

Oracle® Rdb

Oracle SQL/Services Server 構成ガイド

リリース 7.1.6 for OpenVMS Alpha and VAX

部品番号 : E06185-01

2008 年 5 月

このドキュメントは、OpenVMS Alpha および Open VMS VAX
用の Oracle SQL/Services リリース 7.1.6、および OCI Services
for Oracle Rdb に固有の構成情報を記載しています。

Oracle Rdb Oracle SQL/Services Server 構成ガイド, リリース 7.1.6 for OpenVMS Alpha and VAX

部品番号 : E06185-01

原本名 : Oracle Rdb Oracle SQL/Services Server Configuration Guide, Release 7.1.6 for OpenVMS Alpha and VAX

Copyright ©1995, 2005 Oracle. All rights reserved.

制限付権利の説明

このプログラム（ソフトウェアおよびドキュメントを含む）には、オラクル社およびその関連会社に所有権のある情報が含まれています。このプログラムの使用または開示は、オラクル社およびその関連会社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権と工業所有権に関する法律により保護されています。

独立して作成された他のソフトウェアとの互換性を得るために必要な場合、もしくは法律によって規定される場合を除き、このプログラムのリバース・エンジニアリング、逆アセンブル、逆コンパイル等は禁止されています。

このドキュメントの情報は、予告なしに変更される場合があります。オラクル社およびその関連会社は、このドキュメントに誤りが無いことの保証は致し兼ねます。これらのプログラムのライセンス契約で許諾されている場合を除き、プログラムを形式、手段（電子的または機械的）、目的に関係なく、複製または転用することはできません。

このプログラムが米国政府機関、もしくは米国政府機関に代わってこのプログラムをライセンスまたは使用する者に提供される場合は、次の注意が適用されます。

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このプログラムは、核、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションへの用途を目的としておりません。このプログラムをかかるとして使用する際、上述のアプリケーションを安全に使用するために、適切な安全装置、バックアップ、冗長性 (redundancy)、その他の対策を講じることは使用者の責任となります。万一かかるプログラムの使用に起因して損害が発生いたしましても、オラクル社およびその関連会社は一切責任を負いかねます。

Oracle は米国 Oracle Corporation の登録商標です。Oracle Rdb、Oracle SQL/Services、Oracle Rdb7、SQL*Net は米国 Oracle Corporation の商標または登録商標です。その他の名称は、他社の商標の可能性があります。

このプログラムは、第三者の Web サイトへリンクし、第三者のコンテンツ、製品、サービスへアクセスすることがあります。オラクル社およびその関連会社は第三者の Web サイトで提供されるコンテンツについては、一切の責任を負いかねます。当該コンテンツの利用は、お客様の責任になります。第三者の製品またはサービスを購入する場合は、第三者と直接の取引となります。オラクル社およびその関連会社は、第三者の製品およびサービスの品質、契約の履行（製品またはサービスの提供、保証義務を含む）に関しては責任を負いかねます。また、第三者との取引により損失や損害が発生いたしましても、オラクル社およびその関連会社は一切の責任を負いかねます。

目次

はじめに	xiii
対象読者	xiv
オペレーティング・システム情報	xiv
構成	xiv
関連ドキュメント	xiv
表記規則	xv
1 概要	
1.1 Oracle SQL/Services	1-2
1.1.1 サーバー管理ユーティリティ	1-5
1.1.2 サーバーの管理に必要な権限	1-5
1.1.3 SQLSRV_MANAGE ユーティリティの実行	1-6
1.1.4 Oracle SQL/Services Manager GUI の実行	1-6
1.2 オンラインとオフラインのサーバー管理の比較	1-7
1.3 OCI Services for Oracle Rdb	1-9
1.3.1 Oracle Call Interface	1-10
1.3.2 サーバー側のソリューション	1-10
1.3.3 共通のアプリケーション開発	1-11
2 Oracle SQL/Services システムの管理	
2.1 スタート・ガイド	2-2
2.2 Oracle SQL/Services サーバー構成の計画	2-2
2.3 共有メモリー・サイズの設定	2-3
2.4 サーバー・コンポーネントの管理	2-5
2.4.1 サーバーの管理	2-5
2.4.2 ディスパッチャの管理	2-8
2.4.3 サービスの管理	2-10
2.5 ディスパッチャおよびトランスポート選択の設定	2-13
2.6 サービスおよび再利用タイプの設定	2-14
2.6.1 セッションで再利用可能な汎用サービス	2-16
2.6.2 セッションで再利用可能なデータベース・サービス	2-17
2.6.3 トランザクションで再利用可能なデータベース・サービス	2-17
2.6.4 セッションとトランザクションで再利用可能なデータベース・サービスの 使用条件の比較	2-19
2.7 データベース・アクセス認可の設定	2-20

2.7.1	SQL/Services プロトコル用のデフォルトの接続ユーザー名の指定	2-22
2.7.2	サービスへのアクセス権の付与または制限	2-22
2.7.3	任意のデータ・アクセス権または事前定義されたデータ・アクセス権の提供	2-23
2.8	サーバーのセキュリティの設定	2-24
2.8.1	クライアント識別および認証	2-24
2.8.2	サービス・アクセス認可	2-25
2.8.3	データベースおよびデータ・アクセス認可	2-26
2.8.4	SQL/Services プロトコルでのサーバー・セキュリティ層の連携	2-27
2.9	Oracle SQL/Services のデータベース・アクセス認可モデルの理解	2-29
2.9.1	Oracle Rdb データベースへのアクセス	2-29
2.9.2	プロセス・ユーザー名および Oracle Rdb のシステム・ユーザー名の設定	2-31
2.10	サービス所有者ユーザー名を選択する際のセキュリティの検討	2-32
2.10.1	データベース・リクエストの実行環境	2-32
2.10.2	外部ファンクションおよびプロシージャ用の実行環境	2-33
2.11	エグゼキュータ数の属性の設定	2-35
2.11.1	サービス用の固定数のエグゼキュータの構成	2-35
2.11.2	サービス用の可変数のエグゼキュータの構成	2-35
2.12	SQL 初期化ファイルの使用	2-36
2.13	SQL/Services の論理名の使用	2-36

3 Oracle SQL/Services サーバーのメンテナンス

3.1	サーバー・アクティビティの監視	3-2
3.2	クライアント接続の監視	3-2
3.2.1	セッションで再利用可能なサービスのクライアント接続状態	3-2
3.2.2	トランザクションで再利用可能なデータベース・サービスのクライアント接続状態	3-3

4 OCI Services for Oracle Rdb の処理

4.1	OCI Services for Oracle Rdb の機能	4-2
4.1.1	OCI メッセージ・マッピング	4-2
4.1.2	カーソル管理	4-2
4.1.3	データ型	4-2
4.1.4	データ定義言語	4-2
4.1.5	SQL カーソル・セマンティクス	4-2
4.1.6	Oracle SQL ALTER SESSION 文	4-2
4.1.7	データ書式設定	4-3
4.1.8	文の解析	4-3
4.1.9	データ型の説明	4-4
4.1.10	Oracle データ・ディクショナリ	4-5
4.1.11	マルチスキーマ・エミュレーション	4-6
4.2	サーバー環境の初期化	4-6
4.3	Oracle Forms との互換性	4-7
4.4	汎用サービスと組み合わせた Oracle SQL/Services の使用	4-7
4.5	RDB_NATCONN71.COM コマンド・ファイルの使用	4-8
4.5.1	データベースの準備	4-8
4.5.2	データベースのアップグレード	4-9
4.5.3	OCI Services for Oracle Rdb の削除	4-9
4.5.4	ユーザーとパスワードの追加	4-9

4.5.5	パスワードの変更	4-10
4.5.6	ユーザーの削除	4-11
4.5.7	ユーザーの表示	4-11
4.6	キャラクタ・セットの定義	4-12
4.6.1	サーバー・システムでのキャラクタ・セットの定義	4-13
4.6.2	クライアント・システムでのキャラクタ・セットの定義	4-13
4.6.3	規則および推奨事項	4-13
4.7	データベース・リンクとしての Oracle Rdb データベースの参照	4-14
4.7.1	CREATE DATABASE LINK の例	4-14
4.7.2	データベース・リンクの制限事項	4-15
4.8	OCI Services for Oracle Rdb 表での SQL EXPORT/IMPORT の使用	4-16

5 SQL ALTER SESSION 文

ALTER SESSION 文	5-2
-----------------------	-----

6 管理コマンド

6.1	構文表記規則	6-2
6.2	SQLSRV_MANAGE コマンドの動作方法	6-4
	-input スイッチ	6-7
	-output スイッチ	6-8
	@ コマンド	6-9
	ALTER DISPATCHER コマンド	6-10
	ALTER SERVER コマンド	6-14
	ALTER SERVICE コマンド	6-17
	CLOSE コマンド	6-24
	CONNECT TO SERVER コマンド	6-25
	CREATE DISPATCHER コマンド	6-27
	CREATE SERVER コマンド	6-30
	CREATE SERVICE コマンド	6-33
	DISCONNECT SERVER コマンド	6-39
	DROP コマンド	6-40
	DROP SERVER コマンド	6-42
	EXIT コマンド	6-43
	EXTRACT コマンド	6-44
	GRANT USE ON SERVICE コマンド	6-46
	HELP コマンド	6-48
	KILL EXECUTOR コマンド	6-49
	OPEN コマンド	6-50
	RESTART SERVER コマンド	6-51
	REVOKE USE ON SERVICE コマンド	6-52
	SET CONFIGURATION_FILE コマンド	6-54
	SET CONFIRM コマンド	6-55
	SET CONNECTION コマンド	6-56
	SET OUTPUT コマンド	6-58

SET VERIFY コマンド	6-59
SHOW CLIENTS コマンド	6-60
SHOW CONNECTIONS コマンド	6-63
SHOW DISPATCHER コマンド	6-64
SHOW SERVER コマンド	6-66
SHOW SERVICE コマンド	6-67
SHOW SETTINGS コマンド	6-69
SHOW VERSION コマンド	6-70
SHUTDOWN DISPATCHER コマンド	6-71
SHUTDOWN SERVER コマンド	6-72
SHUTDOWN SERVICE コマンド	6-73
START DISPATCHER コマンド	6-74
START SERVER コマンド	6-75
START SERVICE コマンド	6-77

7 ログイングとトラブルシューティング

7.1	問題のレポート	7-2
7.2	エラー・メッセージ	7-2
7.3	サーバーのログ・ファイル	7-3
7.3.1	Oracle SQL/Services モニター・ログ・ファイル	7-3
7.3.2	Oracle SQL/Services ディスパッチャ・ログ・ファイル	7-3
7.3.3	Oracle SQL/Services エグゼキュータ・ログ・ファイル	7-5
7.3.4	OCI Services for Oracle Rdb におけるエグゼキュータ・ログ・ファイルの デバッグ・フラグの使用	7-5
7.3.5	SQL および Oracle Rdb でのログイングの有効化	7-6
7.3.6	SQL/Services でのログイングの無効化	7-6
7.3.7	ディスパッチャ・ログ・ファイルのシステム・ディスクからの移動	7-6
7.4	OCI Services for Oracle Rdb の完全ログイングの有効化	7-7
7.5	SQL/Services API ログ・ファイルの調査	7-7
7.5.1	クライアントおよびドライバ・ログイング	7-7
7.5.2	Winsock ログイング	7-8
7.5.3	Oracle Net ログイング	7-8
7.5.4	ODBC トレース	7-8
7.6	プロセス障害	7-9
7.6.1	モニター・プロセス障害	7-9
7.6.2	ディスパッチャ・プロセス障害	7-9
7.6.3	エグゼキュータ・プロセス障害	7-9
7.7	異なるタイプの問題の調査	7-10
7.7.1	ネットワーク・トランスポートの問題	7-10
7.7.2	ユーザー認証および認可の問題	7-10
7.7.3	サービス起動時のエグゼキュータの障害	7-11
7.7.4	クライアント接続時のエグゼキュータの問題	7-11
7.7.5	クライアント・リクエスト実行時のエグゼキュータの問題	7-11
7.7.6	内部エラーを原因とするサーバーの障害	7-12
7.7.7	Oracle SQL/Services のインストール後に機能しなくなるクライアントからの接続	7-12
7.7.8	ネットワーク・エラー	7-13

7.8	OCI クライアント・アプリケーションに返されるエラー・メッセージ	7-13
7.8.1	ログオン・エラー	7-13
7.8.2	データベース設定エラー	7-14
7.8.3	SQL 初期化ファイル・エラー	7-14
7.8.4	Rdb データベースへのアタッチ・エラーまたは Oracle SQL/Services データベース・サービスのエラー	7-14
7.9	Oracle SQL/Services サーバーまたは OCI ディスパッチャが使用できない場合のエラー	7-15
7.9.1	ERROR: ORA-12203: TNS: 接続先に接続できません。	7-15
7.9.2	Oracle Net サービス名が未定義の場合のエラー	7-15

A 表とリレーション

A.1	Oracle メタデータ・オブジェクト	A-2
A.2	現在のセッション情報	A-5
A.3	31 文字のオブジェクト名の処理	A-6

索引

例一覧

2-1	データベース・アクセス認可が接続ユーザー名に設定されたデフォルトの汎用サービス	2-20
2-2	データベース・アクセス認可がサービス所有者に設定された汎用サービス	2-20
2-3	アクセス認可が接続ユーザー名に設定された、セッションで再利用可能な データベース・サービス	2-21
2-4	アクセス認可がサービス所有者に設定された、トランザクションで再利用可能な データベース・サービス	2-22
4-1	Oracle 日付リテラルの ANSI 日付列への挿入	4-3
4-2	CALL という語のプロシージャ・コールへの挿入	4-3
4-3	西ヨーロッパ言語キャラクタ・セットの指定	4-13
4-4	フランス語と地域の指定	4-13
4-5	TNSNAMES.ORA ファイル	4-14
4-6	データベース・リンクの定義	4-14
4-7	有効な Oracle SQL UPDATE 文	4-15
4-8	無効な Oracle SQL UPDATE 文	4-15
4-9	DML が機能しない場合の PL/SQL の使用方法	4-16

表一覧

1-1	OCI Services for Oracle Rdb の処理機能	1-11
2-1	サーバー・オブジェクト属性のデフォルト設定	2-7
2-2	ディスパッチャ・オブジェクト属性のデフォルト設定	2-9
2-3	サービス・オブジェクト属性のデフォルト設定	2-12
2-4	Oracle SQL/Services サービスの属性	2-16
2-5	セッションとトランザクションで再利用可能なデータベース・サービスの使用条件の比較 ..	2-19
2-6	SQL/Services の論理名	2-36
4-1	サポートされるキャラクタ・セット	4-12
6-1	Oracle SQL/Services オブジェクトとコマンドによる各オブジェクトへの影響	6-4
6-2	SQLSRV_MANAGE の環境コマンドおよびスイッチ	6-6
7-1	DECnet のエラー・コード・ファイル	7-13
7-2	TCP/IP のエラー・コード・ファイル	7-13
A-1	OCI Services for Oracle Rdb のオブジェクト	A-2
A-2	ORA_SESSION 表	A-5

図一覧

1-1	最も単純なクライアント / サーバー・アーキテクチャ	1-2
1-2	Oracle SQL/Services サーバー・システム	1-2
1-3	Oracle SQL/Services のクライアント / サーバー・アーキテクチャ	1-3
1-4	クライアント / サーバー処理	1-9
2-1	Oracle SQL/Services のセッションで再利用可能な汎用サービス	2-16
2-2	Oracle SQL/Services のセッションで再利用可能なデータベース・サービス	2-17
2-3	Oracle SQL/Services のトランザクションで再利用可能なデータベース・サービス	2-18
2-4	Oracle SQL/Services サーバー・セキュリティ	2-27
3-1	セッションで再利用可能なサービスのクライアント接続状態	3-2
3-2	トランザクションで再利用可能なデータベース・サービスのクライアント接続状態	3-3

はじめに

Oracle Rdb は、リレーショナル・データ・モデルに基づいた汎用のデータベース管理システムです。

Oracle Rdb のクライアント / サーバー・コンポーネントである Oracle SQL/Services により、サポート対象のオペレーティング・システムまたはトランスポート上で稼働するリモート・クライアント・コンピュータで起動されたクライアント・アプリケーション・プログラムは、Oracle Rdb データベースにアクセスできます。

OCI Services for Oracle Rdb (旧 SQL*Net for Oracle Rdb) では、既存の OCI アプリケーションを実行して Rdb データベースにアクセスできます。

このマニュアルでは、Oracle SQL/Services サーバーおよび OCI Services for Oracle Rdb システムを管理およびチューニングする方法について説明します。

対象読者

このマニュアルは、Oracle SQL/Services および OCI Services for Oracle Rdb の管理とチューニングを担当するシステム管理者を対象としています。システム管理者は、Oracle SQL/Services システムのインストールに関する重要な情報が記載されているインストレーション・ガイドを参照する必要があります。

オペレーティング・システム情報

このリリースの Oracle Rdb および Oracle SQL/Services と互換性のあるオペレーティング・システムおよびオプション・ソフトウェアのバージョンに関する情報は、Oracle Rdb および Oracle SQL/Services のインストレーション・ガイドとリリース・ノートに記載されています。互換性に関するその他の質問がある場合は、Oracle サポート・サービスまでご連絡ください。

構成

このマニュアルには、次の章が含まれます。

第 1 章	Oracle SQL/Services システムの概要について説明します。
第 2 章	Oracle SQL/Services システムの管理方法について説明します。
第 3 章	Oracle SQL/Services サーバーのメンテナンス方法について説明します。
第 4 章	OCI Services for Oracle Rdb の機能および利点について詳細に説明します。
第 5 章	OCI Services for Oracle Rdb での ALTER SESSION コマンドの使用方法について説明します。
第 6 章	Oracle SQL/Services システムの管理コマンドについて説明します。
第 7 章	Oracle SQL/Services および OCI Services for Oracle Rdb をトラブルシューティングする方法とロギングを有効化する方法について説明します。
付録 A	OCI Services for Oracle Rdb 製品により Rdb データベースに格納される、エミュレートされた Oracle データ・ディクショナリを構成する表、ビュー、およびその他のデータベース・オブジェクトの定義について説明します。この付録には、OCI Services for Oracle Rdb ソフトウェアとともにインストールされるすべてのファイルの名前、ターゲットの場所および用途のリストも含まれます。

関連ドキュメント

詳細は、このドキュメント・セットの他のマニュアル（特に次のマニュアル）を参照してください。

- 『Oracle Rdb New and Changed Features for Oracle Rdb Release 7.1』
- 『Oracle Rdb7 Guide to SQL Programming』
- 『Oracle Rdb SQL リファレンス・マニュアル』
- 『Guide to Using the Oracle SQL/Services Client API』
- Oracle Rdb リリース 7.1 および Oracle SQL/Services リリース 7.1.6 のリリース・ノートとインストレーション・ガイド。インストール時に、PostScript ファイルとして SYS\$HELP ディレクトリに『Oracle Rdb Oracle SQL/Services リリース・ノート』が提供されます。
- 『Oracle Rdb7 Guide to Database Maintenance』

表記規則

このマニュアル中の Oracle Rdb は、Oracle Rdb for OpenVMS ソフトウェアを指します。Oracle Rdb のリリース 7.1 は、V7.1 とも表記されます。

Oracle Rdb への SQL インタフェースは、SQL と表記されます。このインタフェースは、一般に ANSI/ISO SQL 標準または SQL:1999 と呼ばれる 1999 年に採用された SQL 標準の Oracle Rdb の実装です。SQL 標準の詳細は、OpenVMS Alpha の Oracle Rdb リリース・ノートを参照してください。

リリース 7.1 からは、Oracle SQL/Services は複数バージョン専用キットで提供されています。インストール時には、バリエーションのネーミング規則を使用してファイルがインストールされます。つまり、2 桁のバージョン番号が、バリエーションのファイル名またはユーティリティ名の末尾 2 文字として付加されます。たとえば、管理クライアントは SQLSRV_MANAGE71、そのログ・ファイルは *71.log などになります。

OpenVMS は、OpenVMS Alpha または OpenVMS VAX オペレーティング・システムを表します。

このマニュアルでは次の表記規則も使用します。

規則	意味
.	例の中の垂直の省略記号は、例に直接関連しない情報が省略されていることを示します。
\$	ドル記号は、OpenVMS の DIGITAL Command Language プロンプトを表します。
太字のテキスト	本文中の太字は、本文中で定義されている用語を示します。
固定幅フォントの太字	本文中の固定幅フォントの太字は、ユーザー入力を示します。

SQLSRV_MANAGE ユーティリティで使用する構文規則の詳細は、[6.1 項](#)を参照してください。

1

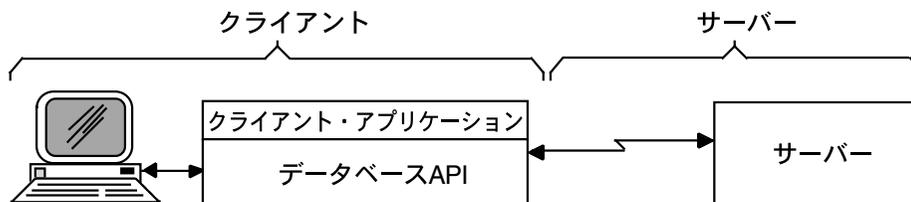
概要

この章では、Oracle SQL/Services および OCI Services for Oracle Rdb (旧 SQL*Net for Oracle Rdb) システムのコンポーネントについて説明し、同時に SQL/Services システムの管理方法の概要について説明します。

1.1 Oracle SQL/Services

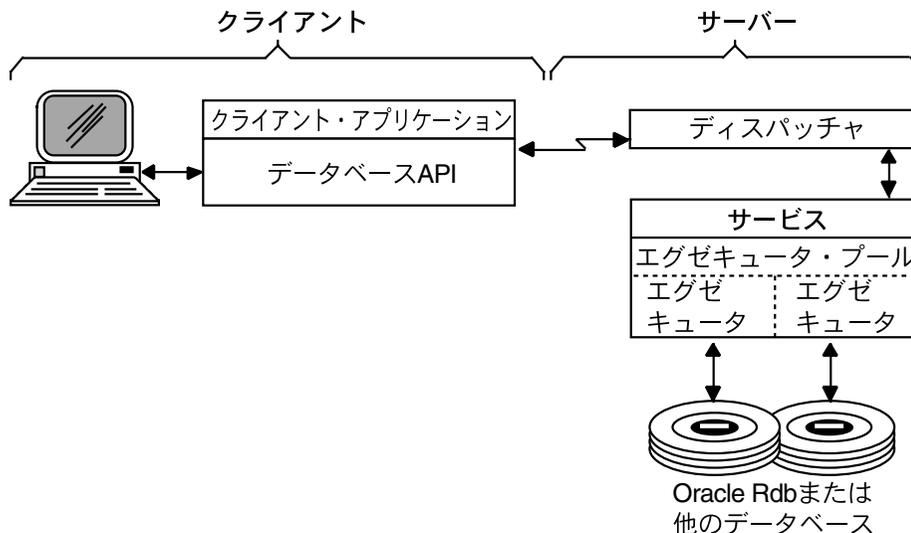
最も単純な形式のクライアント / サーバー・システムは、クライアント、ネットワークおよびサーバー・システムで構成されます。**クライアント**は、データベースの **Application Program Interface (API)** を使用してサーバーのデータベース・リクエストを作成するソフトウェア・プログラムです (図 1-1 を参照)。クライアントは、サーバーと同じプラットフォームに存在することもあります。ただし、通常、クライアント・アプリケーションは、ワークステーションまたは PC 上で実行され、複数のトランスポート・プロトコルをサポートするネットワークを使用して大規模なサーバー・プラットフォーム上のデータベースにアクセスします。

