続試行が失敗しています。この接続試行の結果、[ORA-12525: 「TNS: リスナーは、クライアントのリクエストを許容時間内に受信しませんでした」](#)エラー・メッセージが表示されました。このエラーは、クライアントがlistener.oraファイルのINBOUND_CONNECT_TIMEOUT_listener_nameパラメータで指定された時間内に接続要求を完了するのに失敗すると発生します。このクライアントはリスナーにサービス拒否攻撃をしようとした可能性があります。

例16-6 失敗した接続要求に対するリスナー・ログ・イベント

```
03-MAY-2012 16:41:57 *
(CONNECT_DATA=(CID=(PROGRAM=) (HOST=sales-server) (USER=jdoe)) (COMMAND=status)
(ARGUMENTS=64) (SERVICE=LISTENER) (VERSION=153092352)) * status * 0
03-MAY-2012 16:42:35 * <unknown connect data> *
(ADDRESS=(PROTOCOL=tcp) (HOST=192.0.2.35) (PORT=53208)) * establish *
<unknown sid> * 12525
TNS-12525: TNS:listener has not received client's request in time allowed
TNS-12604: TNS: Application timeout occurred
```

関連項目

- [Oracle Databaseエラー・メッセージ・リファレンス](#)

親トピック: [リスナー・ログ監査証跡](#)

16.6.8.2 リスナーのサービス登録イベント

リスナー・ログ・ファイル内のサービス登録イベント情報を使用すると、インスタンスのサービス名、インスタンス名、サービス・ハンドラ、ロード情報、動的リスニング・エンドポイントなどの登録関連の統計を分析できます。

- [リスナーのサービス登録イベント情報](#)
サービス登録情報は、service_registerやservice_update、service_diedなどのデータベース・サービス登録イベントに関連する統計で構成されています。
- [リスナーのサービス登録情報の書式](#)
これは、リスナー・ログ・ファイルで取得されたデータベース・サービス登録イベント・メッセージの書式です。

親トピック: [リスナー・ログの分析](#)

16.6.8.2.1 リスナーのサービス登録イベント情報

サービス登録情報は、service_registerやservice_update、service_diedなどのデータベース・サービス登録イベントに関連する統計で構成されています。

サービス登録を行うと、リスナー登録(LREG)プロセスからリスナーに次の情報が提供されます。

- データベースにおける稼働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
- 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

- 動的リスニング・エンドポイント

次の表で示されている記録済のサービス登録関連イベントは、listener.logファイル内で一覧で示されています。

表16-15 サービス登録イベントのログ情報

イベント	説明
service_register	リスナーがインスタンスについて受信する登録情報。
service_update	特定インスタンスについてリスナーが受信する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)。
service_died	リスナーが LREG で確立できない接続の接続切断情報。インスタンスの登録情報はすべて廃棄されます。LREG 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。

親トピック: [リスナーのサービス登録イベント](#)

16.6.8.2.2 リスナーのサービス登録情報の書式

これは、リスナー・ログ・ファイルで取得されたデータベース・サービス登録イベント・メッセージの書式です。

```
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(ゼロ)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。

例16-7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イベントの後にはクライアント要求を受信できますが、service_diedイベントの後にはクライアント要求を受信できません。

例16-7 サービス登録イベントが記録されたリスナー・ログ

```
-----
14-MAY-2012 15:28:43 * service_register * sales * 0
14-MAY-2012 15:28:43 * service_register * sales * 0
14-MAY-2012 15:28:58 *
(connect_data=(service_name=sales.us.example.com)
(cid=(program=) (host=sales-server) (user=jdoe)))
* (address=(protocol=tcp) (host=192.0.2.35) (port=41349)) * establish
* sales.us.example.com * 0
14-MAY-2012 15:38:44 * service_update * sales * 0
14-MAY-2012 15:38:44 * service_update * sales * 0
14-MAY-2012 15:48:45 * service_update * sales * 0
14-MAY-2012 15:48:45 * service_update * sales * 0
14-MAY-2012 15:50:57 *
(connect_data=(service_name=sales.us.example.com) (cid=(program=)
(host=sales-server) (user=jdoe)))
```

```
* (address=(protocol=tcp) (host=192.0.2.35) (port=41365)) * establish
* sales.us.example.com * 0
14-MAY-2012 15:51:26 * service_died * sales * 12537
14-MAY-2012 15:51:26 * service_died * sales * 12537
14-MAY-2012 15:52:06 *
(connect_data=(service_name=sales.us.example.com)
(cid=(program=) (host=sales-server) (user=jdoe)))
* (address=(protocol=tcp) (host=192.0.2.35) (port=41406)) * establish
* sales.us.example.com * 12514
TNS-12514: TNS:listener could not resolve SERVICE_NAME given in connect descriptor
-----
```

関連項目

- [Oracle Databaseエラー・メッセージ・リファレンス](#)

親トピック: [リスナーのサービス登録イベント](#)

16.6.8.3 リスナーのダイレクト・ハンドオフ情報

リスナー・ログ・ファイル内のダイレクト・ハンドオフ情報を使用すると、ディスパッチャへのダイレクト・ハンドオフ・イベントを分析できます。

これら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
```

ダイレクト・ハンドオフ・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(ゼロ)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。

次の例では、ログ・ファイル内のダイレクト・ハンドオフ・イベントを示します。

例16-8 ダイレクト・ハンドオフのリスナー・ログ・イベント

```
21-MAY-2012 10:54:55 * oracle.aurora.net.SALESHttp2 * handoff * 0
```

親トピック: [リスナー・ログの分析](#)

16.6.8.4 ONSノード停止イベント情報のリスナー・サブスクリプション

ONS構成ファイルを使用できる場合、リスナーは、起動時にOracle Notification Service (ONS)にノード停止イベントをサブスクライブします。リスナー・ログ・ファイル内のONSノード停止イベント情報を使用すると、これらのメッセージを分析できます。

このサブスクリプションにより、ノード停止イベント通知をONSから受信した場合、リスナーは、影響を受けたサービスを削除できます。リスナーは、イベント通知に非同期サブスクリプションを使用します。

ONSデーモンがホストで実行されていない場合など、サブスクリプションが完了しなかった場合、次の警告メッセージがSTATUSコマンドごとにリスナー・ログ・ファイルに記録されます。

```
WARNING: Subscription for node down event still pending
```

サブスクリプションの保留中は、リスナーはONSイベントを受信できません。それ以外の場合、他のリスナー機能が影響を受けることはありません。

親トピック: [リスナー・ログの分析](#)

16.6.8.5 リスナーのOracle Clusterware通知情報

必要なOracle Clusterwareライブラリがインストールされていて、ホストでOracle Clusterwareが起動されている場合、リスナーは起動プロセスおよび停止プロセスでOracle Clusterwareのステータスを通知します。リスナー・ログ・ファイル内のOracle Clusterware通知情報を使用すると、これらのメッセージを分析できます。

Oracle Clusterware (次のログ・メッセージにCRSとして表示)への通知に成功すると、リスナーはイベントをログに記録します。通知が失敗した場合は、メッセージは記録されません。

```
Listener completed notification to CRS on start  
Listener completed notification to CRS on stop
```

親トピック: [リスナー・ログの分析](#)

16.6.9 Oracle Connection Managerログの分析

Oracle Connection Manager (CMAN)は、指定されたログ・ディレクトリにcman_alias.logファイルを生成します。このログ・ファイルには、CMANリスナー、ゲートウェイ、CMADMINプロセスおよびアラートに関連するメッセージが記録されます。

アラート用ログ・エントリには、重大なエラーが時系列にリストされます。重大なエラーをロギングするだけでなく、インスタンスの起動と停止に関する情報も記録されています。また、セッションの開始と終了時におけるすべての構成パラメータの値も記録されます。

各ログ・エントリは、タイムスタンプとイベントで構成されます。イベントを次のカテゴリにロギングするようにcman.oraファイルを構成できます。

- 初期化と終了
- メモリー操作
- 接続処理
- プロセス管理
- 登録とロード更新
- CMADMIN起動キュー関連のイベント
- ゲートウェイのタイムアウト
- コマンド処理
- 接続制御ブロックに関連付けられたイベント

SET EVENTコマンドを使用して、ロギングするイベントのタイプを指定します。

表16-16 CMADMINおよびゲートウェイ・ログのエントリの詳細

ログ・エントリ	イベント	説明
CMADMIN	Failed to get procedure ID	CMADMIN に接続されていた CMCTL セッションが切断されました。
CMADMIN	GMON attributes validated	情報メッセージです。CMADMIN の起動に必要なパラメータが正しく指定されています。
CMADMIN	Invalid connect data	未知のクライアントが CMADMIN に接続しようとしています。サービス拒否攻撃と考えられます。
CMADMIN	No connect data	未知のクライアントが CMADMIN に接続しようとしています。サービス拒否攻撃と考えられます。
Gateway	Connected to monitor	ゲートウェイが CMADMIN に接続されました。
Gateway	Housekeeping	情報メッセージです。ゲートウェイ・プロセスの内部ハウスキーピングが適切に行われています。ゲートウェイ・プロセスは CMADMIN プロセスに正しく接続されています。
Gateway	Idle timeout	cman. ora ファイルに指定された時間より長い間アイドル状態が続いたため、接続が切断されています。
Gateway	Out of connection control block (CCB)	CMADMIN が接続要求を処理できません。次の 2 つの理由が考えられます。 <ul style="list-style-type: none"> ● CMADMIN とリスナー間のロード更新が不完全 ● CMADMIN に直接接続しようとしている侵入者がいる(サービス拒否攻撃の可能性)
Gateway	Session timeout	cman. ora ファイルに指定されたセッション・タイムアウト時間を超過したため、接続が切断されています。
Gateway	State change from Empty to Init	ゲートウェイからの状態変更メッセージ。Init 状態になると、ゲートウェイは内部データの初期化を開始します。
Gateway	State change from Init to Ready	ゲートウェイからの状態変更メッセージ。ゲートウェイは Ready 状態になると、クライアントからの接続を受け入れます。

例16-9 CMADMINログ・メッセージの例

```

(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:46:40) (EVENT=Parameter list)
(listener_address=(address=(protocol=tcp) (host=sales1) (port=1574)))
(aso_authentication_filter=OFF)
(connection_statistics=ON)
(log_directory=/home/user/network/admin/log)
(log_level=support)
(max_connections=256)
(idle_timeout=5)
(inbound_connect_timeout=0)
(session_timeout=20)
(outbound_connect_timeout=0)
(max_gateway_processes=1)
(min_gateway_processes=1)
(password=OFF)
(trace_directory=/home/user/network/admin/log)
(trace_level=off)
(trace_timestamp=OFF)
(trace_filelen=0)
(trace_fileno=0)
)
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:46:40) (EVENT=Shared Memory Size)
(BYTES=82524))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:46:40) (EVENT=GMON Attributes validated)
(Type=Information))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:46:40) (EVENT=NS Listen Successful)
((ADDRESS=(PROTOCOL=tcp) (HOST=sales1) (PORT=1574))))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:46:44) (EVENT=Received command) (CMD=verify
password))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:46:44) (EVENT=Received command)
(CMD=version))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:46:44) (EVENT=Received command)
(CMD=show status))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:46:44) (EVENT=Failed to get procedure id))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:49:12) (EVENT=Received command) (CMD=verify
password))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:49:15) (EVENT=Failed to get procedure id))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:49:29) (EVENT=Received command) (CMD=verify
password))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:49:46) (EVENT=Failed to get procedure id))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:49:50) (EVENT=Received command) (CMD=verify
password))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:49:50) (EVENT=Received command)
(CMD=probe monitor))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:49:50) (EVENT=Received command)
(CMD=shutdown normal))
-----

```

例16-10 ゲートウェイ・ログ・メッセージの例

```

-----
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:46:41) (EVENT=NS Initialised))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:46:41) (EVENT=Memory Allocated)
(BYTES=1024))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:46:41) (EVENT=NCR Initialised))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:46:41) (EVENT=Connected to Monitor))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:46:41) (EVENT=State Change from Empty to
Init))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:46:41) (EVENT=Memory Allocated)

```

```
(BYTES=251904))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:46:41) (EVENT=Memory Allocated)
(BYTES=2048))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:46:41) (EVENT=CCB Initialised))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:46:41) (EVENT=Started Listening))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:46:41) (EVENT=State Change from Init to
Ready))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:46:47) (EVENT=Housekeeping))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:48:06) (EVENT=Ready) (CONN NO=0))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:48:06) (EVENT=Ready) (CONN NO=0))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:48:07) (EVENT=Housekeeping))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:48:12) (EVENT=Housekeeping))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:48:13) (EVENT=Idle Timeout) (CONN NO=0))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:48:17) (EVENT=Housekeeping))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:48:22) (EVENT=Housekeeping))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:48:25) (EVENT=Ready) (CONN NO=0))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:48:25) (EVENT=Ready) (CONN NO=0))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:48:27) (EVENT=Housekeeping))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:48:30) (EVENT=Idle Timeout) (CONN NO=0))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:48:32) (EVENT=Housekeeping))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:48:37) (EVENT=Housekeeping))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:48:42) (EVENT=Ready) (CONN NO=0))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:48:42) (EVENT=Ready) (CONN NO=0))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:48:42) (EVENT=Housekeeping))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:48:47) (EVENT=Housekeeping))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:48:52) (EVENT=Housekeeping))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:48:57) (EVENT=Housekeeping))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:49:02) (EVENT=Session Timeout) (CONN NO=0))
(LOG_RECORD=(TIMESTAMP=08-MAY-2022 08:49:02) (EVENT=Housekeeping))
-----
```

関連項目

- [『Oracle Database Net Servicesリファレンス・ガイド』](#)

親トピック: [Oracle Net Servicesのエラー情報のロギング](#)

16.7 Oracle Net Servicesの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Oracle Net Services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識別できます。

ノート:



トレース機能には大量のディスク領域を消費するため、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、必要なときのみトレースを行うようにしてください。

この項では、次の項目について説明します。

- [Oracle Net Servicesのトレース・ファイル名の理解](#)
- [トレース・パラメータの設定](#)

- [制御ユーティリティ実行時のトレース設定](#)
- [Oracle Net Servicesのトレース・ファイルの評価](#)
- [トレース・アシスタントを使用したトレース・ファイルの検証](#)

親トピック: [Oracle Net Servicesのトラブルシューティング](#)

16.7.1 Oracle Net Servicesのトレース・ファイル名の理解

Oracle Net Servicesの各コンポーネントは、それぞれ独自のトレース・ファイルを生成します。[表16-17](#)は、デフォルトのトレース・ファイル名と、トレース・ファイルを生成するコンポーネントのリストです。

表16-17 トレース・ファイル名

トレース・ファイル	コンポーネント
instance-name_pid.trc	Oracle Connection Manager リスナー
instance-name_cmgtw_pid.trc	Oracle Connection Manager CMGTW プロセス
instance-name_cmadmin_pid.trc	Oracle Connection Manager CMADMIN プロセス
listener.trc	リスナー
sqlnet.trc	クライアント
svr_pid.trc	データベース・サーバー
tnsping.trc	TNSPING ユーティリティ

親トピック: [Oracle Net Servicesのエラー情報のトレース](#)

16.7.2 トレース・パラメータの設定

トレース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、トレースを制御するパラメータは、[表16-18](#)に示される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。

表16-18 トレース・パラメータの位置

構成ファイル	コンポーネント
cman.ora	Oracle Connection Manager プロセス
listener.ora	リスナー

構成ファイル	コンポーネント
sqlnet.ora	クライアント
	データベース・サーバー
	TNSPING ユーティリティ

この項では、次の項目について説明します。

- [cman.oraのトレース・パラメータ](#)
- [listener.oraのトレース・パラメータ](#)
- [sqlnet.oraのトレース・パラメータ](#)
- [構成ファイルのトレース・パラメータの設定](#)

関連項目:

これらのパラメータの詳細は、『[Oracle Database Net Servicesリファレンス](#)』を参照してください。

親トピック: [Oracle Net Servicesのエラー情報のトレース](#)

16.7.2.1 cman.oraのトレース・パラメータ

[表16-19](#)では、cman.oraファイルでOracle Connection Managerに設定できるトレース・パラメータの設定を示します。

表16-19 cman.oraのトレース・パラメータ

cman.oraのパラメータ	説明
TRACE_DIRECTORY	トレース・ファイルの作成先となるディレクトリ。 デフォルトのディレクトリは ORACLE_HOME/network/trace です。
TRACE_FILELEN	トレース・ファイルのサイズ(KB 単位)。指定のサイズに到達すると、トレース情報は次のファイルに書き込まれます。ファイルの数は、TRACE_FILENO パラメータで指定します。
TRACE_FILENO	トレース用トレース・ファイルの数。このパラメータを TRACE_FILELEN パラメータとともに設定すると、トレース・ファイルは循環方式で使用されます。最初のファイルがいっぱいになると、2 番目のファイルが使用され、順番にファイルが使用されます。最後のファイルがいっぱいになると、最初のファイルが再利用され、再度、順番にファイルが使用されます。 トレース・ファイル名は、順序番号によって識別されます。たとえば、このパラメータを 3 に設定すると、ゲートウェイ・プロセス用の Oracle Connection Manager のトレース・ファイル名は、instance-name_cmgw1_pid.trc、instance-name_cmgw2_pid.trc

cman.oraのパラメータ	説明
	<p>および instance-name_cm gw3_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・イベントの前には、そのファイルの順序番号が付きます。</p>
TRACE_LEVEL	<p>トレース機能によりリスナーで記録される詳細レベル。トレース・レベルは、0 (ゼロ)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、次のいずれかの値を指定します。</p> <ul style="list-style-type: none"> ● off(0 と同じ)はトレースを実行しません。 ● 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。 ● 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。 ● support (16 と同じ)は、Oracle サポート・サービスによるトラブルシューティングのトレース情報を提供します。 <p>Oracle Connection Manager では、リスナー、ゲートウェイおよび CMADMIN の各プロセスで、Linux と Microsoft Windows の両方にトレース・ファイルが作成されます。</p>
TRACE_TIMESTAMP	<p>TRACING パラメータが有効な場合、リスナー・トレース・ファイルのすべてのトレース・イベントに対して、タイムスタンプ(dd-mon-yyyy hh:mi:ss:mi の書式)が作成されます。</p>

親トピック: [トレース・パラメータの設定](#)

16.7.2.2 listener.oraのトレース・パラメータ

[表16-20](#)では、listener.oraファイルで設定できるリスナー用のトレース・パラメータの設定を示します。

表16-20 listener.oraのトレース・パラメータ

Oracle Enterprise Manager Cloud Control/Oracle Net Managerの		
listener.oraのパラメータ	フィールド	説明
TRACE_LEVEL_listener_na	「トレース・レベルを選択します」/「トレース・レベ	トレース機能によりリスナーで記録される詳細レベル。トレース・レベルは、0 (ゼロ)から16 までの数値(0 はトレースなし、16 は最

Oracle Enterprise
Manager Cloud
Control/Oracle
Net Managerの

listener.oraのパラメータ	フィールド	説明
me	ル]	<p>大限のトレースを表す)か、次のいずれかの値を指定します。</p> <ul style="list-style-type: none"> ● off(0と同じ)はトレースを実行しません。 ● 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。 ● 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。 ● support (16と同じ)は、Oracle サポート・サービスによるトラブルシューティングのトレース情報を提供します。
TRACE_DIRECTORY_listener_name	トレース・ファイル	<p>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。デフォルトのディレクトリは ORACLE_HOME/network/trace、デフォルトのファイル名は listener.trc です。</p>
TRACE_FILE_listener_name		
TRACE_FILEAGE_listener_name	このパラメータは手動で設定する必要があります。	<p>リスナー・トレース・ファイルの最大保持期間(分単位)。保持期間制限に達すると、トレース情報は次のファイルに書き込まれます。ファイルの数は、TRACE_FILENO_listener_name パラメータで指定します。</p>
TRACE_FILEAGE_SERVER	このパラメータは手動で設定する必要があります。	<p>トレース・ファイルの最大保持期間を分単位で指定します。保持期間制限に達すると、トレース情報は次のファイルに書き込まれます。ファイルの数は、TRACE_FILENO_SERVER パラメータで指定します。</p>
TRACE_FILELEN_listener_name	このパラメータは手動で設定する必要があります。	<p>リスナー・トレース・ファイルのサイズ(KB 単位)。指定のサイズに到達すると、トレース情報は次のファイルに書き込まれます。ファイルの数は、TRACE_FILENO_listener_name パラメータで指定します。</p>
TRACE_FILENO_listener_name	このパラメータは手動で設定する必要があります。	<p>リスナーのトレース用トレース・ファイルの数。このパラメータが TRACE_FILELEN_listener_name パラメータとともに設定されている場合、トレース・ファイルは循環方式で使用されます。</p>

最初のファイルがいっぱいになると、2 番目のファイルが使用され、順番にファイルが使用されます。最後のファイルがいっぱいになると、最初のファイルが再利用され、再度、順番にファイルが使用されます。

トレース・ファイル名は、順序番号によって識別されます。たとえば、デフォルトのトレース・ファイル listener.trc を使用し、このパラメータを 3 に設定すると、トレース・ファイル名は 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・イベントの前には、そのファイルの順序番号が付きます。

このパラメータが TRACE_FILEAGE_listener_name パラメータとともに設定されている場合、トレース・ファイルはトレース・ファイルの保持期間に基づいて循環されます。最初のファイルが保持期間制限に達するまで使用されると、2 番目のファイルが使用され、以降、同様に繰り返されます。最後のファイルの保持期間制限に達すると、最初のファイルが再利用され、以降、同様に繰り返されます。

このパラメータが TRACE_FILELEN_listener_name と TRACE_FILEAGE_listener_name の両方のパラメータとともに設定されている場合、トレース・ファイルはサイズ制限または保持期間制限に達したときに循環されます。

TRACE_TIMESTAMP_listener_name	このパラメータは手動で設定する必要があります。	リスナー・トレース・ファイルのすべてのトレース・イベントのタイムスタンプ(dd-mon-yyyy hh:mi:ss:mi の書式を使用)。
-------------------------------	-------------------------	-------------------------------------------------------------------------

親トピック: [トレース・パラメータの設定](#)