図 1-1 最も単純なクライアント / サーバー・アーキテクチャ



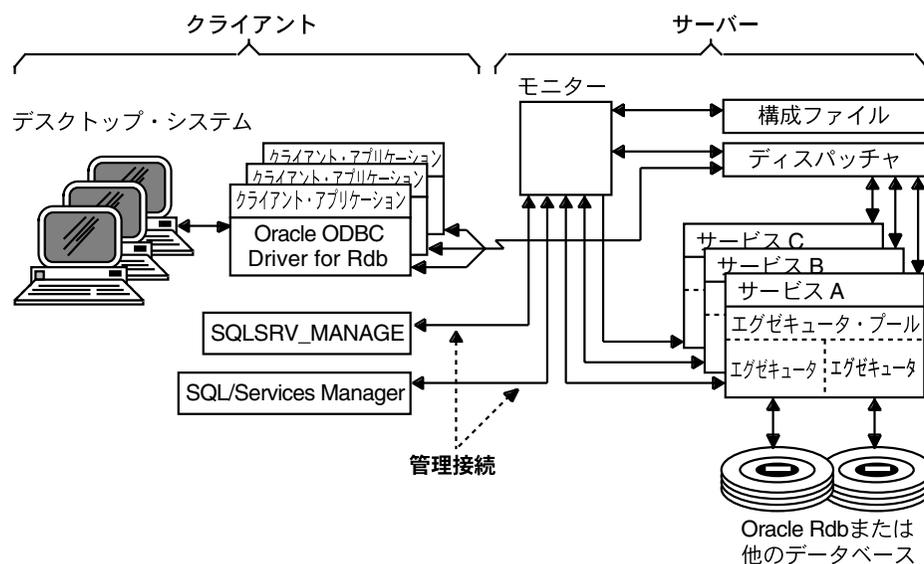
Oracle SQL/Services **サーバー**は、ディスパッチャ・プロセスを含む単一ノード上の連携プロセスのコレクションと、サービスのかわりに動作するエグゼキュータ・プロセスのプールです (図 1-2 を参照)。**ディスパッチャ・プロセス**は、クライアントとサーバー間のすべてのネットワーク通信を処理します。ディスパッチャ・プロセスは、クライアント・リクエストを読み取り、それらのリクエストをサービスのエグゼキュータ用のキューに配置し、サービスのエグゼキュータからのレスポンスをクライアントに返します。**エグゼキュータ・プロセス**は、サービスのかわりに動作し、ディスパッチャのキューからクライアント・リクエストを取得します。その後、データベース・エンジンを起動してそれらのリクエストを処理し、結果をディスパッチャに返します。**サービス**は、クライアントによるデータベースへのアクセス方法を記述した属性のセットです。

図 1-2 Oracle SQL/Services サーバー・システム



Oracle SQL/Services サーバー・システムには、サーバーを管理および制御するためのモニター・プロセス、OpenVMS のローカルまたはリモート・システムで実行される SQLSRV_MANAGE ユーティリティ、または Windows クライアントとして実行される Oracle SQL/Services Manager GUI サーバー管理ユーティリティと、サーバー・コンポーネント定義を格納するための構成ファイルも含まれます (図 1-3 を参照)。

図 1-3 Oracle SQL/Services のクライアント/サーバー・アーキテクチャ



Oracle SQL/Services クライアントは、Oracle SQL/Services API (Microsoft ODBC 接続を含む)、Oracle OCI インタフェース、Oracle RMU インタフェース、Oracle Rdb JDBC インタフェースなどの規約に基づくインタフェースの使用により、サーバーで提供されるサービスを選択してデータにアクセスするソフトウェア・プログラムです。サーバーは、クライアント・リクエストを受信および処理することで応答し、結果をクライアントに返します。

ネットワークは、クライアント API が Oracle SQL/Services サーバーと通信するための通信ハードウェアおよびソフトウェアで構成されます。クライアントからのリクエスト・メッセージとサーバーからのレスポンス・メッセージは、DECnet、Transmission Control Protocol/Internet Protocol (TCP/IP)、NetWare (IPX/SPX)、AppleTalk-DECnet ゲートウェイ、または Oracle Net 通信リンクを介して送受信されます。

サーバー

Oracle SQL/Services サーバーは、1つ以上のサービスを提供する単一ノード上にある連携プロセスのコレクションの属性を表します。一般的に、サーバーには、サーバー・コンポーネントのすべての属性定義が含まれます (属性定義は構成ファイルに格納されます)。これらのサーバー・コンポーネントの詳細は、このリスト内の「サービス」および「ディスパッチャ」を参照してください。サーバー・オブジェクトに限定すると、これらの属性には、サーバーのバージョン、構成ファイルの指定、共有メモリのサイズ、サーバー管理でサポートされるネットワーク・トランスポートの種類などの情報が含まれます。

1つの構成ファイルに対して1つのサーバーのみを定義できます。任意の時点において1つのノードで起動できるのは、Oracle SQL/Services のリリースごとに1つのサーバーのみです。

サービス

Oracle SQL/Services サービスは、クライアントによるデータベースへのアクセス方法を記述した属性のセットです。Oracle SQL/Services は、サーバーにより提供されるサービスごとにカスタマイズされた一連のサービス属性をサポートすることで、様々なクライアントのニーズに対応します。サービスの定義には、サービスを使用できるユーザー、サービスによりアクセスされるデータベース、サービスにより使用されるデータベース・エンジンのバージョン、サービスを同時に使用できるクライアントの数、サービスのかわりに動作するエグゼキュータの数などの情報が含まれます。

エグゼキュータ

Oracle SQL/Services エグゼキュータは、サービスのかわりに動作するプロセスです。

エグゼキュータは、ディスパッチャからクライアント・リクエストを取得し、SQL をコールしてそのリクエストを処理してから、結果をディスパッチャに返します。起動されるサービスごとにエグゼキュータ・プロセスのプールが存在します。

ディスパッチャ

Oracle SQL/Services ディスパッチャは、クライアントのネットワーク通信を処理し、サービスのエグゼキュータを対象にクライアント・リクエストのルーティングとスケジューリングを行うプロセスです。ディスパッチャでは、サーバーに定義されたすべてのサービスがサポートされます。通常、単一のディスパッチャで複数のネットワーク・トランスポートがサポートされますが、サービス・プロトコルについては1つのみがサポートされます。システム管理クライアント以外のすべてのクライアントは、ディスパッチャに直接接続します。

モニター

Oracle SQL/Services モニター・プロセスは、サーバーの全体的な管理および制御機能（サーバーの起動と停止、構成データベースの読取りと書込み、監視機能、およびその他の管理操作など）を提供します。

SQLSRV_MANAGE クライアント

Oracle SQL/Services のサーバー管理コマンドライン・インタフェースにより、OpenVMS システムから Oracle SQL/Services サーバーを管理できます。

Oracle SQL/Services Manager クライアント GUI

Oracle SQL/Services のサーバー管理グラフィカル・ユーザー・インタフェース (GUI) ・ユーティリティにより、任意の Windows クライアントから Oracle SQL/Services サーバーを管理できます。Oracle SQL/Services Manager GUI は、Windows 95、Windows 98、Windows 2000、Windows 2003、Windows NT X86、Windows ME および Windows XP の各オペレーティング・システムで動作します。

構成ファイル

構成ファイルには、1つのサーバーとそのコンポーネントに定義されたすべての属性が含まれます。これには、すべてのサービス定義、ディスパッチャ定義、およびそのサーバーにより提供されるサービスにアクセスするための権限を持つユーザーのリストが含まれます。これは、Oracle SQL/Services サーバー構成とも呼ばれ、サーバー用にまとめて管理されるコンポーネント定義の1つのセットを表します。1つの構成ファイルには1つのサーバーのみを定義できます。通常、サーバー・ノードごとに独自の構成ファイルを用意しますが、OpenVMS クラスターの複数のノード間で1つの構成ファイルを共有することも可能です。

1.1.1 サーバー管理ユーティリティ

Oracle SQL/Services サーバーは、次の方法で管理できます。

- **SQLSRV_MANAGE ユーティリティの使用**

OpenVMS システムのローカルまたはリモート・ノードから **SQLSRV_MANAGE** ユーティリティを使用して、サーバーをオンライン管理またはオフライン管理できます（サーバーをオフライン管理するには、ローカル・ノードで操作する必要があります）。

通常、**SQLSRV_MANAGE** ユーティリティを使用して、サーバー構成をオンライン管理するために実行中のサーバーにシステム管理接続を確立し、実行中のサーバーや構成ファイルを操作するシステム管理機能を実行します。また、**SQLSRV_MANAGE** ユーティリティを使用すると、構成ファイルのサーバー・コンポーネント属性を直接操作することでサーバー構成をオフライン管理できます。システム管理機能のうち、オフラインでのみ実行する必要があるのは、新規サーバー構成を作成してサーバーを起動する操作です。

SQLSRV_MANAGE ユーティリティは、標準の入力デバイスまたはスクリプト・ファイルからコマンドを渡し、OpenVMS システム上で対話的に、またはバッチ・ジョブとして実行できます。

- **Oracle SQL/Services Manager グラフィカル・ユーザー・インタフェース (GUI) の使用**

リモート Windows クライアントから **Oracle SQL/Services Manager GUI** を使用して、Windows 95、Windows 98、Windows 2000、Windows 2003、Windows NT X86、Windows ME または Windows XP のいずれかを通じてサーバーをオンライン管理できます。

Oracle SQL/Services Manager GUI を使用して、サーバー構成をオンライン管理するために実行中のサーバーにシステム管理接続を確立し、実行中のサーバーまたは構成ファイル（あるいはその両方）を操作するシステム管理機能を実行します。

Oracle SQL/Services Manager GUI を使用して、新規サーバー構成の作成、サーバーの起動、または実行中のサーバーの停止といったサーバー管理機能をオフラインで実行することはできません。また、スクリプト・ファイルを実行することもできません。ただし、**Oracle SQL/Services Manager GUI** を使用して、実行中のサーバーを変更および再起動できます。

通常、これら 2 つのユーティリティのいずれかを対話的に使用して、サーバーとそのコンポーネントをオンライン管理します。習得と使用が容易であることから、ほとんどのサーバー管理タスクには、**Oracle SQL/Services Manager GUI** を使用します。**SQLSRV_MANAGE** ユーティリティを使用する必要があるのは、サーバーを起動する場合、または作成した特別なスクリプトを実行する場合のみです。

SQLSRV_MANAGE ユーティリティを使用してサーバーを管理する方法の詳細は、[2.4 項](#)を参照してください。**Oracle SQL/Services Manager GUI** を使用してサーバーを管理する方法の詳細は、Windows クライアントでこのユーティリティを起動し、「Help」プルダウン・メニューを選択してください。

1.1.2 サーバーの管理に必要な権限

OpenVMS で **SQLSRV_MANAGE** ユーティリティを使用してサーバーを起動するには、**SETPRV** 権限を付与されたアカウントか、またはすべての権限を付与されたアカウントを使用する必要があります。OpenVMS で **SQLSRV_MANAGE** ユーティリティを使用してサーバーをオフラインで変更するには、**NETMBX**、**SYSLCK** および **SYSPRV** 権限を付与されたアカウントを使用する必要があります。**SQLSRV_MANAGE** ユーティリティまたは **Oracle SQL/Services Manager GUI** を使用してサーバーをオンラインで変更するには、そのサーバーの **SQLSRV_MANAGE** システム管理サービスの使用権限を付与されたアカウントを使用する必要があります。ただし、高度な権限を付与されたアカウントを使用する必要はありません。サーバー構成ファイルが **SYSS\$MANAGER** ディレクトリ（デフォルトの場所）に存在する場合、**SYSPRV** 権限を付与されたアカウントを使用する必要があります。

これらの権限の要件は、OpenVMS プラットフォームに **Oracle SQL/Services** をインストールするのに必要な要件と同等か、それより制限が緩和されています。詳細は、**Oracle Rdb** および **Oracle SQL/Services** のインストール・ガイドを参照してください。

1.1.3 SQLSRV_MANAGE ユーティリティの実行

SQLSRV_MANAGE ユーティリティを実行するには、次のようにユーティリティを起動するためのシンボルを最初に定義する必要があります。

```
sqlsrv_manage71 := $$SYS$SYSTEM:sqlsrv_manage71
```

これで、コマンド `sqlsrv_manage71` を入力することで `SQLSRV_MANAGE` ユーティリティを起動できます。`SQLSRV_MANAGE` ユーティリティを対話的に使用するには、ユーティリティを起動してから、`SQLSRV>` コマンドライン・プロンプトに応答してシステム管理コマンドを入力します。サーバーをオンライン管理する場合、最初に使用するコマンドは、通常、`CONNECT TO SERVER` コマンドです。サーバー構成ファイルがデフォルトの場所に存在しない場合にサーバーをオフライン管理するには、最初に `SET CONFIGURATION_FILE` コマンドを使用してサーバー構成ファイルの名前を指定します（詳細は「[SET CONFIGURATION_FILE コマンド](#)」を参照）。

`SQLSRV_MANAGE` ユーティリティでスクリプトを使用することもできます。`SQLSRV_MANAGE` スクリプトは、`SQLSRV>` プロンプトに入力するものと同じコマンドを含むファイルです。`@` コマンドを使用して、`SQLSRV>` プロンプトで対話的に `SQLSRV_MANAGE` スクリプトを起動できます。別の方法として、`SQLSRV_MANAGE` ユーティリティを起動してスクリプトから直接システム管理コマンドを読み取ることができます。詳細は、[第 6 章の「-input スイッチ](#)」を参照してください。

スクリプトは、定期的にサーバーに変更を加えるための実用的なツールです。たとえば、ピーク負荷の状態に対応するためにサービスのエグゼキュータの最小数と最大数を増加する場合を考えてみます。1つのスクリプトで値を増加し、もう1つのスクリプトで値を減少できます。バッチ・ジョブを使用して、スクリプトの実行を自動化できます。

1.1.4 Oracle SQL/Services Manager GUI の実行

Oracle SQL/Services Manager GUI を実行するには、Oracle SQL/Services Manager GUI アイコンをクリックして `Connect To Server` ウィンドウを表示し、ノード名、ユーザー名およびパスワードを入力します。接続の確立に使用するトランスポート、および TCP/IP ポート ID または DECnet オブジェクト名を選択します。接続後、サーバーの管理を開始できます。

1.2 オンラインとオフラインのサーバー管理の比較

SQLSRV_MANAGE ユーティリティを使用して、オンラインまたはオフラインでサーバーを管理できます。

オンライン・サーバー管理

通常、サーバーはオンラインで管理します。サーバーをオンライン管理する場合、常に CONNECT TO SERVER コマンドを使用してサーバーに接続します。接続後、サーバーに加えた変更は、すべて構成ファイルに書き込まれます。また、動的属性を変更した場合、その変更は実行中のサーバーに反映されます。動的属性のリストは、[2.4.1 項](#)、[2.4.2 項](#)および [2.4.3 項](#)を参照してください。起動されているオブジェクトの動的属性以外の属性を変更した場合、システム管理ユーティリティにより、変更を反映するにはオブジェクトを再起動する必要があるというメッセージが表示されます。サーバーを再起動する必要があるのは、サーバーの変更がサーバー・オブジェクトそれ自体の動的属性以外の属性を対象とする場合のみです。この場合、変更は、サーバーの再起動操作時に反映されます。

オフライン・サーバー管理

場合によっては、サーバーを使用不能にした変更操作（共有メモリの値の設定を低くしすぎた場合など）からリカバリするために、サーバーをオフラインで管理する必要があります。サーバーをオフライン管理するには、SQLSRV_MANAGE ユーティリティを使用する必要があります。Oracle SQL/Services Manager GUI を使用してサーバーをオフライン管理することはできません。構成ファイルがデフォルトの場所に存在しない場合（詳細は「[SET CONFIGURATION_FILE コマンド](#)」を参照）、最初に SET CONFIGURATION_FILE コマンドを使用して構成ファイルを選択してからシステム管理コマンドを発行する必要があります。通常、サーバーをオフライン管理するのは、サーバーが実行されていない場合のみです。ただし、実行されているサーバーでも、オフライン管理は可能です。サーバー構成に加えられた変更は、すべて構成ファイルに書き込まれますが、変更されたオブジェクトが再起動されるまで実行中のサーバーには影響しません。サーバー・オブジェクトそれ自体の属性に対する変更を反映するには、サーバー全体を再起動する必要があります。ディスパッチャまたはサービス・オブジェクトの属性に対する変更を反映するには、その特定のディスパッチャまたはサービスを停止して起動するだけで済みます。唯一の例外は、サービスの使用権限をユーザー名または識別子を対象に付与または取り消す場合で、変更は即座に反映されます。

表 1-1 に、オンラインまたはオフライン、あるいはその両方で実行可能な Oracle SQL/Services のサーバー管理コマンドと、適用される制限を示します。

コマンド	オンライン	オフライン	コメント
ALTER DISPATCHER	○	○	オフラインの変更は、実行中のディスパッチャに影響しません。
ALTER SERVER	○	○	オフラインの変更は、実行中のサーバーに影響しません。
ALTER SERVICE	○	○	オフラインの変更は、実行中のサービスに影響しません。
CONNECT TO SERVER	○	-	オンライン・サーバー管理専用です。
CREATE DISPATCHER	○	○	オンラインまたはオフラインでディスパッチャを作成できます。
CREATE SERVER	-	○	オフラインでのみサーバーを作成できます。
CREATE SERVICE	○	○	オンラインまたはオフラインでサービスを作成できます。
DISCONNECT SERVER	○	-	オンライン・サーバー管理専用です。
DROP DISPATCHER	○	○	オンラインまたはオフラインでディスパッチャを削除できます。
DROP SERVER	-	○	オフラインでのみサーバーを削除できます。
DROP SERVICE	○	○	オンラインまたはオフラインでサービスを削除できます。

コマンド	オンライン	オフライン	コメント
GRANT USE ON SERVICE	○	○	オフラインの変更は、実行中のサーバーに影響します。
KILL EXECUTOR	○	-	オンラインでのみエグゼキュータを中断できます。
RESTART SERVER	○	-	オンラインでのみサーバーを再起動できません。
REVOKE USE ON SERVICE	○	○	オフラインの変更は、実行中のサーバーに影響します。
SET CONFIGURATION_FILE	-	○	オフライン・サーバー管理専用です。
SET CONNECTION	○	-	オンライン・サーバー管理専用です。
SHOW DISPATCHER	○	○	オンラインまたはオフラインでディスパッチャの定義属性を表示できます。オンラインでのみディスパッチャの実行時属性（その状態など）を表示できます。
SHOW SERVER	○	○	オンラインまたはオフラインでサーバーの定義属性を表示できます。オンラインでのみサーバーの実行時属性（その状態など）を表示できます。
SHOW SERVICE	○	○	オンラインまたはオフラインでサービス・オブジェクトの定義属性を表示できます。オンラインでのみオブジェクトの実行時属性（その状態など）を表示できます。
SHUTDOWN DISPATCHER	○	-	オンラインでのみディスパッチャを停止できます。
SHUTDOWN SERVER	○	-	オンラインでのみサーバーを停止できます。
SHUTDOWN SERVICE	○	-	オンラインでのみサービスを停止できます。
START DISPATCHER	○	-	オンラインでのみディスパッチャを起動できます。
START SERVER	-	○	オフラインでのみサーバーを起動できます。
START SERVICE	○	-	オンラインでのみサービスを起動できます。

第2章と第3章で、サーバーの管理およびメンテナンス方法について説明します。第6章には、SQLSRV_MANAGE コマンドについて記述したリファレンス資料が含まれます。これらの章は、主に SQLSRV_MANAGE ユーティリティとそのコマンドライン・インタフェースを使用する Oracle SQL/Services システム管理者を対象としています。これらの章の本文および例では、必要に応じて SQLSRV_MANAGE コマンドのコマンドライン・インタフェースと構文のみが使用されています。

Windows ベースの PC システムから Oracle SQL/Services Manager GUI クライアントを使用する Oracle SQL/Services システム管理者は、第2章および第3章を参照できますが、コマンドと構文の名前は SQLSRV_MANAGE コマンドライン・インタフェースと Oracle SQL/Services Manager GUI で異なることに注意してください。違いのほとんどは比較的軽微なため、これらの違いに関する説明はありません。より具体的な情報は、Oracle SQL/Services Manager GUI クライアントの Windows ヘルプを参照してください。

1.3 OCI Services for Oracle Rdb

OCI Services for Oracle Rdb (旧 SQL*Net for Oracle Rdb) が提供する環境により、既存の OCI アプリケーションを実行して Oracle Rdb データベースのデータにアクセスできます。OCI アプリケーションでは、Oracle Call Interface (OCI) を使用してデータベースのデータにアクセスし、それらのデータを管理します。

OCI Services for Oracle Rdb は、Oracle クライアントを Oracle Rdb サーバーに接続します。OCI Services for Oracle Rdb により提供される独自のメリットは、Oracle SQL セマンティクスを使用して Oracle Rdb データベースのデータにアクセスできることです。

OCI Services for Oracle Rdb の特長は次のとおりです。

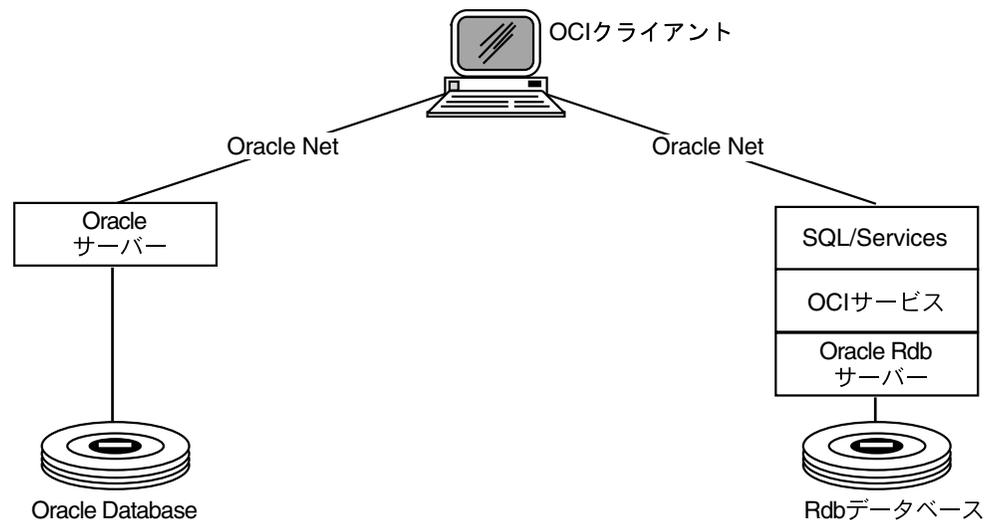
- クライアント・アプリケーションに対して自身を Oracle Database サーバーとみなします。
- Oracle SQL セマンティクスの多くをエミュレートします。
- ネットワーク通信に Oracle SQL/Services を使用します。

OCI Services for Oracle Rdb により、Oracle Rdb または Oracle の RDBMS データベース・インスタンスに対して実行される単一のソース・コード・ストリームを作成することで、クライアント・アプリケーションの適用範囲が広がります。

たとえば、アプリケーションで Oracle Rdb データベースによる機能が必要な場合、Oracle RDBMS サーバーのかわりに Oracle Rdb サーバーを使用できます。Oracle Rdb サーバーのみを使用する場合、OCI Services for Oracle Rdb により、Oracle Rdb アプリケーションに対して OCI アーキテクチャの多くの機能が提供されます。

図 1-4 に、OCI Services for Oracle Rdb 環境でのクライアント / サーバー関係を示します。

図 1-4 クライアント / サーバー処理



1.3.1 Oracle Call Interface

OCI Services for Oracle Rdb の独自機能は、中核となるオープン・テクノロジーの Oracle Call Interface (OCI) を通じて実現されます。

OCI アプリケーションおよびツールは、OCI サービスのクライアント / サーバー環境で実行されます。これにより、サーバーおよびクライアント・ハードウェアとオペレーティング・システム環境の多様な組合せが可能になります。OCI アーキテクチャにより、クライアントのユーザー・インタフェースがサーバー実装から分離されるため、既存のサーバーに異なるユーザー・インタフェースを新規に追加することや、ユーザー・インタフェースに影響を与えることなくサーバー実装を変更することが可能になります。

1.3.2 サーバー側のソリューション

OCI Services for Oracle Rdb は、Oracle クライアント・アプリケーションを直接 Oracle Rdb サーバーに接続することで、OCI の柔軟性を利用します。

OCI Services for Oracle Rdb は、サーバー側のソリューションとして設計されているため、Oracle サーバーをクライアント・プラットフォームの様々なセットと組み合わせて使用できるように、簡単に効率的な方法でクライアント・プラットフォームの様々なセットと組み合わせて使用できます。

クライアント・システムで使用される Application Program Interface (API) ソフトウェアは、OCI Services for Oracle Rdb のサーバー側の実装から明確に分離されています。クライアント・アプリケーションを開発するための API ソフトウェア (OCI や任意の Pro* コンパイラなど) は、クライアント・システムごとに個別に購入してインストールする必要があります。

OCI アプリケーションを新規に作成するには、その新規 OCI アプリケーションの開発と構築に必要な特定の OCI ソフトウェアをインストールする必要があります。既存の OCI アプリケーションは、OCI Services for Oracle Rdb を使用するために追加のクライアント・ソフトウェアを購入、インストール、構成または管理することなく実行できます。

OCI Services for Oracle Rdb は、クライアントにとって Oracle サーバーとして認識されます。クライアントは、Oracle RDBMS サーバーと対話する場合と同じ方法で OCI Services for Oracle Rdb と対話します。通常、クライアントは、Oracle データ・ディクショナリを問い合わせターゲット・データベースのメタデータ情報を取得し、他の OCI コールを何回か実行してデータベースのデータを問い合わせ、それら进行操作します。

注意： Oracle Rdb の SQL 言語のメタデータは、Oracle サーバーのメタデータとは大幅に異なります。付属の Oracle Rdb SQL プログラムを使用して作成された Oracle データ・ディクショナリを使用すると、クライアント・ソフトウェアにとって重要な Oracle データ・ディクショナリのほとんどの要素を OCI Services for Oracle Rdb でエミュレートできます。

1.3.3 共通のアプリケーション開発

OCI Services for Oracle Rdb は、Oracle Rdb サーバーと Oracle RDBMS サーバーの両方で実行できるソフトウェアを作成する SQL プログラマを支援するために開発されました。

Oracle Rdb サーバーに対する OCI Services for Oracle Rdb クライアント・アプリケーションおよびツールの実行を支援するため、OCI Services for Oracle Rdb では、表 1-1 に記載された処理機能により Oracle Rdb SQL の機能が拡張されています。

表 1-1 OCI Services for Oracle Rdb の処理機能

機能	説明
カーソル管理	文ごとに OCI カーソルを管理し、各カーソルを Oracle Rdb SQL 文に関連付けます。
OCI メッセージ・マッピング	OCI コールを Oracle Rdb の動的 SQL コールにマップします。
Oracle データ型	Oracle Rdb データ型を Oracle データ型として記述および変換します。
データ書式設定	Oracle スタイルの書式設定を実行します (Oracle サーバーは、クライアント用にデータを書式設定するか、クライアントから書式設定されたデータを取得します)。書式設定された情報は、Oracle SQL の ALTER SESSION 文によりサーバーに対して示されます。
文の変更	失敗した SQL 文に、Oracle Rdb の構文に準拠するよう必要な変更を加えた後に、その文を再処理します。この処理は、Oracle と Oracle Rdb 間の構文上の相違の数が限定されている場合に実行されます。通常、この再処理によって SQL 文を正常に実行できます。
文のタイプ	Oracle Rdb SQL から解析された SQL 文のタイプを取得し、それをクライアントに返します。
データ定義言語 (DDL)	Oracle サーバーでの動作と同様の動作を実現する DDL および SQL カーソル・セマンティクスが提供されます。たとえば、各 DDL リクエストの前後に COMMIT 文が発行されます。
データ・ディクショナリ	Oracle に特有のメタデータ表のスタイルを提供するために、Oracle データ・ディクショナリをエミュレートするビューおよびストアド・プロシージャのコレクションが提供されます。
マルチスキーマ・エミュレーション	Oracle マルチスキーマ・データベースで実現されるものと同様のマルチスキーマ環境をエミュレートします (すべての Oracle Database はマルチスキーマ・データベースですが、ほとんどの Oracle Rdb データベースはマルチスキーマ・データベースではありません)。表名は、複数のスキーマで使用できませんが、データ・ディクショナリではマルチスキーマ表現が可能です。

これらすべての機能により、OCI Services for Oracle Rdb を使用して、Oracle RDBMS サーバーと Oracle Rdb サーバー間で共通のアプリケーション開発が可能になります。

第 4 章で、OCI Services for Oracle Rdb の処理を詳細に説明します。

Oracle SQL/Services システムの管理

Oracle SQL/Services システムを管理するには、[第 1 章](#)で説明されているディスパッチャ、サービスおよびサーバーに加え、クライアントおよびネットワーク・コンポーネントに関する知識が必要です。ユーザーは、クライアント / サーバー・アーキテクチャで各コンポーネントが他のコンポーネントとどのように連携するか、またサーバー・システム内のコンポーネントがどのように動作するかを理解する必要があります。この章では、サーバー・コンポーネントを作成および管理する方法について説明します。

特に記載のないかぎり、この章の情報は SQL/Services および OCI Services for Oracle Rdb のプロトコルに適用されます。

2.1 スタート・ガイド

デフォルトの Oracle SQL/Services サーバーをインストールして起動したら、最適なパフォーマンスを確保し、問題をトラブルシューティングするために、状況に応じていくつかの追加タスクを実行します。これらのタスクは次のとおりです。

- Oracle SQL/Services サーバー構成の計画
- 共有メモリー・サイズの設定
- サーバー・コンポーネントの管理
- ディスパッチャおよびトランスポート選択の設定
- サービスおよび再利用タイプの設定
- サーバーのセキュリティの設定
- Oracle SQL/Services によるデータベース・アクセス認可モデルの実装方法の理解
- サービス所有者ユーザー名を選択する際のセキュリティの検討
- エグゼキュータ属性の設定
- SQL 初期化ファイルの使用
- SQL/Services の論理名の使用

各内容は、後続の項に記載されています。

2.2 Oracle SQL/Services サーバー構成の計画

Oracle SQL/Services サーバーの初期動作は、構成ファイルにより定義されます。構成ファイルには、サーバー、ディスパッチャ、サービスのオブジェクト定義と特性、および各サービスの認可ユーザー・セットが含まれます。SHOW コマンドで各オブジェクトの現在の定義を表示し、属性設定をすべて確認できます。この基本的な理解に基づいて、サーバー構成を計画するための次の手順を実行します。

1. 使用するサーバー・システムの独自要件を決定します。
2. 各オブジェクトとそれらの最適な管理方法について情報を収集します。
3. サーバー・システムの要件を満たすように、確認した内容を適用します。

サーバー・システム要件の決定

Oracle SQL/Services システム管理者は、使用するサーバー・システムの要件を決定する必要があります。次の点を調査する必要があります。

- Oracle SQL/Services は、単一ノードとクラスタのどちらにインストールされていますか。異なるノードでは異なるディスパッチャおよびサービスが必要ですか。
- ユーザー・コミュニティに関して知っていることは何ですか。クライアントの数は合計でいくつですか。ピーク時にシステムを使用するクライアントの数はいくつですか。
- クライアント / サーバー通信に使用できるトランスポートは何ですか。各トランスポートに使用できるポートの数はいくつですか。
- インストールされている Oracle Rdb のリリースは何ですか。
- ユーザーが実行を希望している特定のアプリケーションは何ですか。ユーザーは、同じデータベースと異なる多くのデータベースのどちらにアタッチされますか。実行されるトランザクションの種類は何ですか。

これらは、回答する必要のある最も重要な質問です。作業を進めるにつれ、独自のサーバー要件の理解に役立つ他の質問が出現する可能性もあります。また、現在のユーザー・コミュニティのサーバー要件を最も適切に満たす方法と、サーバー・システムをチューニングして最大のパフォーマンスを発揮する方法について計画を立てる必要もあります。

サーバー・オブジェクトの情報収集

初めに、各サーバー・オブジェクトに関する次の質問を検討します。

- 監視する必要のある属性はどれですか。
- 管理対象として最も考慮する必要のある属性はどれですか。

これらの質問に回答するには、各属性のデフォルト値の意味を理解し、監視および調整する必要のある属性を決定することが重要です。一般的に、作業を開始する際には、デフォルト・サーバー・システムに対応する属性のすべてのデフォルト設定で十分です。表 2-1、表 2-2 および表 2-3 に、サーバー、ディスパッチャおよびサービス・オブジェクトのデフォルト値のまとめを示します。各表の後に、監視および調整する必要のある属性の簡単な説明があります。

サーバー・システム要件の達成

各サーバー・コンポーネントの最も重要な属性に関する特定の質問に回答することで、サーバー・システムにどのような変更を加える必要があるかを決定できます。計画を実施する過程で、サーバー・コンポーネント・オブジェクトの作成および変更方法を習得し、サーバー・システム要件を満たすためにそれらの変更を適用することになります。

各オブジェクトの監視および調整方法を習得するにつれ、サーバーのパフォーマンスを最適化し、独自のデータベース・クライアント / サーバー環境にとって理想的な状態に Oracle SQL/Services サーバーをカスタマイズできるようになります。たとえば、ユーザーが実行を希望しているアプリケーションを把握したら、それらのクライアント・アプリケーション用に提供する必要のあるサービスの種類を決定できます。

実行サーバーを確立するために検討する必要のある最も重要な項目は、2.4 項から 2.8 項に記載されています。

クライアント / サーバー要件を満たすように Oracle SQL/Services サーバーをカスタマイズしたら、次のタスクは、サーバーをメンテナンスする方法についてより詳しく理解することです（詳細は第 3 章を参照）。

2.3 共有メモリー・サイズの設定

サーバーで使用する共有メモリーのサイズを設定するには、ALTER SERVER コマンドの MAX_SHARED_MEMORY_SIZE 引数の値を指定します。サーバーでは、デフォルトで 2000KB (2MB) の共有メモリーが使用されます。

MAX_SHARED_MEMORY_SIZE 引数を設定することは、サーバー・システムのリソース使用量を最適化するために重要です。この作業の目的は、必要とされるサービスの提供を可能とする最小量の共有メモリーを使用することです。この項では、Oracle SQL/Services サーバーで共有メモリーを使用する方法と、最適なリソース使用量を確保するために MAX_SHARED_MEMORY_SIZE 引数を設定する方法について説明します。

共有メモリーの値は、ALTER SERVER コマンドを使用して変更できます。ただし、これは動的属性ではないため、サーバーを再起動する必要があります。たとえば、値を 4000KB に設定するには、次のようにします。

```
SQLSRV> CONNECT SERVER;
Connecting to server ...
Connected
SQLSRV> ALTER SERVER MAX_SHARED_MEMORY_SIZE 4000;
%DBS-S-ALTER_RESTART, Restart object to have altered settings take effect
SQLSRV> RESTART SERVER;
Disconnected from Server
```

共有メモリーの割当てに関連する2つの主なコストは、次のとおりです。

- システム・ページ・ファイルのディスク領域
共有メモリー・サイズの増加に対応するのに十分なシステム・ページ・ファイル領域が存在することを確認してください。また、`GBLPAGFIL SYSGEN` パラメータも共有メモリー・サイズの増加に対応できる程度に大きな値に設定されている必要があります。
- サーバーの各コンポーネント・プロセス用の仮想メモリー
共有メモリーをマップすると、サーバーの各コンポーネント・プロセスで使用される仮想メモリーが増大するため、オペレーティング・システム・リソースの消費も増大します。

内部的に、Oracle SQL/Services は、共有メモリーを 65,536 バイト単位で管理します。共有メモリーの実際のサイズは、`MAX_SHARED_MEMORY_SIZE` 属性の値未満となる可能性があります。これは、共有メモリーのサイズは、常に最も近似する 65,536 バイト単位の値にまで切り捨てられるためです。

サーバーの共有メモリーは、サーバーの実行に合わせて減少または増大しません。特定の時点では、使用中の値が増減する可能性があります。接続中のサーバーで `SQLSRV_MANAGE` アプリケーションの `SHOW SERVER` コマンドを発行すると、`SQLSRV_MANAGE` により次の3つの値が表示されます。

- 合計メモリー
この値は、サーバーの特定の実行において静的です。サーバーの `MAX_SHARED_MEMORY_SIZE` 引数を変更できます (`RESTART SERVER` コマンドの発行時に反映されます)。合計メモリーは、サーバーによりマップされた共有メモリー単位の数値です。
- 空きメモリー
これは、Oracle SQL/Services サーバーによりまったく使用されていない共有メモリー単位の数値です。
- 部分割当てメモリー
共有メモリー単位は、1つのまとまりとして使用されないことがあります。通常は、より小さなまとまりに分割されます。分割されて部分的に使用されている共有メモリー単位は、部分割当てメモリーとして報告されます。現在のところ、分割された単位内の使用量を表示することは実用的ではありません。

空きメモリーと部分割当てメモリーは、割当て可能な共有メモリー単位を表します。これらの単位を合計単位から差し引くと、完全に使用されている共有メモリー単位を判定できます。

`MAX_SHARED_MEMORY_SIZE` の最小値は、132KB であり、2つ分の共有メモリー単位に対応します。これは、モニターを開始して `SQLSRV_MANAGE` アプリケーションから接続し、1つまたは2つのクライアントを処理する1つまたは2つのエグゼキュータを実行するのに十分な容量です。

`MAX_SHARED_MEMORY_SIZE` の最大値は、2,000,000KB です。通常は、これより小さい値ですべてのアプリケーションに対応できます。

一般的に、共有メモリーの使用量は次のように計画します。

- 実行する予定のエグゼキュータおよびディスパッチャごとに、約 3KB を確保します。
- サポートする予定の Oracle SQL/Services クライアント接続ごとに、クライアントのベース共有メモリーの使用量を考慮し、それに通信バッファで使用されるメモリーを追加する必要があります。

ベース共有メモリーの使用量は、約 11KB です。

Oracle SQL/Services アプリケーションでは、最低でも2つの通信バッファが使用されます。デフォルトのバッファ・サイズは 1.3KB であるため、Oracle SQL/Services クライアント用の最小サイズは、15KB になります (11KB のベース共有メモリー + 4KB のメッセージ・バッファ)。

5KB のメッセージ・バッファ・サイズを使用する場合、最小サイズは約 21KB になります (11KB のベース共有メモリー + 2 * 5KB のメッセージ・バッファ)。

ただし、すべての Oracle SQL/Services アプリケーションが 2 つのバッファのみを使用するわけではありません。マルチタブル・フェッチまたは挿入操作が開始されると、クライアントに追加バッファが割り当てられることがあります。割り当てられる追加バッファの数は、アプリケーションに基づきます。任意の時点で使用できるバッファの数は、ディスクパッチャにより 11 に制限されています。

最適な共有メモリー・サイズを決定するための方針は、次のとおりです。

1. 前述した大まかなサイズ判定方法に基づいて、十分なサイズの共有メモリーを決定します。
2. 通常の負荷でシステムを実行します。
3. 管理対象のサーバーに `SQLSRV_MANAGE` から `SHOW SERVER` コマンドを何回か発行します。これにより、メモリー使用量が表示されます。
4. 共有メモリー・サイズを調整します。
 - 一定数の空きメモリー単位が表示される場合は、サイズを減らします。
 - 空きメモリー単位が表示されない場合は、サイズを増やします。共有メモリーの不足が原因で、クライアント接続がサーバーにより切断されることがあります。これは、ログ・ファイルに記録されます。ごくまれに、不十分な共有メモリーが原因でサーバー全体が停止することもあります。

サーバーに新規ユーザーとアプリケーションを追加するたびに、共有メモリーの使用量を確認してください。

2.4 サーバー・コンポーネントの管理

サーバー・コンポーネントの管理作業は、サーバー、ディスクパッチャおよびサービスの管理と、各オブジェクトの作成、各オブジェクトの起動、停止および再起動、オブジェクト属性の変更、各オブジェクトの削除などのタスクの実行で構成されます。2.4.1 項から 2.4.3 項で、これらの各オブジェクトの管理方法について説明します。

2.4.1 サーバーの管理

サーバーを管理するには、サーバーの作成方法、サーバーの起動、停止および再起動方法に加え、クライアント / サーバー構成の特定の要件を満たすようにサーバーの属性をカスタマイズする方法を理解する必要があります。

サーバーの作成

Oracle SQL/Services をインストールすると、インストール・プロシージャによって自動的にサーバーが作成され、そのノードで起動されます。リカバリできないエラー状態が発生して構成ファイルが使用不能にならないかぎり、通常は Oracle SQL/Services のインストールを実行したノードでサーバーを作成または再作成する必要はありません。ただし、定期的に構成ファイルのバックアップ・コピーを保存する必要があります。構成ファイルのコピーを作成する方法の詳細は、この項の「[構成ファイルのコピー](#)」を参照してください。

ディスク障害などが原因で構成ファイルが破損し、バックアップ・コピーも存在しない場合、`SYSS$MANAGER:SQLSRV_CREATE71.COM` コマンド・プロシージャを使用すると、破損ファイルを削除して初期サーバー構成を再作成できます。

OpenVMS クラスタ環境では、インストール・プロシージャにより、インストールを実行したノードでのみサーバーが作成および起動されます。クラスタ内の他のノードで Oracle SQL/Services を使用する場合、それらの各ノードでサーバーを作成および起動するか、他のノードで使用できる単一の構成ファイルを作成して、それらのノードでサーバーを起動する必要があります。

OpenVMS クラスタの他のノードでサーバーを作成および起動するには、次の2つの方法があります。

- インストールで提供される SYS\$MANAGER:SQLSRV_CREATE71.COM プロシージャを使用する方法

クラスタ内の他のノードでサーバーを作成および起動する際に推奨される方法は、Oracle SQL/Services のインストール・プロシージャにより提供される SYS\$MANAGER:SQLSRV_CREATE71.COM DCL コマンド・プロシージャを起動することです（詳細は『Oracle SQL/Services インストレーション・ガイド』を参照）。このプロシージャは、インストール・プロシージャ自体に使用されるため、最初のインストールが実行されたノードで作成されたサーバーと同じサーバーが作成および起動されます。

- クラスタ内の別のノードから構成ファイルをコピーする方法

クラスタ内の他のノードでサーバーを作成する別の方法では、そのノードに構成ファイルをコピーし、そのノード用に必要な変更を加えてからそのノードでサーバーを起動します。この方法は、エラーが発生しやすいことから前述の方法より困難ですが、それでも選択肢の1つとなります。詳細は、この項の「構成ファイルのコピー」を参照してください。

別の方法として、クラスタ内の複数のノード間で1つの構成ファイルを共有できます。クラスタ内のすべてのノードで使用できる単一の構成ファイルを作成する最も簡単な方法は、インストールを実行したノードのサーバーを停止し、SQLSRV_CONFIG_FILE71.DAT ファイルの名前を SYS\$SPECIFIC:[SYSMGR] ディレクトリから SYS\$COMMON:[SYSMGR] ディレクトリに変更することです。クラスタ内の複数のノード間で単一の構成ファイルを共有する場合、他のノードで任意のオブジェクトを継続して使用するには、いずれかのノードでそのオブジェクトを削除しないよう注意する必要があります。

クラスタ内の各ノードでまったく同じディスパッチャとサービスを使用する場合、追加のタスクを実行する必要はありません。ただし、異なるネットワーク・プロトコルをサポートするか、クラスタ内の異なるノードで特定のサービスを使用する場合は、適宜構成をカスタマイズする必要があります。異なるノードで異なるディスパッチャまたはサービスを使用する場合、一部のノードで起動しないサービスとディスパッチャについて AUTOSTART 属性を OFF に設定し、必要なディスパッチャとサービスのみを起動するノードごとに SQLSRV_MANAGE スクリプトを記述します。異なるノードで異なる属性を持つように共有構成ファイルでサービスまたはディスパッチャ・オブジェクトを構成することはできません。

注意： 同じファイルを使用している実行中のサーバーが存在する場合、構成ファイルをオフラインで変更しないことをお勧めします。この場合、SQLSRV_MANAGE ユーティリティでは、ディスパッチャまたはサービスが実行されているときに、ユーザーがそのディスパッチャまたはサービス・オブジェクトをオフラインで削除することを防止できません。

同様に、SQLSRV_MANAGE ユーティリティまたは Oracle SQL/Services Manager GUI ユーティリティでは、2つ以上のノードで同じ構成ファイルが共有されている環境で、ディスパッチャまたはサービスが異なるノードで実行されているときに、ユーザーがそのディスパッチャまたはサービス・オブジェクトをオンラインで削除することを防止できません。このとき、削除されたがまだ実行中のディスパッチャまたはサービスを表示すると、SQLSRV_MANAGE ユーティリティによって警告メッセージが表示されます。

Oracle SQL/Services Manager GUI ユーティリティでは、削除されたがまだ実行中のディスパッチャまたはサービスに対して、削除アイコン（立入り禁止マーク）が表示されます。

サーバーの起動、停止および再起動

システム起動ファイルに次の Oracle SQL/Services 起動コマンドを追加することをお勧めします。

```
@SYS$STARTUP:SQLSRV$STARTUP71
```

Oracle Rdb の起動コマンド・ファイル RMONSTART の後にこのコマンドを配置します。このコマンドを追加すると、サーバーはシステムの起動時に常に起動されます。

また、システム停止コマンド・プロシージャに次の停止コマンドを追加することをお勧めします。

```
@SYS$STARTUP:SQLSRV$SHUTDOWN71
```

Oracle Rdb の停止プロシージャ RMONSTOP の前にこのコマンドを配置します。このコマンドを追加すると、サーバーはシステムの停止時に常に停止されます。

一般的に、サーバーを再起動する必要があるのは、サーバー・オブジェクトの動的属性以外の属性を変更した場合のみです。この場合、変更を反映するにはサーバーを再起動する必要があります。

サーバーの変更

サーバーを作成したら、状況に応じていくつかのサーバー属性（サーバーで使用できる共有メモリの最大量など）を変更する必要があります。表 2-1 に、サーバーのすべての属性とそのデフォルト値をリストし、その属性を動的に変更できるかどうかを示します。表の後に、主なサーバー属性の簡単な説明を記載します。