16.7.2.3 sqlnet.oraのトレース・パラメータ

[表16-21](#)では、sqlnet.oraファイルで設定できるトレース・パラメータの設定を示します。

表16-21 sqlnet.oraのトレース・パラメータ

sqlnet.oraのパラメータ	Oracle Net Managerのフィールド	説明
TRACE_DIRECTORY_CLIENT	「クライアント情報: トレース・ディレクトリ」	クライアント・トレース出力の作成先となるディレクトリ。デフォルトのクライアント・ディレクトリは ORACLE_HOME/network/trace です。
TRACE_DIRECTORY_SERVER	「サーバー情報: トレース・ディレクトリ」	データベース・サーバー・トレース出力の作成先となるディレクトリ。デフォルトのサーバー・ディレクトリは ORACLE_HOME/network/trace です。
TRACE_FILE_CLIENT	「クライアント情報: トレース・ファイル」	クライアントのトレース・ファイルの名前。デフォルトのトレース・ファイル名は sqlnet.trc です。
TRACE_FILE_SERVER	「サーバー情報: トレース・ファイル」	データベース・サーバーのトレース・ファイルの名前。デフォルトのトレース・ファイル名は svr_pid.trc です。
TRACE_FILEAGE_CLIENT	このパラメータは手動で設定する必要があります。	クライアント・トレース・ファイルの最大保持期間を分単位で指定します。保持期間制限に達すると、トレース情報は次のファイルに書き込まれます。ファイルの数は、TRACE_FILENO_CLIENT パラメータで指定します。
TRACE_FILEAGE_SERVER	このパラメータは手動で設定する必要があります。	トレース・ファイルの最大保持期間を分単位で指定します。保持期間制限に達すると、トレース情報は次のファイルに書き込まれます。ファイルの数は、TRACE_FILENO_SERVER パラメータで指定します。
TRACE_FILELEN_CLIENT	このパラメータは手動で設定する必要があります。	クライアント・トレース・ファイルのサイズ(KB 単位)。指定のサイズに到達すると、トレース情報は次のファイルに書き込まれます。ファイルの数は、TRACE_FILENO_CLIENT パラメータで指定します。
TRACE_FILELEN_SERVER	このパラメータは手動で設定する必要があります。	データベース・サーバー・トレース・ファイルのサイズ(KB 単位)。指定のサイズに到達すると、トレース情報は次のファイルに書き込まれます。ファイルの数は、TRACE_FILENO_SERVER パラメータで指定します。
TRACE_FILENO_CLIENT	このパラメータは手動で設定する必要があります。	クライアント・トレース用のトレース・ファイルの数。このパラメータを TRACE_FILELEN_CLIENT パラメータとともに設定すると、トレース・ファイルは循環方式で使用されます。最初のファイルがいっぱいになると、2 番目のファイルが使用され、順番にファイルが使用されます。最後のファイルがいっぱいになると、最初の

sqlnet.oraのパラメータ	Oracle Net Managerのフィールド	説明
		<p>ファイルが再利用され、再度、順番にファイルが使用されます。</p> <p>トレース・ファイル名は、順序番号によって識別されます。たとえば、デフォルトのトレース・ファイル sqlnet.trc を使用し、このパラメータを 3 に設定すると、トレース・ファイル名は sqlnet1_pid.trc、sqlnet2_pid.trc および sqlnet3_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・イベントの前には、そのファイルの順序番号が付きます。</p> <p>このパラメータが TRACE_FILEAGE_CLIENT パラメータとともに設定されている場合、トレース・ファイルはトレース・ファイルの保持期間に基づいて循環されます。最初のファイルが保持期間制限に達するまで使用されると、2 番目のファイルが使用され、以降、同様に繰り返されます。最後のファイルの保持期間制限に達すると、最初のファイルが再利用され、以降、同様に繰り返されます。</p> <p>このパラメータが TRACE_FILELEN_CLIENT と TRACE_FILEAGE_CLIENT の両方のパラメータとともに設定されている場合、トレース・ファイルはサイズ制限または保持期間制限に達したときに循環されます。</p>
TRACE_FILENO_SERVER	このパラメータは手動で設定する必要があります。	<p>データベース・サーバーのトレース用トレース・ファイルの数。このパラメータを TRACE_FILELEN_SERVER パラメータとともに設定すると、トレース・ファイルは循環方式で使用されます。最初のファイルがいっぱいになると、2 番目のファイルが使用され、順番にファイルが使用されます。最後のファイルがいっぱいになると、最初のファイルが再利用され、再度、順番にファイルが使用されます。</p> <p>トレース・ファイル名は、順序番号によって識別されます。たとえば、デフォルトのトレース・ファイル svr_pid.trc を使用し、このパラメータを 3 に設定すると、トレース・ファイル名は svr1_pid.trc、svr2_pid.trc および svr3_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・イベントの前には、そのファイルの順序番号が付きます。</p>

sqlnet.oraのパラメータ	Oracle Net Managerのフィールド	説明
		<p>このパラメータが TRACE_FILEAGE_SERVER パラメータとともに設定されている場合、トレース・ファイルはトレース・ファイルの保持期間に基づいて循環されます。最初のファイルが保持期間制限に達するまで使用されると、2 番目のファイルが使用され、以降、同様に繰り返されます。最後のファイルの保持期間制限に達すると、最初のファイルが再利用され、以降、同様に繰り返されます。</p> <p>このパラメータが TRACE_FILELEN_SERVER と TRACE_FILEAGE_SERVER の両方のパラメータとともに設定されている場合、トレース・ファイルはサイズ制限または保持期間制限に達したときに循環されます。</p>
TRACE_LEVEL_CLIENT	「クライアント情報: トレース・レベル」	<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。</p> <p>トレース・レベルは、0 (ゼロ)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、次のいずれかの値を指定します。</p> <ul style="list-style-type: none"> ● off(0 と同じ)はトレースを実行しません。 ● 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。 ● 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。 ● support (16 と同じ)は、Oracle サポート・サービスによるトラブルシューティングのトレース情報を提供します。
TRACE_LEVEL_SERVER	「サーバー情報: トレース・レベル」	<p>トレース機能によりデータベース・サーバーで記録される詳細レベル。トレース・レベルは、0 (ゼロ)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、次のいずれかの値を指定します。</p> <ul style="list-style-type: none"> ● off(0 と同じ)はトレースを実行しません。 ● 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。 ● admin(6 と同じ)はインストレーション固有の問題を特

		Oracle Net Managerのフィールド	説明
sqlnet.oraのパラメータ			<p>定するためのトレースを実行します。</p> <ul style="list-style-type: none"> ● support (16と同じ)は、Oracle サポート・サービスによるトラブルシューティングのトレース情報を提供します。
TRACE_TIMESTAMP_CLIENT	このパラメータは手動で設定する必要があります。	クライアント・トレース・ファイル sqlnet. trc のすべてのトレース・イベントのタイムスタンプ(dd-mon-yyyy hh:mi:ss:mi)の書式を使用)。	
TRACE_TIMESTAMP_SERVER	このパラメータは手動で設定する必要があります。	クライアント・トレース・ファイル sqlnet. trc のすべてのトレース・イベントのタイムスタンプ(dd-mon-yyyy hh:mi:ss:mi)の書式を使用)。	
TRACE_UNIQUE_CLIENT	「クライアント情報: 一意のトレース・ファイル名」	値を on に設定すると、複数のファイルが共存できるように、生成された各トレース・ファイルの名前にプロセス識別子が付加され、トレース・セッションごとに一意なファイル名が Oracle Net によって作成されます。たとえば、デフォルトのトレース・ファイル名 sqlnet. trc を使用すると、トレース・ファイル sqlnetpid. trc が作成されます。値を off に設定すると、新規クライアント・トレース・セッションからのデータによって、既存のファイルが上書きされます。	

[表16-22](#)に示されるTNSPINGユーティリティのトレース・パラメータをsqlnet. oraファイルに手動で追加できます。TNSPINGユーティリティでは、Oracle Netネットワーク上のサービス(データベース、その他のTNSサービスなど)に正常に到達できるかどうか判断します。

表16-22 TNSPINGのトレース・パラメータ

sqlnet.oraのパラメータ	説明
TNSPING.TRACE_DIRECTORY	TNSPINGトレース・ファイル tnspring. trc の作成先となるディレクトリ。デフォルトのディレクトリは ORACLE_HOME/network/trace です。
TNSPING.TRACE_LEVEL	<p>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。トレース・レベルは、0 (ゼロ)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、次のいずれかの値を指定します。</p> <ul style="list-style-type: none"> ● off(0と同じ)はトレースを実行しません。 ● 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

sqlnet.oraのパラメータ	説明
	<p>します。</p> <ul style="list-style-type: none"> ● admin(6と同じ)はインストール固有の問題を特定するためのトレースを実行します。 ● support (16と同じ)は、Oracle サポート・サービスによるトラブルシューティングのトレース情報を提供します。

親トピック: [トレース・パラメータの設定](#)

16.7.2.4 構成ファイルのトレース・パラメータの設定

sqlnet.oraファイルのトレース・パラメータはOracle Net Managerで設定し、listener.oraファイルのトレース・パラメータはOracle Enterprise Manager Cloud ControlまたはOracle Net Managerで設定します。cman.oraファイルのトレース・パラメータは、手動で設定する必要があります。

- [Oracle Net Managerを使用したsqlnet.oraファイルのトレース・パラメータの設定](#)
- [Oracle Enterprise Manager Cloud Controlを使用したリスナーのトレース・パラメータの設定](#)
- [Oracle Net Managerを使用したリスナーのトレース・パラメータの設定](#)

親トピック: [トレース・パラメータの設定](#)

16.7.2.4.1 Oracle Net Managerを使用したsqlnet.oraファイルのトレース・パラメータの設定

次の手順では、Oracle Net Managerを使用してsqlnet.oraファイルでトレース・パラメータを設定する方法について説明します。

1. Oracle Net Managerを起動します。

関連項目:

[「Oracle Net Managerを使用したOracle Net Servicesの構成」](#)

2. ナビゲータ・ペインで、「ローカル」見出しから「プロファイル」を展開します。
3. 右ペインのリストから、「一般」を選択します。
4. 「トレース」タブをクリックします。
5. 設定を指定します。
6. 「ファイル」メニューから「ネットワーク構成の保存」を選択します。

クライアントのトレース・ファイルの名前はsqlnet.trcです。サーバーのトレース・ファイルの名前はsvr_pid.trcです。

親トピック: [構成ファイルのトレース・パラメータの設定](#)

16.7.2.4.2 Oracle Enterprise Manager Cloud Controlを使用したリスナーのトレース・パラメータの設定

次の手順では、Oracle Enterprise Manager Cloud Controlを使用してリスナーのトレース・パラメータを設定する方法について説明します。

1. Oracle Enterprise Manager Cloud Controlの「Net Services管理」ページにアクセスします。

関連項目:

[「Oracle Enterprise Manager Cloud Controlを使用したOracle Net Servicesの構成」](#)

2. 「管理」リストから「リスナー」を選択し、構成ファイルの場所を含むOracleホームを選択します。
3. 「実行」をクリックし、「リスナー」ページを表示します。
4. リスナーを選択し、「編集」をクリックし、「リスナーの編集」ページを表示します。
5. 「ロギングとトレース」タブをクリックします。
6. 設定を指定します。
7. 「OK」をクリックします。

トレース・ファイルの名前はlistener.trcです。

親トピック: [構成ファイルのトレース・パラメータの設定](#)

16.7.2.4.3 Oracle Net Managerを使用したリスナーのトレース・パラメータの設定

次の手順では、Oracle Net Managerを使用してリスナーのトレース・パラメータを設定する方法について説明します。

1. Oracle Net Managerを起動します。

関連項目:

[「Oracle Net Managerを使用したOracle Net Servicesの構成」](#)

2. ナビゲータ・ペインで、「ローカル」見出しから「リスナー」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、「一般」を選択します。
5. 「ロギングとトレース」タブをクリックします。
6. 設定を指定します。
7. 「ファイル」メニューから「ネットワーク構成の保存」を選択します。

親トピック: [構成ファイルのトレース・パラメータの設定](#)

16.7.3 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時に設定できます。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。

- リスナーの場合は、リスナー制御ユーティリティのSET TRC_DIRECTORY、SET TRC_FILEおよびSET TRC_LEVELコマンドを使用します。
- Oracle Connection Managerの場合は、Oracle Connection Manager制御ユーティリティのSET TRACE_DIRECTORY、SET TRACE_LEVELおよびSET TRACE_TIMESTAMPコマンドを使用します。

親トピック: [Oracle Net Servicesのエラー情報のトレース](#)

16.7.4 Oracle Net Servicesのトレース・ファイルの評価

トレース・ファイルは、Oracleサポート・サービスが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際に役立ちます。この項では、トレース・ファイルの基本的な分析を実行する方法について説明します。

- [ネットワーク・ノード間のデータ・パケットの流れ](#)
- [Oracle Netのデータ・パケットのフォーマット](#)
- [Oracle Netの関連トレース・エラー出力](#)

親トピック: [Oracle Net Servicesのエラー情報のトレース](#)

16.7.4.1 ネットワーク・ノード間のデータ・パケットの流れ

Oracle Net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Oracle Net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。

親トピック: [Oracle Net Servicesのトレース・ファイルの評価](#)

16.7.4.2 Oracle Netのデータ・パケットのフォーマット

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。

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。これは、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に役立ちます。トレース・ファイルで使用されるキーワードは次のとおりです。

- NSPTCN: 接続パケット・タイプで使用します。
- NSPTAC: 受入れパケット・タイプで使用します。
- NSPTRF: 拒否パケット・タイプで使用します。
- NSPTRS: 再送パケット・タイプで使用します。
- NSPTDA: データ・パケット・タイプで使用します。
- NSPCNL: 制御パケット・タイプで使用します。
- NSPTMK: マーカー・パケット・タイプで使用します。

[例16-11](#)では、一般的なパケット情報を示します。この例では、nsconプロシージャはネットワークを通じてNSPTCNパケットを送信します。

例16-11 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
npsend:00 BB 00 00 01 00 00 00 |.....|
npsend:01 33 01 2C 0C 01 08 00 |.3.,....|
npsend:7F FF 7F 08 00 00 00 01 |.....|
npsend:00 99 00 22 00 00 08 00 |..."....|
npsend:01 01 28 44 45 53 43 52 |.. (DESCR|
npsend:49 50 54 49 4F 4E 3D 28 |PTION=(|
npsend:43 4F 4E 4E 45 43 54 5F |CONNECT_|
npsend:44 41 54 41 3D 28 53 49 |DATA=(SI|
npsend:44 3D 61 70 33 34 37 64 |D=ap347d|
npsend:62 31 29 28 43 49 44 3D |b1) (CID=|
npsend:28 50 52 4F 47 52 41 4D |(PROGRAM|
npsend:3D 29 28 48 4F 53 54 3D |=) (HOST=|
npsend:61 70 32 30 37 73 75 6E |sales-12|
npsend:29 28 55 53 45 52 3D 6D |) (USER=m|
npsend:77 61 72 72 65 6E 29 29 |scott)) |
npsend:29 28 41 44 44 52 45 53 |) (ADDRES|
npsend:53 5F 4C 49 53 54 3D 28 |S_LIST=(|
npsend:41 44 44 52 45 53 53 3D |ADDRESS=|
npsend:28 50 52 4F 54 4F 43 4F |(PROTOCO|
npsend:4C 3D 74 63 70 29 28 48 |L=tcp) (H|
npsend:4F 53 54 3D 61 70 33 34 |OST=sale|
npsend:37 73 75 6E 29 28 50 4F |s-12) (PO|
npsend:52 54 3D 31 35 32 31 29 |RT=1521)|
npsend:29 29 29 00 00 00 00 00 |))).....|
npsend: normal exit
nscon: exit (0)

```

親トピック: [Oracle Net Servicesのトレース・ファイルの評価](#)

16.7.4.3 Oracle Netの関連トレース・エラー出力

問題があると、トレース・ファイルにエラー・コードがロギングされます。[例16-12](#)は、データベース・サーバーへのSQL*Plus接続の障害に関する代表的なトレース・ファイル出力です。エラー・メッセージおよびエラー・スタックは太字で示されています。

例16-12 トレースの例

```

[22-MAY-2012 13:34:07:687] nsprecv: entry
[22-MAY-2012 13:34:07:687] nsbal: entry
[22-MAY-2012 13:34:07:687] nsbgetfl: entry
[22-MAY-2012 13:34:07:687] nsbgetfl: normal exit
[22-MAY-2012 13:34:07:687] nsmal: entry
[22-MAY-2012 13:34:07:687] nsmal: 44 bytes at 0x132d90
[22-MAY-2012 13:34:07:687] nsmal: normal exit
[22-MAY-2012 13:34:07:687] nsbal: normal exit
[22-MAY-2012 13:34:07:687] nsprecv: reading from transport...
[22-MAY-2012 13:34:07:687] nttrd: entry
[22-MAY-2012 13:35:09:625] nttrd: exit
[22-MAY-2012 13:35:09:625] ntt2err: entry
[22-MAY-2012 13:35:09:625] ntt2err: Read unexpected EOF ERROR on 10
[22-MAY-2012 13:35:09:625] ntt2err: exit

```

```

[22-MAY-2012 13:35:09:625] nsprecv: transport read error
[22-MAY-2012 13:35:09:625] nsprecv: error exit
[22-MAY-2012 13:35:09:625] nserror: entry
[22-MAY-2012 13:35:09:625] nserror: nsres: id=0, op=68, ns=12537, ns2=12560:
nt[0]=507, nt[1]=0, nt[2]=0; ora[0]=0, ora[1]=0, ora[2]=0
[22-MAY-2012 13:35:09:625] nscon: error exit
[22-MAY-2012 13:35:09:625] nsdo: nsctxrnk=0
[22-MAY-2012 13:35:09:625] nsdo: error exit
[22-MAY-2012 13:35:09:625] nscall: unexpected response
[22-MAY-2012 13:35:09:625] nsclose: entry
[22-MAY-2012 13:35:09:625] nstimarmed: entry
[22-MAY-2012 13:35:09:625] nstimarmed: no timer allocated
[22-MAY-2012 13:35:09:625] nstimarmed: normal exit
[22-MAY-2012 13:35:09:625] nsdo: entry
[22-MAY-2012 13:35:09:625] nsdo: cid=0, opcode=98, *bl=0, *what=0,
uflgs=0x440, cflgs=0x2
[22-MAY-2012 13:35:09:625] nsdo: rank=64, nsctxrnk=0
[22-MAY-2012 13:35:09:625] nsdo: nsctx: state=1, flg=0x4201, mvd=0
[22-MAY-2012 13:35:09:625] nsbfr: entry
[22-MAY-2012 13:35:09:625] nsbaddfl: entry
[22-MAY-2012 13:35:09:625] nsbaddfl: normal exit
[22-MAY-2012 13:35:09:625] nsbfr: normal exit
[22-MAY-2012 13:35:09:625] nsbfr: entry
[22-MAY-2012 13:35:09:625] nsbaddfl: entry
[22-MAY-2012 13:35:09:625] nsbaddfl: normal exit
[22-MAY-2012 13:35:09:625] nsbfr: normal exit
[22-MAY-2012 13:35:09:625] nsdo: nsctxrnk=0
[22-MAY-2012 13:35:09:625] nsdo: normal exit
[22-MAY-2012 13:35:09:625] nsclose: closing transport
[22-MAY-2012 13:35:09:625] nttdisc: entry
[22-MAY-2012 13:35:09:625] nttdisc: Closed socket 10
[22-MAY-2012 13:35:09:625] nttdisc: exit
[22-MAY-2012 13:35:09:625] nsclose: global context check-out (from slot 0)
complete
[22-MAY-2012 13:35:09:703] nsnadisc: entry
[22-MAY-2012 13:35:09:703] nadisc: entry
[22-MAY-2012 13:35:09:703] nacomtm: entry
[22-MAY-2012 13:35:09:703] nacompd: entry
[22-MAY-2012 13:35:09:703] nacompd: exit
[22-MAY-2012 13:35:09:703] nacompd: entry
[22-MAY-2012 13:35:09:703] nacompd: exit
[22-MAY-2012 13:35:09:703] nacomtm: exit
[22-MAY-2012 13:35:09:703] nas_dis: entry
[22-MAY-2012 13:35:09:703] nas_dis: exit
[22-MAY-2012 13:35:09:703] nau_dis: entry
[22-MAY-2012 13:35:09:703] nau_dis: exit
[22-MAY-2012 13:35:09:703] naeetrm: entry
[22-MAY-2012 13:35:09:703] naeetrm: exit
[22-MAY-2012 13:35:09:703] naectrm: entry
[22-MAY-2012 13:35:09:703] naectrm: exit
[22-MAY-2012 13:35:09:703] nagbltrm: entry
[22-MAY-2012 13:35:09:703] nau_gtm: entry
[22-MAY-2012 13:35:09:703] nau_gtm: exit
[22-MAY-2012 13:35:09:703] nagbltrm: exit
[22-MAY-2012 13:35:09:703] nadisc: exit
[22-MAY-2012 13:35:09:703] nsnadisc: normal exit
[22-MAY-2012 13:35:09:703] nsbfr: entry
[22-MAY-2012 13:35:09:703] nsbaddfl: entry

```

```
[22-MAY-2012 13:35:09:703] nsbaddfl: normal exit
[22-MAY-2012 13:35:09:703] nsbfr: normal exit
[22-MAY-2012 13:35:09:703] nsmfr: entry
[22-MAY-2012 13:35:09:703] nsmfr: 2256 bytes at 0x130508
[22-MAY-2012 13:35:09:703] nsmfr: normal exit
[22-MAY-2012 13:35:09:703] nsmfr: entry
[22-MAY-2012 13:35:09:703] nsmfr: 484 bytes at 0x1398a8
[22-MAY-2012 13:35:09:703] nsmfr: normal exit
[22-MAY-2012 13:35:09:703] nsclose: normal exit
[22-MAY-2012 13:35:09:703] nscall: connecting...
[22-MAY-2012 13:35:09:703] nsclose: entry
[22-MAY-2012 13:35:09:703] nsclose: normal exit
[22-MAY-2012 13:35:09:703] nladget: entry
[22-MAY-2012 13:35:09:734] nladget: exit
[22-MAY-2012 13:35:09:734] nsmfr: entry
[22-MAY-2012 13:35:09:734] nsmfr: 144 bytes at 0x132cf8
[22-MAY-2012 13:35:09:734] nsmfr: normal exit
[22-MAY-2012 13:35:09:734] nsmfr: entry
[22-MAY-2012 13:35:09:734] nsmfr: 156 bytes at 0x138e70
[22-MAY-2012 13:35:09:734] nsmfr: normal exit
[22-MAY-2012 13:35:09:734] nladtrm: entry
[22-MAY-2012 13:35:09:734] nladtrm: exit
[22-MAY-2012 13:35:09:734] nscall: error exit
[22-MAY-2012 13:35:09:734] nioqper: error from nscall
[22-MAY-2012 13:35:09:734] nioqper: ns main err code: 12537
[22-MAY-2012 13:35:09:734] nioqper: ns (2) err code: 12560
[22-MAY-2012 13:35:09:734] nioqper: nt main err code: 507
[22-MAY-2012 13:35:09:734] nioqper: nt (2) err code: 0
[22-MAY-2012 13:35:09:734] nioqper: nt OS err code: 0
[22-MAY-2012 13:35:09:734] niomapnerror: entry
[22-MAY-2012 13:35:09:734] niqme: entry
[22-MAY-2012 13:35:09:734] niqme: reporting NS-12537 error as ORA-12537
[22-MAY-2012 13:35:09:734] niqme: exit
[22-MAY-2012 13:35:09:734] niomapnerror: returning error 12537
[22-MAY-2012 13:35:09:734] niomapnerror: exit
[22-MAY-2012 13:35:09:734] niotns: Couldn't connect, returning 12537
[22-MAY-2012 13:35:10:734] niotns: exit
[22-MAY-2012 13:35:10:734] nsbfrfl: entry
[22-MAY-2012 13:35:10:734] nsbrfr: entry
[22-MAY-2012 13:35:10:734] nsbrfr: nsbfs at 0x132d90, data at 0x132dc8.
[22-MAY-2012 13:35:10:734] nsbrfr: normal exit
[22-MAY-2012 13:35:10:734] nsbrfr: entry
[22-MAY-2012 13:35:10:734] nsbrfr: nsbfs at 0x1248d8, data at 0x132210.
[22-MAY-2012 13:35:10:734] nsbrfr: normal exit
[22-MAY-2012 13:35:10:734] nsbrfr: entry
[22-MAY-2012 13:35:10:734] nsbrfr: nsbfs at 0x12d820, data at 0x1319f0.
[22-MAY-2012 13:35:10:734] nsbrfr: normal exit
[22-MAY-2012 13:35:10:734] nsbfrfl: normal exit
[22-MAY-2012 13:35:10:734] nigtrm: Count in the NI global area is now 1
[22-MAY-2012 13:35:10:734] nigtrm: Count in the NL global area is now 1
```

ノート:



前述の例のエラー・スタックには、オペレーティング・システムのエラー・コードが示されています。各オペレーティング・システムには独自のエラー・コードがあります(オペレーティング・システムのエラー・コードの詳細は、システムのドキュメン

トを参照してください)。

エラー・コードを評価する最も効率的な方法は、ロギングされている最新のnerrorエントリを検索することです。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。これらは最新のエラーであり、接続の問題の原因を示しています。

特定のリターン・コードの情報に関しては、Oracleエラー・ツールのoerrを使用します。このためには、任意のコマンドラインで次のように入力します。

```
oerr tns error_number
```

例として、[例16-12](#)に示されるトレース・ファイルにロギングされた、次のnerrorエントリを考えます。

```
[22-MAY-2012 13:35:09:625] nerror: nsres: id=0, op=68, ns=12537, ns2=12560;  
nt[0]=507, nt[1]=0, nt[2]=0; ora[0]=0, ora[1]=0, ora[2]=0
```

前述の例では、メインTNSエラーは12537で、セカンダリ・エラーは12560です。プロトコル・アダプタ・エラーは507です。oerrを使用することによって、リターン・コード12537、12560、および507の詳細を検索できます。次の例ではユーザー入力は太字で示されています。

```
oerr tns 12537  
12537, 00000, "TNS:connection closed"  
// *Cause: "End of file" condition has been reached; partner has disconnected.  
// *Action: None needed; this is an information message.  
oerr tns 12560  
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 reexecute the  
// operation. Turn off tracing when the operation is complete.  
oerr tns 507  
00507, 00000, "Connection closed"  
// *Cause: Normal "end of file" condition has been reached; partner has  
// disconnected.  
// *Action: None needed; this is an information message.
```

親トピック: [Oracle Net Servicesのトレース・ファイルの評価](#)

16.7.5 トレース・アシスタントを使用したトレース・ファイルの検証

Oracle Net Servicesでは、トレース・アシスタントという名前のツールを提供しており、このツールは、トレース・ファイル・テキストの既存の行をわかりやすい段落に変換することによって、トレース・ファイル内の情報を理解する際に役立ちます。トレース・アシスタントは、レベル16(support)のOracle Net Servicesトレース・ファイルでのみ機能します。



ノート:

トレース・アシスタントは、DIAG_ADR_ENABLED パラメータが off に設定されている場合のみ使用できま

す。[「自動診断リポジトリの理解」](#)を参照してください。

- [トレース・アシスタントの構文](#)
- [パケットに関する出力例](#)
- [Two-Task Common \(TTC\)パケットの例](#)
- [接続に関する出力例](#)
- [統計に関する出力例](#)

親トピック: [Oracle Net Servicesのエラー情報のトレース](#)

16.7.5.1 トレース・アシスタントの構文

トレース・アシスタントを実行するには、コマンドライン・プロンプトで次のコマンドを入力します。