表 2-1 サーバー・オブジェクト属性のデフォルト設定

属性	デフォルト設定	動的属性
MAX_SHARED_MEMORY_SIZE	2000KB	
構成ファイル	SYS\$MANAGER:SQLSRV_CONFIG_FILE71.DAT	
PROCESS_STARTUP_TIMEOUT	0	○
PROCESS_SHUTDOWN_TIMEOUT	0	○
ネットワーク・ポート	DECnet - SQLSRV_SERVER	
ネットワーク・ポート	TCP/IP - 2199	

Oracle SQL/Services では、プロセス間通信に共有メモリーを使用します。

MAX_SHARED_MEMORY_SIZE 属性は、**SHOW SERVER コマンド**を使用して定期的に監視する必要のある唯一のサーバー属性です。2.3 項で、監視対象と調整の実施時期について説明します。サーバーでは、システム管理クライアントをリスニングするためにネットワーク・ポートを使用します。デフォルトのネットワーク・ポートを使用できるのは Oracle SQL/Services の 1 つのリリースのみであるため、マルチリリース環境ではこれらのネットワーク・ポートは一意である必要があります。マルチリリースのインストール中に、サーバーで使用する代替ネットワーク・ポートを指定する必要があります。現在のリリースの Oracle SQL/Services をデフォルトにせず、デフォルトのシステム管理ネットワーク・ポートを使用しない場合は、これらのネットワーク・ポートに変更を加える必要はありません。システム管理クライアントで接続問題が発生した場合は、**SHOW SERVER コマンド**を使用してこれらのネットワーク・ポートを監視し、各ポートが実行中であることを確認します。

サーバー属性の変更方法の詳細は、「**ALTER SERVER コマンド**」を参照してください。

実行中のサーバーの動的属性をオンラインで変更すると、その変更は即座に反映されます。ただし、実行中のサーバーの動的属性以外の属性をオンラインで変更する場合、変更を反映するにはサーバーを再起動する必要があります。

実行中のサーバーの動的属性以外の属性を SQLSRV_MANAGE ユーティリティを使用して変更すると、変更を反映するにはサーバーを再起動する必要があるという内容の成功ステータスが表示されます。次に例を示します。

```
SQLSRV> ALTER SERVER MAXIMUM_SHARED_MEMORY_SIZE 4000;
%DBS-S-ALTER_RESTART, Restart object to have altered settings take effect
SQLSRV>
```

SQLSRV_MANAGE ユーティリティでは、サーバーの再起動時に反映される、変更済の動的属性以外の属性の値が表示されます。次に例を示します。

```
** The Server will be updated as follows when it is restarted **
    Max Shared memory size: 4000
```

実行中のサーバーの動的属性以外の属性を Oracle SQL/Services Manager GUI ユーティリティを使用して変更すると、変更を反映するにはサーバーを再起動する必要があるという内容の成功ステータスが表示され、即座にサーバーを再起動するかどうかを尋ねられます。Oracle SQL/Services Manager GUI ユーティリティでは、サーバーの再起動時に反映される、変更済の動的属性以外の属性の値が、各属性の横に旗の形のアイコンが表示されることで強調されます。

サーバーを再起動すると、サーバーのすべてのディスパッチャとサービスも再起動されます。また、サーバーに対するすべてのクライアント・ネットワーク接続は切断されます。したがって、サーバー・オブジェクトの変更は、サーバーを使用するクライアントが少ないか、またはまったく存在しないときにスケジュールする必要があります。

構成ファイルのコピー

DCL COPY コマンドを使用すると、構成ファイルのコピーを作成できます（そのファイルを使用する実行中のサーバーが存在しない場合のみ）。実行中のサーバーにより現在使用されている構成ファイルのコピーを作成するには、DCL BACKUP/IGNORE=INTERLOCK コマンドを使用する必要があります。

サーバーの削除

サーバーを削除する必要があるのは、ディスク障害などが発生して構成ファイルが破損し、完全に使用不能になった場合のみです。サーバーの削除は、オフライン操作であり、構成ファイルを削除します（「DROP SERVER コマンド」を参照）。かわりに、DCL DELETE コマンドを使用することもできます。

実行中のサーバーを削除する必要がある場合、最初にそのサーバーをオンラインで停止してから（「SHUTDOWN SERVER コマンド」を参照）、DROP SERVER コマンドを使用してオフラインで削除します。

2.4.2 ディスパッチャの管理

ディスパッチャを管理するには、ディスパッチャの作成方法、ディスパッチャの起動、停止および再起動方法に加え、クライアント / サーバー構成の特定の要件を満たすようにディスパッチャの属性をカスタマイズする方法を理解する必要があります。

ディスパッチャの作成

Oracle SQL/Services のインストール・プロシージャと SYS\$MANAGER:SQLSRV_CREATE71.COM コマンド・プロシージャにより、3つのディスパッチャが作成および起動されます。これらのディスパッチャは、Oracle SQL/Services (ODBC for Oracle Rdb) クライアントに使用される SQLSRV_DISP、OCI クライアントに使用される OCI_DISP、および Oracle RMU クライアントに使用される RMU_DISP の3つです。

Oracle Net ネットワーク・トランスポートを使用する予定の場合、使用するネットワーク・ポートの決定後に別のディスパッチャを作成します。ネットワークで使用可能なトランスポートごとに個別のディスパッチャ・プロセスを提供する場合、他のディスパッチャを作成することも可能です。新規ディスパッチャを作成する場合、指定するネットワーク・ポートがノード上の他のディスパッチャに使用されていないことを確認する必要があります。ディスパッチャでどのネットワーク・ポートもリスニングできない場合、ログ・ファイルにエラー・メッセージが書き込まれ、ディスパッチャは停止します。

ディスパッチャの起動、停止および再起動

AUTOSTART 属性が ON に設定されているディスパッチャは、Oracle SQL/Services のインストール時と、サーバーの起動時に常に自動的に起動されます。必要な場合は、START SERVER AUTOSTART OFF コマンドを使用してサーバーを起動することで、この動作を無効化できます。ディスパッチャは、サーバーの停止時に自動的に停止します。ごくまれに、起動に失敗したディスパッチャを停止する必要があります。失敗したディスパッチャは、常に失敗状態のまま残されます。失敗の原因（定義内に指定された引数の値が間違っていることが多い）は、ALTER DISPATCHER コマンドを使用して訂正できます。ディスパッチャを停止して訂正し、START DISPATCHER コマンドを使用してディスパッチャを起動するか、またはサービスが失敗状態にあるときに変更を加え、変更の完了後にディスパッチャを停止して再起動できます。

一般的に、ディスパッチャを再起動する必要があるのは、ディスパッチャ・オブジェクトの動的属性以外の属性を変更した場合のみです。この場合、変更を反映するにはディスパッチャを再起動する必要があります。

ディスパッチャの変更

環境が変化するにつれ、一部のディスパッチャ属性を変更する必要があります。たとえば、追加ユーザーをサポートするには、状況に応じてディスパッチャに許可される接続の最大数を増加する必要があります。パフォーマンスを向上するには、最大クライアント・バッファ・サイズを増加する必要があります。

Oracle SQL/Services サーバーの複数のリリースを実行する場合、Oracle SQL/Services の 1 つのリリースの使用を停止したときに、デフォルトのネットワーク・ポートを使用するようにネットワーク・ポート指定を変更できます。

表 2-2 に、ディスパッチャのすべての属性とそのデフォルト値をリストし、その属性を動的に変更できるかどうかを示します。表の後に、主なディスパッチャ属性の簡単な説明を記載します。

表 2-2 ディスパッチャ・オブジェクト属性のデフォルト設定

属性	デフォルト設定	動的属性
AUTOSTART	ON	
MAX_CONNECTIONS	100	
IDLE_USER_TIMEOUT	0	○
MAX_CLIENT_BUFFER_SIZE	5000	
ログ・ファイル	SYS\$MANAGER:<dispatcher-name>.LOG	
ダンプ・ファイル	SYS\$MANAGER:<dispatcher-name>.DMP	
メッセージ・プロトコル	SQLSERVICES	
ネットワーク・ポート	DECnet - 81	
ネットワーク・ポート	TCP/IP - 118	
ネットワーク・ポート	IPX/SPX - 0x84b1	
ネットワーク・ポート	Oracle Net (デフォルトなし。使用できるリスナー・オブジェクトのリストは、listener.ora ファイルを参照。)	

ディスパッチャに同時に接続するクライアントが 100 を超えると予想される場合、MAX_CONNECTIONS 引数の値を大きめに設定します。

大きなバッファ・サイズの使用により特定のアプリケーションにメリットがあることが判明している場合、MAX_CLIENT_BUFFER_SIZE 引数の値を大きめに設定します。

ディスパッチャでは、Oracle SQL/Services、Oracle ODBC Driver for Oracle Rdb、Oracle RMU、Oracle OCI および Oracle Rdb JDBC クライアントをリスニングするためにネットワーク・ポートを使用します。ノード上のディスパッチャに対してデフォルトのネットワーク・ポートを使用できるのは Oracle SQL/Services の 1 つのリリースのみであるため、マルチリリース環境ではこれらのネットワーク・ポートは一意である必要があります。マルチリリース

のインストール中に、ディスパッチャで使用する代替ネットワーク・ポートを指定する必要があります。DECnet ネットワーク・ポート専用リスニングする任意のディスパッチャを作成し、TCP/IP ネットワーク・ポート専用リスニングする別のディスパッチャを作成するなどの場合を除き、ネットワーク通信量を節約するため、これらのネットワーク・ポートに変更を加える必要はありません。クライアントでディスパッチャに対する接続問題が発生した場合は、SHOW DISPATCHER コマンドを使用してこれらのネットワーク・ポートを監視し、各ポートが実行中であることを確認します。

ディスパッチャ属性の変更方法の詳細は、「ALTER DISPATCHER コマンド」を参照してください。

実行中のディスパッチャの動的属性をオンラインで変更すると、その変更は即座に反映されません。ただし、実行中のディスパッチャの動的属性以外の属性をオンラインで変更する場合、変更を反映するにはディスパッチャを再起動する必要があります。

実行中のディスパッチャの動的属性以外の属性を SQLSRV_MANAGE ユーティリティを使用して変更すると、変更を反映するにはディスパッチャを再起動する必要があるという内容の成功ステータスが表示されます。次に例を示します。

```
SQLSRV> ALTER DISPATCHER sqlsrv_disp MAX_CLIENT_BUFFER_SIZE 10000;  
%DBS-S-ALTER_RESTART, Restart object to have altered settings take effect
```

SQLSRV_MANAGE ユーティリティでは、ディスパッチャの再起動時に反映される、変更済の動的属性以外の属性の値が表示されます。次に例を示します。

```
** This Dispatcher will be updated as follows when it is restarted **  
Max client buffer size: 10000 bytes
```

実行中のディスパッチャの動的属性以外の属性を Oracle SQL/Services Manager GUI ユーティリティを使用して変更すると、変更を反映するにはディスパッチャを再起動する必要があるという内容の成功ステータスが表示され、即座にディスパッチャを再起動するかどうかを尋ねられます。Oracle SQL/Services Manager GUI ユーティリティでは、ディスパッチャの再起動時に反映される、変更済の動的属性以外の属性の値が、各属性の横に旗の形のアイコンが表示されることで強調されます。

ディスパッチャを再起動すると、ディスパッチャに対するすべてのクライアント・ネットワーク接続は切断されます。したがって、ディスパッチャの変更は、そのディスパッチャの管理対象のネットワーク・ポートを使用するクライアントが少ないか、またはまったく存在しないときにスケジュールする必要があります。

ディスパッチャの削除

ディスパッチャをオンライン操作で削除するには、最初にそのディスパッチャを停止する必要があります（第6章の「SHUTDOWN DISPATCHER コマンド」および「DROP コマンド」を参照）。ディスパッチャを削除するのは、ディスパッチャが不要になった場合のみです。

2.4.3 サービスの管理

サービスを管理するには、サービスの作成方法、サービスの起動、停止および再起動方法に加え、クライアント / サーバー構成の特定の要件を満たすようにサービスの属性をカスタマイズする方法を理解する必要があります。

サービスの作成

Oracle SQL/Services のインストール・プロシージャと SYSSMANAGER:SQLSRV_CREATE71.COM コマンド・プロシージャにより、3つのサービスが作成および起動されます。これらのサービスは、Oracle SQL/Services (ODBC for Oracle Rdb) クライアントに使用される GENERIC という汎用サービス、Oracle OCI クライアントに使用される OCI_SAMPLE というデータベース・サービス、および Oracle RMU クライアントに使用される RMU_SERVICE という Oracle RMU サービスの3つです。

サーバー管理者は、状況に応じて、異なるリリースの Oracle Rdb 用に他のサービスを作成する必要があります。同様に、システム上の特定の Oracle Rdb データベース用に 1 つ以上のデータベース・サービスを作成することが可能です。サービスを作成する場合、サービスへのアクセスを許可するユーザーや、サービスを使用するクライアントをサポートするのに必要なエグゼキュータの数などを決定する必要があります。汎用サービスとデータベース・サービスの詳細は、2.6 項も参照してください。

サービスの起動、停止および再起動

AUTOSTART 属性が ON に設定されているサービスは、Oracle SQL/Services のインストール時と、サーバーの起動時に常に自動的に起動されます。必要な場合は、START SERVER AUTOSTART OFF コマンドを使用してサーバーを起動することで、この動作を無効化できます。サービスは、システムの停止時に自動的に停止します。常時クライアントで使用可能にするため、通常は作成するほとんどのサービスで AUTOSTART 属性を ON に設定します。

ただし、一部のサービスは手動で起動することもできます。たとえば、特定のデータベース用にトランザクションで再利用可能なサービスを作成し、セッションで再利用可能なサービスを使用するよりパフォーマンスが向上するかどうかを確認できます。この場合、新規サービスのテスト中は、AUTOSTART 属性を OFF に設定できます。

サービスを停止する理由の 1 つとして、サービスにより提供されるデータベースへのクライアント・アクセスを防止することがあげられます。たとえば、ディスク障害の発生後にデータベースをリストアする場合、その間はサービスを停止します。サービスの起動の失敗は、サービスを停止するもう 1 つの理由になります。失敗したサービスは、常に失敗状態のまま残されます。失敗の原因（定義内に指定された引数の値が間違っていることが多い）は、ALTER SERVICE コマンドを使用して訂正できます。サービスを停止して訂正し、START SERVICE コマンドを使用してサービスを起動するか、またはサービスが失敗状態にあるときに変更を加え、変更の完了後にサービスを停止して再起動できます。

一般的に、サービスを再起動する必要があるのは、サービス・オブジェクトの動的属性以外の属性を変更した場合のみです。この場合、変更を反映するにはサービスを再起動する必要があります。

サービスの変更

サービスの作成後、状況によっては、システムのパフォーマンスをチューニングするためにエグゼキュータ数またはサービス用のエグゼキュータ当たりのクライアント数を調整する必要があります。新規ユーザーをネットワークに追加する場合、サービスへのアクセス権をそれらのユーザーに付与する必要があります。データベースをアップグレードしてより新しいリリースの Oracle Rdb に移行する場合、サービスのエグゼキュータに使用される新規 SQL パージョンを指定するようサービスを変更する必要があります。表 2-3 に、サービスのすべての属性とそのデフォルト値をリストし、その属性を動的に変更できるかどうかを示します。表の後に、主なサービス属性の簡単な説明を記載します。

表 2-3 サービス・オブジェクト属性のデフォルト設定

属性	デフォルト設定	動的属性
AUTOSTART	ON	
DEFAULT_CONNECT_USERNAME	なし	○
REUSE_SCOPE	SESSION	
SQL_VERSION	STANDARD	
PROTOCOL	SQL/Services	
PROCESS_INITIALIZATION	なし	
ATTACH	なし	
OWNER	なし	
SCHEMA	なし	
SQL_INIT_FILE	なし	
DATABASE_AUTHORIZATION	CONNECT USERNAME	
APPLICATION_TRANSACTION_USAGE	SERIAL	○
IDLE_USER_TIMEOUT	0	○
IDLE_EXECUTOR_TIMEOUT	1800	○
MIN_EXECUTORS	0	○
MAX_EXECUTORS	1	○
CLIENTS_PER_EXECUTOR	1	○

ユーザー・コミュニティが実行するアプリケーションによりアクセスされるデータベースに対応できる程度の数のサービス・オブジェクトを作成します。詳細は、[2.6 項](#)、[2.8 項](#)および[2.9 項](#)を参照してください。

クライアントに効率的なサービスを提供するため、ユーザー・アクティビティの推移に基づいてサービスごとに MIN_EXECUTORS、MAX_EXECUTORS および IDLE_EXECUTOR_TIMEOUT 属性を設定します。詳細は、[2.11 項](#)を参照してください。

トランザクションで再利用可能なサービスをチューニングする場合、最高のパフォーマンスを確保するには、CLIENTS_PER_EXECUTOR 属性の値を調整する必要があります。

継続的に実行する必要がある別のタスクとして、サービスへのアクセス権をユーザーまたは識別子に付与することや、現在のアクセス権を変更することがあげられます。これらのタスクを実行するには、GRANT USE ON SERVICE および REVOKE USE ON SERVICE コマンドを使用します。SHOW SERVICE コマンドを使用すると、特定のサービスへのアクセス権を現在保持しているユーザーまたは識別子を確認できます。

サービス属性の変更方法の詳細は、「[ALTER SERVICE コマンド](#)」を参照してください。

実行中のサービスの動的属性をオンラインで変更すると、その変更は即座に反映されます。ただし、実行中のサービスの動的属性以外の属性をオンラインで変更する場合、変更を反映するにはサービスを再起動する必要があります。

実行中のサービスの動的属性以外の属性を SQLSRV_MANAGE ユーティリティを使用して変更すると、変更を反映するにはサービスを再起動する必要があるという内容の成功ステータスが表示されます。次に例を示します。

```
SQLSRV> ALTER SERVICE payroll SQL_INIT_FILE PAYROLL_DIR:PAYROLL.SQLINIT;
%DBS-S-ALTER_RESTART, Restart object to have altered settings take effect
```

SQLSRV_MANAGE ユーティリティでは、サービスの再起動時に反映される、変更済の動的属性以外の属性の値が表示されます。次に例を示します。

```
** This Service will be updated as follows when it is restarted **
   SQL init file:                payroll_dir:payroll.sqlinit
```

実行中のサービスの動的属性以外の属性を Oracle SQL/Services Manager GUI ユーティリティを使用して変更すると、変更を反映するにはサービスを再起動する必要があるという内容の成功ステータスが表示され、即座にサービスを再起動するかどうかを尋ねられます。Oracle SQL/Services Manager GUI ユーティリティでは、サービスの再起動時に反映される、変更済の動的属性以外の属性の値は、各属性の横に旗の形のアイコンが表示されることで強調されます。

サービスを再起動すると、サービスを使用しているアプリケーションからのすべてのクライアント・ネットワーク接続は切断されます。したがって、サービスの変更は、サービスを使用するクライアントが少ないか、またはまったく存在しないときにスケジュールする必要があります。

サービスの削除

汎用サービスを削除する必要があるのは、使用されている Oracle Rdb の特定のリリースに対応するデータベースが不要になった場合のみです。同様に、サービスが不要になった場合や、サービスを使用するユーザーが少なすぎて現在のタイプのサービスを維持する必要がなくなった場合、データベース・サービスを削除できます。どちらの場合も、オンラインでサービスを削除するには、最初にそのサービスを停止する必要があります（第 6 章の「SHUTDOWN SERVICE コマンド」および「DROP コマンド」を参照）。

2.5 ディスパッチャおよびトランスポート選択の設定

クライアントは、システムでサポートされるネットワーク・トランスポートを使用して、サーバーおよびディスパッチャと通信します。Oracle SQL/Services リリース 7.1 以上のサーバーでは、TCP/IP、DECnet、Oracle Net および NetWare (IPX/SPX) トランスポートがサポートされます。

ディスパッチャ・オブジェクトを作成する場合、ディスパッチャで 1 つ以上のトランスポートをサポートするかどうかを指定できます。1 つのディスパッチャで 1 つのトランスポートのみをサポートする場合、Oracle SQL/Services クライアントで使用される各トランスポートをサポートするために追加のディスパッチャを作成する必要があります。Oracle SQL/Services では、サーバー構成ごとに 1 つ以上のディスパッチャを使用できます。定義された各ディスパッチャは、1 つ以上の一意のネットワーク・ポート ID またはオブジェクトをリスニングする必要があります。

次の例は、Oracle Net トランスポートをサポートするディスパッチャの作成方法を示しています。

```
SQLSRV> CONNECT SERVER;
Connecting to server ...
Connected
SQLSRV> CREATE DISPATCHER sqlnet_disp NETWORK_PORT SQLNET_LISTENER_NAME 'LISTENER';
SQLSRV> START DISPATCHER sqlnet_disp;
```

Oracle Net トランスポートの使用の詳細は、『Guide to Using the Oracle SQL/Services Client API』を参照してください。

次の例は、2 つのトランスポートをサポートするディスパッチャを停止および削除し、それぞれが 1 つのトランスポートのみをサポートする別の 2 つのディスパッチャを作成する方法を示しています。最初に、削除する予定のディスパッチャによりサポートされているトランスポートを使用するクライアントが存在しないことを確認します。ディスパッチャを停止すると、そのディスパッチャを使用しているクライアントからのネットワーク接続は切断されます。ディスパッチャを使用中のクライアントが存在しない場合、ディスパッチャを停止して削除します。最後に、新規ディスパッチャを作成して起動します。

```

SQLSRV> CONNECT SERVER;
Connecting to server ...
Connected
SQLSRV> SHOW CLIENTS;

Service: SQLSRV_MANAGE

   Connect      Client
   Username    Node           State          Executor
Application
   root        127.0.0.1      RUNNING BOUND  00000ec3
SQLSRV_MANAGE
SQLSRV> SHUTDOWN DISPATCHER sqlsrv_disp;
SQLSRV> DROP DISPATCHER sqlsrv_disp;
SQLSRV> CREATE DISPATCHER sqlsrv_tcpip NETWORK_PORT TCP/IP;
SQLSRV> CREATE DISPATCHER sqlsrv_decnet NETWORK_PORT DECNET;
SQLSRV> START DISPATCHER sqlsrv_tcpip;
SQLSRV> START DISPATCHER sqlsrv_decnet;

```

注意：OCI Services for Oracle Rdb を使用するには、LISTENER.ORA ファイルに LISTENER_NAME を定義する必要があります。詳細は、『Oracle Rdb Oracle SQL/Services インストレーション・ガイド』の第4章を参照してください。

2.6 サービスおよび再利用タイプの設定

Oracle SQL/Services サーバーでは、汎用サービスとデータベース・サービスが提供されます。特に記載のないかぎり、この項の情報は Oracle SQL/Services および OCI Services for Oracle Rdb のプロトコルに適用されます。RMU サービスの詳細は、『Oracle Rdb for OpenVMS Oracle RMU リファレンス・マニュアル』の RMU バックアップ・コマンドを参照してください。

汎用サービス

汎用サービスの場合、クライアント・アプリケーションは、アクセス先のデータベースを決定できます。したがって、汎用サービスのエグゼキュータ・プロセスは、特定のデータベースに事前にアタッチされていません。クライアント・アプリケーションが汎用サービスに接続するたびに、1つ以上のデータベース・アタッチ文を発行してから、データ・アクセス操作を実行する必要があります。