```
trcasst [options] filename
```

[表16-23](#)では、オプションを説明します。

表16-23 トレース・アシスタントの構文

オプション	説明
-elevel	エラー情報を表示します。-e の後に、0、1 または 2 を使用してエラー・デコーディング・レベルを指定します。 <ul style="list-style-type: none">● 0 または何もない場合は、nserror ファンクションからダンプされた NS エラー番号が変換され、他のすべてのエラーがリスト表示されます。● 1 の場合は、nserror ファンクションから変換された NS エラーのみが表示されます。● 2 の場合は、変換されていないエラー番号が表示されます。
-la	接続 ID が NS 接続パケットに存在する場合、出力結果には接続 ID が表示されます。接続 ID は、16 進数の 8 バイトの ID として表示されます。パケットが接続に関連付けられていない場合、生成される ID はトレース・アシスタントによって作成されます。つまり、トレース・ファイル内の接続パケットは上書きされます。これは、循環方式のトレース・ファイルで発生する可能性があります。 <p>出力には、ID ごとに次の内容が表示されます。</p> <ul style="list-style-type: none">● ソケット ID(接続で使用する場合)。● 接続パケットの送受信操作。● 初期化パラメータ・ファイル内にある DISPATCHERS パラメータの MULTIPLEX 属性の現行設定。MULTIPLEX を ON に設定すると、セッションの多重化が使用可能になります。● セッション ID(MULTIPLEX が ON に設定されている場合)。

オプション	説明
	<ul style="list-style-type: none"> ● 接続データ情報。 <p>ノート:</p> <ul style="list-style-type: none"> ● このオプションは、他のオプションと一緒に使用しないでください。 ● トレース・アシスタントで生成された ID は、クライアント/サーバー・トレース・ファイルと相関関係はありません。
-li ID	<p>-la の出力から、特定の ID に関するトレースを表示します。</p> <p>ノート: このオプションは、必ず -la オプションの出力とともに使用します。</p>
-otype	<p>出力する情報の種類と量を表示します。-o の後には、次のオプションを使用できます。</p> <ul style="list-style-type: none"> ● c を指定すると、接続サマリー情報を表示します。 ● d を指定すると、接続詳細情報を表示します。 ● u を指定すると、Two-Task Common (TTC) サマリー情報を表示します。 ● t を指定すると、TTC 詳細情報を表示します。 ● q を指定すると、TTC サマリー情報を強化する SQL コマンドが表示されます。このオプションは、-ouq のように、u とともに使用します。 <p>ノート: d の出力には c を指定した場合の情報も含まれるため、c と d の両方を指定しないでください。両方のオプションを指定すると、d の出力のみが表示されます。</p>
-s	<p>次の統計情報を表示します。</p> <ul style="list-style-type: none"> ● 送受信の合計バイト数。 ● オープン・カーソルの最大数。 ● 現在のオープン・カーソル数。 ● 操作の件数と率。 ● PL/SQL に関する解析と実行の件数。 ● 送受信コールの合計数。

オプション	説明
-------	----

- 送受信バイトの合計、平均および最大数。
- 現行の転送およびセッションの合計数。
- タイムスタンプ情報(ある場合)。
- 順序番号(ある場合)。

オプションを指定しない場合、デフォルトは`-odt -e0 -s`です。この場合、トレース・ファイル内の接続情報とTTCイベントの詳細、エラー・レベル0(ゼロ)のエラー情報、および統計情報が表示されます。

[例16-13](#)は、トレース・アシスタントが`-e1`オプションを使用して、トレース・ファイル情報を読みやすい書式に変換する方法を示しています。

例16-13 `trcasst -e1`の出力

```
*****
*                               Trace Assistant                               *
*****
ntus2err: exit
ntuscni: exit
ntusconn: exit
nserver: entry
-<ERROR>- nserver: nsres: id=0, op=65, ns=12541, ns2=12560; nt[0]=511, nt[1]=2, nt[2]=0
////////////////////////////////////
Error found. Error Stack follows:
      id:0
  Operation code:65
      NS Error 1:12541
      NS Error 2:12560
NT Generic Error:511
  Protocol Error:2
      OS Error:0
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 reexecute 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 (or TNSNAV.ORA if the connection is to go
// by way of an Interchange. Start the listener on the remote machine.
/
////////////////////////////////////
*****
*                               Trace Assistant has completed                               *
*****
```

ただし、nerrorファンクションでロギングされなかった他のエラーがトレース・ファイル内に存在する場合があります。

親トピック: [トレース・アシスタントを使用したトレース・ファイルの検証](#)

16.7.5.2 パケットに関する出力例

トレース・アシスタントでは、Oracle NetレイヤーとTTC通信レイヤーの両方からデータ・パケットを表示することもできます。トレース・アシスタントでは、これらのパケットを表示するために、次の2つのオプションを使用できます。

- 接続性のサマリー(-ocオプションを使用)
- 接続性の詳細(-odオプションを使用)

[例16-14](#)は、-ocオプションを使用した場合のサマリー情報の出力を示します。

例16-14 trcasst -ocの出力のサマリー情報

```
*****
*                               Trace Assistant                               *
*****
---> Send 198 bytes - Connect packet
Connect data length: 140
Connect Data:
  (DESCRIPTION= (ADDRESS= (PROTOCOL=TCP) (HOST=sales-server) (PORT=1521))
  (CONNECT_DATA= (SERVICE_NAME=sales.us.example.com) (CID= (PROGRAM=
  (HOST=sales-server) (USER=joe))))

<--- Received 76 bytes - Redirect packet
Redirect data length: 66
Redirect Data:
  (ADDRESS= (PROTOCOL=tcp) (HOST=sales-server) (PORT=1521))
---> Send 198 bytes - Connect packet
Connect data length: 140
Connect Data:
  (DESCRIPTION= (ADDRESS= (PROTOCOL=TCP) (HOST=sales-server) (PORT=1521))
  (CONNECT_DATA= (SERVICE_NAME=sales.us.example.com) (CID= (PROGRAM=
  (HOST=sales-server) (USER=joe))))

<--- Received 32 bytes - Accept packet
Connect data length: 0
---> Send 153 bytes - Data packet
  Native Services negotiation packet
<--- Received 127 bytes - Data packet
  Native Services negotiation packet
---> Send 32 bytes - Data packet
<--- Received 140 bytes - Data packet
```

```
*****
*                               Trace Assistant has completed                               *
*****
```

送信または受信されるパケットには、それぞれ、---> Send nnn bytesまたは<--- Received nnn bytesという接頭辞が付きます。これは、このノードがnnnバイトの特定のタイプのパケットを送信または受信していることを示しています。この接頭辞によって、ノードがクライアントかデータベース・サーバーかを判断できます。接続要求は、常にクライアントから送信され、データベース・サーバーまたはリスナーに受信されます。

[例16-15](#)は、-odオプションを使用した場合の詳細情報の出力を示します。この出力には、接続のネゴシエーション時に接続データとともに送信されたすべての詳細情報が表示されます。

例16-15 trcasst -odの出力の詳細情報

```
*****
*                               Trace Assistant                               *
*****
---> Send 241 bytes - Connect packet
Current NS version number is: 311.
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:8192
  Maximum TDU size:32767
  NT protocol characteristics:
    Test for more data
    Test operation
    Full duplex I/O
    Urgent data support
    Generate SIGURG signal
    Generate SIGPIPE signal
    Generate SIGIO signal
    Handoff connection to another
  Line turnaround value :0
  Connect data length :183
  Connect data offset :58
  Connect data maximum size :512
    Native Services wanted
    NAU doing O3LOGON - DH key foldedin
    Native Services wanted
    NAU doing O3LOGON - DH key foldedin
  Cross facility item 1: 0
  Cross facility item 2: 0
  Connection id : 0x000059F70000004C
  (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=sales-server) (PORT=1521))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com) (SRVR=SHARED) (CID=(PROGRAM=)
  (HOST=sales-server) (USER=joe))))
<--- Received 76 bytes - Redirect packet
  Redirect data length: 66
  (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521))
---> Send 241 bytes - Connect packet
Current NS version number is: 311.
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:8192
Maximum TDU size:32767
NT protocol characteristics:
Test for more data
Test operation
Full duplex I/O
Urgent data support
Generate SIGURG signal
Generate SIGPIPE signal
Generate SIGIO signal
Handoff connection to another
Line turnaround value :0
Connect data length :183
Connect data offset :58
Connect data maximum size :512
    Native Services wanted
    NAU doing O3LOGON - DH key foldedin
    Native Services wanted
    NAU doing O3LOGON - DH key foldedin
Cross facility item 1: 0
Cross facility item 2: 0
Connection id : 0x000059F70000007A
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=sales-server) (PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com) (SRVR=SHARED) (CID=(PROGRAM=
(HOST=sales-server) (USER=joe))))
<--- Received 32 bytes - Accept packet
    Accepted NS version number is: 310.
Global options for the connection:
no attention processing
Don't care
Accepted maximum SDU size: 8192
Accepted maximum TDU size: 32767
Connect data length: 0
    Native Services wanted
    NAU doing O3LOGON - DH key foldedin
    Native Services wanted
    NAU doing O3LOGON - DH key foldedin
---> Send 153 bytes - Data packet
    Native Services negotiation packet version#: 150999040
    Service data packet #0 for Supervisor has 3 subpackets
        Subpacket #0: Version #150999040
        Subpacket #1: 0000000000000000
        Subpacket #2: DEADBEEF0003000000040004000100010002
    Service data packet #1 for Authentication has 3 subpackets
        Subpacket #0: Version #150999040
        Subpacket #1: UB2: 57569
        Subpacket #2: FCFF
    Service data packet #2 for Encryption has 2 subpackets
        Subpacket #0: Version #150999040
        Subpacket #1: 0000000000000000
    Service data packet #3 for Data Integrity has 2 subpackets
        Subpacket #0: Version #150999040
        Subpacket #1: 000000
<--- Received 127 bytes - Data packet
    Native Services negotiation packet version#: 135290880

```

```

Service data packet #0 for Supervisor has 3 subpackets
  Subpacket #0: Version #135290880
  Subpacket #1: 0000
  Subpacket #2: DEADBEEF00030000000200040001
Service data packet #1 for Authentication has 2 subpackets
  Subpacket #0: Version #135290880
  Subpacket #1: FBFF
Service data packet #2 for Encryption has 2 subpackets
  Subpacket #0: Version #135290880
  Subpacket #1: UB1: 0
Service data packet #3 for Data Integrity has 2 subpackets
  Subpacket #0: Version #135290880
  Subpacket #1: UB1: 0
....
--> Send 11 bytes - Marker packet
    One data byte.
    Hex character sent over to the server: 2
<--- Received 11 bytes - Marker packet
    One data byte.
    Hex character sent over to the server: 2
<--- Received 155 bytes - Data packet
--> Send 25 bytes - Data packet
<--- Received 11 bytes - Data packet
--> Send 13 bytes - Data packet
<--- Received 11 bytes - Data packet
--> Send 10 bytes - Data packet
    Data Packet flags:
    End of file
*****
*                               Trace Assistant                               *
*****

```

親トピック: [トレース・アシスタントを使用したトレース・ファイルの検証](#)

16.7.5.3 Two-Task Common (TTC)パケットの例

TTCは、カーソルのオープン、行の選択、行の更新など、データベース・サーバーに送られる要求を処理します。データベース・サーバーは、すべての要求に応答します。ログオンを要求すると、その要求を完了したデータベース・サーバーから応答が返されます。

-ouオプションを使用した場合のTTCサマリー情報は、他の表示とは異なり、行ごとに1つのパケットではなく2つのパケットが表示されます。これは、TTCで処理された要求と応答のペアをミラー化するためです。

出力は次の書式で表示されます。

```
description TTC_message cursor_number SQL_statement bytes_sent bytes_received
```

[例16-16](#)では、接続のネゴシエーション時に接続データとともに送信されたすべての詳細情報を示します。

例16-16 trcasst -ouの出力

```

*****
*                               Trace Assistant                               *
*****
                                           Bytes Bytes

```

		Sent	Rcvd
Send operation(TTIPRO)		32	140
Send operation(TTIDTY)		33	22
Get the session key (OESSKEY)		229	145
Generic authentication call (OAUTH)		368	1001
Send operation(TTIPFN)		44	144
Send operation(TTIPFN)		36	16
Parse a statement (OSQL)	# 1 SELECT USER FROM ...	47	100
Fast upi calls to opial7 (OALL7)	# 1	130	111
Fetch row (OFETCH)	# 1	21	137
Close cursor (OCLOSE)	# 1	17	11
New v8 bundled call (OALL8)	# 0 !Keep Parse BEGI...	156	145
Send operation(TTIPFN)		51	16
Parse a statement (OSQL)	# 1 SELECT ATTRIBUTE,...	186	100
Fast upi calls to opial7 (OALL7)	# 1	246	111
Fetch row (OFETCH)	# 1	21	126
Close cursor (OCLOSE)	# 1	17	11
Send operation(TTIPFN)		36	16
Parse a statement (OSQL)	# 1 SELECT CHAR_VALUE...	208	100
Fast upi calls to opial7 (OALL7)	# 1	130	111
Fetch row (OFETCH)	# 1	21	126
Close cursor (OCLOSE)	# 1	17	11
Send operation(TTIPFN)		36	16
Fast upi calls to opial7 (OALL7)	# 1 !Keep Parse BEGI...	183	41
Send operation(TTIRXD)		20	111
Close cursor (OCLOSE)	# 1	17	11
New v8 bundled call (OALL8)	# 0 Parse Fetch SELE...	165	278
Send operation(TTIPFN)		51	16
Parse a statement (OSQL)	# 1 commit	31	100
Execute statement (OEXEC)	# 1 number of rows: 1	25	100
Close cursor (OCLOSE)	# 1	17	11
Send operation(TTIPFN)		36	16
Fast upi calls to opial7 (OALL7)	# 1 !Keep Parse BEGI...	183	41
Send operation(TTIRXD)		60	111
Close cursor (OCLOSE)	# 1	17	11
Send operation(TTIPFN)		36	16
Fast upi calls to opial7 (OALL7)	# 1 !Keep Parse BEGI...	183	41
Send operation(TTIRXD)		20	111
Close cursor (OCLOSE)	# 1	17	11
New v8 bundled call (OALL8)	# 0 Parse Fetch sele...	144	383
New v8 bundled call (OALL8)	# 1 !Keep Fetch	121	315
Logoff off of Oracle (OLOGOFF)		13	11

```

*****
*                               Trace Assistant                               *
*****

```

例16-17は、-otオプションを使用した場合のTTC詳細情報の出力を示します。

例16-17 trcasst -otの出力の詳細なTTC情報

```

*****
*                               Trace Assistant                               *
*****
Set protocol (TTIPRO)
  Operation 01 (con) Send protocol version=6
  Originating platform: x86_64/Linux 2.4.xx
Set protocol (TTIPRO)
  Operation 01 (con) Receive protocol version=6

```

```

Destination platform: x86_64/Linux 2.4.xx
Set datatypes (TTIDTY)
Set datatypes (TTIDTY)
Start of user function (TTIFUN)
    Get the session key (OSESKEY)
Return opi parameter (TTIRPA)
Start of user function (TTIFUN)
    Generic authentication call (OAUTH)
Return opi parameter (TTIRPA)
Piggyback function follows (TTIPFN)
Start of user function (TTIFUN)
    V8 session switching piggyback (O80SES)
Start of user function (TTIFUN)
    Get Oracle version/date (OVERSION)
Return opi parameter (TTIRPA)
Oracle Database 18c Enterprise Edition Release 18.0.0.0.0 - 64bit Production
Version 18.1.0.0.0
Start of user function (TTIFUN)
    New v8 bundled call (OALL8) Cursor # 0
    Parse Fetch
Describe information (TTIDCB)
Start of user function (TTIFUN)
    Fetch row (OFETCH) Cursor # 3
ORACLE function complete (TTIOER)
ORA-01403: no data found
Piggyback function follows (TTIPFN)
Start of user function (TTIFUN)
    Cursor Close All (OCCA)
Start of user function (TTIFUN)
    New v8 bundled call (OALL8) Cursor # 0
    Parse Fetch
Describe information (TTIDCB)
Piggyback function follows (TTIPFN)
Start of user function (TTIFUN)
    Cursor Close All (OCCA)
Start of user function (TTIFUN)
    New v8 bundled call (OALL8) Cursor # 0
    Parse Fetch
Describe information (TTIDCB)
Piggyback function follows (TTIPFN)
Start of user function (TTIFUN)
    Cursor Close All (OCCA)
Start of user function (TTIFUN)
    New v8 bundled call (OALL8) Cursor # 0
    !Keep Parse
Sending the I/O vector only for fast upi (TTIIOV)
Piggyback function follows (TTIPFN)
Start of user function (TTIFUN)
    Cursor Close All (OCCA)
Start of user function (TTIFUN)
    New v8 bundled call (OALL8) Cursor # 0
    Parse Fetch
Describe information (TTIDCB)
Piggyback function follows (TTIPFN)
Start of user function (TTIFUN)
    Cursor Close All (OCCA)
Start of user function (TTIFUN)
    Commit (OCOMMIT)

```

```

V6 Oracle func complete (TTISTA)
Start of user function (TTIFUN)
  Commit (OCOMMIT)
V6 Oracle func complete (TTISTA)
Start of user function (TTIFUN)
  Logoff off of Oracle (OLOGOFF)
  MAXIMUM OPEN CURSORS: 0
  CURSORS NOT CLOSED: 0
V6 Oracle func complete (TTISTA)
  Succeeded
*****
*                Trace Assistant has completed                *
*****

```

例16-18は、-ouqオプションを使用した場合のSQL詳細情報の出力を示します。出力の各行の最初に表示される項目は、実際に送信された要求です。2番目に表示される項目は、操作が実行されたカーソルの位置です。3番目に表示される項目は、SQLコマンドのリスト、または応答中を示すフラグのいずれかです。送受信されたバイト数は、最も右側に表示されます。フラグは次のいずれかです。

- !PL/SQL = PL/SQL要求以外
- COM = Commit
- IOV = Get I/O Vector
- DEFN = Define
- EXEC = Execute
- FETCH = Fetch
- CAN = Cancel
- DESCSEL = Describe select
- DESCBND = Describe Bind
- BND = Bind
- PARSE = Parse
- EXACT = Exact

例16-18 trcasst -ouqの出力の詳細なSQL情報

```

*****
*                Trace Assistant                *
*****

```

		Bytes	Bytes
		Sent	Rcvd
Send operation (TTIPRO)		32	140
Send operation (TTIDTY)		33	22
Get the session key (OESSKEY)		229	145
Generic authentication call (OAUTH)		368	1001
Send operation (TTIPFN)		44	144
Send operation (TTIPFN)		36	16
Parse a statement (OSQL)	# 1	47	100
SELECT USER FROM DUAL			
Fast upi calls to opial7 (OALL7)	# 1	130	111

Fetch row (OFETCH)	# 1	21	137
Close cursor (OCLOSE)	# 1	17	11
New v8 bundled call (OALL8)	# 0 !Keep Parse	156	145
BEGIN DBMS_OUTPUT.DISABLE; END;			
Send operation(TTIPFN)		51	16
Parse a statement (OSQL)	# 1	186	100
SELECT ATTRIBUTE, SCOPE, NUMERIC_VALUE, CHAR_VALUE, DATE_VALUE FROM SYSTEM.PRODUCT_PRIVS WHERE (UPPER('SQL*Plus') LIKE UPPER(PRODUCT)) AND (UPPER(USER) LIKE USERID)			
Fast upi calls to opial7 (OALL7)	# 1	246	111
Fetch row (OFETCH)	# 1	21	126
Close cursor (OCLOSE)	# 1	17	11
Send operation(TTIPFN)		36	16
Parse a statement (OSQL)	# 1	208	100
SELECT CHAR_VALUE FROM SYSTEM.PRODUCT_PRIVS WHERE (UPPER('SQL*Plus') LIKE UPPER(PRODUCT)) AND ((UPPER(USER) LIKE USERID) OR (USERID = 'PUBLIC')) AND (UPPER(ATTRIBUTE) = 'ROLES')			
Fast upi calls to opial7 (OALL7)	# 1	130	111
Fetch row (OFETCH)	# 1	21	126
Close cursor (OCLOSE)	# 1	17	11
Send operation(TTIPFN)		36	16
Fast upi calls to opial7 (OALL7)	# 1 !Keep Parse	183	41
BEGIN DBMS_APPLICATION_INFO.SET_MODULE(:1, NULL); END;			
Send operation(TTIRXD)		20	111
Close cursor (OCLOSE)	# 1	17	11
New v8 bundled call (OALL8)	# 0 Parse Fetch	165	278
SELECT DECODE('A', 'A', '1', '2') FROM DUAL			
Send operation(TTIPFN)		51	16
Parse a statement (OSQL)	# 1	31	100
commit			
Execute statement (OEXEC)	# 1 number of rows: 1	25	100
Close cursor (OCLOSE)	# 1	17	11
Send operation(TTIPFN)		36	16
Fast upi calls to opial7 (OALL7)	# 1 !Keep Parse	183	41
BEGIN DBMS_APPLICATION_INFO.SET_MODULE(:1, NULL); END;			
Send operation(TTIRXD)		60	111
Close cursor (OCLOSE)	# 1	17	11
Send operation(TTIPFN)		36	16
Fast upi calls to opial7 (OALL7)	# 1 !Keep Parse	183	41
BEGIN DBMS_APPLICATION_INFO.SET_MODULE(:1, NULL); END;			
Send operation(TTIRXD)		20	111
Close cursor (OCLOSE)	# 1	17	11
New v8 bundled call (OALL8)	# 0 Parse Fetch	144	383
select * from dept			
New v8 bundled call (OALL8)	# 1 !Keep Fetch	121	315
Logoff off of Oracle (OLOGOFF)		13	11

* Trace Assistant has completed *			

親トピック: [トレース・アシスタントを使用したトレース・ファイルの検証](#)

16.7.5.4 接続に関する出力例

例16-19は、-laオプションを使用した場合の出力を示します。出力には次の情報が表示されます。

- 受信した接続ID
- この接続を受信したソケットID。
- 操作
 - Receiveはトレースがデータベース・サーバー・トレースであることを示します。この例では、Receiveが操作です。
 - Sendはトレースがクライアント・トレースであることを示します。
- DISPATCHERSパラメータのMULTIPLEX属性は、ONに設定されています。
- 32ビットのセッションID
- 受信した接続データ情報。

例16-19 trcasst -laの出力

```
*****
*                               Trace Assistant                               *
*****
Connection ID: 00000B270000000B
  Socket Id: 15
  Operation: Receive
  Multiplex: ON
  Session Id: 8362785DE4FC0B19E034080020F793E1
  Connect Data:
  (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=sales-server) (PORT=1521))
  (CONNECT_DATA=(SERVER=shared)
  (SERVICE_NAME=sales.us.example.com) (CID=(PROGRAM=) (HOST=sales-server)
  (USER=oracle))))
Connection ID: 00000B240000000B
  Socket Id: 15
  Operation: Receive
  Multiplex: ON
  Session Id: 8362785DE4FB0B19E034080020F793E1
  Connect Data:
  (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=sales-server) (PORT=1521))
  (CONNECT_DATA=(SERVER=shared)
  (SERVICE_NAME=sales.us.example.com) (CID=(PROGRAM=) (HOST=sales-server)
  (USER=oracle))))
Connection ID: 00000B1F00000008
  Socket Id: 15
  Operation: Receive
  Multiplex: ON
  Session Id: 8362785DE4F90B19E034080020F793E1
  Connect Data:
  (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=sales-server) (PORT=1521))
  (CONNECT_DATA=(SERVER=shared)
  (SERVICE_NAME=sales.us.example.com) (CID=(PROGRAM=) (HOST=sales-server)
  (USER=oracle))))
*****
*                               Trace Assistant has completed                               *
*****
```

例16-20は、-li 00000B1F00000008オプションを使用した場合の接続ID 00000B1F00000008に関する出力を示します。

例16-20 trcasst -liの出力