汎用サービスと Oracle Rdb を組み合わせて使用することで、ローカルおよびリモートの Oracle Rdb データベースにアクセスできます。

データベース・サービス

データベース・サービスの場合、クライアント・アプリケーションは、特定のデータベース内のデータにアクセスできます。データベース・サービスのエグゼキュータ・プロセスは、単一のデータベースに事前にアタッチされています。クライアントがデータベース・サービスに接続した場合、事前にアタッチされたデータベースのデータに対するアクセスを即座に開始できます。

データベース・サービスと Oracle Rdb を組み合わせて使用することで、ローカルおよびリモートのデータベースにアクセスできますが、リモート・データベースにアクセスするサービス所有者にはデータベース認可属性を設定する必要があります（詳細は 2.7 項を参照）。

エグゼキュータ・プロセスにより、Oracle Rdb リリース 6.1 以上を使用するローカルまたはリモート・データベースに対してサービスが提供される場合、次の SQL 文は準備できません。

- ATTACH
- DECLARE DATABASE
- CREATE DATABASE
- ALTER DATABASE
- DROP DATABASE
- CONNECT
- SET CONNECT
- DISCONNECT

注意：データベース・サービスに接続されたクライアントは、事前にアタッチされたデータベースのデータにのみアクセスできます。他のデータベースのデータにはアクセスできません。

再利用タイプ

Oracle SQL/Services サーバーでは、セッション再利用属性またはトランザクション再利用属性のいずれかを持つサービスが提供されます。

セッション再利用 セッションで再利用可能なサービスのエグゼキュータは、一度に1つのクライアント・セッションに対するリクエストを処理します。ODBC 接続関数の `sqlsrv_associate` コールまたは OCI 接続を使用してクライアントがサービスに接続すると、セッションが開始され、接続はエグゼキュータ・プロセスにバインドされます。クライアントがサービスから切断されると、セッションは終了し、接続はエグゼキュータ・プロセスからバインド解除されます。セッションで再利用可能なサービスは、現在のセッションが終了するまでエグゼキュータで新しいセッションの処理が開始されないため、その名前が付けられました。セッション再利用属性は、汎用サービスまたはデータベース・サービスに適用可能です。この属性は、汎用サービスに適用できる唯一の属性です。詳細は、[2.6.1 項](#)および[2.6.2 項](#)を参照してください。

トランザクション再利用 トランザクションで再利用可能なサービスのエグゼキュータは、一度に1つのクライアントのトランザクションに対するリクエストを処理します。ただし、このエグゼキュータは、多くの同時クライアント・セッションにより共有されます。暗黙的または明示的にトランザクションを開始する SQL 文がクライアントによって発行されると、トランザクションが起動します。クライアントが `SQL COMMIT` または `ROLLBACK` 文を正常に発行した場合、あるいはトランザクションをコミットまたはロールバックするストアード・プロシージャを実行した場合、トランザクションは終了します。トランザクションで再利用可能なサービスは、現在のトランザクションが終了するまでエグゼキュータで新しいトランザクションの処理が開始されないため、その名前が付けられました。トランザクション再利用属性は、データベース・サービスにのみ適用可能です。詳細は、[2.6.3 項](#)を参照してください。

注意：トランザクションで再利用可能なデータベース・サービスは、OCI Services for Oracle Rdb でサポートされません。

表 2-4 に、各サービスに関連する属性と設定を示します。

表 2-4 Oracle SQL/Services サービスの属性

サービス定義属性	サービス		
	セッションで再利用可能 (汎用)	セッションで再利用可能 (データベース)	トランザクションで再利用可能 (データベース)
事前起動	○	○	○
事前アタッチ	×	○	○
ATTACH 文の実行	○	×	×
複数アタッチの実行	○	×	×
エグゼキュータごとのクライアント数	1	1	>1

2.6.1 セッションで再利用可能な汎用サービス

セッションで再利用可能な汎用サービスのエグゼキュータは、一度に 1 つのクライアント・セッションに対するリクエストを処理します。このエグゼキュータは、特定のデータベースに事前にアタッチされていません。

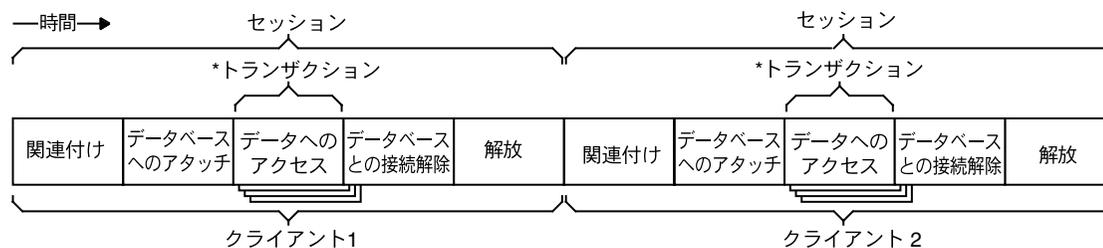
次のいずれかの条件が成立する場合、汎用サービスを使用します。

- 使用するデータベースをクライアント・アプリケーションで決定する場合
- アクセス頻度の低い多くのデータベースを含むノードがあり、個別のデータベース・サービスを用意するのが実用的ではない場合
- 汎用サービスに接続して SQL ATTACH 文を実行することによってのみ使用するデータベースを選択できるレガシー・アプリケーションまたはサード・パーティ・アプリケーションがある場合
- 使用するデータベースをアプリケーション開発者が完全に制御する必要がある場合

汎用サービスのエグゼキュータ・プロセスは、必要に応じて事前起動または起動できます。汎用サービスのエグゼキュータ・プロセスを事前に十分な数だけ起動しておくことで、クライアントでは、サービスへの接続時にプロセス起動の遅延を回避できます。汎用サービスを使用すると、クライアントでは、必要なデータベースにアタッチするためのオーバーヘッドが常に発生します。

図 2-1 に、汎用サービスの動作を示します。クライアントが汎用サービスに接続すると、クライアント接続がエグゼキュータ・プロセスに割り当てられ、バインドされます。バインド後、クライアントは 1 つ以上のデータベースにアタッチされ、データにアクセスします。最終的には、アタッチされたデータベースとの接続が解除されます。クライアントが接続を解放すると、エグゼキュータ・プロセスは、別のクライアントにより使用可能となります。

図 2-1 Oracle SQL/Services のセッションで再利用可能な汎用サービス



* 単数または複数可

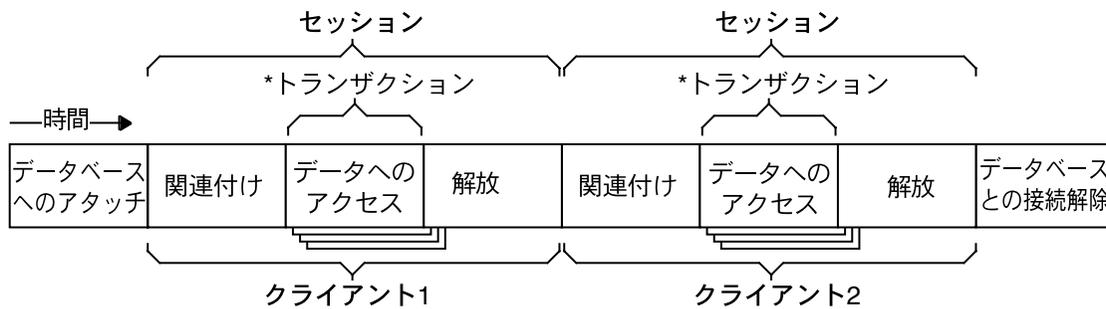
2.6.2 セッションで再利用可能なデータベース・サービス

セッションで再利用可能なデータベース・サービスのエグゼキュータは、一度に1つのクライアント・セッションに対するリクエストを処理します。このエグゼキュータは、単一のデータベースに事前にアタッチされています。

トランザクションの継続期間が長い、または不明である特定のデータベースにアクセスするためのサービスをクライアントに提供する場合、セッションで再利用可能なデータベース・サービスを使用します。セッションで再利用可能なデータベース・サービスのエグゼキュータ・プロセスは、必要に応じて事前起動または起動できます。セッションで再利用可能なデータベース・サービスのエグゼキュータ・プロセスを事前に十分な数だけ起動しておくことで、クライアントでは、サービスへの接続時にプロセス起動およびデータベース・アタッチの遅延を回避できます。

図 2-2 に、セッションで再利用可能なデータベース・サービスの動作を示します。クライアントがセッションで再利用可能なデータベース・サービスに接続すると、クライアント接続がエグゼキュータ・プロセスに割り当てられ、バインドされます。エグゼキュータはデータベースに事前にアタッチされているため、バインド後にクライアントはデータベースのデータに即座にアクセスできます。クライアントが接続を解放すると、エグゼキュータ・プロセスはクライアント接続とのバインドを解除します。その後、エグゼキュータ・プロセスは、別のクライアントにより使用可能となります。

図 2-2 Oracle SQL/Services のセッションで再利用可能なデータベース・サービス



* 単数または複数可

2.6.3 トランザクションで再利用可能なデータベース・サービス

注意： トランザクションで再利用可能なデータベース・サービスは、OCI Services for Oracle Rdb でサポートされません。

トランザクションで再利用可能なデータベース・サービスのエグゼキュータは、単一のデータベースに事前にアタッチされ、一度に1つのクライアントのトランザクションに対するリクエストを処理します。このエグゼキュータは、多くの同時クライアント・セッションにより共有されます。特定のエグゼキュータ・プロセスに割り当てられた特定のクライアント接続は、クライアント・アプリケーションの接続がサービスから切断されるまで、そのエグゼキュータ・プロセスに割り当てられたままとなります。

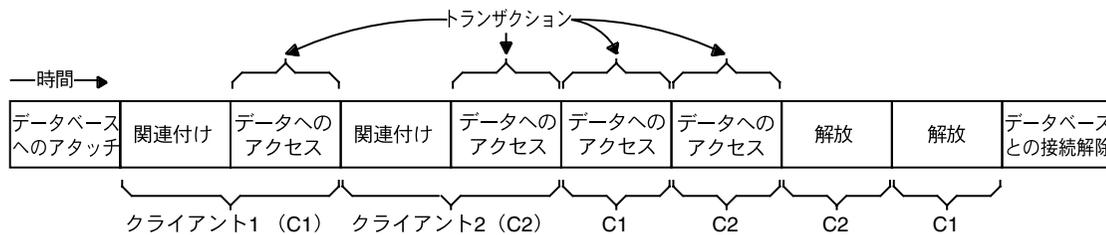
データベース・ワークロードが比較的短い期間の既知のトランザクションで構成される特定のデータベースにアクセスするためのサービスをクライアントに提供する場合、トランザクションで再利用可能なデータベース・サービスを使用します。トランザクションで再利用可能なデータベース・サービスを適切な状況で使用すると、システム・リソースの使用量とデータベースの競合を減らすことで、パフォーマンスを向上できます。その理由は、複数のクライアントで単一のエグゼキュータ・プロセスを共有することにより、システムで必要とされるエグゼキュータ・プロセスの合計数を削減できるためです。

トランザクションで再利用可能なデータベース・サービスは、トランザクションの継続期間が長い、または変化する場合にはあまり効果がありません。このような状況で、トランザクションで再利用可能なデータベース・サービスを使用すると、ユーザーはレスポンス時間を予測できないことが多くなります。これは、継続期間の長いトランザクションを実行するクライアントがエグゼキュータ・プロセスを独占してしまい、他のユーザーが使用できなくなるためです。

トランザクションで再利用可能なデータベース・サービスのエグゼキュータ・プロセスは、常に事前に起動されているため、サーバーではサービス用に起動されたエグゼキュータ・プロセスのセット全体にクライアント接続を均等に分散できます。トランザクションで再利用可能な単一のエグゼキュータ・プロセスを複数のクライアント接続で共有するため、セッションで再利用可能なエグゼキュータを使用する場合と同じ数のエグゼキュータ・プロセスを事前に起動しておく必要はありません。継続期間の非常に短いトランザクションでワークロードが構成される場合、エグゼキュータ当たりのクライアントの数が多くても、必要なエグゼキュータ・プロセスは少なく済みます。トランザクションの継続期間が増加するにつれ、エグゼキュータ当たりのクライアントの数が減少し、多くのエグゼキュータ・プロセスが必要になります。

図 2-3 に、トランザクションで再利用可能なデータベース・サービスの動作を示します。クライアントがトランザクションで再利用可能なデータベース・サービスに接続すると、クライアント接続がエグゼキュータ・プロセスに割り当てられます。ただし、クライアント接続は、初期接続がエグゼキュータにより処理された後、エグゼキュータ・プロセスにバインドされたままになりません。複数のクライアント接続が単一のエグゼキュータ・プロセスに割り当てられます。トランザクションが開始されると、クライアント接続はエグゼキュータ・プロセスにバインドされ、その時点でクライアントはデータベースのデータにアクセスできます。クライアントが SQL COMMIT または ROLLBACK 文を使用するか、トランザクションをコミットまたはロールバックするストアド・プロシージャを実行することでトランザクションを終了した場合、エグゼキュータ・プロセスはクライアント接続とのバインドを解除します。その後、エグゼキュータ・プロセスは、別のクライアント接続により使用可能となります。クライアントが接続を解放すると、クライアント接続のエグゼキュータ・プロセスへの割当ては解除されます。

図 2-3 Oracle SQL/Services のトランザクションで再利用可能なデータベース・サービス



2.6.4 セッションとトランザクションで再利用可能なデータベース・サービスの使用条件の比較

注意：この項は、OCI Services for Oracle Rdb には適用されません。トランザクションで再利用可能なデータベース・サービスは、このプロトコルでは使用できないためです。

表 2-5 に、セッションで再利用可能なデータベース・サービスとトランザクションで再利用可能なデータベース・サービスのどちらを使用するかを決定する際に検討する必要のある条件を示します。

表 2-5 セッションとトランザクションで再利用可能なデータベース・サービスの使用条件の比較

属性	データベース・サービス	
	セッションで再利用可能	トランザクションで再利用可能
クライアント・トランザクションの条件	継続期間が長い場合 長さが不明の場合	継続期間が短い場合 長さが既知の場合
サービス使用頻度の条件	頻度が低い場合 エグゼキュータ数の設定： Min=0 Max= 大きな値	エグゼキュータ数の設定： Min=Max (必須)
	頻度が高い場合 エグゼキュータ数の設定： Min=Max	エグゼキュータ数の設定： Min=Max (必須)
エグゼキュータ当たりのクライアント数の条件	1 (必須)	>1 継続期間の短いトランザクションの場合、大きな値に設定 継続期間の長いトランザクションの場合、小さな値に設定

2.7 データベース・アクセス認可の設定

OCI Services for Oracle Rdb では、データベース・アクセス認可は接続ユーザー名である必要があります。

SQL/Services プロトコルでは、クライアントに提供するサービスのタイプと、データベース・アクセス認可を接続ユーザー名とサービス所有者のどちらに設定するかを理解および決定する際に、次のガイドラインが役立ちます。

汎用サービス

データベース・アクセス認可を接続ユーザー名に設定する場合 汎用サービスを使用するクライアントでは、クライアント提供のユーザー名、DECnet プロキシ・ユーザー名またはデフォルトの接続ユーザー名を使用してクライアント・アプリケーションでデータベースにアタッチおよびアクセスする場合、データベース・アクセス認可を接続ユーザー名に設定します。データベース・アクセス認可が接続ユーザー名に設定されている場合、データベースへのクライアント・アクセスは、基礎となるデータベースのセキュリティ・メカニズムを使用して個々のユーザーまたはユーザー・グループに付与された使用権限に基づきます。

例 2-1 は、GENERIC という汎用サービスを作成する方法を示しています。GENERIC は、サービス名が指定されない場合に Oracle SQL/Services または Oracle ODBC Driver for Oracle Rdb クライアントによりデフォルトで使用されるサービス名です。この汎用サービスでは、データベース・アクセス認可が接続ユーザー名に設定され、すべてのユーザーにアクセス権が付与されます。また、エグゼキュータ・プロセスの最小数は 1、最大数は 20 です。

例 2-1 データベース・アクセス認可が接続ユーザー名に設定されたデフォルトの汎用サービス

```
SQLSRV> CREATE SERVICE GENERIC
_SQLSRV>      OWNER 'SQLSRV$DEFLT'
_SQLSRV>      DATABASE AUTHORIZATION CONNECT USERNAME
_SQLSRV>      MIN_EXECUTORS 1
_SQLSRV>      MAX_EXECUTORS 20;
SQLSRV> GRANT USE ON SERVICE GENERIC TO PUBLIC;
SQLSRV> START SERVICE GENERIC;
```

データベース・アクセス認可をサービス所有者に設定する場合 汎用サービスを使用するクライアントでは、単一の固定ユーザー名またはサービス所有者ユーザー名を使用してクライアント・アプリケーションでデータベースにアタッチおよびアクセスする場合にのみ、データベース・アクセス認可をサービス所有者に設定します。GRANT USE ON SERVICE コマンドを使用すると、このようなサービスにアクセス可能なユーザーを制限できます。

注意： データベース・アクセス認可を汎用サービスのサービス所有者に設定する場合、セキュア・データまたは機密データ（通常であれば権限のないユーザーによるアクセスから保護されるデータ）を含むデータベースに対するアクセス権を、そのサービス所有者ユーザー名が保持していないことを確認してください。

通常、データベース認可を汎用サービスのサービス所有者に設定することはありません。

例 2-2 は、データベース・アクセス認可がサービス所有者に設定された、テスト目的で使用可能な汎用サービスを作成する方法を示しています。サービスを使用するための権限は、サービス所有者ユーザー名のアカウントに加え、2つの開発者アカウントにのみ付与されます。

例 2-2 データベース・アクセス認可がサービス所有者に設定された汎用サービス

```
SQLSRV> CREATE SERVICE GEN_DEVEL OWNER 'noprivs'
_SQLSRV>      DATABASE AUTHORIZATION SERVICE OWNER
_SQLSRV>      MIN_EXECUTORS 0
_SQLSRV>      MAX_EXECUTORS 5;
SQLSRV> GRANT USE ON SERVICE GEN_DEVEL TO 'develop', 'test';
SQLSRV> START SERVICE GEN_DEVEL;
```

データベース・サービス

データベース・アクセス認可を接続ユーザー名に設定する場合 データベース・サービスを使用するクライアントでは、クライアント提供のユーザー名、DECnet プロキシ・ユーザー名またはデフォルトの接続ユーザー名を使用してクライアントでデータベースにアクセスする場合、データベース・アクセス認可を接続ユーザー名に設定します。データベース・アクセス認可が接続ユーザー名に設定されている場合、データベースへのクライアント・アクセスは、基礎となるデータベースのセキュリティ・メカニズムを使用して個々のユーザーまたはユーザー・グループに付与された使用権限に基づきます。

例 2-3 は、企業のポリシーおよび手続きデータベースにアクセスするためのデータベース・サービスを作成する方法を示しています（データベースへのアクセスはクライアントのユーザー名に基づきます）。サービスへのアクセス権はすべてのユーザーに付与されますが、データベースのデータへのアクセスは、基礎となるデータベースのセキュリティ・メカニズムに基づきます。不明なユーザーは、デフォルトの接続ユーザー名 `readpp` に基づいてサービスの使用を認可されます。`readpp` は、データベースのデータに対する読取り専用アクセス権を保持します。サービスは、エグゼキュータ・プロセスの起動時にデータベースにアタッチするために使用される `ppdb` アカウントにより所有されます。

例 2-3 アクセス認可が接続ユーザー名に設定された、セッションで再利用可能なデータベース・サービス

```
SQLSRV> CREATE SERVICE P_AND_P
_SQLSRV>          ATTACH 'FILENAME pp_disk:[pp]pp_database'
_SQLSRV>          OWNER ppdb
_SQLSRV>          DATABASE_AUTHORIZATION CONNECT USERNAME
_SQLSRV>          DEFAULT_CONNECT_USERNAME readpp
_SQLSRV>          MIN_EXECUTORS 0
_SQLSRV>          MAX_EXECUTORS 10;
SQLSRV> GRANT USE ON SERVICE P_AND_P TO PUBLIC;
SQLSRV> START SERVICE P_AND_P;
```

データベース・アクセス認可をサービス所有者に設定する場合 データベース・サービスを使用するクライアントでは、サービス所有者ユーザー名を使用してクライアント・アプリケーションでデータベースにアクセスする場合、データベース・アクセス認可をサービス所有者に設定します。基礎となるデータベースのセキュリティ・メカニズムを使用してデータベース内の特定のデータおよび特定のデータベース操作に対するアクセス権を単一のユーザー名に付与する場合、この方法を使用します。`GRANT USE ON SERVICE` コマンドを使用すると、ユーザー名の制限されたセットにサービスの使用権限を付与できます。

例 2-4 は、企業の受注データベースにアクセスするためのデータベース・サービスを作成する方法を示しています（データベースへのアクセスはサービス所有者ユーザー名 `ordent` に基づきます）。

サービスへのアクセス権は、サービス所有者と `SYSPRV` 権限を保持する権限付きユーザーに加え、`ordent1`、`ordent2`、`ordent3` および `ordmgr` ユーザーにのみ付与されます。データベース名 `OE_DISK:[OE]OE_DATABASE` は、論理名 `oe_database` として定義されます。そのため、必要な場合は、サービス定義を変更することなくデータベースを物理的に移動できます。

データベースのトランザクション・ワークロード特性により、サービスはトランザクションで再利用可能です。また、分散された最大 100 のユーザーを 5 つのエグゼキュータ・プロセスでサポートし、プロセス当たり最大 20 のユーザーを処理できます。

例 2-4 アクセス認可がサービス所有者に設定された、トランザクションで再利用可能なデータベース・サービス

```

SQLSRV> CREATE SERVICE ORD_ENT REUSE SCOPE IS TRANSACTION
SQLSRV> ATTACH 'FILENAME OE_DATABASE'
SQLSRV> OWNER ordent
SQLSRV> DATABASE AUTHORIZATION SERVICE OWNER
SQLSRV> MIN_EXECUTORS 5
SQLSRV> MAX_EXECUTORS 5
SQLSRV> CLIENTS_PER_EXECUTOR 20;
SQLSRV> GRANT USE ON SERVICE ORD_ENT TO ordent1,
SQLSRV> ordent2,
SQLSRV> ordent3,
SQLSRV> ordmgr;
SQLSRV> START SERVICE ORD_ENT;

```

2.7.1 SQL/Services プロトコル用のデフォルトの接続ユーザー名の指定

次のガイドラインは、汎用サービスまたはデータベース・サービスを通じてシステム上のデータベースにアクセスする不明なユーザーを認可するために、デフォルトの接続ユーザー名を指定するかどうかを決定する際に役立ちます。

汎用サービスの使用

不明なユーザーがシステム上のデータベースにアクセスすることを許可する必要がある場合にのみ、汎用サービスでデフォルトの接続ユーザー名を指定します。開発環境で、テストおよびデバッグに使用されるデータベースへの簡易アクセスを許可する場合にこの方法を選択できます。

注意： デフォルトの接続ユーザー名を指定して不明なユーザーに汎用サービスの使用権限を付与する場合、セキュア・データまたは機密データを含むデータベースがデータベース・レベルで適切なアクセス制限により保護されていることを確認してください。

通常、デフォルトの接続ユーザー名を指定して不明なユーザーに汎用サービスの使用権限を付与することはありません。

データベース・サービスの使用

ユーザー名とパスワードを要求せずに特定のデータベースのデータに対するアクセスを許可する場合、デフォルトの接続ユーザー名を指定して不明なユーザーにデータベース・サービスの使用権限を付与します。たとえば、接続ユーザー名に設定されたデータベース・アクセス認可とこのメカニズムを組み合わせることで、データベースに含まれる機密データ以外の公開アクセス・データに対するアクセス権を付与できます（例 2-3 を参照）。

2.7.2 サービスへのアクセス権の付与または制限

次のガイドラインは、サービスへのアクセス権をすべてのユーザーに付与するか、サービスへのアクセス権をユーザーの指定されたリストに制限するかを決定する際に役立ちます。

サービスへのアクセス権をすべてのユーザーに付与する場合

- データベース・アクセス認可が接続ユーザー名に設定された汎用サービスまたはデータベース・サービスが存在し、個々のデータベースの基礎となるデータベース・セキュリティに基づいてシステム上のデータベースのデータにアクセスする最大限柔軟な方法をすべてのユーザーに提供する場合。
- データベース・アクセス認可がサービス所有者に設定されたデータベース・サービスが存在するが、サービス所有者ユーザー名を使用したデータベースへのアクセスは、すべてのユーザーが使用できる機密データ以外の公開アクセス・データに制限する場合。

サービスへのアクセス権をユーザーの指定されたリストに制限する場合

- データベース・アクセス認可がサービス所有者に設定された汎用サービスが存在し、固定ユーザー名を使用してデータベースのセットにアクセスする場合。
- データベース・アクセス認可がサービス所有者に設定されたデータベース・サービスが存在し、基礎となるデータベースのセキュリティ・メカニズムを使用して単一のユーザー名にデータベースのデータに対するアクセス権を付与し、次に Oracle SQL/Services のセキュリティ・メカニズムを使用してそのデータへのアクセスを制御する場合。

注意： Oracle SQL/Services を使用してサービスへのアクセス権をユーザー名の指定されたリストに制限しても、他のユーザーがシステムにログインし、それらのサービスにより提供される同じデータベースに対して対話型 SQL などのツールを使用してアクセスすることを防止することはできません。サービスへのアクセス権をユーザー名の指定されたリストに制限しても、データベースのセキュア・データや機密データは、基礎となるデータベースのセキュリティ・メカニズムを使用して従来どおり保護する必要があります。

2.7.3 任意のデータ・アクセス権または事前定義されたデータ・アクセス権の提供

次のガイドラインは、任意のデータ・アクセス権と事前定義されたデータ・アクセス権のどちらを提供するかを決定する際に役立ちます。

任意のデータ・アクセス権

基礎となるデータベースのセキュリティ・メカニズムを単独で使用するか、Oracle SQL/Services のセキュリティ・メカニズムと組み合わせることで、ユーザーがアクセスできる表や、それらの表に対して実行できる操作を制限できます。ただし、データ・アクセス権が一度付与されると、ユーザーは、付与されたアクセス権に基づいてそのデータに対して任意の SQL 文を実行できます。たとえば、ユーザーは表に対して INSERT アクセスを行うことで、希望するデータをその表に挿入できます。状況によっては、データベースのデータに対する任意アクセスを許可することは、不適切な場合があります。

事前定義されたデータ・アクセス権

状況によっては、ユーザーのデータ操作権限を、事前定義された操作のセットに制限する方が適切です。この作業は、定義者の権限ストア・プロシージャのセットを作成することにより、データベース設定の一環として行います。これらのプロシージャにより、1つ以上の表のデータに対する必要なアクセス権のすべてが提供されます。表へのアクセス権をストア・プロシージャの定義者のユーザー名に制限することで、他のすべてのユーザーからのアクセスを防止できます。次に、基礎となるデータベースのセキュリティ・メカニズムを単独で使用するか、Oracle SQL/Services のセキュリティ・メカニズムと組み合わせることで、ストア・プロシージャへのアクセス権を付与します。

2.8 サーバーのセキュリティの設定

Oracle SQL/Services を基礎となるデータベース・エンジンと組み合わせることで、様々なセキュリティ・メカニズムを提供できます。これらのセキュリティ・メカニズムを使用して、ユーザーにアクセスを許可するサービスやデータを制御できます。Oracle SQL/Services サーバー環境は、セキュリティのチェックが行われる 3 つの層に分割されます。各層は次のとおりです。

- **第 1 層: クライアント識別および認証**

最初に、Oracle SQL/Services サーバーでは、サーバーへのアクセスをリクエストしているユーザーの識別と認証を行います。このチェックは、クライアントが最初にサーバーに接続したときに行われます。

- **第 2 層: サービス・アクセス認可**

次に、Oracle SQL/Services サーバーでは、特定のサービスへのアクセスをリクエストしている各ユーザーが、そのサービスの使用を認可されているかどうかをチェックします。

- **第 3 層: データベースおよびデータ・アクセス認可**

最後に、基礎となるデータベース・エンジンが、エグゼキュータ・プロセスにより作成された各データベース・アクセス・リクエストをチェックします。

これらの各セキュリティ層について、後続の各項で説明します。

2.8.1 クライアント識別および認証

サーバー・セキュリティの第 1 層は、クライアント識別および認証です。このチェックは、クライアント・アプリケーションが最初にサーバーに接続したときに行われます。第 1 層の処理が正常に完了したときの結果は、後続の層の認可チェックで使用される接続ユーザー名です。

ユーザー名とパスワードの検証は、次のいずれかの方法で行われます。

- **OCI Services for Oracle Rdb**

OCI Services for Oracle Rdb の場合、クライアントがデータベースの `USER$` 表に存在するユーザー名を提供します。`USER$` 表には、暗号化されたパスワードも含まれます。パスワードは、パスワードを復号化するための鍵を返すクライアントに返されます。その後、ユーザー名とパスワードは、認可のために Oracle SQL/Services に渡されます。OCI サービスに対するアクセスは、接続ユーザー名別に行われる必要があります。

- **Oracle SQL/Services**

Oracle SQL/Services では、サービスへのアクセス時にユーザーがユーザー名とパスワードを提供します。クライアントが Oracle SQL/Services サーバーに接続すると、ユーザーのアカウントがシステムに存在し、パスワードが有効であることがサーバーによって確認されます。認証の成功後は、クライアント提供のユーザー名が接続ユーザー名として使用されます。ユーザー名とパスワードのチェックに失敗した場合、またはパスワードの有効期限が切れている場合、接続は拒否されてクライアントにエラー・メッセージが返されます。ユーザーがユーザー名とパスワードを提供しない場合、サーバーは接続のネットワーク・トランスポートをチェックします。

クライアントが DECnet トランスポートを選択した場合、サーバーは、クライアントのノード名またはユーザー名（あるいはその両方）に対応するプロキシが存在するかどうかをチェックします。サーバーは、最初にクライアントの DECnet ノード名と DECnet ユーザー名を検索します（Oracle SQL/Services のプロキシ・ファイル `SYS$STARTUP:SQLSRV$PROXY.DAT` に存在する場合）。一致する項目が検出されると、ローカル・プロキシ・ユーザー名が接続ユーザー名として使用されます。一致する項目が検出されないが、クライアントがサーバーと同じノード上に存在する場合、クライアント・プロセスのユーザー名が接続ユーザー名として使用されます。

システム管理クライアントにかぎり、サーバーは、ユーザーが次の操作を実行した場合にクライアント・プロセスのユーザー名を接続ユーザー名として使用します。

- TCP/IP トランスポートを選択した場合
- サーバー・ノードにログインした場合
- SYSPRV または BYPASS 権限を保持している場合

システム管理者は、デフォルトの接続ユーザー名を指定することで、ユーザー名とパスワードを要求せずにサービスへのアクセスを許可できます。ユーザーがユーザー名とパスワードを提供せず、デフォルトの接続ユーザー名も指定されていない場合（不明なユーザーにはサービスへのアクセス権が付与されない場合）、接続は拒否されてクライアントにエラー・メッセージが返されます。クライアントがユーザー名とパスワードを提供せず、デフォルトの接続ユーザー名が指定されている場合（不明なユーザーにはサービスへのアクセス権が付与される場合）、接続ユーザー名はデフォルトの接続ユーザー名に設定されます。クライアントがユーザー名を提供する場合、デフォルトの接続ユーザー名が指定されているかどうかにかかわらず、その提供されたユーザー名が接続ユーザー名として使用されます。

システム管理者がサーバーのシステム管理サービスに接続すると、サーバーは、クライアントがサービスに接続した場合と同じようにユーザー名とパスワードのチェックを行います。ユーザー名とパスワードのチェックに失敗すると、接続は拒否され、システム管理アプリケーションにエラー・メッセージが返されます。システム管理サービスにデフォルトの接続ユーザー名を指定することはできないため、不明なユーザーによるシステム管理サービスへのアクセスは認可されません。

2.8.2 サービス・アクセス認可

サーバー・セキュリティの第2層では、選択されたサービスへのアクセス権がユーザーに付与されているかどうかを検証します。

各サービスには、そのサービスへのアクセス権が付与されたユーザー名と識別子のリストがあります。新規サービスを作成した場合、そのサービスへのアクセス権は、サービス所有者にのみ付与されます。システム管理者は、サーバーにより提供されるサービスへの適切なユーザー・アクセス権を付与する必要があります。個々のユーザー名または識別子に基づいてサービスへのアクセス権を付与するか、すべてのユーザーにサービスへのアクセス権を付与できません（GRANT USE ON SERVICE GENERIC TO PUBLIC などを使用）。

ユーザーがサービスに接続すると、サーバーは、接続ユーザー名（または接続ユーザー名に保持されている識別子）にサービスを使用する権限が付与されているかどうか、またはすべてのユーザーにサービスを使用するアクセス権が付与されているかどうかをチェックします。サービスへのアクセス権がユーザーに付与されていない場合、接続は拒否されてクライアントにエラー・メッセージが返されます。

サーバーのシステム管理サービスにも、そのサービスへのアクセス権が付与された（つまり、サーバーの管理権限が付与された）ユーザー名と識別子のリストがあります。サーバーを（通常はインストールの一環として）作成したときには、SYSPRV 権限を保持する権限付きユーザーのみがサーバーを管理できます。システム管理者は、サーバーを管理するその他のユーザーにアクセス権を付与する必要があります。権限のないユーザーがサーバーのシステム管理サービスに接続しようとする、接続は拒否され、システム管理アプリケーションにエラー・メッセージが返されます。

2.8.3 データベースおよびデータ・アクセス認可

サーバー・セキュリティの最後の第3層は、エグゼキュータ・プロセスのデータベース・レベルに存在します。エグゼキュータ・プロセスがSQL文を実行すると、常に基礎となるデータベース・エンジンにより、リクエストを実行するユーザー名にその実行権限が付与されているかどうかを確認するためのセキュリティ・チェックが実行されます。Oracle SQL/Servicesでは、提供されるサービスのタイプおよびそのサービスに指定されたOracle Rdbのリリースに応じて、接続ユーザー名またはサービス所有者を使用してデータベース・リクエストを実行できます。システム管理者は、各サービスのデータベース・アクセス認可属性を接続ユーザー名またはサービス所有者に指定して、データベース・エンジンにより認可されるユーザー名を決定します。

- データベース・アクセス認可を接続ユーザー名に設定する場合

データベース・アクセス認可を接続ユーザー名に設定すると、基礎となるデータベースにより、データベース・リクエストの実行権限がクライアントに付与されているかどうかを判断するのに接続ユーザー名が使用されます。接続ユーザー名は、クライアント指定のユーザー名、DECnet プロキシ・ユーザー名またはデフォルトの接続ユーザー名です。

注意：データベース・アクセス認可が接続ユーザー名に設定されたデータベース・サービスを使用して、リモートOracle Rdb データベースへのアクセス権を付与することはできません。ただし、データベース・アクセス認可がサービス所有者に設定されたサービスを作成すれば、Oracle SQL/Services でリモートOracle Rdb データベース用のデータベース・サービスを提供できます。

- データベース・アクセス認可をサービス所有者に設定する場合

データベース・アクセス認可をサービス所有者に設定すると、基礎となるデータベースにより、データベース・リクエストの実行権限がクライアントに付与されているかどうかを判断するのにサービス所有者のユーザー名が使用されます。サービス所有者アカウントは、アタッチ先のデータベースへのSELECTアクセス権を保持している必要があります。エグゼキュータ・プロセスは、サービス所有者のユーザーIDからOpenVMSの権限を継承します。NETMBXおよびTMPMBXは、AUTHORIZED権限として必要なすべての権限です。Oracle SQL/Servicesでは、DEFAULT権限リストのかわりとなる権限リストを使用します。

2.8.4 SQL/Services プロトコルでのサーバー・セキュリティ層の連携

図 2-4 は、クライアントがシステムにログインする 3 つの接続例で、3 つのサーバー・セキュリティ層がどのように連携するかを示しています。各例では、サービスごとのサービス定義に基づいて、クライアント識別および認証、サービス・アクセス認可、結果のデータベースおよびデータ・アクセス認可を示します。

図 2-4 Oracle SQL/Services サーバー・セキュリティ

	接続1	接続2	接続3
サービス定義	サービス名=X 所有者=fred アタッチ=payroll_db データベース認可= サービス所有者 デフォルトの接続 ユーザー名引数なし freda、nedに使用権限を付与	サービス名=Y 所有者=bert アタッチ引数なし データベース認可= 接続ユーザー名 デフォルトの接続 ユーザー名引数なし PUBLICに使用権限を付与	サービス名=Z 所有者=joe アタッチ=account_db データベース認可= 接続ユーザー名 デフォルトの接続 ユーザー名=jane janet、janeに使用権限を付与
サーバーへのクライアント接続	ユーザー名=ned パスワード=pwned サービス名=X	ユーザー名=holly パスワード=pwholly サービス名=Y	ユーザー名なし パスワードなし サービス名=Z
第1層: クライアント識別 および認証	ユーザー名=ned パスワードpwnedを使用して認証 接続ユーザー名をnedに設定	ユーザー名=holly パスワードpwhollyを使用して認証 接続ユーザー名をhollyに設定	デフォルトの 接続ユーザー名=jane 接続ユーザー名をjaneに設定
第2層: サービス・アクセス認可	nedを使用して サービスXへのアクセス権を 接続ユーザー名nedに付与	PUBLICを使用して サービスYへのアクセス権を 接続ユーザー名hollyに付与	janeを使用して サービスZへのアクセス権を 接続ユーザー名janeに付与
第3層: データベース およびデータ・ アクセス認可	サービス所有者ユーザー名 fredを使用して データベースにアタッチ サービス所有者ユーザー名 fredを使用して データにアクセス	接続ユーザー名 hollyを使用して データベースにアタッチ 接続ユーザー名 hollyを使用して データにアクセス	サービス所有者 ユーザー名joeを使用して データベースにアタッチ 接続ユーザー名 janeを使用して データにアクセス

第 1 の接続例

第 1 の接続例では、ユーザーはサービス X へのアクセスをリクエストします。ユーザーは、ユーザー名とパスワードを指定しているため、それらはセキュリティの第 1 層でサーバーにより認証され、接続ユーザー名は ned に設定されます。デフォルトの接続ユーザー名は指定されていないため、不明なユーザーはサービス X へのアクセスを許可されません。したがって、サービス X へのアクセスをリクエストするすべてのユーザーは、有効なユーザー名とパスワードを指定する必要があります。セキュリティの第 2 層で、接続ユーザー名 (この例では ned) にサービスへのアクセス権が付与されているかどうかサーバーによりチェックされます。ユーザー freda と ned には、SYSPRV 権限を保持する権限付きユーザーとともに、サービス X の使用権限が付与されています。そのため、ユーザー ned には、サービスへのアクセス権が付与されます。サービス X はデータベース・サービスであるため、エグゼキュータ・プロセスは、サービス所有者のユーザー名 fred を使用して payroll_db データベースにアタッチします。サービス X のデータベース・アクセス認可は、サービス所有者に設定されているため、すべてのデータベース・アタッチおよびデータ・アクセス・リクエストも、サービス所有者のユーザー名 (この例では fred) に基づいて作成されます。

第2の接続例

第2の接続例では、ユーザーはサービス Y へのアクセスをリクエストします。ユーザーは、ユーザー名とパスワードを指定しているため、それらはセキュリティの第1層でサーバーにより認証され、接続ユーザー名は **holly** に設定されます。デフォルトの接続ユーザー名は指定されていないため、不明なユーザーはサービス Y へのアクセスを許可されません。したがって、サービス Y へのアクセスをリクエストするすべてのユーザーは、有効なユーザー名とパスワードを指定する必要があります。セキュリティの第2層で、接続ユーザー名（この例では **holly**）にサービスへのアクセス権が付与されているかどうかサーバーによりチェックされます。サービス Y を使用する権限はすべてのユーザーに付与されているため、ユーザー **holly** にはサービスへのアクセス権が付与されます。サービス Y は汎用サービスであるため、エグゼキュータは特定のデータベースに事前にアタッチされていません。サービス Y のデータベース・アクセス認可は、接続ユーザー名に設定されているため、すべてのデータベース・アタッチおよびデータ・アクセス・リクエストは、接続ユーザー名（この例では **holly**）に基づいて作成されます。

第3の接続例

第3の接続例では、ユーザーはサービス Z へのアクセスをリクエストします。ユーザーは、ユーザー名とパスワードを指定していないため、リクエストされたサービスへのアクセス権が不明なユーザーに付与されているかどうかサーバーによりチェックされます。デフォルトの接続ユーザー名が指定されているため、不明なユーザーはユーザー **jane** としてサービス Z へのアクセスを許可されます。したがって、接続ユーザー名は **jane** に設定されます。セキュリティの第2層で、接続ユーザー名（この例では **jane**）にサービスへのアクセス権が付与されているかどうかサーバーによりチェックされます。ユーザー **janet** と **jane** には、SYSPRV 権限を保持する権限付きユーザーとともに、サービス Z の使用権限が付与されています。そのため、ユーザー **jane** には、サービスへのアクセス権が付与されます。サービス Z はデータベース・サービスであるため、エグゼキュータ・プロセスは、サービス所有者のユーザー名 **joe** を使用して **account_db** データベースにアタッチします。ただし、サービス Z のデータベース・アクセス認可は、接続ユーザー名に設定されているため、すべてのデータ・アクセス・リクエストは、接続ユーザー名（この例では **jane**）に基づいて作成されます。

2.9 Oracle SQL/Services のデータベース・アクセス認可モデルの理解

2.7 項で、Oracle SQL/Services により、サービス所有者ユーザー名または接続ユーザー名を使用してデータベース・アクセスを認可できるということを説明しました。また、データベース・リクエストや外部ファンクションが実行される環境にこれらのモデルがどのように影響するかについても説明しました。この項では、Oracle SQL/Services による接続ユーザー別およびサービス所有者別のデータベース認可の実装方法について詳細に説明します。

2.9.1 Oracle Rdb データベースへのアクセス

Oracle SQL/Services による接続ユーザー名別およびサービス所有者別のデータベース認可の実装方法を理解するには、最初に、Oracle SQL/Services 環境の Oracle Rdb データベースへのアクセスに関連する次の 4 つのユーザー名を理解する必要があります。

- オペレーティング・システムのプロセス・ユーザー名
- Oracle Rdb のシステム・ユーザー名
- Oracle Rdb のセッション・ユーザー名
- Oracle Rdb の現行ユーザー名

注意： OCI サービスに対するアクセスは、接続ユーザー名別に行われる必要があります。

次に、4 つのユーザー名について説明します。

2.9.1.1 オペレーティング・システムのプロセス・ユーザー名

プロセス・ユーザー名は、Oracle SQL/Services のエグゼキュータ・プロセスがローカル・アタッチを実行する際に使用されるユーザー名です。または、リモート・アタッチにおける Oracle Rdb リモート・サーバー・プロセスのユーザー名です。

プロセス・ユーザー名は、ローカル・アタッチでは SERVICE OWNER サービス属性に基づいて設定されますが、リモート・アタッチでは ATTACH 文とリモート Oracle Rdb サーバー・ノードの構成に基づいて設定されます。プロセス・ユーザー名には、多くのプロセス属性が関連付けられています。これらの属性は次のとおりです。

- UIC
- 権限
- 権限リスト
- アカウント名
- デフォルト・ディレクトリ
- 次のような論理名
 - SYS\$DISK
 - SYS\$LOGIN_DEVICE
 - SYS\$LOGIN
 - SYS\$SCRATCH
 - LNM\$GROUP (グループ論理名表用)

2.9.1.2 Oracle Rdb のシステム・ユーザー名

エグゼキュータ・プロセスのアタッチされた各データベースには、システム・ユーザー名の値が含まれます。Oracle Rdb のシステム・ユーザー名は、プロセスがデータベースにアタッチする権限を付与されているかどうかを判別する際に使用されます。また、Oracle Rdb のセッション・ユーザー名のデフォルト値としても機能します。

アタッチされたデータベースに対する Oracle Rdb のシステム・ユーザー名は、デフォルトでプロセス・ユーザー名に設定されますが、提供されるサービスのタイプ、サービスの属性、および使用される Oracle Rdb のリリースに応じて、データベース・サービスの SQL ATTACH 文の属性または汎用サービスにアクセスするクライアント・アプリケーションによって上書きされる場合があります。

アタッチされたデータベースに対する Oracle Rdb のシステム・ユーザー名は、データベースへのアタッチの際に確立され、アタッチの継続中は固定されたままとなります。システム・ユーザー名のデフォルト値は、Oracle Rdb リリース 6.1 以上の使用時に、SQL ATTACH 文のアタッチ文字列引数または SQL CONNECT 文の接続文字列引数にユーザー名とパスワードを指定することで上書きできます。SQL の ATTACH 文および CONNECT 文の詳細は、『Oracle Rdb SQL リファレンス・マニュアル』を参照してください。

汎用サービスを提供するエグゼキュータ・プロセスのアタッチされたデータベースの数は、クライアント・アプリケーションによって決定されます。アタッチされた異なるデータベースでは、異なるシステム・ユーザー名を使用できます。

データベース・サービスを提供するエグゼキュータ・プロセスでは、アタッチされたデータベースを 1 つだけ使用します。

2.9.1.3 Oracle Rdb のセッション・ユーザー名

すべてのデータベース・リクエストは、SQL 接続のコンテキスト内で実行されます。エグゼキュータ・プロセスの各 SQL 接続には、セッション・ユーザー名の値が含まれます。SQL 接続のセッション・ユーザー名は、デフォルトで Oracle Rdb のシステム・ユーザー名に設定されますが、提供されるサービスのタイプ、サービスの属性、および使用される Oracle Rdb のリリースに応じて、Oracle SQL/Services またはクライアント・アプリケーションによって上書きされる場合があります。

SQL 接続のセッション・ユーザー名は、SQL 接続の確立時に決定され、SQL 接続の継続中は固定されたままとなります。セッション・ユーザー名のデフォルト値は、Oracle Rdb リリース 6.1 以上の汎用サービスの使用時に、ユーザー名およびパスワードを引数として SQL CONNECT 文に指定することで上書きできます。SQL CONNECT 文の詳細は、『Oracle Rdb SQL リファレンス・マニュアル』を参照してください。