```
*****
*                               Trace Assistant                               *
*****
<--- Received 246 bytes - Connect packet
Current NS version number is: 310.
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: 8192
  Maximum TDU size: 32767
  NT protocol characteristics:
    Test for more data
    Test operation
    Full duplex I/O
    Urgent data support
    Generate SIGURG signal
    Generate SIGPIPE signal
    Generate SIGIO signal
    Handoff connection to another
  Line turnaround value: 0
  Connect data length: 188
  Connect data offset: 58
  Connect data maximum size: 512
    Native Services wanted
    NAU doing O3LOGON - DH key foldedin
    Native Services wanted
    NAU doing O3LOGON - DH key foldedin
  Cross facility item 1: 0
  Cross facility item 2: 0
  Connection id: 0x00000B1F00000008
  (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=sales-server) (PORT=1521))
  (CONNECT_DATA=(SERVER=shared) (SERVICE_NAME=sales.us.example.com)
  (CID=(PROGRAM=) (HOST=sales-server) (USER=oracle))))
---> Send 114 bytes - Accept packet
Accepted NS version number is: 310.
Global options for the connection:
  No attention processing
  Don't care
  Accepted maximum SDU size: 8192
  Accepted maximum TDU size: 32767
  Connect data length: 0
    Native Services wanted
    NAU doing O3LOGON - DH key foldedin
    Native Services wanted
    NAU doing O3LOGON - DH key foldedin
  Connection Time out: 1000
  Tick Size: 100
  Reconnect Data: (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=34454))
  Session Id: 8362785DE4F90B19E034080020F793E1
<--- Received 164 bytes - Data packet
  Native Services negotiation packet version#: 135290880
    Service data packet #0 for Supervisor has 3 subpackets
```

```

        Subpacket #0: Version #135290880
        Subpacket #1: 0000000000000000
        Subpacket #2: DEADBEEF0003000000040004000100010002
Service data packet #1 for Authentication has 3 subpackets
        Subpacket #0: Version #135290880
        Subpacket #1: UB2: 57569
        Subpacket #2: FCFF
Service data packet #2 for Encryption has 2 subpackets
        Subpacket #0: Version #135290880
        Subpacket #1: 0000000000
Service data packet #3 for Data Integrity has 2 subpackets
        Subpacket #0: Version #135290880
        Subpacket #1: 0000
----> Send 143 bytes - Data packet
    Native Services negotiation packet version#: 135290880
        Service data packet #0 for Supervisor has 3 subpackets
            Subpacket #0: Version #135290880
            Subpacket #1: 0000
            Subpacket #2: DEADBEEF00030000000200040001
        Service data packet #1 for Authentication has 2 subpackets
            Subpacket #0: Version #135290880
            Subpacket #1: FBFF
        Service data packet #2 for Encryption has 2 subpackets
            Subpacket #0: Version #135290880
            Subpacket #1: UB1: 0
        Service data packet #3 for Data Integrity has 2 subpackets
            Subpacket #0: Version #135290880
            Subpacket #1: UB1: 0
<--- Received 48 bytes - Data packet
Set protocol (TTIPRO)
    Operation 01 (con) Receive protocol version=6
    Destination platform: SVR4-be-8.1.0
----> Send 156 bytes - Data packet
Set protocol (TTIPRO)
    Operation 01 (con) Send protocol version=6
    Originating platform: SVR4-be-8.1.0
<--- Received 49 bytes - Data packet
Set datatypes (TTIDTY)
----> Send 38 bytes - Data packet
Set datatypes (TTIDTY)
<--- Received 245 bytes - Data packet
Start of user function (TTIFUN)
    Get the session key (OSESSKEY)
----> Send 161 bytes - Data packet
Return opi parameter (TTIRPA)
...
*****
*                               Trace Assistant has completed                               *
*****

```

親トピック: [トレース・アシスタントを使用したトレース・ファイルの検証](#)

16.7.5.5 統計に関する出力例

収集される統計の種類は、ほとんどの場合、ネットワーク・パートナー間で送受信されたTTCコール数、パケット数およびバイト数です。[例16-21](#)では、`-s`オプションを使用した場合の代表的なトレース・ファイルの統計を示します。

例16-21 trcasst -sの出力

```
*****
*                               Trace Assistant                               *
*****

-----
Trace File Statistics:
-----

Total number of Sessions: 3
DATABASE:
Operation Count:    0 OPENS,    21 PARSES,    21 EXECUTES,    9 FETCHES
Parse Counts:
  9 PL/SQL,    9 SELECT,    0 INSERT,    0 UPDATE,    0 DELETE,
  0 LOCK,    3 TRANSACT,    0 DEFINE,    0 SECURE,    0 OTHER
Execute counts with SQL data:
  9 PL/SQL,    0 SELECT,    0 INSERT,    0 UPDATE,    0 DELETE,
  0 LOCK,    0 TRANSACT,    0 DEFINE,    0 SECURE,    0 OTHER
Packet Ratio: 6.142857142857143 packets sent per operation
Currently opened Cursors: 0
Maximum opened Cursors : 0
ORACLE NET SERVICES:
Total Calls :      129 sent,      132 received,      83 oci
Total Bytes :     15796 sent,     13551 received
Average Bytes:     122 sent per packet,     102 received per packet
Maximum Bytes:     1018 sent,      384 received
Grand Total Packets: 129 sent,    132 received

*****
*                               Trace Assistant has completed                               *
*****
```

親トピック: [トレース・アシスタントを使用したトレース・ファイルの検証](#)

16.8 Oracleサポート・サービスへの連絡

一部のメッセージについては、Oracleサポート・サービスに問題を報告することをお勧めします。Oracleサポート・サービスに連絡する場合は、次の情報を用意してください。

- ハードウェア、オペレーティング・システムおよびOracle Databaseを実行しているオペレーティング・システムのリリース番号。
- Oracle Databaseの完全なリリース番号(12.2.0.1.0など)。
- エラーが発生したときに使用していたすべてのOracleプログラムとリリース番号(SQL*Plusリリース12.2.0.1.0など)。
- 複数のエラー・コードまたはメッセージが検出された場合の表示順の正確なコード番号およびメッセージ・テキスト。
- 次の規約に従った問題の重大度。
 - 1: プログラムが使用できません。操作に重大な影響を与えます。
 - 2: プログラムが使用できます。操作が大幅に制限されます。
 - 3: 機能が制限されて使用できるプログラム。全体の操作に重大な影響はありません。
 - 4: お客様によって回避された問題。発生しても操作への影響は最小限です。

次の項目も必要です。

- 連絡者の名前
- 連絡者の組織の名前
- 連絡者のOracle Support ID番号
- 連絡者の電話番号

親トピック: [Oracle Net Servicesのトラブルシューティング](#)

用語集

アクセス制御リスト(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に対して、特定のデータへのアクセス・レベルを付与します。

親トピック: [用語集](#)

ACL

[「アクセス制御リスト\(ACL\)」](#)を参照してください。

親トピック: [用語集](#)

アクセス制御

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規則を設定するOracle Connection Managerの機能。

親トピック: [用語集](#)

アドレス

[「プロトコル・アドレス」](#)を参照してください。

親トピック: [用語集](#)

ADR

[「自動診断リポジット\(ADR\)」](#)を参照してください。

親トピック: [用語集](#)

別名

サーバー内のネットワーク・オブジェクトの代替名です。別名には、参照するオブジェクトの名前が格納されます。クライアントが別名の参照をリクエストすると、Oracleは、その別名があたかも参照されているオブジェクトであるかのようにしてその参照を完了します。

親トピック: [用語集](#)

アプリケーション・ゲートウェイ

[Oracle Netファイアウォール・プロキシ](#)を実行するホスト・コンピュータです。アプリケーション・ゲートウェイは、クライアント側から見

ると実際のサーバーのように機能し、サーバー側から見ると実際のクライアントのように機能します。アプリケーション・ゲートウェイは、インターネットと企業の内部ネットワークの間に位置し、双方のユーザーに仲介サービス(プロキシ・サービス)を提供します。

親トピック: [用語集](#)

ASCII文字セット

American Standard Code for Information Interchange文字セットの略で、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です。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序です。

親トピック: [用語集](#)

属性

ディレクトリ・エントリの性質を説明する情報項目の1つです。エントリは属性のセットから構成され、それぞれの属性が[オブジェクト・クラス](#)に属する。さらに、各属性にはタイプと値があり、タイプは属性の情報の種類を説明するもので、値には実際のデータが格納されています。

親トピック: [用語集](#)

認証方式

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です。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ます。次の認証方法がサポートされています。

- [Kerberos](#)
- Oracle Autonomous Cloud DatabaseのMicrosoft Azure Active Directory (Azure AD)ユーザー認証および認可([Oracle Databaseセキュリティ・ガイド](#)を参照)
- [Microsoft Windows NTのシステム固有の認証](#)
- Oracle Autonomous Cloud DatabaseのOracle Cloud Infrastructure (OCI) Identity and Access Management (IAM)ユーザー認証および認可([Oracle Databaseセキュリティ・ガイド](#)を参照)
- [RADIUS](#)
- [Transport Layer Security\(TLS\)](#)

親トピック: [用語集](#)

自動診断リポジトリ(ADR)

自動診断リポジトリ(ADR)は、システム全体のファイルのトレースおよびロギング用中央リポジトリです。このリポジトリは、診断情報を格納するファイル・ベースの階層データ・ストアです。

親トピック: [用語集](#)

キャッシュ

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です。

親トピック: [用語集](#)

CIDR

Classless Inter-Domain Routing。CIDR表記では、IPv6サブネットは、スラッシュ(/)文字で区切ったサブネット接頭辞と接頭辞のサイズ(10進数で表したビット数)で表されます。たとえば、2001:0db8:0000:0000::/64は

2001:0db8:000:0000:0000:0000:0000:0000から2001:0db8:000:0000:FFFF:FFFF:FFFF:FFFFのアドレスを持つサブネットを表します。CIDR表記法には、IPv4アドレスのサポートが含まれます。たとえば、192.0.2.1/24は、アドレス192.0.2.1から192.0.2.255までのサブネットを示します。

親トピック: [用語集](#)

Classless Inter-Domain Routing (CIDR)

[\[CIDR\]](#)を参照してください。

親トピック: [用語集](#)

クライアント

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です。クライアントはユーザー・プロセスです。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ります。

親トピック: [用語集](#)

クライアント・ロード・バランシング

ロード・バランシング機能では、複数のリスナーが単一データベースにサービスを提供する場合に、クライアントはリスナー間で接続リクエストをランダムに選択できます。このランダム化によって、すべてのリスナーに着信接続リクエストの処理負荷を分散できます。

親トピック: [用語集](#)

クライアント・プロファイル

クライアントのプロパティ。[ネーミング・メソッド](#)の優先順位、クライアントとサーバーの[ロギング](#)と[トレース](#)、名前の要求元であるドメインやその他のクライアント・オプションなどがあります。

親トピック: [用語集](#)

クライアント/サーバー・アーキテクチャ

2台のCPU間で処理を分割するソフトウェア・アーキテクチャです。1つのCPUは、トランザクションでクライアントとして機能し、サービスを要求して受け取ります。もう1つは、要求を処理するサーバーとして機能します。

親トピック: [用語集](#)

cman.oraファイル

着信リクエストおよび管理コマンドのプロトコル・アドレス、Oracle Connection Managerのパラメータおよび[アクセス制御](#)の規則を指定するOracle Connection Managerの構成ファイルです。

親トピック: [用語集](#)

CMADMIN (Oracle Connection Manager Administration)

リスナーとOracle Connection Managerゲートウェイ・プロセスの状態を監視する[Oracle Connection Manager](#)プロセスで、必要に応じてプロセスを停止および起動します。CMADMINは、リスナーによるゲートウェイ・プロセスに関する情報を登録し、Oracle Connection Manager Controlユーティリティで実行されるコマンドを処理します。

親トピック: [用語集](#)

CMGW (Oracle Connection Manager Gateway)

Oracle Connection Managerインスタンス上のリスナーによって選別され、転送されたクライアント接続を受信する[Oracle Connection Manager](#)プロセスです。ゲートウェイ・プロセスは、要求をデータベース・サーバーへ転送します。さらに、1つのプロトコル接続で、複数のクライアント接続を多重化し、集中化します。

親トピック: [用語集](#)

接続データ

接続先データベースの[サービス名](#)または[Oracleシステム識別子\(SID\)](#)を定義する[接続記述子](#)の一部です。次の例では、SERVICE_NAMEにsales.us.example.comというデータベース・サービスが定義されています。

```
(DESCRIPTION=
  (ADDRESS= (PROTOCOL=tcp) (HOST=sales-server) (PORT=1521)
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.example.com)))
```

親トピック: [用語集](#)

接続記述子

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です。接続記述子には、宛先サービスおよびネットワーク・ルート情報が含まれています。

宛先サービスは、その[サービス名](#)で示されます。ネットワーク・ルートは、少なくとも、ネットワーク・アドレスを使用してリスナーの場所を示します。

親トピック: [用語集](#)

接続識別子

[接続記述子](#)または接続記述子にマップされる名前。接続識別子は、[ネットワーク・サービス名](#)、[データベース・サービス名](#)、[ネットワーク・サービス別名](#)のいずれかです。次のように、ユーザーは、接続するサービスに対して、接続文字列内の接続識別子とともにユーザー名とパスワードを渡して、接続要求を実行します。

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

親トピック: [用語集](#)

接続文字列

ユーザー名、[接続識別子](#)など、ユーザーが接続するサービスに渡す情報。

```
CONNECT username@net_service_name
```

親トピック: [用語集](#)

接続時フェイルオーバー

リスナーが応答しない場合に、クライアントの接続要求が別のリスナーに転送される機能です。リスナーは接続を試行するためにインスタンスが起動しているかどうかを認識するため、接続時フェイルオーバーは[サービス登録](#)によって使用可能になります。

親トピック: [用語集](#)

接続

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)との接続をリクエストする起動側(クライアント)によって開始されます。

親トピック: [用語集](#)

接続ロード・バランシング

インスタンスやディスパッチャ間で同じサービスに対するアクティブな接続数を均等化する方法です。接続ロード・バランシングにより、リスナーはディスパッチャごとの接続数およびノードのロードに基づいて、ルーティングを決定できます。

親トピック: [用語集](#)

接続リクエスト

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。

親トピック: [用語集](#)

データ・パケット

[「パケット」](#)を参照してください。

親トピック: [用語集](#)

データベース・リンク

Oracleデータベース・サーバーから別のデータベース・サーバーへの一方向の接続パスを定義するポインタ。パブリックおよびプライベート・データベースのリンクは、データ・ディクショナリ表で定義済みのエン트리です。グローバル・データベース・リンクは、LDAPディレクトリに格納されており、ネットワーク上のすべてのユーザーがアクセスできます。パブリックおよびプライベート・リンクにアクセスするには、データ・ディクショナリ・エントリを含むローカル・データベースに接続する必要があります。

ローカル・データベースAに接続したクライアントは、データベースAに格納されているパブリックまたはプライベート・リンクを使用して、リモート・データベースBの情報にアクセスできます。しかし、データベースBに接続されたユーザーは同じリンクを使用してデータベースAのデータにアクセスできません。データベースB上のローカル・ユーザーがデータベースA上のデータにアクセスする場合、データベースBのデータ・ディクショナリに格納されるリンクを定義する必要があります。グローバル・リンクは、ネットワーク上の任意のクライアントとデータベースの間で使用できます。

次のデータベース・リンクがサポートされます。

- データベースの特定のスキーマにある[プライベート・データベース・リンク](#)。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。
- データベースの[パブリック・データベース・リンク](#)。そのデータベースのすべてのユーザーが使用できます。
- [グローバル・データベース・リンク](#)は、LDAPディレクトリに格納されたデータベース・リンクです。

親トピック: [用語集](#)

専用接続

データベース・セッションが含まれる専用サーバーです。

親トピック: [用語集](#)

専用サーバー

1つのクライアント接続専用のサーバー・プロセスです。[「共有サーバー」](#)と対比してください。

親トピック: [用語集](#)

デフォルト・ドメイン

ほとんどのクライアント要求が実行される[ドメイン](#)です。クライアントが存在しているドメイン、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です。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る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ります。名前の要求に(.)が入っていない場合は、未修飾の要求です。

親トピック: [用語集](#)

ディレクトリ情報ツリー(DIT)

エントリの識別名(DN)で構成される、[ディレクトリ・サーバー](#)内のツリー形式の階層構造です。この構造は、x500およびLDAP固有です。

親トピック: [用語集](#)

ディレクトリ・ネーミング

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、[ネットワーク・サービス名](#)または[ネットワーク・サービス別名](#)を[接続記述子](#)に解決する[ネーミング・メソッド](#)です。[ディレクトリ・サーバー](#)によって、ディレクトリ・ネーミング・オブジェクトの集中管理が可能となり、サービスの追加または再配置に伴う作業負荷を軽減できます。

親トピック: [用語集](#)

ディレクトリ・サーバー

[Lightweight Directory Access Protocol \(LDAP\)](#)でアクセスするディレクトリ・サーバーです。LDAP準拠のディレクトリ・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のための集中化された媒体が提供されます。このディレクトリ・サーバーによって、クライアント側とサーバー側にあるローカルのtnsnames.oraファイルを置換できます。

親トピック: [用語集](#)

ディスパッチャ

多くのクライアントが、クライアントごとの専用サーバー・プロセスなしで同じサーバーに接続できるようにするプロセスです。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・プロセスに送ります。「[共有サーバー](#)」も参照してください。

親トピック: [用語集](#)

識別名(DN)

[ディレクトリ・サーバー](#)内の場所の名前です。ディレクトリ・パスによってファイルの正確な位置が指定されるのと同様に、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ます。

親トピック: [用語集](#)

分散処理

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実行します。Oracle Net Services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します。

親トピック: [用語集](#)

ドメイン

[ドメイン・ネーム・システム\(DNS\)](#)ネームスペース内の任意のツリーまたはサブツリー。一般にこの用語は、ホスト名が共通の接尾辞(ドメイン名)を共有するコンピュータのグループを表します。

親トピック: [用語集](#)

ドメイン・ネーム・システム(DNS)

[ドメイン](#)に階層化されたコンピュータやネットワーク・サービスのネーミングに関するシステムです。DNSは、ユーザーにわかりやすい名前がコンピュータの位置を特定するためにTCP/IPネットワークで使用される。DNSは、ユーザー・フレンドリな名前をコンピュータが理解できる[IPアドレス](#)に解決します。

Oracle Net Servicesの場合、DNSは、TCP/IPアドレス内のホスト名をIPアドレスに変換します。

親トピック: [用語集](#)

DNS

「[ドメイン・ネーム・システム\(DNS\)](#)」を参照してください。

親トピック: [用語集](#)

エンタープライズ・ロール

エンタープライズ・ロールは、通常のデータベース・ロールに類似していますが、認証が複数のデータベースにまたがる点が異なります。エンタープライズ・ロールは、特定のデータベース上で権限を定義するロールの一種です。エンタープライズ・ロールは、あるデータベースのデータベース管理者により作成されます。1人または複数のエンタープライズ・ユーザーから、エンタープライズ・ロールの権限付与または取消しを行うことが可能です。これらのロールの権限付与および取消しについての情報も、ディレクトリ・サーバー内に格納されます。

親トピック: [用語集](#)

エンタープライズ・ユーザー

企業内で固有の識別情報を持つユーザー。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します。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ます。

親トピック: [用語集](#)

エントリ

ディレクトリ・サーバーの基本単位で、ディレクトリ・ユーザーに関係のあるオブジェクトに関する情報が含まれています。

親トピック: [用語集](#)

外部ネーミング

[Network Information Service \(NIS\)](#)などのサード・パーティ製のネーミング・サービスを使用する[ネーミング・メソッド](#)。

親トピック: [用語集](#)

外部プロシージャ

PL/SQLコードから呼び出す、第3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ます。

親トピック: [用語集](#)

フェイルオーバー

[「接続時フェイルオーバー」](#)を参照してください。

親トピック: [用語集](#)

ファイアウォール・サポート

[「アクセス制御」](#)を参照してください。

親トピック: [用語集](#)

FTP

File Transfer Protocol。一方のコンピュータのユーザーが、TCP/IPネットワークを介して他方のコンピュータとファイルを送受信できるようにするクライアント/サーバー・プロトコル。

親トピック: [用語集](#)

グローバル・データベース・リンク

LDAPディレクトリに格納されているデータベース・リンクの定義で、ネットワーク上のすべてのユーザーがアクセスできます。この定義は、クライアントのデータベース接続に使用されるものと同じです(名前/接続記述子)。

グローバル・データベース・リンクには、ユーザー句またはパスワード句を含めることはできません。これらは、リンクを起動するデータベースが、既存のクライアントの識別情報を使用してそのリンクを確立するときのみ機能します。

親トピック: [用語集](#)

グローバル・データベース名

他のデータベースから一意に識別される、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`です。たとえば、`sales.us.example.com`となります。

データベース名の部分`sales`は、データベースを参照するときに使用する単純な名前です。データベース・ドメイン部分`us.example.com`によって、データベースが位置するデータベース・ドメインを指定し、一意のグローバル・データベース名にします。可能なかぎり、データベース・ドメインにはネットワーク・ドメインを反映する名前を使用してください。

グローバル・データベース名は、初期化パラメータ・ファイルの`SERVICE_NAMES`パラメータで指定したデータベースのデフォルト・サービス名です。

親トピック: [用語集](#)

異機種間サービス

Oracle Databaseサーバーからサード・パーティのシステムにアクセスするための基本的な技術を提供する統合コンポーネントです。異機種間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、サード・パーティ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分散環境からサード・パーティのシステム、サービスまたはApplication Program Interface (API)に透過的にアクセスすること

親トピック: [用語集](#)

階層ネーミング・モデル

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のインフラストラクチャ。

親トピック: [用語集](#)

ホスト・ネーミング

[ネーミング・メソッド](#)の1つで、これにより、TCP/IP環境のユーザーは既存の名前解決サービスで名前を解決できます。この名前解決サービスには、[ドメイン・ネーム・システム\(DNS\)](#)、[Network Information Service \(NIS\)](#)または集中管理された`/etc/hosts`ファイル・セットがあります。ホスト・ネーミングによって、ユーザーはサーバー・コンピュータのホスト名またはホスト名の別名を指定してOracleデータベース・サーバーに接続できます。この機能を利用するために、クライアント構成は不要です。単純なTCP/IP環境の場合は、このメソッドをお勧めします。

親トピック: [用語集](#)

HTTP

Hypertext Transfer Protocolの略です。WebブラウザとアプリケーションWebサーバーの通信を可能にする言語を提供するプロトコルです。

親トピック: [用語集](#)

アイデンティティ管理レلم

IDの集合で、同じ管理ポリシーによって制御されるすべてのIDです。企業で、同じイントラネットにアクセスするすべての従業員は1つのレلمに属し、その企業の公開アプリケーションにアクセスするすべての外部ユーザーは別のレلمに属します。アイデンティティ管理レلمは、ディレクトリ上で特別なオブジェクト・クラスが関連付けられている特定のエントリによって表されます。

親トピック: [用語集](#)

インスタンス

[システム・グローバル領域\(SGA\)](#)およびOracleバックグラウンド・プロセスの組合せです。データベース・サーバー上でデータベースが起動している場合(コンピュータのタイプに関係なく)、OracleではSGAと呼ばれるメモリー領域が割り当てられ、1つ以上のOracleプロセスが開始されます。インスタンスのメモリーおよびプロセスは、関連付けられたデータベースのデータを効率よく管理し、データベース・ユーザーに提供します。ユーザーは、任意のインスタンスに接続し、クラスタ・データベース内の情報にアクセスできます。

親トピック: [用語集](#)

インスタンス名

Oracle Databaseインスタンスの名前です。インスタンス名は、データベース初期化パラメータ・ファイルのINSTANCE_NAMEパラメータによって識別されます。INSTANCE_NAMEは、インスタンスの[Oracleシステム識別子\(SID\)](#)に対応しています。クライアントは、接続記述子にINSTANCE_NAMEパラメータを指定すると特定のインスタンスに接続できます。

インスタンス名は、[接続記述子](#)の[接続データ](#)部分に記述されています。

親トピック: [用語集](#)

IPアドレス

ネットワーク上のノードを識別するために使用します。ネットワーク上の各コンピュータには、ネットワークIDと一意なホストIDで構成される一意なインターネット・プロトコル(IP)アドレスが割り当てられています。通常、このアドレスは、8ビットずつの10進値をピリオドで区切ったドット10進表記(192. 0. 2. 22など)で表されます。

親トピック: [用語集](#)

IPC

プロセス間通信です。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、データベースとの通信に使用するプロトコルの1つです。IPCは、TCP/IPより速いローカル接続を提供できます。

親トピック: [用語集](#)

IPv4

インターネット・プロトコル・バージョン4。IPv4は、現在の標準のIPプロトコルです。IPv4は32ビット(4バイト)アドレスを使用します。通常、ドット区切りの10進数表記で表現されます。各オクテットの10進数の値は、192. 0. 2. 22のようにピリオドで区切られます。

親トピック: [用語集](#)

IPv6

インターネット・プロトコル・バージョン6。[IPv4](#)の後継として設計されたプロトコルです。IPv6では、IPアドレスは通常、2001:0db8:0000:0000:0000:1428:57ABのように、コロンで区切られた8つのフィールドの16進数の値で表現されます。2001:DB8::1428:57ABのように、値が0のフィールドを圧縮することもあります。

親トピック: [用語集](#)

IP Version 4 (IPv4)

[「IPv4」](#)を参照してください。

親トピック: [用語集](#)

IP Version 6 (IPv6)

[「IPv6」](#)を参照してください。

親トピック: [用語集](#)

Java Database Connectivity(JDBC)ドライバ

JavaアプリケーションおよびアプレットによるOracleデータベースへのアクセスを可能にするドライバ。

親トピック: [用語集](#)

JDBC OCIドライバ

クライアント/サーバーのJavaアプリケーションで使用されるType IIドライバ。このドライバを使用するには、Oracleクライアントのインストールが必要です。

親トピック: [用語集](#)

JDBC Thinドライバ

Oracle JDBCアプレットとアプリケーション用のType 4ドライバです。このドライバはすべてJavaで記述されているため、プラットフォームに依存しません。また、クライアント側に追加のOracleソフトウェアは必要ありません。Thinドライバは、サーバーとの通信に、[Two-Task Common \(TTC\)](#) (データベース・サーバーへのアクセス用にOracleが開発したプロトコル)を使用します。

親トピック: [用語集](#)

Kerberos

分散環境におけるセキュリティを強化するネットワーク認証サービス。Kerberosは、共有秘密に依存してサード・パーティがセキュアであると見なす、信頼性のあるサード・パーティ認証システムです。ユーザーにシングル・サインオン機能とデータベース・リンク認証(MIT Kerberosのみ)を提供します。パスワード・ストレージを一元化し、PCのセキュリティを拡張できます。

親トピック: [用語集](#)

キーワード値ペア

キーワードと値の組合せです。接続記述子および多数の構成ファイルで標準的な情報単位として使用されます。キーワード値ペアはネストされることがあり、キーワードは他のキーワード値ペアを値として持つことができます。

親トピック: [用語集](#)

待機時間

ネットワークのラウンドトリップ時間です。

親トピック: [用語集](#)

LDAP Data Interchange Format (LDIF)

LDAP Data Interchange Formatは、LDAPコマンドライン・ユーティリティに使用する入力ファイルの形式を指定するための一連の規格です。

親トピック: [用語集](#)

ldap.oraファイル

Oracle Internet Directoryコンフィギュレーション・アシスタントまたはOracle Net Configuration Assistantによって作成されるファイル。次のディレクトリ・サーバー・アクセス情報が記述されています。

- ディレクトリ・サーバーのタイプ。
- ディレクトリ・サーバーの場所。

- クライアントまたはサーバーが、データベース・サービスに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Oracleコンテキスト。

Oracle Internet Directoryコンフィギュレーション・アシスタントで作成されると、ldap.oraファイルはORACLE_HOME/ldap/adminディレクトリに配置されます。Oracle Net Configuration Assistantで作成されたldap.oraファイルはORACLE_HOME/network/adminディレクトリに配置されます。

親トピック: [用語集](#)

LDIF

[「LDAP Data Interchange Format\(LDIF\)」](#)を参照してください。

親トピック: [用語集](#)

Lightweight Directory Access Protocol (LDAP)

標準的で、拡張可能なディレクトリ・アクセス・プロトコル。LDAPクライアントとサーバーが通信に使用する共通言語。業界標準の[ディレクトリ・サーバー](#)をサポートしている設計規則に関するフレームワークです。

親トピック: [用語集](#)

リンク修飾子

データベース・ユーザー名およびパスワード資格証明の代替設定を提供するために、グローバル・データベース・リンクに追加される修飾子。たとえば、グローバル・データベース・リンクsales.us.example.comにリンク修飾子fieldrepを追加できます。