汎用サービスを提供するエグゼキュータ・プロセスの SQL 接続の数は、クライアント・アプリケーションによって決定されます。異なる SQL 接続では、異なるセッション・ユーザー名を使用できます。汎用サービスを提供するエグゼキュータ・プロセスの SQL 接続では、1 つ以上のデータベース・アタッチを参照できます。

データベース・サービスを提供するエグゼキュータ・プロセスの SQL 接続の数は、使用中の Oracle Rdb のリリースと、サービス再利用属性によって決定されます。再利用属性の詳細は、[2.6 項](#)を参照してください。

データベース・サービスの ATTACH 文は、常にデフォルトの SQL 接続のコンテキスト内で実行されます。データベース・サービスを使用したデータベースへのアタッチに、SQL CONNECT 文を使用することはできません。デフォルト接続のセッション・ユーザー名は、デフォルトでシステム・ユーザー名に設定されます。サービスに SQL 初期化ファイルが指定されている場合、それに含まれる文は、SQL ATTACH 文の後にあるデフォルトの SQL 接続のコンテキスト内で実行されます。

Oracle Rdb リリース 6.1 以上では、サービスに接続するクライアント・アプリケーションごとに新規 SQL 接続が作成されます。サービスがサービス所有者によるデータベース認可で定義されている場合、クライアント・アプリケーションの各 SQL 接続のセッション・ユーザー名は、デフォルトでシステム・ユーザー名に設定されます。サービスが接続ユーザーによるデータベース認可で定義されている場合、クライアント・アプリケーションの各 SQL 接続は、個々のクライアント接続ごとの接続ユーザー名を使用して作成されます。クライアント・アプリケーションの接続がサービスから切断されると、クライアント・アプリケーションの SQL 接続は削

除されます。セッションで再利用可能なデータベース・サービスの場合、エグゼキュータごとに最大で1つのクライアント・アプリケーション SQL 接続が存在します。トランザクションで再利用可能なデータベース・サービスの場合、同時クライアント接続ごとに1つのクライアント・アプリケーション SQL 接続が存在します。

2.9.1.4 Oracle Rdb の現行ユーザー名

現行ユーザー名は、定義者の権限ストアド・プロシージャの実行時を除き、常にセッション・ユーザー名の値に設定されます。定義者の権限ストアド・プロシージャの実行時には、現行ユーザー名は定義者のユーザー名に設定されます。

データベース・リクエストが開始されると、Oracle Rdb では、リクエストを発行したプロセスにそのリクエストの実行権限が付与されているかどうかを判別する必要があります。このチェックを実行するため、Oracle Rdb は、まずデータベースにアクセスするプロセスのシステム権限と現行ユーザー名のデータベース権限をマージします。ローカル・アタッチの場合、データベースにアクセスするプロセスは、Oracle SQL/Services のエグゼキュータ・プロセスです。リモート・アタッチの場合、データベースにアクセスするプロセスは、Oracle Rdb のサーバー・プロセスです。

オペレーティング・システムのプロセス権限マスクは、エグゼキュータ・プロセスのシステム権限として使用されます。

Oracle Rdb は、権限のマージ後に、その権限の組合せがリクエストを実行するのに十分であるかどうかを判別します。Oracle Rdb ではこのように権限が結合されるため、データベース・サービスのサービス所有者ユーザー名は慎重に選択する必要があります。詳細は、2.10 項を参照してください。

例として、SYSTEM というサービス所有者ユーザー名が定義されており、データベース認可が接続ユーザー名に設定されている PAYROLL というデータベース・サービスについて検討します。ユーザー SMITH は、通常は権限がないため給与データベースの EMPLOYEE_PAY という表を更新できません。ただし、ユーザー SMITH が PAYROLL サービスを使用して給与データベースにアクセスした場合、ユーザー名 SMITH のデータベース権限が SYSPRV および BYPASS を含む SYSTEM ユーザー名のシステム権限と結合された時点で、このユーザーは EMPLOYEE_PAY 表とデータベース内の他のすべての表にフル・アクセスすることが可能になります。

2.9.2 プロセス・ユーザー名および Oracle Rdb のシステム・ユーザー名の設定

Oracle Rdb のシステム・ユーザー名を設定するため、Oracle SQL/Services では、プロセス・ユーザー名の偽装メカニズムを使用して、プロセス・ユーザー名およびエグゼキュータ・プロセスのすべての関連プロセス属性を設定します。プロセス・ユーザー名を設定することで、Oracle SQL/Services では、Oracle Rdb のシステム・ユーザー名の適切なデフォルト値を自動的に確定します。さらに、プロセス・ユーザー名を設定することで、Oracle SQL/Services では、エグゼキュータ・プロセスのコンテキスト内で外部ファンクションおよびプロシージャを一貫して実行するために適切な環境を確立します。

Oracle SQL/Services では、提供されるサービスのタイプに基づいて、次のように異なる時点でプロセス・ユーザー名が設定されます。

- 汎用サービス

Oracle SQL/Services では、汎用サービスのエグゼキュータ・プロセスに新規クライアント接続が割り当てられるたびに、プロセス・ユーザー名が設定されます。これにより、エグゼキュータ・プロセスのコンテキスト内で実行される外部ファンクションおよびプロシージャに対して常に適切な実行環境が提供されます。

- データベース・サービス

Oracle SQL/Services では、エグゼキュータ・プロセスが最初に起動された時点で、データベース・サービスのエグゼキュータ・プロセスに対してプロセス・ユーザー名が1回設定されます。ただし、エグゼキュータがデータベースにアタッチされると、データベース・リクエストを適切かつ正常に実行するため、Oracle SQL/Services では、新規クライアント接続のエグゼキュータ・プロセスへの割当て時にプロセス・ユーザー名をリセットできません。この動作により、エグゼキュータ・プロセスのコンテキスト内で実行される外部

ファンクションおよびプロシージャに対して一貫した実行環境が提供されます。ただし、このことは、このようなすべてのファンクションおよびプロシージャが実行される場合、データベース認可属性が接続ユーザーに設定されたサービスの接続ユーザー名ではなく、サービス所有者ユーザー名が使用されることを意味します。Oracle SQL/Services で外部ファンクションおよびプロシージャを使用する方法の詳細は、2.10.2 項を参照してください。この項には、接続ユーザー名の各権限を使用して独立型のサーバー・プロセスのコンテキスト内で実行するように外部ファンクションおよびプロシージャを定義する方法も含まれます。

Oracle SQL/Services でエグゼキュータ・プロセスを作成する場合、Oracle SQL/Services モニター・プロセスにより、サービス所有者アカウントの認可権限とデフォルト権限がマージされます。これらの権限の組合せは、エグゼキュータ・プロセスの認可権限マスクとなります。Oracle SQL/Services がエグゼキュータ・プロセスのプロセス・ユーザー名をリセットする場合、新規プロセス・ユーザー名の認可権限とデフォルト権限をマージすることで、エグゼキュータ・プロセスのプロセス権限マスクと現行権限マスクを設定します。ただし、Oracle SQL/Services では、エグゼキュータ・プロセスの認可権限マスクを設定できません。したがって、サービス所有者アカウントに過度な認可権限またはデフォルト権限が付与されていないことを確認する必要があります。通常、サービス所有者アカウントには、TMPMBX および NETMBX 権限のみを付与します。

2.10 サービス所有者ユーザー名を選択する際のセキュリティの検討

サービス所有者アカウントとして使用するアカウントの選択および構成のためのセキュリティ基準は、サーバーが稼働するプラットフォーム、提供するサービスのタイプ、およびサービスのデータベース認可属性に基づきます。

2.10.1 データベース・リクエストの実行環境

次のガイドラインは、サービス・タイプとサービスのデータベース認可属性に基づいて、OpenVMS システムでサービス所有者アカウントとして使用するアカウントを選択および構成する際に役立ちます。

汎用サービス

データベース・アクセス認可を接続ユーザー名に設定する場合 データベース認可が接続ユーザー名に設定された汎用サービスの場合、最小の権限を備えた非システム・ユーザー識別コード (UIC) を保持するアカウントを選択する必要があります。Oracle SQL/Services のインストール・プロシージャにより、データベース認可が接続ユーザー名に設定されたすべての汎用サービスで使用できる `SQLSRV$DEFLT` という権限のないアカウントが作成されます。

データベース・アクセス認可をサービス所有者に設定する場合 データベース認可がサービス所有者に設定された汎用サービスの場合、最小の権限を備え、サービスのアクセス用として設計されたデータベースにのみアクセスするのに必要なデータベース・アクセス権を付与されている非システム UIC を保持するアカウントを選択する必要があります。データのセキュリティと整合性を確保するため、選択するアカウントは、通常、システム上のデータベースとそれに含まれるデータに対するアクセス権に関して厳しく制限されます。

データベース・サービス

データベース・アクセス認可を接続ユーザー名に設定する場合 データベース認可が接続ユーザー名に設定されたデータベース・サービスの場合、最小の権限を備えた非システム UIC を保持するアカウントを選択する必要があります。すべてのデータベース・リクエストは接続ユーザー名を使用して実行されるため、サービス所有者ユーザー名として選択するアカウントには、データベースにアタッチする権限のみが付与されている必要があります。たとえば、データベースへのアタッチ権限のみを `SQLSRV$DEFLT` アカウントに付与することで、Oracle SQL/Services のインストール・プロシージャにより作成された権限のないアカウントを使用できます。

データベース・アクセス認可をサービス所有者に設定する場合 データベース認可がサービス所有者に設定されたデータベース・サービスの場合、最小の権限を備え、データベース内の特定のデータにアクセスするための権限と、そのデータに対して特定の操作を実行するための権限を付与されている非システム UIC を保持するアカウントを選択する必要があります。サービス所有者アカウントに付与するアクセス権の範囲は、データベース認可がサービス所有者に設定されたデータベース・サービスを提供する各データベースに固有です。

各サービス所有者アカウントは、次の最小の権限および割当て制限を使用して構成する必要があります。

- TMPMBX および NETMBX 権限
- デフォルトのアカウント割当て制限値で十分ですが、次の例外があります。
 - ENQLM: 割当て制限を 2000 に設定
 - JTQUOTA: 割当て制限を 4096 に設定

現在のデータベースに数百の記憶域が含まれる場合、状況によっては、AUTHORIZE を使用してプロセス用の PGFLQUOTA（ページング・ファイル制限）を増加するか、システム生成（SYSGEN）ユーティリティを使用して PAGFILCNT および VIRTUALPAGECNT システム・パラメータ値を増加する必要があります。記憶域当たり 60 ページを確保できます。

AUTHORIZE を使用してユーザー・アカウントを検証および変更します。AUTHORIZE を使用するシステム権限を保持している必要があります。AUTHORIZE プロンプト（UAF>）で、アカウント名を使用して SHOW コマンドを入力し、その特定のアカウントをチェックします。次に例を示します。

```
$ SET DEFAULT SYSS$SYSTEM
$ RUN AUTHORIZE
UAF> SHOW SMITH
```

割当て制限と権限を変更するには、MODIFY コマンドを使用します。

```
MODIFY account-name /quota-name=NNN /PRIVILEGE=(priv-name) /DEFPRIV=(priv-name)
```

次の例では、SMITH アカウントの FILLM 割当て制限を変更し、SMITH アカウントに TMPMBX および NETMBX 権限を付与します。

```
UAF> MODIFY SMITH /FILLM=300 -
_UAF> /PRIVILEGE=(TMPMBX,NETMBX) /DEFPRIV=(TMPMBX,NETMBX)
```

ユーザーは、AUTHORIZE での変更を反映するため、一度ログアウトして再度ログインする必要があります。アカウントの割当て制限を変更する方法の詳細は、OpenVMS システム管理ドキュメントの OpenVMS Authorize ユーティリティの説明を参照してください。

Oracle Rdb の要件の詳細は、Oracle Rdb のインストレーションおよび構成ガイドを参照してください。

2.10.2 外部ファンクションおよびプロシージャ用の実行環境

外部ファンクションおよびプロシージャは、エグゼキュータ・プロセスのコンテキスト内で、または Oracle Rdb が外部ファンクションおよびプロシージャを実行するために特に作成した独立型のサーバー・プロセスで実行するよう定義できます。

エグゼキュータ・プロセスのコンテキスト内で実行するよう外部ファンクションまたはプロシージャを定義するには、SQL BIND ON CLIENT SITE 構文を使用します。Oracle Rdb データベース・エンジンの観点からすると、データベース・クライアントは、Oracle SQL/Services クライアントではなく Oracle SQL/Services のエグゼキュータ・プロセスです。独立型のサーバー・プロセスで実行するよう外部ファンクションまたはプロシージャを定義するには、SQL BIND ON SERVER SITE 構文を使用します。

Oracle SQL/Services によるデータベース・アクセス認可モデルの実装方法の詳細は、2.7 項を参照してください。外部ファンクションおよびプロシージャの定義方法の詳細は、『Oracle Rdb SQL リファレンス・マニュアル』および『Oracle Rdb7 Guide to SQL Programming』を参照してください。

2.10.2.1 エグゼキュータ・プロセスのコンテキストで実行される外部ファンクションおよびプロシージャ

外部ファンクションおよびプロシージャはエグゼキュータ・プロセスのコンテキスト内で実行するよう定義できるため、サービス所有者アカウントを構成する場合にはそのことを考慮する必要があります。

汎用サービス

データベース・アクセス認可を接続ユーザー名に設定する場合 エグゼキュータ・プロセスのコンテキストで実行するよう定義された外部ファンクションおよびプロシージャは、常にこのタイプのサービスを使用する接続ユーザー名の各権限に基づいて実行されます。

データベース・アクセス認可をサービス所有者に設定する場合 エグゼキュータ・プロセスのコンテキストで実行するよう定義された外部ファンクションおよびプロシージャは、常にこのタイプのサービスを使用するサービス所有者ユーザー名の各権限に基づいて実行されます。

データベース・サービス

データベース・アクセス認可を接続ユーザー名に設定する場合 エグゼキュータ・プロセスのコンテキストで実行するよう定義された外部ファンクションおよびプロシージャは、常にこのタイプのサービスを使用するサービス所有者ユーザー名の各権限に基づいて実行されます。この理由は、一度データベースにアタッチすると、Oracle SQL/Services ではエグゼキュータ・プロセスを再構成できないためです。接続ユーザー名の各権限に基づいて外部ファンクションおよびプロシージャを実行するには、SQL BIND ON SERVER SITE 構文を使用して独立型のサーバー・プロセスで実行するよう外部ファンクションおよびプロシージャを定義する必要があります。

データベース・アクセス認可をサービス所有者に設定する場合 エグゼキュータ・プロセスのコンテキストで実行するよう定義された外部ファンクションおよびプロシージャは、常にこのタイプのサービスを使用するサービス所有者ユーザー名の各権限に基づいて実行されます。

2.10.2.2 独立型プロセスのコンテキストで実行される外部ファンクションおよびプロシージャ

外部ファンクションおよびプロシージャは、独立型のサーバー・プロセスのコンテキスト内で実行するよう定義できます。このモデルの場合、外部ファンクションおよびプロシージャの実行環境は、汎用サービスとデータベース・サービスのどちらを使用するかにかかわらず、データベース認可属性に基づきます。

汎用サービスとデータベース・サービス

データベース・アクセス認可を接続ユーザー名に設定する場合

独立型のサーバー・プロセスのコンテキストで実行するよう定義された外部ファンクションおよびプロシージャは、常にこのタイプの認可を使用する接続ユーザー名の各権限に基づいて実行されます。

データベース・アクセス認可をサービス所有者に設定する場合

独立型のサーバー・プロセスのコンテキストで実行するよう定義された外部ファンクションおよびプロシージャは、常にこのタイプの認可を使用するサービス所有者ユーザー名の各権限に基づいて実行されます。

2.11 エグゼキュータ数の属性の設定

システムで一部のサービスの使用状況は、時間が経過してもほとんど変わりませんが、一部のサービスの使用状況は、ユーザー・アクティビティの増大と減少に伴って時間とともに変化します。サービスの MIN_EXECUTORS、MAX_EXECUTORS および IDLE_EXECUTOR_TIMEOUT 属性の値を適切に設定することで、クライアントに効率的にサービスを提供できます。

MIN_EXECUTORS 属性は、トランザクションで再利用可能なサービスの MAX_EXECUTORS 属性と常に同じ値に設定する必要があります。これにより、Oracle SQL/Services では、サービスの使用可能なエグゼキュータ・プロセスのプールに新規クライアント接続を均等に分散できます。

2.11.1 サービス用の固定数のエグゼキュータの構成

時間が経過しても接続しているユーザーの数がほとんど変わらないサービスの場合、MIN_EXECUTORS 属性を MAX_EXECUTORS 属性と同じ値に設定することをお勧めします。これにより、固定数のエグゼキュータが事前に起動され、クライアント・アプリケーションで常時使用できます。これにより、新規クライアント接続用にエグゼキュータ・プロセスを起動するときの遅延を避けることができます。

2.11.2 サービス用の可変数のエグゼキュータの構成

時間が経過するにつれて接続ユーザーの数が変化するサービスの場合（ピーク時にユーザーが増加し、閑散時にユーザーが減少する場合）、負荷に合わせてエグゼキュータの数を調整できます。これを行うには、必要に応じて Oracle SQL/Services サーバーで自動的に新規エグゼキュータ・プロセスを起動するか、ピーク時に増大する要求数を予測して新規エグゼキュータ・プロセスを事前に起動します。

2.11.2.1 必要に応じた新規エグゼキュータ・プロセスの起動

サービス要求の増大と減少に対応する簡単な方法は、MIN_EXECUTORS 属性を平常時のサービスのアクティビティをサポートする値に設定し、MAX_EXECUTORS 属性をピーク時のサービスのアクティビティをサポートする値に設定することです。これで、Oracle SQL/Services により、使用量のピーク時に増大する要求数に合わせて新規エグゼキュータ・プロセスが作成されます。IDLE_EXECUTOR_TIMEOUT 属性の適切な値を選択することで、要求数が少しの間減少しても、起動済のエグゼキュータをアクティブなまま維持できます。この方法では、構成と管理を非常に簡単に行うことができますが、新規エグゼキュータ・プロセスを作成する必要がある場合、ピーク期間の最初にサービスに接続した新規ユーザーに対して若干の遅延が発生するというデメリットがあります。

2.11.2.2 増大する要求数を予測した新規エグゼキュータ・プロセスの事前起動

サービス要求の増大と減少に対応するより複雑な方法は、MAX_EXECUTORS 属性をピーク時のサービスのアクティビティをサポートする値に設定し、次に SQLSRV_MANAGE スクリプトを作成することです。このスクリプトは、ピーク時にサービスの MIN_EXECUTORS 属性の値を増加し、サービスに対する要求が減少し始めた時点でサービスの MIN_EXECUTORS 属性の値を減少するのに使用します。

バッチ・ジョブのコマンド・プロシージャを記述することで、サービスのユーザー数の増加を予測して MIN_EXECUTORS 属性の値を増加し、ピーク期間の最後に MIN_EXECUTORS 属性の値を減少する SQLSRV_MANAGE スクリプトを自動的に起動できます。要求が増大する前にエグゼキュータを事前に起動しておくこの方法のメリットは、ピーク期間の最初にサービスに接続した新規ユーザーに対して、エグゼキュータ・プロセスの作成による遅延が発生しないことです。この方法のデメリットは、管理がより複雑になることです。

2.12 SQL 初期化ファイルの使用

CREATE SERVICE または ALTER SERVICE コマンドの SQL_INIT_FILE 引数を使用して、クライアント接続の SQL 環境をカスタマイズする SQL 文が含まれるファイルを指定できます。たとえば、SQL 初期化ファイルの使用により、SQL 言語とデフォルト・キャラクタ・セットを設定できます。SQL 初期化ファイルの文は、クライアントがサービスに接続するたびに実行されます。

SQL 初期化ファイルで使用される構文表記規則の詳細は、6.1 項を参照してください。

2.13 SQL/Services の論理名の使用

表 2-6 に、Oracle SQL/Services および OCI Services for Oracle Rdb で使用できる論理名を示します。

表 2-6 SQL/Services の論理名

論理名	説明
SQLSRV_EXEC_LOG	すべてのサービスのサービス・ログを無効化します。
SQLSRV_DISP_LOGPATH	ディスパッチャ・ログ・ディレクトリの場所を指定します。
SQLSRV_DISP_DUMP_PATH	ダンプ・ファイル・ディレクトリの場所を指定します。
SQLNET_DEBUG_FLAGS	SQL/Services エグゼキュータ・ログ・ファイルに追加情報を記録します。
SQLNET_RECO_USER	データベース・リンクから OCI Services for Oracle Rdb を使用しており、2 フェーズ・コミットを実行する場合、トランザクションの途中で Oracle システムまたは Oracle Rdb システムに障害が発生すると、Oracle Transaction Manager は、トランザクションの結果を検証するために Oracle Rdb データベースへの再接続を試行します。OCI Services for Oracle Rdb でこの接続を作成するには、ユーザー名とパスワードが必要です。ユーザーは、進行中のトランザクションの結果を記録する ORA_COMM_TRANS 表にのみアクセスする必要があります。
SQLNET_BLOB	以前のリリースの OCI Services for Oracle Rdb では、Oracle Rdb のセグメント化された文字列は、Oracle ツールにより LONG 型の文字列として扱われていました。現在では、OCI LOB インタフェースの大部分が実装されたため、セグメント化された文字列は Oracle LOB として扱うことが可能です。この機能を有効化するには、この論理名を Y または y に定義する必要があります。論理名を別の名前で定義した場合や、論理名を定義しない場合、Oracle Rdb のセグメント化された文字列は、以前のリリースと同様に LONG 型の文字列として扱われます。
SQLNET_DOMAIN	OCI Services for Oracle Rdb では、デフォルトのドメイン名が .WORLD であると仮定されます。論理名 SQLNET_DOMAIN を定義して、ドメイン名を変更できます。この機能は、特に Oracle Database のデータベース・リンクを使用して OCI Services for Oracle Rdb にアクセスするユーザーに適用されます。論理名は、サービスのプロセス初期化ファイルに定義できます。
SQLNET_BUGCHECK_FILE	論理名 SQLNET_BUGCHECK_FILE を定義して、ダンプ・ファイルの名前を指定できます。
SQLSRV\$MAX_EXECUTOR_FAILURES	論理名 SQLSRV\$MAX_EXECUTOR_FAILURES を定義して、エグゼキュータが失敗する前に許容する障害の最大数を変更できます。論理名に割り当てられた値により、デフォルト値の 2 が上書きされます。このようにして、定期的なデータベース・メンテナンス中にエグゼキュータとサービスを停止する頻度を制御できます。

表 2-6 SQL/Services の論理名 (続き)