```
SQL> SELECT * FROM emp@sales.us.example.com@fieldrep
```

親トピック: [用語集](#)

リスナー

[「Oracle Net Listener」](#)を参照してください。

親トピック: [用語集](#)

リスナー制御ユーティリティ

Oracle Net Servicesに含まれているユーティリティ。リスナーの起動、停止および状態の取得などの機能を制御します。

親トピック: [用語集](#)

listener.oraファイル

次の内容を識別するOracle Net Listenerの構成ファイルです。

- 一意名

- 接続リクエストを受け入れるプロトコル・アドレス
- リスニングの対象となるサービス

通常listener.oraファイルはORACLE_HOME/network/adminディレクトリに配置されます。

Oracle Databaseでは、[サービス登録](#)があるため、データベース・サービスの識別を必要としません。ただし、Oracle Enterprise Manager Cloud Controlを使用する場合は、静的サービス構成が必要となります。

親トピック: [用語集](#)

リスナー登録(LREG)

[サービス登録](#)の一部として、LREGはリスナーにインスタンス情報を登録します。LREGは、データベース初期化パラメータ・ファイルで構成される各データベース・インスタンスのインスタンス・バックグラウンド・プロセスです。

親トピック: [用語集](#)

ロード・バランシング

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への負荷を軽くする機能。

Oracle Net Servicesは、[クライアント・ロード・バランシング](#)と[接続ロード・バランシング](#)をサポートしています。

親トピック: [用語集](#)

ローカル・ネーミング

[ネーミング・メソッド](#)の1つです。これにより、各クライアントのtnsnames.oraファイルで構成および格納された情報を使用して、ネットワーク・アドレスを検索します。ローカル・ネーミングは、変更が少なく、サービスの数が少ない単純な分散ネットワークに非常に適しています。

親トピック: [用語集](#)

位置の透過性

アプリケーションがデータ表の場所を認識していなくても、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です。すべてのデータ表は単一データベースに存在している見なされ、表名に基づいて実際のデータの場所が判別されます。ユーザーは1つの文で複数のノードにあるデータを参照できます。必要に応じて、SQL文(一部)が自動的および透過的にリモート・ノードに渡されます。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ます。

親トピック: [用語集](#)

ロギング

エラー、サービス・アクティビティおよび統計情報をログ・ファイルに書き込む機能。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、管理者はログ・ファイルから追加情報を取得できます。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。

[「トレース」](#)も参照してください。

親トピック: [用語集](#)

ループバック・テスト

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、Oracle Netがデータベース・サーバーで機能していることが確認できます。

親トピック: [用語集](#)

マップ

[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です。

親トピック: [用語集](#)

Microsoft Active Directory

Microsoft Windows Serverに含まれているLDAP準拠のディレクトリ・サーバーの1つです。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します。Active Directoryでは、シングル・ログオン・プロセスを使用してネットワーク上のリソースにアクセスできます。

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ます。

親トピック: [用語集](#)

Microsoft Windows NTのシステム固有の認証

[認証方式](#)の1つです。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・ログオン・アクセスが可能になります。

親トピック: [用語集](#)

Named Pipesプロトコル

分散アプリケーションを使用するクライアント/サーバー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つです。

親トピック: [用語集](#)

ネーミング・コンテキスト

全体が、1つのディレクトリ・サーバーに常駐するサブツリーです。サブツリーは、連続したものです。つまり、サブツリーの頂点となる

エントリから始まり、下位方向にリーフ・エントリまたは従属ネーミング・コンテキストへの参照のいずれかまでを範囲とする必要があります。単一エントリから[ディレクトリ情報ツリー\(DIT\)](#)全体までをサイズの範囲とすることができます。

[Oracleコンテキスト](#)は、ネーミング・コンテキストの下に作成できます。

親トピック: [用語集](#)

ネーミング・メソッド

クライアント・アプリケーションがデータベース・サービスへの接続を試みるときに、[接続識別子](#)を[接続記述子](#)に変換するために使用する解決メソッド。Oracle Netでは、次の4種類のネーミング・メソッドを提供しています。

- [ローカル・ネーミング](#)
- [ディレクトリ・ネーミング](#)
- 簡易接続ネーミング
- [外部ネーミング](#)

親トピック: [用語集](#)

ネットワーク・サービスの別名

ディレクトリ・サーバーの[ディレクトリ・ネーミング](#)・オブジェクトの代替名。ディレクトリ・サーバーは、定義された[ネットワーク・サービス名](#)またはデータベース・サービスのネットワーク・サービス別名を格納します。ネットワーク・サービス別名のエントリには、接続記述子情報は含まれていません。かわりに、別名の対象のオブジェクトの場所のみ参照する。クライアントがネットワーク・サービス別名のディレクトリ検索を要求した場合、ディレクトリは、エントリがネットワーク・サービス別名であることを判断し、そのネットワーク・サービス別名が実際に参照しているエントリであるかのように検索を実行します。

親トピック: [用語集](#)

ネットワーク・サービス名

[接続記述子](#)に変換するサービスの単純な名前。ユーザーは、接続するサービスに対する接続文字列に、ネットワーク・サービス名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。

```
CONNECT username@net_service_name
```

ネットワーク・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ます。

- 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
- [Network Information Service \(NIS\)](#)などの外部ネーミング・サービス

親トピック: [用語集](#)

ネットワーク

ハードウェアとソフトウェアによってリンクされ、データおよび周辺装置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です。

親トピック: [用語集](#)

ネットワーク管理者

ネットワーク・コンポーネントのインストール、構成およびテストなどの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・リンクなどを管理します。

親トピック: [用語集](#)

ネットワーク文字セット

Oracleの定義では、キーワードと値のペア(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。セットには、大文字と小文字の英数字およびいくつかの特殊文字が含まれます。

親トピック: [用語集](#)

Network Information Service (NIS)

ユーザー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に配布するためのクライアント/サーバー・プロトコルです。このサービスは、以前は「Sun社のYellow Pages (yp)」と呼ばれていました。

親トピック: [用語集](#)

ネットワーク・インタフェース(NI)

Oracleのクライアント、サーバーまたは外部プロセスがOracle Net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。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。

親トピック: [用語集](#)

ネットワーク・リスナー

[「リスナー」](#)を参照してください。

親トピック: [用語集](#)

ネットワーク・オブジェクト

ネットワーク上でアドレスを直接指定できるサービス。リスナーなど。

親トピック: [用語集](#)

ネットワーク・プロトコル

「[Oracle protocol support](#)」を参照してください。

親トピック: [用語集](#)

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。[Oracle Call Interface \(OCI\)](#)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リクエストを作成できるようにします。

親トピック: [用語集](#)

ネットワーク・セッション(NS)

標準的なOracle Net接続で使用される[セッション・レイヤー](#)で、クライアント・アプリケーションとデータベース・サーバー間で接続を確立およびメンテナンスします。

親トピック: [用語集](#)

NIS

「[Network Information Service\(NIS\)](#)」を参照してください。

親トピック: [用語集](#)

ノード

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。

親トピック: [用語集](#)

オブジェクト・クラス

ディレクトリ・サーバーにおいて、名前を持った属性のグループ。属性をエントリに割り当てるときは、その属性を保持しているオブジェクト・クラスをそのエントリに割り当てます。

同じオブジェクト・クラスに関連するオブジェクトはすべて、そのオブジェクト・クラスの属性を共有します。

親トピック: [用語集](#)

OCI

Oracle Call Interface。 [Oracle Call Interface \(OCI\)](#)を参照してください。

親トピック: [用語集](#)

OPI

「[Oracleプログラム・インタフェース\(OPI\)](#)」を参照してください。

親トピック: [用語集](#)

Open Systems Interconnection (OSI)

国際標準化機構(ISO)によって開発されたネットワーク・アーキテクチャ・モデルです。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フレームワークです。

OSIアーキテクチャは、最下位レイヤーから最上位レイヤーまで、次の7つのレイヤーに分けられています。

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

各レイヤーは、1つ上位のレイヤーを使用し、下位レイヤーにサービスを提供します。

親トピック: [用語集](#)

Oracle Advanced Security

透過的データ暗号化(TDE)およびデータ・リダクションを提供するOracle製品。

親トピック: [用語集](#)

Oracle Call Interface (OCI)

Application Program Interface(API)の1つ。これにより、第3世代言語のネイティブ・プロシージャやファンクション・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ます。OCIは、C、C++、COBOL、FORTRANなど、多数の第3世代言語のデータ型、コール規則、構文およびセマンティクスをサポートします。

親トピック: [用語集](#)

Oracle Connection Manager

クライアント接続要求をその次のホップに送信するか、または直接データベース・サーバーに送信する際の経路となるルーター。Oracle Connection Managerを介して接続要求をルート指定するクライアントは、そのOracle Connection Managerで構成されている[セッションの多重化](#)、[アクセス制御](#)または[プロトコル変換](#)機能を利用できます。

親トピック: [用語集](#)

Oracle Connection Manager制御ユーティリティ

Oracle Net Servicesに含まれているユーティリティで、Oracle Connection Managerの起動、停止、状態の取得などの様々な機能を制御します。

親トピック: [用語集](#)

Oracleコンテキスト

LDAP準拠のインターネット・ディレクトリ内にあるcn=OracleContextというエントリです。このディレクトリには、Oracle Net Servicesディレクトリ・ネーミングおよびチェックサムセキュリティのエントリなど、Oracleソフトウェア関連のすべての情報が格納されています。1つのディレクトリ内に1つ以上のOracleコンテキストが存在する場合があります。Oracleコンテキスト・エントリは、ディレクトリ・ネーミング・コンテキストと関連付けることができます。

[Oracle Internet Directory](#)は、[ディレクトリ情報ツリー\(DIT\)](#)構造のルートにOracleコンテキストを自動的に作成します。このルートOracleコンテキストには、[識別名\(DN\)](#)(dn:cn=OracleContext)があります。

親トピック: [用語集](#)

Oracle Enterprise Manager Cloud Control

独立したOracle製品。Oracle製品群の管理に使用する統合的、包括的なシステム管理プラットフォームを提供するために、グラフィカル・コンソール、エージェント、共通サービスおよびツール類を統合します。

親トピック: [用語集](#)

Oracle Identity Management

インフラストラクチャ対応の展開により、すべての企業IDと、それらによる企業内の各種アプリケーションへのアクセスを集中的かつ安全に管理します。

親トピック: [用語集](#)

Oracle Internet Directory

Oracleデータベース上のアプリケーションとして実装されているディレクトリ・サーバー。分散ユーザーおよびネットワーク・リソースの情報を検索できます。Oracle Internet Directoryは、オープンなインターネット規格のディレクトリ・サーバー・アクセス・プロト

コルである[Lightweight Directory Access Protocol\(LDAP\)](#)バージョン3とOracleデータベースの高度のパフォーマンス、拡張性、耐久性および可用性を組み合わせています。

親トピック: [用語集](#)

Oracle Net

クライアント・アプリケーションからOracleデータベース・サーバーへのネットワーク・セッションを可能にする通信ソフトウェア。ネットワーク・セッションの確立後、Oracle Netはクライアント・アプリケーションとデータベース・サーバーのためのデータ伝達手段として機能します。クライアント・アプリケーションとデータベース・サーバー間でのメッセージの交換に加え、これらの間の接続を確立および維持します。このようなジョブの実行は、Oracle Netがネットワーク上の各コンピュータに配置されることにより可能になります。

親トピック: [用語集](#)

Oracle Net Configuration Assistant

インストール後に、次の内容を含む基本ネットワーク・コンポーネントを構成する処理ツール。

- 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](#)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・サーバーの使用

親トピック: [用語集](#)

Oracle Netファイアウォール・プロキシ

ファイアウォール・ベンダーによって提供される製品です。[Oracle Connection Manager](#)の機能を補います。

親トピック: [用語集](#)

Oracle Net Foundationレイヤー

クライアント・アプリケーションとサーバー間でのメッセージの交換に加え、これらの間の接続を確立および維持するネットワーク通信レイヤー。

親トピック: [用語集](#)

Oracle Net Listener

サーバー上に常駐するプロセスで、クライアントの着信接続リクエストをリスニングし、サーバーへの通信量を管理します。

クライアントがデータベース・サーバーとのネットワーク・セッションをリクエストすると、リスナーは実際のリクエストを受信します。クライアントの情報がリスナーの情報と一致すると、リスナーはデータベース・サーバーへの接続を認めます。

親トピック: [用語集](#)

Oracle Net Manager

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、Oracle Net Servicesの構成や管理のための統合環境を提供するツール。

Oracle Net Manager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

[接続識別子](#)を定義して[接続記述子](#)にマップし、ネットワークの位置とサービスの識別情報を識別します。Oracle Net Managerでは、ローカルのtnsnames.oraファイルまたはディレクトリ・サーバー内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変換する様々な方法を構成します。

- リスナー

リスナーを作成および構成し、クライアント接続を受け取ります。

親トピック: [用語集](#)

Oracle Net Services

分散された異機種間コンピューティング環境においてエンタープライズ・レベルの接続性の解決策を提供する一連のネットワーク・コンポーネント。Oracle Net Servicesは、[Oracle Net](#)、[リスナー](#)、[Oracle Connection Manager](#)、[Oracle Net Configuration Assistant](#)および[Oracle Net Manager](#)で構成されます。

親トピック: [用語集](#)

Oracleプログラム・インタフェース(OPI)

[Oracle Call Interface \(OCI\)](#)によって送信された適切な各メッセージに対する応答を受け持つネットワーク・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に戻します。

親トピック: [用語集](#)

Oracleプロトコル・サポート

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[Transparent Network Substrate\(TNS\)](#)の機能をマッピングするソフトウェア・レイヤーです。

親トピック: [用語集](#)

Oracle Real Application Clusters (Oracle RAC)

複数のインスタンスがデータ・ファイルの共有データベースにアクセスできるアーキテクチャです。Oracle RACは、Oracle Enterprise EditionおよびOracle RACで必要なクラスタ・データベース・スクリプト、初期化ファイルおよびデータ・ファイルを提

供するソフトウェア・コンポーネントでもあります。

親トピック: [用語集](#)

Oracle Rdb

Digital社の64ビット・プラットフォーム用データベースです。Oracle Rdbには独自のリスナーがあるため、クライアントはOracle Databaseと通信する場合と同じようにOracle Rdbと通信できます。

親トピック: [用語集](#)

Oracleスキーマ

[ディレクトリ・サーバー](#)に格納できる内容を判断する一連の規則。Oracleには、Oracle Net Servicesエントリなど、多くのOracleエントリ・タイプに適用される独自のスキーマがあります。Oracle Net ServicesのエントリのOracleスキーマには、そのエントリに登録される属性があります。

親トピック: [用語集](#)

Oracleシステム識別子(SID)

Oracleデータベースの特定のインスタンスを識別する名前。どのデータベースでも、そのデータベースを参照するインスタンスが1つ以上存在します。

親トピック: [用語集](#)

Oracle XML DB

Oracleデータベース・サーバーが提供する高性能なXMLストレージおよび検索テクノロジー。W3C XMLデータ・モデルに基づきます。

親トピック: [用語集](#)

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。

親トピック: [用語集](#)

OSI

「[Open Systems Interconnection\(OSI\)](#)」を参照してください。

親トピック: [用語集](#)

パケット

接続またはデータ転送がリクエストされるたびに、ネットワーク上で送信される情報のブロック。パケットに記述される情報は、connect、accept、redirect、dataなど、パケット・タイプによって異なります。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ります。

親トピック: [用語集](#)

PMONプロセス

クライアント・プロセスが失敗したときにプロセス・リカバリを実行するプロセス・モニターのデータベース・プロセスです。PMONは、キャッシュをクリーン・アップし、プロセスで使用されていたリソースを解放する役割を持ちます。PMONはディスパッチャおよびサーバー・プロセスも確認し、それらに障害が発生した場合は再起動します。

親トピック: [用語集](#)

プレゼンテーション・レイヤー

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ネットワーク通信レイヤー。[Two-Task Common\(TTC\)](#)は、プレゼンテーション・レイヤーの一例。

親トピック: [用語集](#)

プライベート・データベース・リンク

あるユーザーが排他的に使用するデータベース・リンクです。

「[データベース・リンク](#)」および「[パブリック・データベース・リンク](#)」も参照してください。

親トピック: [用語集](#)

プロファイル

クライアントまたはサーバー上でのOracle Net Services機能の有効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います。

親トピック: [用語集](#)

プロトコル

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。

親トピック: [用語集](#)

プロトコル・アドレス

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、クライアントとリクエストの受信者([リスナー](#)、[Oracle Connection Manager](#)など)が同じプロトコル・アドレスで構成されます。クライアントは、このアドレスを使用して接続リクエストを特定のネットワーク・オブジェクト位置に送信し、受信者はこのアドレスでリクエストのリスニングを行います。クライアントと接続受信者に対して同じプロトコルをインストールし、同じアドレスを構成することが重要です。

親トピック: [用語集](#)

プロトコル変換

Oracle Connection Managerの機能です。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します。

親トピック: [用語集](#)

プロトコル・スタック

特定の[プレゼンテーション・レイヤー](#)と[セッション・レイヤー](#)の組合せを示します。

親トピック: [用語集](#)

プロキシ・サーバー

実際のサーバーのかわりとなるサーバーで、クライアントの接続要求を実際のサーバーや他のプロキシ・サーバーに転送します。プロキシ・サーバーは、アクセス制御、データとシステムのセキュリティ、監視およびキャッシュ機能を提供します。

親トピック: [用語集](#)

パブリック・データベース・リンク

すべてのユーザーがアクセスできるローカル・データベース上に、DBAが作成するデータベース・リンク。

「[データベース・リンク](#)」および「[プライベート・データベース・リンク](#)」も参照してください。

親トピック: [用語集](#)

RADIUS

Remote Authentication Dial-In User Service (RADIUS)は、リモート・アクセス・サーバーと中央サーバーとの通信を可能にするクライアント・サーバー・プロトコルおよびソフトウェアです。これは、ダイヤルイン・ユーザーを認証し、要求されたシステムまたはサービスへのアクセスを承認するために役立ちます。

親トピック: [用語集](#)

レルムOracleコンテキスト

各[アイデンティティ管理レルム](#)に含まれるOracleコンテキスト。次の情報が格納されます。

- アイデンティティ管理レルムのユーザー・ネーミング・ポリシー(ユーザーのネーミングおよび検索方法)。
- 必須認証属性。
- アイデンティティ管理レルムでのグループの場所
- アイデンティティ管理レルムに関する権限の割当て(誰がレルムに別のユーザーを追加する権限を持つかなど)。
- 認可など、そのレルムのアプリケーション固有のデータ

親トピック: [用語集](#)

RDBMS

リレーショナル・データベース管理システム。

親トピック: [用語集](#)

RDN

[「相対識別名\(RDN\)」](#)を参照してください。

親トピック: [用語集](#)

相対識別名(RDN)

ローカルの、最も細かいレベルのエントリ名。エントリを一意に指定するための修飾子を持ちません。完全修飾のX.500名です。たとえば、cn=sales, dc=us, dc=example, dc=com, cn=salesはRDNです。

親トピック: [用語集](#)

ルートOracleコンテキスト

[Oracle Identity Management](#)インフラストラクチャにおいて、ルートOracleコンテキストは、インフラストラクチャのデフォルトの[アイデンティティ管理レルム](#)へのポインタを含むOracle Net Services内のエントリです。また、指定されたレルムの簡単な名前によって、アイデンティティ管理レルムを検索する方法に関する情報を含んでいます。

親トピック: [用語集](#)

RPC

リモート・プロシージャ・コール。

親トピック: [用語集](#)

SDP

Sockets Direct Protocol(SDP)の略です。

親トピック: [用語集](#)

Secure Sockets Layer (SSL)

ネットワーク接続を保護するための業界標準プロトコル。SSLは、公開キーインフラストラクチャ(PKI)を使用して、認証、暗号化およびデータ整合性を提供します。[Transport Layer Security \(TLS\)](#)プロトコルは、SSLプロトコルの後継です。

[Transport Layer SecurityとSecure Sockets Layerの違い](#)を参照してください。

親トピック: [用語集](#)

サーバー・パラメータ・ファイル

Oracle Databaseホストでメンテナンスされる初期化パラメータ設定を含むバイナリ・ファイルです。このファイルをテキスト・エディタで手動で編集することはできません。サーバー・パラメータ・ファイルは、初めCREATE SPFILE文を使用して、テキスト形式の初期化パラメータ・ファイルから作成するか、直接作成します。

親トピック: [用語集](#)

サーバー・プロセス

データベースのかわりにクライアント・リクエストを処理するデータベース・プロセス。

親トピック: [用語集](#)

サービス

様々なクライアントからのリクエストに回答したり、操作を実行したりするプログラム。データベースでのサービスは、クライアントのためにデータの格納と取出しを行うことです。

親トピック: [用語集](#)

サービス・ハンドラ

リスナーからデータベース・サーバーへの接続ポイントとして機能するプロセスです。[ディスパッチャ](#)または[専用サーバー](#)があります。

親トピック: [用語集](#)

サービス名

データベースの論理的表現で、クライアントに対してデータベースはこの形式で表現されます。サービス名は、インストールまたはデータベースの作成時に入力された、データベース名とドメイン名の組合せである[グローバル・データベース名](#)の文字列です。グ

ローバル・データベース名がわからない場合は、初期化パラメータ・ファイルのSERVICE_NAMESパラメータの値から確認できません。

サービス名は、[接続記述子](#)の[接続データ](#)部分に含まれています。

親トピック: [用語集](#)

サービス登録

[リスナー登録\(LREG\)](#)プロセスが[リスナー](#)に自動的に情報を登録する機能です。この情報はリスナーに登録されるため、listener.oraファイルはこの静的情報で構成する必要はありません。

サービス登録では、次の情報をリスナーに提供します。

- 実行中の各データベース・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](#)または[専用サーバー](#))。これによって、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リクエストを最適に処理できるディスパッチャを判断します。すべてのディスパッチャがブロックされている場合、リスナーは接続用の専用サーバーを生成できます。

親トピック: [用語集](#)

セッション・データ・ユニット(SDU)

Oracle Net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。Oracle Netがバッファ内のデータを送信するのは、データ送信がリクエストされたとき、またはバッファがデータでいっぱいになったときです。

親トピック: [用語集](#)

セッション・レイヤー

[プロトコル・アドレス](#)のエンティティが必要とするサービスを提供するネットワーク・レイヤーであり、エンティティで対話の編成と同期およびデータ交換の管理を行えるようにします。このレイヤーは、クライアントとサーバー間のネットワーク・セッションを確立、管理および終了します。セッション・レイヤーの例には、[ネットワーク・セッション\(NS\)](#)があります。

親トピック: [用語集](#)

セッションの多重化

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つのネットワーク接続で転送します。

親トピック: [用語集](#)

共有サーバー

サポートされるユーザー数を増やすために、多数のクライアント・プロセスがごく少数のサーバー・プロセスを共有できるように構成されたデータベース・サーバーです。共有サーバー構成では、多数のクライアント・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ります。サーバー・プロセスの共有プールの中のアイドル状態の共有サーバー・プロセスは、共通キューから要求を取り出します。したがって、サーバー・プロセスの小規模プールによる多数のクライアントの処理が可能になります。[専用サーバー](#)と対比される機能です。

親トピック: [用語集](#)

共有サーバー・プロセス

[共有サーバー](#)構成で使用するプロセス・タイプです。

親トピック: [用語集](#)

SID

「[Oracleシステム識別子\(SID\)](#)」を参照してください。

親トピック: [用語集](#)

単一クライアント・アクセス名(SCAN)

Oracle Real Application Clusters環境で使用され、クライアントに単一の名前を与えて、クラスタ内で実行される任意のOracle Databaseにアクセスできるようにする機能。

親トピック: [用語集](#)

シングル・サインオン

ユーザーが単一のパスワードを使用して異なるサーバーにログインできる機能です。ユーザーがアクセス権限を持つすべてのサーバーに対して認証が可能。

親トピック: [用語集](#)

sqlnet.oraファイル

次の内容を指定する、クライアントまたはサーバー用の構成ファイル。

- 未修飾サービス名またはネットワーク・サービス名に追加するクライアント・ドメイン。
- 名前を解決する際にクライアントで使用するネーミング・メソッドの順序。
- 使用するロギング機能とトレース機能。
- 接続のルート。

- 外部ネーミング・パラメータ。
- セキュリティ・パラメータ。

通常sqlnet.oraファイルはORACLE_HOME/network/adminディレクトリに配置されます。

親トピック: [用語集](#)

SSL

「[Secure Sockets Layer\(SSL\)](#)」を参照してください。

親トピック: [用語集](#)

システム・グローバル領域(SGA)

Oracle [インスタンス](#)のデータおよび制御情報を含む共有メモリー構造のグループ。

親トピック: [用語集](#)

TCP/IP

Transmission Control Protocol/Internet Protocol。ネットワークを介したクライアント/サーバー間の通信に使用する、標準の通信プロトコル。

親トピック: [用語集](#)

TLSプロトコル付きTCP/IP

クライアント上のOracleアプリケーションを、[TCP/IP](#)および[Transport Layer Security \(TLS\)](#)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するプロトコルです。

親トピック: [用語集](#)

ティック

メッセージをクライアントからサーバーまたはサーバーからクライアントに送信し、処理するために必要な時間。

親トピック: [用語集](#)

Transport Layer Security(TLS)

ネットワーク接続を保護するための業界標準プロトコル。TLSプロトコルはSecure Sockets Layer (SSL)プロトコルの後継です。公開キー・インフラストラクチャ(PKI)を使用した認証、暗号化およびデータの整合性を提供します。

[Transport Layer SecurityとSecure Sockets Layerの違い](#)を参照してください。

親トピック: [用語集](#)

TNS

[「Transparent Network Substrate\(TNS\)」](#)を参照してください。

親トピック: [用語集](#)

tnsnames.oraファイル

[ネットワーク・サービス名](#)を[接続記述子](#)にマップする構成ファイル。このファイルは、[ローカル・ネーミング](#)・メソッドで使用されます。tnsnames.oraファイルは、通常はORACLE_HOME/network/adminディレクトリに存在します。

親トピック: [用語集](#)

トレース

操作に関する詳細情報を出力ファイルに書き込むユーティリティです。トレース・ユーティリティによって、操作が実行されるときのイベントを示す詳細な文が生成されます。管理者は、異常発生時の診断にトレース・ユーティリティを使用します。通常、トレース機能はオンになっていません。

[「ロギング」](#)も参照してください。

親トピック: [用語集](#)

透過的アプリケーション・フェイルオーバー(TAF)

Oracle RACやOracle Fail Safeなどの高可用性環境を対象としたランタイム・フェイルオーバーであり、アプリケーションからサービスへの接続のフェイルオーバーおよび再確立を意味します。TAFにより、クライアント・アプリケーションは接続障害の発生時にデータベースに自動的に再接続でき、実行中だったSELECT文を再開することも可能です。この再接続は、Oracle Call Interface(OCI)ライブラリ内から自動的に実行されます。

親トピック: [用語集](#)

Transparent Network Substrate (TNS)

基盤となるテクノロジーの1つです。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[Oracle Net Foundationレイヤー](#)に組み込まれています。

親トピック: [用語集](#)

トランスポート

データ・フロー制御とエラー・リカバリ方式を通じてエンドツーエンドの信頼性を維持するネットワーキング・レイヤー。[Oracle Net Foundationレイヤー](#)では、[Oracleプロトコル・サポート](#)をトランスポート・レイヤーに使用します。

親トピック: [用語集](#)

TTC

「[Two-Task Common\(TTC\)](#)」を参照してください。

親トピック: [用語集](#)

Two-Task Common (TTC)

標準的なOracle Net接続で使用される[プレゼンテーション・レイヤー](#)です。クライアントとサーバーの異なる文字セットや型式の、文字セットとデータ型の変換を提供します。

親トピック: [用語集](#)

UPI

ユーザー・プログラム・インタフェース。

親トピック: [用語集](#)

バーチャル・サーキット

[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リクエストおよび応答を目的とします。ディスパッチャは、リクエストが到着するとバーチャル・サーキットを共通キューに配置します。アイドル状態の共有サーバーは、共通キューからバーチャル・サーキットを取り出してリクエストを処理し、そのバーチャル・サーキットが、共通キューから別のバーチャル・サーキットを取り出す前に、取り出したバーチャル・サーキットを放棄します。

親トピック: [用語集](#)

WebDAV°プロトコル

World Wide Web分散オーサリングおよびバージョンングです。[HTTP](#)に対する一連の拡張機能を持つプロトコルで、ユーザーがリモートWebサーバー上のファイルを管理できます。

親トピック: [用語集](#)

索引

数字 [A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [Y](#)

数字

- 1521 ポート [9.4.1.1](#)
 - 使用していない場合 [9.2.2](#)
-

A

- アクセス制御リスト(ACL)
 - 説明 [3.2.6.2.2](#)
 - データベースに対するアクセス制御
 - Oracle Connection Manager [1.1.5.1](#), [10.2.1.2](#)
 - sqlnet.oraパラメータ [12.4](#)
 - ACTION_LISTネットワーク・パラメータ [10.1](#), [10.2.1](#)
 - ADAPTERSユーティリティ [16.2.2](#)
 - 「アドレス・リスト・オプション」ダイアログ・ボックス [13.1.2](#)
 - ADDRESSネットワーク・パラメータ [10.2.1](#)
 - ADR
 - 「自動診断リポジトリ」を参照
 - ADR_BASE_listener_nameログ・パラメータ [16.6.4.2](#)
 - ADR_BASEログ・パラメータ [16.6.4.1](#), [16.6.4.3](#)
 - ADRCI [16.1.1](#)
 - アプリケーションWebサーバー [1.1.1.2](#)
 - アプリケーションWebサーバー [1.1.1.2.2](#)
 - アーキテクチャ
 - Oracle Connection Manager [5.6](#)
 - 属性
 - 定義 [3.2.1](#)
 - 監査証跡
 - 説明 [16.6.8.1.1](#)
 - 情報を使用するためのスクリプト [16.6.8.1.1](#)
 - 自動診断リポジトリ [16.1](#)
 - 自動開始リスナー [9.5.1](#)
-

B

- BACKUPネットワーク・パラメータ [13.3.3](#)
- BEQUEATH_DETACHネットワーク・パラメータ [12.5](#)

- Bequeathプロトコル [2.4.2](#)
 - バッファ・フラッシング、説明 [1.1.4.3](#)
-

C

- 集中化されたネーミング [3](#)
 - 接続の確立 [3](#)
- Oracle Net Managerの「TCP/IPクライアント・アクセス権のチェック」オプション [12.4](#)
- クライアントの構成
 - 接続時フェイルオーバー [2.6.1](#)
 - デフォルト・ドメイン [12.3.1](#)
 - リスナーのロード・バランシング要求 [2.6.3](#), [13.1.2](#)
 - ローカル・ネーミング [8.2.2](#)
 - ログ・ファイル [16.6.4.1](#)
 - ネットワーク・サービス名 [6.6](#)
 - Oracle Connection Managerアドレス [10.2.2](#)
 - Oracle Rdb接続 [13.6.3](#)
 - 共有サーバー [11.3](#)
 - sqlnet.logファイル [16.6.4.1](#)
 - トレース・ファイル [16.7.2.3](#)
 - 透過的アプリケーション・フェイルオーバー(TAF) [2.6.2](#)
- クライアント・ロード・バランシング
 - 構成 [13.1.2](#)
 - 説明 [2.6.3](#)
- Oracle Net Managerの「クライアント登録ID」フィールド [12.5](#)
- Oracle Net Managerの「アクセスを許可されるクライアント」フィールド [12.4](#)
- クライアント/サーバー接続
 - ルーティング [14.8.2](#)
- Oracle Net Managerの「アクセスを許可されないクライアント」フィールド [12.4](#)
- クライアントのトラブルシューティング [16.2.2](#)
- CMADMIN(Connection Manager Administration) [5.6](#)
- cman.logファイル [16.6.3](#)
- cman.oraファイル
 - ログ・パラメータ
 - ADR_BASE [16.6.4.3](#)
 - DIAG_ADR_ENABLED [16.6.4.3](#)
 - EVENT_GROUP [16.6.4.3](#)
 - LOG_DIRECTORY [16.6.4.3](#)
 - LOG_LEVEL [16.6.4.3](#)
 - パラメータ
 - ACTION_LIST [10.1](#), [10.2.1](#)
 - ADDRESS [10.2.1](#)
 - RULE_LIST [10.2.1.2](#)

- 設定 [10.1](#), [10.2.1](#)
- トレース・パラメータ
 - TRACE_DIRECTORY [16.7.2.1](#)
 - TRACE_FILELEN [16.7.2.1](#)
 - TRACE_FILENO [16.7.2.1](#)
 - TRACE_LEVEL [16.7.2.1](#)
 - TRACE_TIMESTAMP [16.7.2.1](#)
- CMAN DISPATCHERSパラメータ [10.2.3.2](#)
 - 「DISPATCHERSパラメータ」も参照
- CMGW(Connection Manager Gateway) [5.6](#)
- COLOCATION_TAGパラメータ [13.2.3](#)
- 構成
 - Traffic DirectorモードのOracle Connection Manager [10.2.4.11](#)
 - Transport Layer Security [10.2.1.1](#), [10.2.4.10](#)
- 構成
 - アクセス制御 [10.2.1.2](#), [12.4](#)
 - クライアント
 - Exadirect [14.6.3](#)
 - クライアント
 - クライアント・ロード・バランシング [13.1.2](#)
 - 接続時フェイルオーバー [2.6.1](#), [13.1.2](#)
 - デフォルト・ドメイン [12.3.1](#)
 - ローカル・ネーミング [8.2.2](#)
 - プロファイルのネーミング・メソッド [12.3.2](#)
 - ネットワーク・サービス名 [6.6](#)
 - 透過的アプリケーション・フェイルオーバー(TAF) [2.6.2](#)
 - 接続ロード・バランシング [13.2](#)
 - プロファイルでの接続要求 [12.3.3](#)
 - 接続要求のタイムアウト [14.7](#)
 - データベース・クライアント
 - Exadirect [14.6.3](#)
 - データベース・サーバー
 - アクセス制御 [12.4](#)
 - 接続要求のタイムアウト [14.7](#)
 - Exadirect [14.6.2](#)
 - セッション・データ・ユニット(SDU) [14.5.2](#), [14.5.3](#)
 - 共有サーバー [11.1](#)
 - ディレクトリ・ネーミング [8.3](#)
 - Exadirectプロトコル [14.6](#)
 - 外部ネーミング [8.7](#)
 - 外部プロシージャ接続 [13.6.1](#)
 - 異機種間サービス接続 [13.6.2](#), [13.6.2.1](#)
 - インスタンス・ロール [13.4](#)

- リスナー [9](#)
 - 接続要求のタイムアウト [14.7](#)
 - ディレクトリ・ネーミング [8.3](#)
 - 外部プロシージャ [13.6.1.1.1](#)
 - ローカル・ネーミング [8.2.2](#)
 - キュー・サイズ [9.4.2](#)
 - ローカル管理 [3](#)
 - ローカル・ネーミング [8.2](#)
 - 複数のプロトコル・アドレス [13.1.1](#)
 - ネーミング・メソッド [8](#)
 - ネットワーク・ドメイン、デフォルト [12.3.1](#)
 - ネットワーク・サービスの別名 [8.3](#)
 - ネットワーク・サービス名
 - ディレクトリ・ネーミング [8.3](#)
 - 外部ネーミング [8.7](#)
 - ローカル・ネーミング [8.2.2](#)
 - Oracle Advanced Security [12.7](#)
 - Oracle Rdbデータベース接続 [13.6.3](#)
 - プライマリ・インスタンスおよびセカンダリ・インスタンス [13.4](#)
 - プロトコル・アドレス [9.4.1](#)
 - クイック・リファレンス [7.4](#)
 - SDPプロトコル [14.5](#)
 - サーバー
 - 接続要求のタイムアウト [14.7](#)
 - Exadirect [14.6.2](#)
 - セッション・データ・ユニット(SDU) [14.5.2](#), [14.5.3](#)
 - 共有サーバー [11.1](#)
 - サービス登録 [10.2.3.1](#), [10.2.4.5](#)
 - セッション・データ・ユニット(SDU)サイズ [13.1.3](#)
 - セッションの多重化 [10.2.3.2](#)
- 接続ブローカ [2.4.3](#)
 - 接続ロード・バランシング [13.2](#)
 - 構成
 - 専用サーバー [13.2.2](#)
 - 共有サーバー [13.2.1](#)
 - 説明 [2.6.4](#)
 - サービス登録 [9.1](#)
 - 接続 [2.3.3.2](#)
 - エラー回避のためのリスナー・キュー・サイズの調整 [1.1.4.1](#), [9.4.2](#)
 - bequeath [2.4.2](#)
 - 同時、接続数の増加 [9.4.2](#)
 - 接続文字列、入力 [2.5.3](#)
 - 専用サーバー [12.3.3](#)

- ディレクトリ・ネーミング [8.3](#)
- 外部プロシージャ [13.6.1](#)
- 異機種間サービス [13.6.2](#)
- ローカル・ネーミング [8.2.2](#)
- NIS外部ネーミング [8.7](#)
- Oracle Connection Manager [12.3.3](#)
- Oracle Rdbデータベース [13.6.3](#)
- CONNECTIONS(CONまたはCONN)属性 [11.1](#)
- Oracle Net Managerの「接続タイムアウト」フィールド [12.5](#)
- 接続時フェイルオーバー
 - およびGLOBAL_DNAMEパラメータ [13.5.1](#)
 - 構成 [13.1.2](#)
 - サービス登録 [9.1](#)

D

- データベース常駐接続プーリング [2.4.3](#)
- データベース・サーバーの構成
 - アクセス制御 [12.4](#)
 - 共有サーバーにリソースの割当て [11.1.1](#)
 - 接続要求のタイムアウト [14.7](#)
 - ローカル・ネーミング [8.2.2](#)
 - ログ・ファイル [16.6.4.1](#)
 - ループバック・テスト [16.2.1](#)
 - 共有サーバー [11.1](#)
 - トレース・ファイル [16.7.2.3](#)
- データ転送、最大化 [1.1.4.2](#)
- DBMS_SFW_ACL_ADMINパラメータ [9.4.3](#)
- 専用接続 [2.4.3](#)
- 専用サーバー
 - 構成 [11.3](#)
- 専用サーバー
 - bequeathedセッション [2.4.2](#)
 - 接続記述子構成 [2.3.3.2](#)
 - 定義 [1.1.3](#)
 - 説明 [2.4.2](#), [5.5](#)
 - 共有サーバーとの違い [1.1.3](#)
 - 接続のルーティング [12.3.3](#)
- サービス拒否攻撃 [14.7](#)
- DIAG_ADR_ENABLED_listener_nameログ・パラメータ [16.6.4.2](#)
- DIAG_ADR_ENABLEDログ・パラメータ [16.6.4.1](#), [16.6.4.3](#)
- 診断
 - 「トラブルシューティング」を参照

- ダイレクト・ハンドオフ
 - 説明 [2.4.1](#)
 - イベントのlistener.log格納 [16.6.8.3](#)
- ディレクトリの構成
 - OracleNetAdminsグループへのユーザーの追加 [7.2.1](#)
 - エクスポート
 - tnsnames.oraファイルからのネットワーク・サービス名 [8.5](#)
 - 移行
 - tnsnames.oraファイルからのネットワーク・サービス名 [8.5](#)
- ディレクトリ情報ツリー(DIT)、定義 [3.2.1](#)
- ディレクトリ・ネーミング [16.3.14](#)
 - メリットおよびデメリット [2.5.1](#)
 - アーキテクチャ [3.2.1](#)
 - 認証方式
 - ネイティブ [3.2.6.2.1](#)
 - 簡易 [3.2.6.2.1](#)
 - 厳密 [3.2.6.2.1](#)
 - TLS [3.2.6.2.1](#)
 - 構成 [8.3](#)
 - データベース・サービスへの接続 [3.2.5](#)
 - 接続 [8.3](#)
 - 接続
 - エントリの完全修飾名の使用 [3.2.1.4](#)
 - エントリの相対名の使用 [3.2.1.3](#)
 - 接続の確立 [8.3](#)
 - エクスポート
 - tnsnames.oraファイルからのネットワーク・サービス名 [8.5](#)
 - 完全修飾ネーミング [3.2.1.4](#)
 - Java Database Connectivity(JDBC)
 - OCIドライバ [3.2.1.3](#), [3.2.1.4](#)
 - Thinドライバ [3.2.1.3](#), [3.2.1.4](#)
 - Java Database Connectivity(JDBC)シン・ドライバ [3.2](#), [3.2.1.4](#)
 - リスナー構成 [8.3](#)
 - 移行
 - tnsnames.oraファイルからのネットワーク・サービス名 [8.5](#)
 - オブジェクト・クラス [3.2.6.3](#)
 - オブジェクト
 - データベース・サービス [3.2.1](#)
 - ネットワーク・サービスの別名 [3.2.3](#)
 - ネットワーク・サービス名 [3.2.1](#)
 - Oracleコンテキスト [3.2.2](#)
 - OracleContextAdminsグループ [3.2.4](#), [3.2.6.2.2](#)
 - OracleDBCreatorsグループ [3.2.4](#), [3.2.6.2.2](#)

- OracleNetAdminsグループ [3.2.6.2.2](#), [7.2.1](#), [8.3](#)
- Oracle Net Manager [3.2.4](#)
- Oracleスキーマ [3.2.6.3](#)
- 概要 [3.2](#)
- 推奨 [2.5.1](#)
- セキュリティ [3.2.6.2](#)
- トラブルシューティング [16.3.14](#)
- 相対ネーミングを使用 [3.2.1.3](#)
- ディレクトリ・サーバー移行ウィザード [7.1.2.3.2](#), [8.5](#)
- ディレクトリ・サーバー
 - 属性 [3.2.1](#)
 - ディレクトリ情報ツリー [3.2.1](#)
 - 識別名(DN) [3.2.1](#)
 - エントリ [3.2.1](#)
 - Microsoft Active Directory [3.2.7](#)
 - Oracleコンテキスト [3.2.2](#)
 - 概要 [1.1.2.2](#)
 - パフォーマンス [3.2.6.1](#)
 - 相対識別名(RDN) [3.2.1](#)
 - セキュリティ [3.2.6.2](#)
- Oracle Net Configuration Assistantの「ディレクトリ使用構成」オプション [7.1.4](#)
- DISABLE_OOBネットワークング・パラメータ [12.5](#)
- Oracle Net Managerの「アウト・オブ・バンド・ブレイクの使用禁止」オプション [12.5](#)
- ディスパッチャ [1.1.3](#), [5.5.1](#)
 - 説明 [1.1.3](#), [2.4.1](#), [5.5.1](#)
- DISPATCHERS(DISまたはDISP)属性 [11.1](#)
- DISPATCHERS初期化パラメータ [11.1](#)
 - CONNECTIONS属性 [11.1](#)
 - DISPATCHERS属性 [11.1](#)
 - LISTENER属性 [9.2.2](#), [9.2.3](#), [11.1](#)
 - MULTIPLEX属性 [11.1](#)
 - SESSIONS属性 [11.1](#)
- DISPATCHERSパラメータ
 - MULTIPLEX属性 [10.2.3.2](#)
 - PROTOCOL属性 [10.2.3.2](#)
- 識別名(DN)、定義 [3.2.1](#)
- ネットワーク管理者のタスク [7.4](#)
- 動的な登録 [2.4.3](#)

E

- 簡易接続ネーミング
 - メリットおよびデメリット [2.5.1](#)

- 推奨 [2.5.1](#)
- 構文例 [8.1.3](#)
- 簡易接続ネーミング・メソッド [8.1.1](#)
- 簡易接続プラス [8.1.2](#)
- 有効化
 - 外部パスワード・ストア [10.2.4.3.1](#)
- エントリ, 定義 [3.2.1](#)
- Oracle Net Managerの「環境」フィールド [13.6.1](#)
- ENVISネットワークング・パラメータ [13.6.1](#)
- エラー・メッセージ
 - Oracleサポート・サービスへの連絡 [16.8](#)
 - トレース・ファイル [16.7.4.3](#)
 - ORA-12154 [16.3.2](#)
 - ORA-12170 [14.7](#), [16.3.3](#)
 - ORA-12203
 - サンプルのエラー・スタック [16.6.1.1](#)
 - ORA-12514 [16.3.5](#)
 - ORA-12520 [16.3.6](#)
 - ORA-12521 [16.3.7](#)
 - ORA-12525 [14.7](#), [16.3.8](#)
 - ORA-12526 [5.4](#)
 - ORA-12527 [5.4](#)
 - ORA-12528 [5.4](#)
 - ORA-12533 [16.3.9](#)
 - ORA-12535 [12.5](#)
 - ORA-12547 [14.7](#)
 - ORA-12608 [12.5](#)
 - ORA-12609 [12.5](#)
 - ORA-12637 [14.7](#)
 - 解決 [16](#)
 - ログ・ファイルによる追跡 [16.6.7](#)
- エラー・スタック
 - 説明 [16.6.1](#)
 - ログ・ファイルのエントリ [16.6.7](#)
 - サンプル [16.6.1.1](#)
 - 代表的なレイヤー [16.6.1](#)
- EVENT_GROUPログ・パラメータ [16.6.4.3](#)
- Exadirectプロトコル
 - 構成 [14.6](#)
- エクスポート
 - tnsnames.oraファイルからディレクトリへのネットワーク・サービス名 [8.5](#)
- 外部ネーミング
 - メリットおよびデメリット [2.5.1](#)

- 構成 [8.7](#)
 - Network Information Service [2.5.1](#)
 - NIS [12.3.2](#)
 - 推奨 [2.5.1](#)
 - 外部パスワード・ストア
 - 有効化 [10.2.4.3.1](#)
 - 外部プロシージャ
 - 接続の構成 [13.6.1](#)
 - 説明 [13.6.1](#)
 - extprocエージェント [13.6.1](#)
 - EXTPROC_DLLS環境変数 [13.6.1](#)
 - extprocエージェント [13.6.1](#)
-

F

- ファイルオーバー
 - 接続時 [2.6.1](#), [13.1.2](#)
 - 透過的アプリケーション・ファイルオーバー(TAF) [2.6.2](#), [13.3](#)
 - FAILOVER_MODEネットワーク・パラメータ [13.3](#), [13.3.3](#)
 - FAILOVERネットワーク・パラメータ [13.1.2](#), [13.2.2](#)
 - ファイアウォール
 - ファイアウォール・アクセス制御 [1.1.5.1](#)
 - FIREWALLパラメータ [9.4.3](#)
 - FTPプレゼンテーション [4.3](#)
 - 専用サーバー構成 [5.5.2](#)
 - 共有サーバー構成 [5.5.1](#)
 - FTPプロトコル [1.1.1.2.2](#)
 - 完全修飾ネーミング
 - Java Database Connectivity(JDBC)
 - OCIドライバ [3.2.1.3](#), [3.2.1.4](#)
 - Thinドライバ [3.2.1.4](#)
 - ディレクトリ・ネーミングを使用 [3.2.1.4](#)
-

G

- GLOBAL_NAMEネットワーク・パラメータ [13.1.3](#), [13.6.3](#)
 - グローバル・データベース名
 - リスナーの構成 [9.2.1](#)
 - 定義 [2.3.1](#)
 - 説明 [9.2.1](#)
 - Oracle Net Managerの「グローバル・データベース名」フィールド [13.1.3](#), [13.6.3](#)
-

H

- 異機種間サービス
 - 接続の構成 [13.6.2](#)
 - 説明 [13.6.2](#)
 - ホスト・ネーミング
 - 要件 [8.1.4](#)
 - HSネットワーク・パラメータ [13.1.3](#)
 - HTTPプレゼンテーション [4.3](#)
 - 専用サーバー構成 [5.5.2](#)
 - 共有サーバー構成 [5.5.1](#)
 - HTTPプロトコル [1.1.1.2](#)
-

I

- INBOUND_CONNECT_TIMEPUT_listener_nameネットワーク・パラメータ [14.7](#)
- InfiniBandネットワーク通信
 - Exadirectサポートの構成 [14.6](#)
 - SDPプロトコル・サポートの構成 [14.5](#)
- 初期化パラメータ・ファイル
 - DISPATCHERSパラメータ [11.1](#)
 - INSTANCE_NAMEパラメータ [1.2](#), [2.1](#)
 - LOCAL_LISTENERパラメータ [9.2.2](#), [9.4.1.1](#)
 - REMOTE_LISTENERパラメータ [9.2.3](#), [9.2.4](#)
 - SERVICE_NAMESパラメータ [2.2](#), [2.3.1](#)
- インストール
 - デフォルトの構成
 - リスナー [9.3](#)
 - ローカル・ネーミング [8.2.1](#)
 - プロファイル [12.2](#)
- INSTANCE_NAMEネットワーク・パラメータ [1.2](#), [2.1](#), [2.3.1](#), [13.1.3](#), [13.4](#)
- Oracle Net Managerの「インスタンス名」フィールド [13.1.3](#)
- インスタンス・ロールの構成 [13.4](#)
 - TAFの接続 [13.4](#)
 - プライマリおよびセカンダリ・インスタンスへの接続 [13.4](#)
 - 特定のインスタンスへの接続 [13.4](#)
- IPv6
 - アドレス [4.4.1.1](#), [4.4.1.1.2](#)
 - 構成 [4.4.1.2](#)
 - 接続記述子 [2.3.1.1](#)
 - ネットワーク・プロトコル [4.1.1.4](#)
 - Oracle Databaseコンポーネント [4.4.1.4](#)
 - TCP/IPプロトコル [4.4.1](#)

J

- Java Database Connectivity(JDBC)
 - ドライバ [1.1.1.1.1](#)
 - OCIドライバ [4.2](#)
 - 完全修飾ネーミング・サポート [3.2.1.3](#), [3.2.1.4](#)
 - 相対ネーミング・サポート [3.2.1.3](#)
 - Thinドライバ [4.2](#)
 - ディレクトリ・ネーミング・サポート [3.2](#), [3.2.1.4](#)
 - 完全修飾ネーミング・サポート [3.2.1.4](#)
 - 相対ネーミング・サポート [3.2.1.3](#)
 - JavaNet [4.2](#)
 - JavaNetレイヤー [1.1.1.2.2](#)
 - JavaTTC [4.2](#)
 - JDBC
 - 「Java Database Connectivity (JDBC)」を参照
-

K

- キー・ベースのルーティング
 - 構成 [14.8.2](#)
 - データベース要求 [14.8.1](#)
-

L

- listener.logファイル [16.6.2](#), [16.6.3](#)
- listener.oraファイル
 - ログ・パラメータ
 - ADR_BASE_listener_name [16.6.4.2](#)
 - DIAG_ADR_ENABLED_listener_name [16.6.4.2](#)
 - LOG_DIRECTORY_listener_name [16.6.4.2](#)
 - LOG_FILE_listener_name [16.6.4.2](#)
 - パラメータ
 - ENV5 [13.6.1](#)
 - INBOUND_CONNECT_TIMEOUT_listener_name [14.7](#)
 - ORACLE_HOME [13.6.1](#), [13.6.2.1](#)
 - PROGRAM [13.6.1](#), [13.6.2.1](#)
 - SID_NAME [13.6.1](#), [13.6.2.1](#)
 - トレース・パラメータ
 - TRACE_DIRECTORY_listener_name [16.7.2.2](#)
 - TRACE_FILE_listener_name [16.7.2.2](#)

- TRACE_FILEAGE_listener_name [16.7.2.2](#)
 - TRACE_FILEAGE_SERVER [16.7.2.2](#)
 - TRACE_FILELEN_listener_name [16.7.2.2](#)
 - TRACE_FILENO_listener_name [16.7.2.2](#)
 - TRACE_LEVEL_listener_name [16.7.2.2](#)
 - TRACE_TIMESTAMP_listener_name [16.7.2.2](#)
- listener.trcファイル [16.7.1](#)
- LISTENER(LISまたはLIST)属性 [9.2.2](#), [9.2.3](#), [11.1](#)
- Oracle Net Configuration Assistantの「リスナー構成」オプション [7.1.4](#)
- リスナー制御ユーティリティ
 - コマンド
 - SERVICES [6.3](#), [9.5.4](#)
 - START [6.3](#)
 - STATUS [9.5.3.1](#)
 - リスナーの起動 [9.5.1.1](#)
 - リスナーの停止 [9.5.1.1](#)
 - 使用 [7.3](#)
- リスナーのリダイレクト [9.6](#)
- リスナー [2.3.2](#), [13.6.2.1](#)
 - キュー・サイズの調整 [9.4.2](#)
 - 自動開始 [9.5.1](#)
 - クライアント・ロード・バランシング [2.6.3](#)
 - 構成 [9](#)
 - アドレス・リスト [13.1.1](#)
 - ディレクトリ・ネーミング・メソッド [8.3](#)
 - 外部プロシージャ [13.6.1.1.1](#)
 - グローバル・データベース名 [9.2.1](#)
 - ローカル・ネーミング・メソッド [8.2.2](#)
 - 非デフォルトのアドレス [9.2.2](#)
 - プロトコル・アドレス [9.4.1](#)
 - 接続ロード・バランシング [2.6.4](#), [13.2](#)
 - 接続時フェイルオーバー [2.6.1](#)
 - デフォルトのアドレス [9.2.2](#)
 - デフォルト構成 [9.3](#)
 - ダイレクト・ハンドオフ [2.4.1](#)
 - 同時接続の処理 [9.4.2](#)
 - キュー・サイズの増大 [9.4.2](#)
 - ログ・ファイル [9.5.5](#), [16.6.4.2](#), [16.6.8](#)
 - 監査証跡 [16.6.8.1.1](#)
 - ダイレクト・ハンドオフ・イベント情報 [16.6.8.3](#)
 - サービス登録イベント情報 [16.6.8.2.1](#)
 - 監視 [9.5.3.1](#), [9.5.4](#), [9.5.5](#)
 - 複数のアドレス [13.1.1](#)

- Oracle Net Listener [1.3.2](#)
- キュー・サイズ [1.1.4.1](#)
- リダイレクト接続 [2.4.1](#)
- セキュリティ
 - 接続要求のタイムアウト [14.7](#)
 - トレース・ファイル [16.7.2.2](#)
 - 透過的アプリケーション・フェイルオーバー(TAF) [2.6.2](#)
- LOAD_BALANCEネットワーク・パラメータ [13.1.2](#)
- ロード・バランシング
 - クライアント [2.6.3](#)
 - 接続 [2.6.4](#), [13.2](#)
- LOCAL_LISTENER初期化パラメータ [9.2.2](#), [9.4.1.1](#), [11.1](#)
- LOCAL_REGISTRATION_ADDRESS_listener_nameパラメータ [9.4.3](#)
- LOCAL環境変数 [2.5.3](#)
- ローカル・ネーミング
 - メリットおよびデメリット [2.5.1](#)
 - クライアントの構成 [8.2.2](#)
 - 構成 [8.2](#)
 - 接続 [8.2.2](#)
 - データベース・サーバーの構成 [8.2.2](#)
 - デフォルト構成 [8.2.1](#)
 - 接続の確立 [8.2.2](#)
 - リスナー構成 [8.2.2](#)
 - 推奨 [2.5.1](#)
- Oracle Net Configuration Assistantの「ローカル・ネット・サービス名構成」オプション [7.1.4](#)
- LOCALレジストリ・エントリ [2.5.3](#)
- LOG_DIRECTORY_CLIENTログ・パラメータ [16.6.4.1](#)
- LOG_DIRECTORY_listener_nameログ・パラメータ [16.6.4.2](#)
- LOG_DIRECTORY_SERVERログ・パラメータ [16.6.4.1](#)
- LOG_DIRECTORYログ・パラメータ [16.6.4.3](#)
- LOG_FILE_CLIENTログ・パラメータ [16.6.4.1](#)
- LOG_FILE_listener_nameログ・パラメータ [16.6.4.2](#)
- LOG_FILE_SERVERログ・パラメータ [16.6.4.1](#)
- LOG_LEVELログ・パラメータ [16.6.4.3](#)
- ログ・ファイル [9.5.5](#), [16.6.4.1](#), [16.6.8](#)
 - デフォルト名 [16.6.2](#)
 - listener.log [16.6.2](#), [16.6.4.2](#)
 - Oracle Connection Manager [16.6.9](#)
 - セグメンテーション [16.6.3](#)
 - sqlnet.log [16.6.2](#)
 - クライアントとデータベース・サーバーのsqlnet.log [16.6.4.1](#)
 - エラーの追跡に使用 [16.6.7](#)
- Oracle Net Managerの「ログオン認証プロトコルのバージョン」フィールド [12.5](#)

- ログ・パラメータ
 - cman.ora
 - ADR_BASE [16.6.4.3](#)
 - DIAG_ADR_ENABLED [16.6.4.3](#)
 - EVENT_GROUP [16.6.4.3](#)
 - LOG_DIRECTORY [16.6.4.3](#)
 - LOG_LEVEL [16.6.4.3](#)
 - listener.ora
 - ADR_BASE_listener_name [16.6.4.2](#)
 - DIAG_ADR_ENABLED_listener_name [16.6.4.2](#)
 - LOG_DIRECTORY_listener_name [16.6.4.2](#)
 - LOG_FILE_listener_name [16.6.4.2](#)
 - sqlnet.ora
 - ADR_BASE [16.6.4.1](#)
 - DIAG_ADR_ENABLED [16.6.4.1](#)
 - LOG_DIRECTORY_CLIENT [16.6.4.1](#)
 - LOG_DIRECTORY_SERVER [16.6.4.1](#)
 - LOG_FILE_CLIENT [16.6.4.1](#)
 - LOG_FILE_SERVER [16.6.4.1](#)
 - ループバック・テスト [16.2.1](#)
 - LREGプロセス [9.1](#), [10.2.3.1](#), [16.6.8.2.1](#)
-

M

- データ転送の最大化、SDUサイズの調整 [1.1.4.2](#)
 - METHODネットワーク・パラメータ [13.3.3](#)
 - Microsoft Active Directory [3.2.7](#)
 - 移行
 - tnsnames.oraファイルからディレクトリへのネットワーク・サービス名 [8.5](#)
 - 複数のアドレス [13.1.2](#)
 - クライアント・ロード・バランシングの構成 [13.1.2](#)
 - 接続時フェイルオーバーの構成 [13.1.2](#)
 - 複数のプロトコル・アドレス [13.1.1](#)
 - MULTIPLEX(MULまたはMULT)属性 [10.2.3.2](#), [11.1](#)
 - 相互TLS (mTLS)認証 [4.4.2](#)
-

N

- Named Pipesプロトコル
 - 説明 [4.4.3](#)
- NAMES.DEFAULT_DOMAINネットワーク・パラメータ [12.3.1.1](#)
- NAMES.DIRECTORY_PATHネットワーク・パラメータ [12.3.2](#)

- ezconnect [12.3.2](#)
- ホスト名 [12.3.2](#)
- ldap [12.3.2](#)
- nis [12.3.2](#)
- tnsnames [12.3.2](#)
- ネーミング・メソッド [12.3.2](#)
- ネーミング・メソッド [2.5.1](#)
 - 説明 [2.5.1](#)
 - ディレクトリ・ネーミング [8.3](#)
 - 外部ネーミング [8.7](#)
 - ローカライズ [3](#)
 - ローカル・ネーミング [8.2](#)
 - 概要 [1.1.2.2](#)
- Oracle Net Configuration Assistantの「ネーミング・メソッドの構成」オプション [7.1.4](#)
- ネット・サービス名
 - TNSPINGを使用するテスト [15.2](#)
- Netサービス名ウィザード [7.1.2.3.1](#), [8.2.2](#), [10.2.2](#)
- ネットワーク管理者のタスク [7.4](#)
- ネットワーク認証(NA)
 - エラー・スタックのレイヤー [16.6.1](#)
- ネットワークの可用性、判断 [6.2](#)
- ネットワークの構成
 - 集中管理 [3](#)
 - ローカル管理 [3](#)
- ネットワーク・ドメイン、デフォルト構成 [12.3.1](#)
- Network Information Service
 - 「NIS」を参照
- ネットワーキング計画
 - 内部ネットワーク
 - 可用性 [1.1.4.5](#)
 - クライアント・ロード・バランシング [1.1.4.5](#)
 - 接続時フェイルオーバー [1.1.4.5](#)
- 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 [16.6.1](#)
- ネットワーク・パフォーマンス, 向上
 - SDUサイズの調整 [1.1.4.2](#)
 - クライアント・ロード・バランシング [2.6.3](#)
 - リスナー・キュー・サイズ [1.1.4.1](#)
- ネットワーキング計画
 - セッション・データ・ユニット(SDU)サイズ [1.1.4.2](#)
- ネットワーク・サービスの別名
 - 構成 [8.3](#)
 - 説明 [3.2.3](#)

- ディレクトリ・ネーミング [3.2.3](#)
 - 使用方法 [3.2.3](#)
 - ネットワーク・サービス名
 - アドレスの追加 [13.1.1](#)
 - ネットワーク・サービス名
 - 構成 [6.6](#)
 - ディレクトリ・ネーミング [8.3](#)
 - 外部ネーミング [8.7](#)
 - ローカル・ネーミング [8.2.2](#)
 - 複数のアドレス [13.1.1](#), [13.1.2](#)
 - ネーミング・メソッドの優先順位 [12.3.2](#)
 - TNSPINGを使用するテスト [15.2](#)
 - ネットワーク・セッション(NS)、エラー・スタックのレイヤー [16.6.1](#)
 - 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 [16.6.1](#)
 - NIS外部ネーミング [2.5.1](#), [12.3.2](#)
 - 構成 [8.7](#)
 - 接続 [8.7](#)
 - 接続の確立 [8.7](#)
 - マップ [8.7](#)
-

O

- オブジェクト・クラス
 - 説明 [3.2.6.3](#)
- Open Systems Interconnection(OSI)
 - 説明 [4.1](#)
- ORA-12154エラー・メッセージ, [16.3.2](#)
- ORA-12170エラー・メッセージ [14.7](#), [16.3.3](#)
- ORA-12203エラー・メッセージ
 - サンプルのエラー・スタック [16.6.1.1](#)
- ORA-12514エラー・メッセージ [16.3.5](#)
- ORA-12520エラー・メッセージ [16.3.6](#)
- ORA-12521エラー・メッセージ [16.3.7](#)
- ORA-12525エラー・メッセージ [14.7](#), [16.3.8](#)
- ORA-12526エラー・メッセージ [5.4](#)
- ORA-12527エラー・メッセージ [5.4](#)
- ORA-12528エラー・メッセージ [5.4](#)
- ORA-12533エラー・メッセージ [16.3.9](#)
- ORA-12535エラー・メッセージ [12.5](#)
- ORA-12547エラー・メッセージ [14.7](#)
- ORA-12637エラー・メッセージ [14.7](#)
- ORACLE_HOMEネットワーク・パラメータ [13.6.1](#), [13.6.2.1](#)
- Oracle9i Real Application Clusters

- 接続時フェイルオーバー [13.1.2](#)
- FAILOVER_MODEネットワーク・パラメータ [13.3.3](#)
- FAILOVERネットワーク・パラメータ [13.1.2](#)
- 透過的アプリケーション・フェイルオーバー(TAF) [13.3](#)
- Oracle Advanced Security
 - Oracle Net Managerによる構成 [12.7](#)
 - 概要 [1.3.5](#)
- Oracle Call Interface(OCI)レイヤー、説明 [4.1.1.1](#)
- Oracle Clusterware
 - 通知情報 [16.6.8.5](#)
- Oracle Connection Manager
 - アーキテクチャ [5.6](#)
 - CMADMINプロセス [5.6](#)
 - 構成
 - Transport Layer Security [10.2.1.1](#)
 - 構成
 - アクセス制御 [10.2.1.2](#)
 - クライアント [10.2.2](#)
 - データベース・サーバー [10.2.3](#), [10.2.3.2](#)
 - Oracle Connection Managerコンピュータ [10.2.1](#)
 - Oracle Connection Managerの Protokol・アドレス [10.2.2](#)
 - サービス登録 [10.2.3.1](#)
 - セッションの多重化 [10.2.3.2](#)
 - ゲートウェイ・プロセス [5.6](#)
 - リスナー [5.6](#)
 - ログ・ファイル [16.6.4.3](#)
 - 名前 [16.6.2](#)
 - 理解 [16.6.9](#)
 - 概要 [1.3.3](#)
 - 接続のルーティング [12.3.3](#)
 - セッションの多重化 [5.6](#)
 - トレース・ファイル [16.7.2.1](#)
 - 構成 [16.7.2.4](#)
 - 名前 [16.7.1](#)
- Oracle Connection Manager制御ユーティリティ
 - コマンド
 - ADMINISTER [6.4](#)
 - EXIT [6.4](#)
 - STARTUP [6.4](#)
 - 使用 [10.4](#)
- Traffic DirectorモードのOracle Connection Manager [10.2.4.1](#)
 - 構成
 - Transport Layer Security [10.2.4.10](#)

- 構成 [10.2.4](#)
 - Oracle Connection Managerコンピュータ [10.2.4.2](#)
 - サービス登録 [10.2.4.5](#)
- プロキシ常駐接続プールの構成 [10.2.4.6](#)
- 有効化
 - 外部パスワード・ストア [10.2.4.3.1](#)
- 制限事項
 - Traffic DirectorモードのOracle Connection Managerの制限事項 [10.2.4.11](#)
- Oracleコンテキスト
 - 定義 [3.2.2](#)
- OracleContextAdminsグループ [3.2.4](#), [3.2.6.2.2](#)
- OracleDBCreatorsグループ [3.2.4](#), [3.2.6.2.2](#)
- OracleHOME_NAMECManサービス [6.4](#)
- Oracle Net Managerの「Oracleホーム・ディレクトリ」フィールド [13.6.1](#)
- Oracle Net
 - バッファ [1.1.4.2](#), [1.1.4.3](#)
 - 定義 [1.3.1](#)
 - Oracle Net Foundationレイヤー [1.3.1.1](#)
 - Oracleプロトコル・サポート [1.3.1.2](#)
 - 概要 [1.1](#)
 - 拡張機能 [1.1.3](#)
 - 共有サーバー [1.1.3](#)
 - 理解 [1.1](#)
- OracleNetAdminsグループ [3.2.6.2.2](#), [7.2.1](#), [8.3](#)
- Oracle Netコンフィギュレーション・アシスタント
 - 説明 [7.1.4](#)
 - 「ディレクトリ使用構成」オプション [7.1.4](#)
 - リスナー構成 [9.3](#)
 - 「リスナー構成」オプション [7.1.4](#)
 - ローカル・ネーミング・メソッド [8.2.2](#)
 - 「ローカル・ネット・サービス名構成」オプション [7.1.4](#)
 - 「ネーミング・メソッドの構成」オプション [7.1.4](#)
 - ネットワーク・サービス名 [8.2.2](#)
 - OracleContextAdminsグループ [3.2.6.2.2](#)
 - OracleDBCreatorsグループ [3.2.4](#), [3.2.6.2.2](#)
 - OracleNetAdminsグループ [3.2.6.2.2](#)
 - サーバー
 - リスナー構成 [7.1.4](#)
 - 起動 [7.1.4](#)
- Oracle Net Foundationレイヤー [1.3.1.1](#), [4.1.1.3](#)
- Oracle Net Listener
 - 「リスナー」を参照:
 - 説明 [1.3.2](#)

- 起動 [6.3](#)
- Oracle Net Manager
 - アドレスの追加 [13.1.1](#)
 - 「アドレス・リスト・オプション」ダイアログ・ボックス [13.1.2](#)
 - クライアント
 - クライアント・ロード・バランシング [13.1.2](#)
 - 接続時フェイルオーバー [13.1.2](#)
 - デフォルト・ネットワーク・ドメイン [12.3.1](#)
 - ローカル・ネーミング・メソッド [8.2.2](#)
 - Oracle Connection Manager [10.2.2](#)
 - 説明 [7.1.2](#)
 - ディレクトリ・ネーミング [3.2.4](#)
 - ディレクトリ・サーバー移行ウィザード [8.5](#)
 - 外部プロシージャ接続 [13.6.1](#)
 - 異機種間サービス接続 [13.6.2](#)
 - 「インスタンス名」フィールド [13.1.3](#)
 - リスナー
 - 「環境」フィールド [13.6.1](#)
 - 「グローバル・データベース名」フィールド [13.1.3](#)
 - 「Oracleホーム・ディレクトリ」フィールド [13.6.1](#)
 - Oracleシステム識別子フィールド [13.6.1](#)
 - 「プログラム名」フィールド [13.6.1](#)
 - プロトコル・アドレス [9.4.1](#)
 - ローカル・ネーミング・メソッド [8.2.2](#)
 - 複数のアドレス・オプション [13.1.2](#)
 - ナビゲート [7.1.2.2](#)
 - Netサービス名ウィザード [8.2.2](#), [10.2.2](#)
 - ネットワーク・サービスの別名 [8.3](#)
 - ネットワーク・サービス名 [8.2.2](#)
 - 「Oracle RDBデータベース」フィールド [13.1.3](#)
 - Oracle Rdbデータベース [13.6.3](#)
 - 「グローバル・データベース名」フィールド [13.6.3](#)
 - 「RDBデータベース」フィールド [13.6.3](#)
 - 「サービスのタイプ」フィールド [13.6.3](#)
 - プロファイル [12.7](#)
 - 詳細オプション [12.5](#)
 - 「TCP/IPクライアント・アクセス権のチェック」オプション [12.4](#)
 - 「クライアント登録ID」フィールド [12.5](#)
 - 「アクセスを許可されるクライアント」フィールド [12.4](#)
 - 「アクセスを許可されないクライアント」フィールド [12.4](#)
 - 「接続タイムアウト」フィールド [12.5](#)
 - 「アウト・オブ・バンド・ブレイクの使用禁止」オプション [12.5](#)
 - 「ログオン認証プロトコルのバージョン」フィールド [12.5](#)

- 「受信操作タイムアウト」フィールド [12.5](#)
 - 「送信操作タイムアウト」フィールド [12.5](#)
 - 「TNSタイムアウト値」オプション [12.5](#)
 - 「合計受信バッファ・サイズ」フィールド [12.5](#)
 - 「合計送信バッファ・サイズ」フィールド [12.5](#)
 - 「UNIXシグナル処理をオフにする」オプション [12.5](#)
- 接続要求のルーティング [12.3.3](#)
- Oracle Net Managerのセッション・データ・ユニット(SDU)フィールド [13.1.3](#)
- ネーミング・メソッドの指定 [12.3.2](#)
- 起動 [7.1.2.1](#)
- 「サービスのタイプ」フィールド [13.1.3](#)
- 「異機種間サービスを使用」オプション [13.1.3](#), [13.6.2.1](#)
- ウィザード [7.1.2.3](#)
- Oracle Net Services
 - コンポーネント
 - Oracle Connection Manager [1.3.3](#)
 - Oracle Net [1.3.1](#)
 - Oracle Net Listener [1.3.2](#)
 - 説明 [1.3](#)
- Oracleプロトコル・サポート
 - 説明 [1.3.1.2](#), [4.1.1.4](#)
 - Named Pipes [4.4.3](#)
 - TCP/IP [4.4.1](#)
 - TLS付きTCP/IP [4.4.2](#)
 - websocketプロトコル [4.4.6](#)
- Oracle Rdbデータベース
 - 接続の構成 [13.6.3](#)
 - 説明 [13.6.3](#)
- Oracle Net Managerの「Oracle RDBデータベース」フィールド [13.1.3](#)
- Real Application Clusters
 - 接続時フェイルオーバー [2.6.1](#)
- Oracleスキーマ
 - 説明 [3.2.6.3](#)
- Oracleサポート・サービス
 - 連絡 [16.8](#)
- Oracle Net Managerの「Oracleシステム識別子(SID)」フィールド [13.6.1](#)
- OSI
 - 「Open System Interconnection (OSI)」を参照

P

- パケット
 - トレース・データの検証 [16.7.5.2](#), [16.7.5.4](#)

- タイプ [16.7.4.2](#)
- 計画
 - 内部ネットワーク
 - 可用性 [1.1.4.5](#)
 - 接続時フェイルオーバー [1.1.4.5](#)
 - セッション・データ・ユニット(SDU)サイズ [1.1.4.2](#)
- プール・サーバー [2.4.3](#)
- ポート 1521
 - 使用していない場合 [9.2.2](#)
- ポート
 - 権限 [9.4.1.1](#)
- プレゼンテーション・レイヤー
 - FTP [4.3](#)
 - HTTP [4.3](#)
 - JavaTTC [4.2](#)
 - Two-Task Common(TTC) [4.1.1.2](#)
 - WebDAV [4.3](#)
- プライマリ・インスタンスおよびセカンダリ・インスタンス [13.4](#)
- 特権ポート [9.4.1.1](#)
- プロファイル(sqlnet.ora)
 - 構成
 - 詳細オプション [12.5](#)
 - デフォルト・ドメイン [12.3.1](#)
 - デフォルト構成 [12.2](#)
 - ネーミング・メソッド、指定 [12.3.2](#)
 - 接続要求のルーティング [12.3.3](#)
- Oracle Net Managerの「プログラム名」フィールド [13.6.1](#)
- PROGRAMネットワーク・パラメータ [13.6.1](#), [13.6.2.1](#)
- PROTOCOL(PROまたはPROT)属性 [10.2.3.2](#)
- プロトコル・アドレス [2.3.2](#)
- プロトコル
 - FTP [1.1.1.2.2](#)
 - HTTP [1.1.1.2](#), [1.1.1.2.2](#)
 - Named Pipes [4.4.3](#)
 - Oracleサポート [1.3.1.2](#)
 - TCP/IP [4.4.1](#)
 - TLS付きTCP/IP [4.4.2](#)
 - WebDAV [1.1.1.2.2](#)
 - websocket [4.4.6](#)
- プロキシ認証
 - 構成
 - Traffic DirectorモードのOracle Connection Manager [10.2.4.4](#)
- プロキシ・サーバー [10](#)

Q

- キュー・サイズ [1.1.4.1](#), [9.4.2](#)
 - QUEUESIZEパラメータ [9.4.2](#)
 - リスナー・キュー・サイズの調整 [1.1.4.1](#), [9.4.2](#)
-

R

- リスナー間での要求の無作為化 [2.6.3](#)
 - RDB_DATABASEネットワーク・パラメータ [13.1.3](#), [13.6.3](#)
 - 「RDBデータベース」フィールド [13.6.3](#)
 - Oracle Net Managerの「受信操作タイムアウト」フィールド [12.5](#)
 - リダイレクト接続 [2.4.1](#)
 - 相対識別名(RDN) [3.2.1](#)
 - 相対ネーミング
 - ディレクトリ・ネーミング [3.2.1.3](#)
 - Java Database Connectivity(JDBC)
 - OCIドライバ [3.2.1.3](#)
 - Thinドライバ [3.2.1.3](#)
 - 要求キュー [5.5.1](#)
 - 解決
 - エラー
 - 「トラブルシューティング」を参照 [16](#)
 - レスポンス・キュー [5.5.1](#)
 - Traffic DirectorモードのConnection Managerの制限事項
 - Traffic DirectorモードのConnection Manager [10.2.4.11](#)
 - ルーティング
 - キー値 [14.8.1](#)
 - 接続のルーティング [12.3.3](#)
 - RULE_LISTネットワーク・パラメータ [10.2.1.2](#)
-

S

- SDPプロトコル
 - 構成 [14.5](#)
- セキュリティ
 - データベース・サーバー
 - アクセス制御の構成 [12.4](#)
 - 接続要求のタイムアウト [14.7](#)
 - リスナー
 - 接続要求のタイムアウト [14.7](#)

- Oracle Net Listenerのセキュリティの管理 [9.4.3](#)
- Oracle Net Managerの「送信操作タイムアウト」フィールド [12.5](#)
- サーバーの構成
 - アクセス制御 [12.4](#)
 - 共有サーバーにリソースの割当て [11.1.1](#)
 - 接続要求のタイムアウト [14.7](#)
 - ローカル・ネーミング [8.2.2](#)
 - ログ・ファイル [16.6.4.1](#)
 - ループバック・テスト [16.2.1](#)
 - 共有サーバー [11.1](#)
 - トレース・ファイル [16.7.2.3](#)
- SERVERネットワーク・パラメータ [2.3.3.2](#)
- サーバー
 - アクセス制御 [12.4](#)
- サーバーのトラブルシューティング [16.2.1](#)
- service_diedリスナー・ログ・イベント [16.6.8.2.1](#)
- SERVICE_NAMEネットワーク・パラメータ [2.3.1](#)
- SERVICE_NAMES初期化パラメータ [2.2](#), [2.3.1](#)
- service_registerリスナー・ログ・イベント [16.6.8.2.1](#)
- service_updateリスナー・ログ・イベント [16.6.8.2.1](#)
- サービス・ハンドラ
 - 専用サーバー [2.4.2](#), [5.5](#)
 - デイスパッチャ [2.4.1](#)
- サービス名
 - 構成 [2.3.1](#)
 - 説明 [2.2](#)
- サービス登録
 - メリット [9.1](#)
 - 構成 [10.2.3.1](#), [10.2.4.5](#)
 - 接続ロード・バランシング [2.6.4](#), [9.1](#), [13.2](#)
 - 接続時フェイルオーバー [9.1](#)
 - 定義 [2.3.3](#)
 - listener.logのイベント [16.6.8.2.1](#)
 - service_diedリスナー・ログ・イベント [16.6.8.2.1](#)
 - service_registerリスナー・ログ・イベント [16.6.8.2.1](#)
 - service_updateリスナー・ログ・イベント [16.6.8.2.1](#)
- SERVICESコマンド [9.5.4](#)
 - リスナー制御ユーティリティ [6.3](#), [9.5.4](#)
- セッション・データ・ユニット(SDU) [1.1.4.2](#), [1.1.4.3](#), [13.1.3](#)
 - ネットワーク・パフォーマンス向上のために調整 [1.1.4.2](#)
 - 構成 [14.2](#)
 - Oracle Net Managerのフィールド [13.1.3](#)
- セッションの多重化 [1.1.3](#), [5.6](#), [10.2.3.2](#)

- SESSIONS(SESまたはSESS)属性 [11.1](#)
- 共有サーバー
 - リソースの割当て [11.1.1](#)
 - 専用サーバーとの比較 [1.1.3](#)
 - 構成 [11.3](#)
 - 接続記述子構成パラメータ [2.3.3.2](#)
 - 接続ロード・バランシング [2.6.4](#), [13.2](#)
 - 定義 [1.1.3](#)
 - 説明 [5.5.1](#)
 - デイスパッチャ [1.1.3](#), [2.4.1](#), [5.5.1](#)
 - 要求キュー [5.5.1](#)
 - レスポンス・キュー [5.5.1](#)
 - バーチャル・サーキット [5.5.1](#)
- SID_LIST_listener_nameパラメータ
 - 外部プロシージャ [13.6.1.1.2](#)
- SID_NAMEネットワーク・パラメータ [13.6.1](#), [13.6.2.1](#)
- ディレクトリ・ネーミングの簡易認証 [3.2.6.2.1](#)
- SOURCE_ROUTEネットワーク・パラメータ [13.1.2](#)
- SQLNET.ALLOWED_LOGON_VERSION_CLIENTネットワーク・パラメータ [12.5](#)
- SQLNET.ALLOWED_LOGON_VERSION_SERVERネットワーク [12.5](#)
- SQLNET.CLIENT_REGISTRATIONネットワーク・パラメータ [12.5](#)
- SQLNET.EXPIRE_TIMEネットワーク・パラメータ [12.5](#)
- SQLNET.INBOUND_CONNECT_TIMEOUTネットワーク・パラメータ [12.5](#), [14.7](#)
- sqlnet.logファイル [16.6.2](#)
- sqlnet.oraファイル
 - ログ・パラメータ
 - ADR_BASE [16.6.4.1](#)
 - DIAG_ADR_ENABLED [16.6.4.1](#)
 - LOG_DIRECTORY_CLIENT [16.6.4.1](#)
 - LOG_DIRECTORY_SERVER [16.6.4.1](#)
 - LOG_FILE_CLIENT [16.6.4.1](#)
 - LOG_FILE_SERVER [16.6.4.1](#)
 - パラメータ
 - NAMES.DEFAULT_DOMAIN [12.3.1](#)
 - NAMES.DIRECTORY_PATH [12.3.2](#)
 - SQLNET.ALLOWED_LOGON_VERSION_CLIENT [12.5](#)
 - SQLNET.ALLOWED_LOGON_VERSION_SERVER [12.5](#)
 - SQLNET.INBOUND_CONNECT_TIMEOUT [14.7](#)
 - TCP.EXCLUDED_NODES [12.4](#)
 - TCP.INVITED_NODES [12.4](#)
 - TCP.VALIDNODE_CHECKING [12.4](#)
 - トレース・パラメータ
 - TNSPING.TRACE_DIRECTORY [16.7.2.3](#)

- TNSPING.TRACE_LEVEL [16.7.2.3](#)
- TRACE_DIRECTORY_CLIENT [16.7.2.3](#)
- TRACE_DIRECTORY_SERVER [16.7.2.3](#)
- TRACE_FILE_CLIENT [16.7.2.3](#)
- TRACE_FILE_SERVER [16.7.2.3](#)
- TRACE_FILEAGE_CLIENT [16.7.2.3](#)
- TRACE_FILEAGE_SERVER [16.7.2.3](#)
- TRACE_FILELEN_CLIENT [16.7.2.3](#)
- TRACE_FILELEN_SERVER [16.7.2.3](#)
- TRACE_FILENO_CLIENT [16.7.2.3](#)
- TRACE_FILENO_SERVER [16.7.2.3](#)
- TRACE_LEVEL_CLIENT [16.7.2.3](#)
- TRACE_LEVEL_SERVER [16.7.2.3](#)
- TRACE_TIMESTAMP_CLIENT [16.7.2.3](#)
- TRACE_TIMESTAMP_SERVER [16.7.2.3](#)
- TRACE_UNIQUE_CLIENT [16.7.2.3](#)
- SQLNET.RECV_BUF_SIZEネットワーク・パラメータ [12.5](#)
- SQLNET.RECV_TIMEOUTネットワーク・パラメータ [12.5](#)
- SQLNET.SEND_BUF_SIZEネットワーク・パラメータ [12.5](#)
- SQLNET.SEND_TIMEOUTネットワーク・パラメータ [12.5](#)
- sqlnet.trcファイル [16.7.1](#)
- STARTコマンド
 - リスナー制御ユーティリティ [6.3](#), [9.5.1.1](#)
- 起動
 - データベース [6.3](#)
 - Oracle Netコンフィギュレーション・アシスタント [7.1.4](#)
 - Oracle Net Listener [6.3](#)
 - Oracle Net Manager [7.1.2.1](#)
- STATUSコマンド
 - リスナー制御ユーティリティ [9.5.3.1](#)
- STOPコマンド
 - リスナー制御ユーティリティ [9.5.1.1](#)
- ディレクトリ・ネーミングの厳密認証 [3.2.6.2.1](#)
- svr_pid.trcファイル [16.7.1](#)
- 構文
 - リスナー制御ユーティリティ [7.3](#)
 - Oracle Connection Manager制御ユーティリティ [10.4](#)

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

- 説明 [4.4.1](#)
- TLSプロトコル付きTCP/IP
 - 説明 [4.4.2](#)
 - mTLSまたは双方向認証 [4.4.2](#)
 - TLSまたは一方向認証 [4.4.2](#)
- TCP.EXCLUDED_NODESネットワーク・パラメータ [12.4](#)
- TCP.INVITED_NODESネットワーク・パラメータ [12.4](#)
- TCP.INVALIDNODE_CHECKINGネットワーク・パラメータ [12.4](#)
- 終了済接続の検出
 - 構成 [12.5](#)
 - 制限事項 [12.5](#)
- 終了済接続のタイムアウト
 - 「終了済接続のタイムアウト」を参照
- テスト
 - クライアントの構成
 - TCROUTE使用 [15.3](#)
 - TNSPING使用 [15.2](#)
 - 制御ユーティリティ使用 [7.3](#)
- ディレクトリ・ネーミングのTLS認証 [3.2.6.2.1](#)
- TNS
 - 「Transparent Network Substrate (TNS)」を参照
- TNS_ADMIN環境変数 [3.1](#)
- tnsnames.oraファイル
 - ディレクトリ・サーバーへのエントリのエクスポート [8.5](#)
 - ディレクトリ・サーバーへのエントリの移行 [8.5](#)
 - パラメータ
 - BACKUPパラメータ [13.3.3](#)
 - FAILOVER [13.1.2](#)
 - FAILOVER_MODE [13.3.3](#)
 - GLOBAL_NAME [13.1.3](#)
 - HS [13.1.3](#)
 - INSTANCE_NAME [13.1.3](#), [13.4](#)
 - LOAD_BALANCE [13.1.2](#)
 - METHOD [13.3.3](#)
 - RDB_DATABASE [13.1.3](#)
 - SDU [13.1.3](#)
 - SOURCE_ROUTE [13.1.2](#)
 - TYPE [13.3.3](#)
 - TYPE_OF_SERVICE [13.1.3](#)
- TNSPING.TRACE_DIRECTORYトレース・パラメータ [16.7.2.3](#)
- TNSPING.TRACE_LEVELトレース・パラメータ [16.7.2.3](#)
- TNSPINGユーティリティ [15.2](#)
 - TRCROUTEユーティリティとの比較 [15.3](#)

- Oracle Net Managerの「TNSタイムアウト値」オプション [12.5](#)
- Oracle Net Managerの「合計受信バッファ・サイズ」フィールド [12.5](#)
- Oracle Net Managerの「合計送信バッファ・サイズ」フィールド [12.5](#)
- TRACE_DIRECTORY_CLIENTトレース・パラメータ [16.7.2.3](#)
- TRACE_DIRECTORY_listener_nameトレース・パラメータ [16.7.2.2](#)
- TRACE_DIRECTORY_SERVERトレース・パラメータ [16.7.2.3](#)
- TRACE_DIRECTORYトレース・パラメータ [16.7.2.1](#)
- TRACE_FILE_CLIENTトレース・パラメータ [16.7.2.3](#)
- TRACE_FILE_listener_nameトレース・パラメータ [16.7.2.2](#)
- TRACE_FILE_SERVERトレース・パラメータ [16.7.2.3](#)
- TRACE_FILEAGE_CLIENTトレース・パラメータ [16.7.2.3](#)
- TRACE_FILEAGE_listener_nameトレース・パラメータ [16.7.2.2](#)
- TRACE_FILEAGE_SERVERトレース・パラメータ [16.7.2.2](#), [16.7.2.3](#)
- TRACE_FILELEN_CLIENTトレース・パラメータ [16.7.2.3](#)
- TRACE_FILELEN_listener_nameトレース・パラメータ [16.7.2.2](#)
- TRACE_FILELEN_SERVERトレース・パラメータ [16.7.2.3](#)
- TRACE_FILELENTトレース・パラメータ [16.7.2.1](#)
- TRACE_FILENO_CLIENTトレース・パラメータ [16.7.2.3](#)
- TRACE_FILENO_listener_nameトレース・パラメータ [16.7.2.2](#)
- TRACE_FILENO_SERVERトレース・パラメータ [16.7.2.3](#)
- TRACE_FILENOTトレース・パラメータ [16.7.2.1](#)
- TRACE_LEVEL_CLIENTトレース・パラメータ [16.7.2.3](#)
- TRACE_LEVEL_listener_nameトレース・パラメータ [16.7.2.2](#)
- TRACE_LEVEL_SERVERトレース・パラメータ [16.7.2.3](#)
- TRACE_LEVELトレース・パラメータ [16.7.2.1](#)
- TRACE_TIMESTAMP_CLIENTトレース・パラメータ [16.7.2.3](#)
- TRACE_TIMESTAMP_listener_nameトレース・パラメータ [16.7.2.2](#)
- TRACE_TIMESTAMP_SERVERトレース・パラメータ [16.7.2.3](#)
- TRACE_TIMESTAMPトレース・パラメータ [16.7.2.1](#)
- TRACE_UNIQUE_CLIENTトレース・パラメータ [16.7.2.3](#)
- トレース・アシスタント
 - トレース・ファイルの検証 [16.7.5](#)
 - 機能 [16.7.5](#)
 - オプションのリファレンス [16.7.5.1](#)
 - IDのトレース・データ [16.7.5.4](#)
 - パケットのトレース・データ [16.7.5.2](#)
 - トレース・データの統計 [16.7.5.5](#)
- トレース・ファイル
 - トレース・アシスタントを使用した分析 [16.7.5](#)
 - デフォルト名 [16.7.1](#)
 - エラー・メッセージ情報 [16.7.4.3](#)
 - トレース・アシスタントを使用した検証 [16.7.5](#)
 - listener.trc [16.7.2.2](#)

- クライアントのsqlnet.trc [16.7.2.3](#)
- サーバーのsvr_pid.trc [16.7.2.3](#)
- トレース・パラメータ
 - cman.ora
 - TRACE_DIRECTORY [16.7.2.1](#)
 - TRACE_FILELEN [16.7.2.1](#)
 - TRACE_FILENO [16.7.2.1](#)
 - TRACE_LEVEL [16.7.2.1](#)
 - TRACE_TIMESTAMP [16.7.2.1](#)
 - listener.ora
 - TRACE_DIRECTORY_listener_name [16.7.2.2](#)
 - TRACE_FILE_listener_name [16.7.2.2](#)
 - TRACE_FILEAGE_listener_name [16.7.2.2](#)
 - TRACE_FILEAGE_SERVER [16.7.2.2](#)
 - TRACE_FILELEN_listener_name [16.7.2.2](#)
 - TRACE_FILENO_listener_name [16.7.2.2](#)
 - TRACE_LEVEL_listener_name [16.7.2.2](#)
 - TRACE_TIMESTAMP_listener_name [16.7.2.2](#)
 - sqlnet.ora
 - TNSPING.TRACE_DIRECTORY [16.7.2.3](#)
 - TNSPING.TRACE_LEVEL [16.7.2.3](#)
 - TRACE_DIRECTORY_CLIENT [16.7.2.3](#)
 - TRACE_DIRECTORY_SERVER [16.7.2.3](#)
 - TRACE_FILE_CLIENT [16.7.2.3](#)
 - TRACE_FILE_SERVER [16.7.2.3](#)
 - TRACE_FILEAGE_CLIENT [16.7.2.3](#)
 - TRACE_FILEAGE_SERVER [16.7.2.3](#)
 - TRACE_FILELEN_CLIENT [16.7.2.3](#)
 - TRACE_FILELEN_SERVER [16.7.2.3](#)
 - TRACE_FILENO_CLIENT [16.7.2.3](#)
 - TRACE_FILENO_SERVER [16.7.2.3](#)
 - TRACE_LEVEL_CLIENT [16.7.2.3](#)
 - TRACE_LEVEL_SERVER [16.7.2.3](#)
 - TRACE_TIMESTAMP_CLIENT [16.7.2.3](#)
 - TRACE_TIMESTAMP_SERVER [16.7.2.3](#)
 - TRACE_UNIQUE_CLIENT [16.7.2.3](#)
- 透過的アプリケーション・フェイルオーバー(TAF)
 - およびGLOBAL_DBNAMEパラメータ [13.5.1](#)
 - 構成 [13.3](#)
 - listener.ora内のGLOBAL_DBNAMEネットワーク・パラメータ [13.3.4](#), [13.5.1](#)
 - 概要 [2.6.2](#)
 - インスタンス・ロール [13.4](#)
- Transparent Network Substrate (TNS)

- メリット [4.1.1.3](#)
 - 説明 [4.1.1.3](#)
 - CMANでのTransport Layer Security
 - Oracle Connection Manager [10.2.1.1](#)
 - Traffic DirectorモードのConnection ManagerでのTransport Layer Security
 - Traffic DirectorモードのConnection Manager [10.2.4.10](#)
 - TRCROUTEユーティリティ
 - 説明 [15.3](#)
 - トラブルシューティング [16](#)
 - クライアント [16.2.2](#)
 - ループバック・テスト [16.2.1](#)
 - 質問 [16.4.1](#)
 - サーバー [16.2.1](#)
 - TTC
 - 「Two-Task Common (TTC)」を参照
 - Oracle Net Managerの「UNIXシグナル処理をオフにする」オプション [12.5](#)
 - TWO_TASK環境変数 [2.5.3](#)
 - Two-Task Common(TTC)プレゼンテーション
 - 専用サーバー構成 [5.5.2](#)
 - 説明 [4.1.1.2](#)
 - 共有サーバー構成 [5.5.1](#)
 - TYPE_OF_SERVICEネットワーク・パラメータ [13.1.3](#)、[13.6.3](#)
 - TYPEネットワーク・パラメータ [13.3.3](#)
 - Oracle Net Managerの「サービスのタイプ」フィールド [13.1.3](#)、[13.6.3](#)
-

U

- Oracle Net Managerの「異機種間サービスを使用」オプション [13.1.3](#)、[13.6.2.1](#)
-

V

- バーチャル・サーキット [5.5.1](#)
-

W

- ウォレット
 - 構成
 - Traffic DirectorモードのOracle Connection Manager [10.2.4.3](#)
- WebDAVプレゼンテーション [4.3](#)
 - 専用サーバー構成 [5.5.2](#)
 - 共有サーバー構成 [5.5.1](#)

- WebDAVプロトコル [1.1.1.2.2](#)
 - websocketプロトコル
 - 説明 [4.4.6](#)
 - Windows NTのサービス
 - OracleHOME_NAMECManサービス [6.4](#)
 - ウィザード
 - ディレクトリ・サーバー移行 [7.1.2.3.2](#)
 - ネット・サービス名 [7.1.2.3.1](#)
 - Oracle Net Manager [7.1.2.3](#)
-

Y

- ypservプログラム [8.7](#)