論理名	説明
SQLSRV\$LOG_CONNECTIONS	論理名 SQLSRV\$LOG_CONNECTIONS を NO に定義すると、成功した接続がディスパッチャ・ログ・ファイルに記録されないため、ディスパッチャ・ログ・ファイルのサイズが減少します。論理名を定義しないか、別の値を割り当てると、成功した接続が記録されます。この論理名は、ディスパッチャの起動時に評価されるため、論理名を変更したらディスパッチャを再起動する必要があります。
RDB\$DDTM_XG_INFO	論理名 RDB\$DDTM_XG_INFO を SYSTEM\$<nodename> に定義すると、XA ゲートウェイを 2 フェーズ・コミット・トランザクションで使用するよう指定できます。この論理名を定義する場合、論理名 SQLNET_RECO_USER も定義する必要があります。
SQLNET_VALIDATE_PROGRAM	論理名 SQLNET_VALIDATE_PROGRAM を Y に定義すると、OCI Services for Oracle Rdb により、ログオン処理中に各ユーザー名とプログラムが ORA_VALID_PROGRAMS 表のエントリに対して検証され、一致エントリのない項目は拒否されます。
SQLNET_STRUCTURED_DATE_TYPES	この論理名を使用して構造日付型を有効化し、正しいデータ型を OCI クライアントに返します。OCI Services for Oracle Rdb により返される実際のデータ型は、4.1.9 項を参照してください。
SQLNET_TIMESTAMP_DATE_TYPE	この論理名を使用して、タイムスタンプ・データ型を有効化します。この論理名を Y に設定すると、TIME および TIMESTAMP データ型が TIMESTAMP として返されます。設定しない場合、データ型は DATE として返されます。

すべてのサービスのサービス・ログを無効化する場合、サービスの起動前に次の例のように論理名 SQLSRV_EXEC_LOG を定義します。

```
$DEFINE/SYSTEM SQLSRV_EXEC_LOG NOLOG
```

サービス・ログは、問題が発生した場合に必要なため、無効化しないことをお勧めします。

次の例は、ディスパッチャ・ログ・ディレクトリの場所の指定方法を示しています。

```
$ DEFINE/SYSTEM/EXEC SQLSRV_DISP_LOGPATH DKA100:[USER1.LOG]
```

次の例は、ダンプ・ファイル・ディレクトリの場所の指定方法を示しています。

```
$ DEFINE/SYSTEM/EXEC SQLSRV_DISP_DUMP_PATH DKA100:[USER1.DUMP]
```

論理名 SQLSRV_DISP_LOGPATH および SQLSRV_DISP_DUMP_PATH は、システム論理名として定義する必要があります。論理名 SQLSRV_DISP_LOGPATH または SQLSRV_DISP_DUMP_PATH を定義しない場合、ディスパッチャ・ログ・ファイルとダンプ・ファイルのデフォルト・ディレクトリは、SYS\$MANAGER ディレクトリになります。

論理名 SQLSRV_DISP_LOGPATH または SQLSRV_DISP_DUMP_PATH のいずれかを定義したら、ディスパッチャを再起動する必要があります。

Oracle SQL/Services サーバーの メンテナンス

Oracle SQL/Services を設定し、1 つ以上のサーバーを構成したら、次のようなメンテナンス・タスクを定期的に行う必要があります。

- サーバー・アクティビティの監視
- クライアント接続の監視

これらの各内容について、後続の各項で説明します。

3.1 サーバー・アクティビティの監視

サーバー・アクティビティの監視は、SHOW コマンドを使用してオブジェクトの稼働状態を表示する作業の一部を構成します。たとえば、サービスおよびディスパッチャ・オブジェクトについて、SHOW コマンドを使用してオブジェクトが実行中であるかどうかを確認できます。サービスまたはディスパッチャ・オブジェクトが停止しており、実行する必要がある場合、ログ・ファイルとダンプ・ファイルを確認してオブジェクトの実行が停止した理由を判断する必要があります。問題を解決したら、START SERVICE または START DISPATCHER コマンドを発行し、起動するオブジェクトのサービス名またはディスパッチャ名を指定します。別の SHOW コマンドを発行して、サービスまたはディスパッチャ・オブジェクトが実行中であることを確認します。

SHOW SERVICES コマンドを使用すると、特定のサーバー上で提供されているすべてのサービスのピーク負荷期間中にクライアント・アクティビティを監視することもできます。たとえば、アクティブ・クライアントの数が最大許容数に到達した場合、クライアント接続障害の可能性を低下させるために、クライアントの最大許容数の増加を検討する必要があります。ALTER SERVICE コマンドを使用すると、特定のサービスの MAX_EXECUTORS 値を動的に増加できます。

3.2 クライアント接続の監視

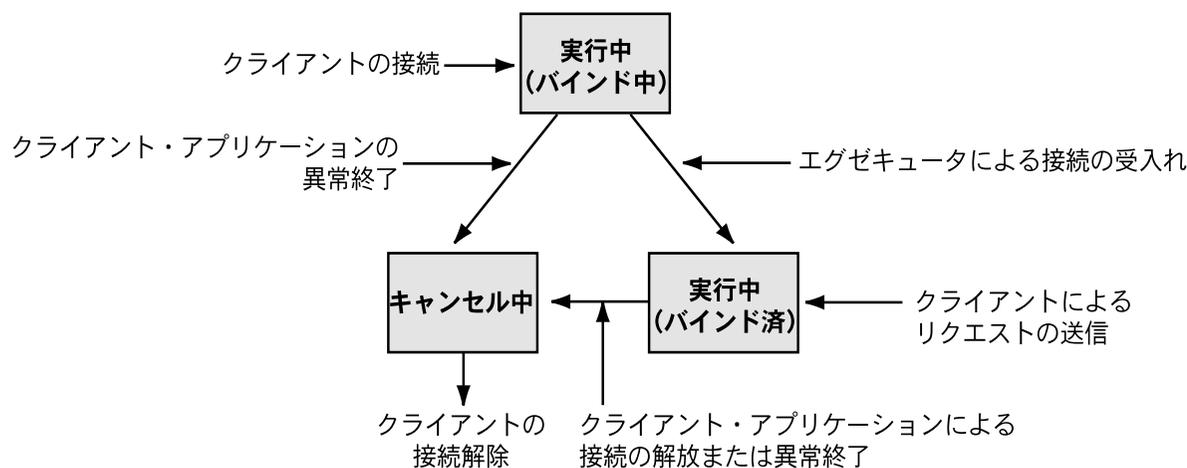
SHOW CLIENTS コマンドを使用すると、各クライアントのサービスへの接続、リクエストの送信、および接続の解放に応じて、クライアントの状態を表示できます。接続状態の発生、順序および継続期間は、サービスのタイプごとに異なります。クライアントの状態は、各接続の実行している処理の内容や、接続が通常どおり提供されているかどうかを判断するのに役立ちます。ただし、接続状態の情報それ自体は、すべての問題をトラブルシューティングするのに十分ではないことがあります。問題のトラブルシューティング方法の詳細は、7.7 項を参照してください。

3.2.1 項および 3.2.2 項で、クライアント接続が表示する状態、発生する順序、およびセッションで再利用可能なサービスまたはトランザクションで再利用可能なデータベース・サービスによるサービス提供時の各状態の相対的継続期間について説明します。

3.2.1 セッションで再利用可能なサービスのクライアント接続状態

図 3-1 に、クライアントおよびエグゼキュータ・イベントに関連して、セッションで再利用可能なサービスのエグゼキュータ・プロセスによるサービス提供時に SHOW CLIENTS コマンドでクライアント接続に関して表示可能な 3 つの接続状態を示します。

図 3-1 セッションで再利用可能なサービスのクライアント接続状態



セッションで再利用可能なサービスへの接続を試行するクライアントからの接続は、エグゼキュータによる接続の受入れの待機中は、実行中（バインド中）の状態にあります。空いているエグゼキュータ・プロセスが接続を受入れ可能な場合、接続はごく短時間のみ実行中（バインド中）の状態になります。ただし、接続用に新規エグゼキュータ・プロセスを作成する必要がある場合、接続はより長い時間（数秒程度）実行中（バインド中）の状態のままとなります。

エグゼキュータ・プロセスが接続を受け入れると、接続状態は実行中（バインド中）から実行中（バインド済）に遷移します。セッションで再利用可能なサービスのエグゼキュータが接続を受け入れると、エグゼキュータは接続期間中その接続にバインドされたままとなります。

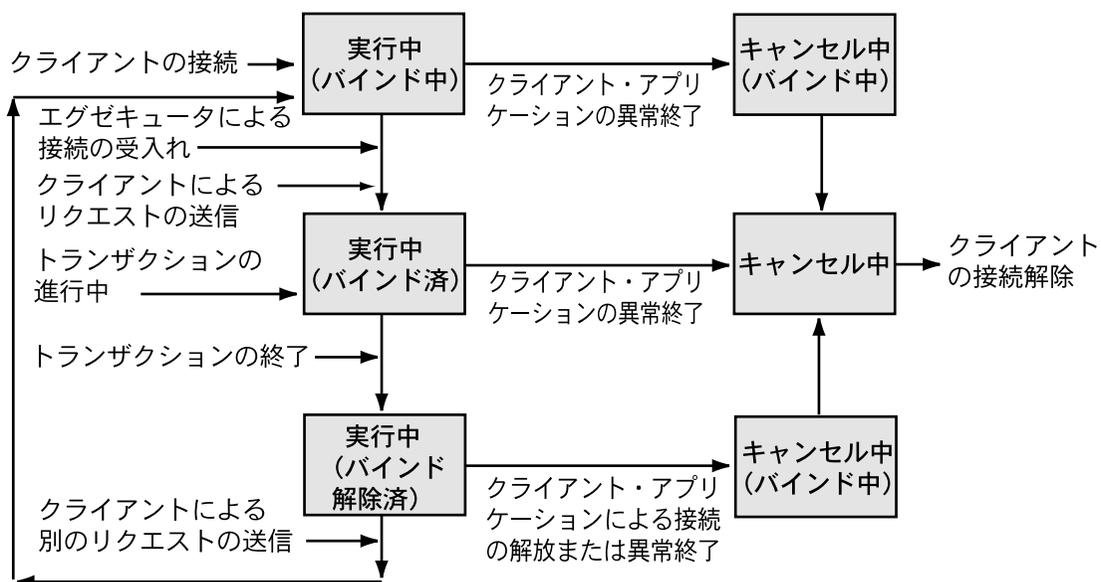
アプリケーションが接続を正常に解放するか、アプリケーションが異常終了すると、接続はキャンセル中の状態に遷移します。接続のキャンセル中の状態が続くのは、通常、ごく短時間です。ただし、他のデータベース・アクティビティが原因で未処理のデータベース・トランザクションのクリーンアップに遅延が発生する場合、接続はより長い時間キャンセル中の状態のままとなります。

3.2.2 トランザクションで再利用可能なデータベース・サービスのクライアント接続状態

トランザクションで再利用可能なサービスのエグゼキュータは、一度に1つのクライアントのトランザクションに対するリクエストを処理します。ただし、このエグゼキュータは、多くの同時クライアント接続により共有されます。暗黙的または明示的にトランザクションを開始する SQL 文がクライアントによって発行されると、トランザクションが起動します。クライアントが SQL COMMIT または ROLLBACK 文を正常に発行した場合、あるいはトランザクションをコミットまたはロールバックするストアード・プロシージャを実行した場合、トランザクションは終了します。クライアント接続は、一度エグゼキュータ・プロセスに割り当てられると、接続の継続中はそのプロセスに関連付けられたままとなります。他のエグゼキュータ・プロセスを使用して、特定の接続のかわりにトランザクションを処理することはできません。新規クライアント接続は、通常、既存接続の最小限の数に基づいてエグゼキュータに割り当てられます。ただし、アプリケーションによっては、ALTER SERVICE APPLICATION TRANSACTION USAGE CONCURRENT 属性を使用してこの動作を変更する必要があります。

図 3-2 に、クライアントおよびエグゼキュータ・イベントに関連して、トランザクションで再利用可能なデータベース・サービスのエグゼキュータによるサービス提供時に SHOW CLIENTS コマンドでクライアント接続に関して表示可能な 5 つの接続状態を示します。

図 3-2 トランザクションで再利用可能なデータベース・サービスのクライアント接続状態



トランザクションで再利用可能なサービスへの接続を試行する新規クライアントからの接続は、割当て済エグゼキュータによる接続の受入れの待機中は、実行中（バインド中）の状態にあります。同様に、既存のクライアントが新規トランザクションを開始した場合、割当て済エグゼキュータによる新規トランザクションの処理の待機中は、接続は実行中（バインド中）の状態にあります。継続期間の短いトランザクションを実行するよう適切に設計されたアプリケーションの場合、接続は短時間のみ実行中（バインド中）の状態のままとなります。ただし、クライアントがトランザクションを実行する速度が向上し、トランザクションの平均の長さが増大するにつれ、この時間も増加します。

エグゼキュータ・プロセスが新規または既存の接続にバインドされると、接続状態は実行中（バインド中）から実行中（バインド済）に遷移します。エグゼキュータは、一度接続にバインドされると、トランザクションの最後までその接続にバインドされたままとなります。新規接続では、エグゼキュータは、新規データベース・セッションを確立するのに必要な期間のみ接続にバインドされたままとなります。トランザクションの最後に、または新規接続の受入れ後に、エグゼキュータは接続とのバインドを解除され、接続状態は実行中（バインド済）から実行中（バインド解除済）に遷移します。

アプリケーションが接続を正常に解放するか、アプリケーションが異常終了すると、実行中（バインド中）または実行中（バインド解除済）の状態にある接続は、キャンセル中（バインド中）の状態に遷移します。エグゼキュータが、現在バインドされている接続のトランザクションと、キュー内でエグゼキュータを待機している接続の他のトランザクションをすべて完了すると、キャンセル中（バインド中）の接続は、その状態からキャンセル中の状態に遷移します。アプリケーションが接続を正常に解放するか、アプリケーションが異常終了すると、実行中（バインド済）の状態にある接続は、直接キャンセル中の状態に遷移します。キャンセル中の状態のときに、エグゼキュータは接続のデータベース・セッションをクリーンアップし、最終的に接続とのバインドを解除します。

OCI Services for Oracle Rdb の処理

OCI Services for Oracle Rdb（旧 SQL*Net for Oracle Rdb）が提供する環境により、既存の OCI アプリケーションを実行して Oracle Rdb データベースのデータにアクセスできます。OCI Services for Oracle Rdb により提供される独自のメリットは、Oracle SQL セマンティクスを使用して Oracle Rdb データベースのデータにアクセスできることです。

Oracle Rdb SQL 内では、特に OCI Services for Oracle Rdb により Oracle レベル 1 言語、Oracle レベル 2 言語および Oracle 関数が作成されています。Oracle Rdb SQL 機能は、SQL プログラマが Oracle Rdb サーバーと Oracle サーバーの両方で実行できるクライアント・アプリケーションを作成する際に役立ちます。

この章では、OCI Services for Oracle Rdb の処理における次の各機能の詳細を説明します。

- **4.1 項**では、Oracle RDBMS サーバーと Oracle Rdb サーバー間で共通のアプリケーション開発を可能にするため、OCI Services for Oracle Rdb が多くの処理機能を使用して Oracle Rdb SQL を拡張する方法について説明します。
- **4.2 項**では、Oracle SQL/Services の実行環境を初期化するために特定のセッション・パラメータを設定する SQL 初期化ファイルについて説明します。
- **4.3 項**では、Oracle Forms との互換性を確立する方法について説明します。
- **4.4 項**では、Oracle SQL/Services の汎用サービスの使用方法について説明します。
- **4.5 項**では、RDB_NATCONN71 コマンド・ファイルを使用して、Oracle Rdb データベースを準備またはアップグレードする方法、Oracle Rdb データベースから OCI 関連の表およびファンクションを削除する方法、または準備された Oracle Rdb データベースの暗号化されたパスワードを使用してユーザーの追加、削除、変更、表示を行う方法について説明します。
- **4.6 項**では、キャラクタ・セットの使用方法について説明します。
- **4.7 項**では、Oracle Rdb データベースからのデータベース・リンクとして Oracle Rdb データベースを参照する方法について説明します。
- **4.8 項**では、IMPORT/EXPORT サイクルの後に OCI Services for Oracle Rdb の機能を維持する方法について説明します。

4.1 OCI Services for Oracle Rdb の機能

次の各項では、OCI Services for Oracle Rdb の処理機能の詳細を説明します。

4.1.1 OCI メッセージ・マッピング

OCI では、単一のコールで複数の行をオープンおよびフェッチできます。この機能をエミュレートするため、OCI Services for Oracle Rdb では、次のように同じ結果が取得される複数のステップを暗黙的に実行します。

1. カーソルをオープンします。
2. 指定された数の行をフェッチします。
3. データとともにレスポンスを返します。

Oracle Rdb の動的 SQL 文の使用時と比較すると、OCI の使用時にはカーソルがオープンおよび記述される順序が異なります。OCI Services for Oracle Rdb では、メッセージ順序の操作とクライアントへの OCI メッセージ順序の提供によりその差異が隠蔽されます。

4.1.2 カーソル管理

OCI ではすべての文にカーソルが使用されるため、OCI Services for Oracle Rdb では、文ごとに仮想 OCI カーソルが管理されます。OCI Services for Oracle Rdb は、これらの仮想 OCI カーソルを Oracle Rdb の動的 SQL 文 ID (ほとんどの文の場合) または Oracle Rdb の動的 SQL カーソル (SELECT 文の場合) に関連付けます。

4.1.3 データ型

OCI Services for Oracle Rdb は、マシンのネイティブ・データ型でデータをフェッチし、次にそのデータをクライアントに送信する前に Oracle の移植可能データ型に変換します。

逆に、OCI Services for Oracle Rdb は、移植可能データ型でデータを受信し、そのデータをマシンのネイティブ・データ型に変換して Oracle Rdb の動的 SQL に渡します。

4.1.4 データ定義言語

OCI Services for Oracle Rdb は、各 DDL リクエストの前後に COMMIT 文を発行することで Oracle の動作を模倣します。

4.1.5 SQL カーソル・セマンティクス

すべての動的 SQL 宣言のカーソル文に対して、OCI Services for Oracle Rdb では WITH HOLD PRESERVE ALL 句が追加されます。この操作のメリットは、複数のトランザクション間でカーソルがオープンされたままになることです。この動作は、Oracle サーバーの動作の模倣です。

4.1.6 Oracle SQL ALTER SESSION 文

ALTER SESSION 文は、『Oracle Database SQL 言語リファレンス』に記載されているとおり、様々な情報を提供するために OCI Services for Oracle Rdb によって処理されます。OCI Services for Oracle Rdb では、第 5 章に記載されている ALTER SESSION 文の構文のみがサポートされます。

4.1.7 データ書式設定

Oracle サーバーでは、クライアント用にデータを書式設定し、クライアントからは書式設定済のデータを受信します。書式設定された情報は、ALTER SESSION 文により設定されたデフォルトのデータ形式を使用してサーバーに渡されます。これにより、次の3つの重要な Oracle 機能が有効化されます。

- TO_DATE、TO_CHAR および TO_NUMBER 関数の適切な処理
- ALTER SESSION 文により指定された形式での日付リテラルの適切な処理
- ALTER SESSION 文で指定された情報に従ってテキストとしてフェッチされた場合の日付および数値データの適切な書式設定

SQL ALTER SESSION 文の詳細は、[第5章](#)を参照してください。

4.1.8 文の解析

Oracle Rdb SQL では、Oracle レベル1 言語と Oracle レベル2 言語が提供されます。ただし、これらの言語を使用しても、Oracle Rdb SQL では対応できない複数の Oracle 構成メンバーが存在します。日付変換エラーなどの特定の理由で Oracle Rdb SQL により文が拒否された場合、OCI Services for Oracle Rdb では、その文を調査し、Oracle 形式の日付リテラルを Oracle Rdb SQL で対応可能な同等の文に置き換えます。

注意： 解析は、文が失敗するまで実行されません。たとえば、有効な OpenVMS 日付リテラルが提供されると、その日付リテラルは OCI Services for Oracle Rdb の支援なしに処理されます。

次に、文の解析の例を示します。

例 4-1 Oracle 日付リテラルの ANSI 日付列への挿入

この例では、Oracle 日付リテラルの ANSI 日付列への挿入を試みます。

```
INSERT INTO ATABLE VALUES ('3-AUG-46');
```

この文は Oracle Rdb SQL により拒否されるため、OCI Services for Oracle Rdb によって次のように置き換えられます。

```
INSERT INTO ATABLE VALUES (CAST (TO_DATE('3-AUG-46') AS DATE ANSI));
```

OCI Services for Oracle Rdb により提供される TO_DATE 関数は、文字列を日付に書式設定するのに使用される Oracle TO_DATE 関数に似ています。OCI Services for Oracle Rdb により提供される TO_DATE 関数では、DATE VMS 日付が返されるため、CAST 関数を使用して DATE ANSI 形式に一致させる必要があります。

注意： 使用可能であれば、常に DATE VMS 日付を使用することをお勧めします。DATE VMS は、Oracle DATE データ型に最も類似しています。

例 4-2 CALL という語のプロシージャ・コールへの挿入

次の例は、Oracle プロシージャ・コールを示しています。

```
BEGIN UPDATE_EMPLOYEE_NAME('FIRST_NAME', 'LAST_NAME'); END;
```

この文は Oracle Rdb に認識されないため、エラーが返されます。OCI Services for Oracle Rdb では、この文がプロシージャ・コールとして認識され、CALL という語が文に挿入されます。結果として生成される文は、Oracle Rdb によって正しく実行されます。

```
BEGIN CALL UPDATE_EMPLOYEE_NAME('FIRST_NAME', 'LAST_NAME'); END;
```

OCI Services for Oracle Rdb による SQL 文の変更方法を参照するには、ALTER SESSION LOG BRIEF 文（第 5 章を参照）を使用してロギングを有効化します。識別しやすいように、変更された文には -- GTW Fixed up というコメントが追加されます。

4.1.9 データ型の説明

OCI Services for Oracle Rdb では、すべての Oracle Rdb データ型を Oracle データ型に基づいて表現します。

- OpenVMS Alpha では、論理名 SQLNET_STRUCTURED_DATE_TYPES を Y または ON に設定するか、ALTER SESSION SET SQLNET_STRUCTURED_DATE_TYPES コマンドを発行することで、正しいデータ型を OCI クライアントに返すようリクエストできます。次の表に、クライアントに返されるデータ型を示します。

Oracle Rdb データ型	クライアントに返されるデータ型	
	SQLNET_STRUCTURED_DATE_TYPES = ON または YES	SQLNET_STRUCTURED_DATE_TYPES = OFF または NO
DATE	DATE	DATE
TIME	DATE	DATE
TIMESTAMP	TIMESTAMP	DATE
INTERVAL YEAR TO MONTH	INTERVAL YEAR TO MONTH	CHAR
INTERVAL DAY TO SECOND	INTERVAL DAY TO SECOND	CHAR

この引数が ON または YES に設定され、OCI クライアントで構造化データ型が指定されると、クライアントに構造化データ型が返されます。

- OpenVMS Alpha では、論理名 SQLNET_TIMESTAMP_DATE_TYPE を Y に設定するか、ALTER SESSION SET SQLNET_TIMESTAMP_DATE_TYPE ON コマンドを発行することで、タイムスタンプ・データ型を有効化できます。TIMESTAMP データ型を有効化した場合、TIME および TIMESTAMP データ型は TIMESTAMP として返されます。TIMESTAMP データ型を有効化していない場合、データ型は DATE として返されます。
- 2000 バイトを超えるすべての CHAR データ型、および 4000 バイトを超える VARCHAR データ型は、LONG として表現されます。
- LIST OF BYTE VARYING 列は、LONG として表現されますが、論理名 SQLNET_BLOB が Y に定義されている場合は BLOB として表現されます。
- すべての数値データ型は、Oracle の数値として表現されます。
 - INTEGER は NUMBER(10,0) として表現されます。
 - BIGINT(2) は NUMBER(19,2) として表現されます。

データ型の精度は、実際に表現可能な精度より 1 桁多く記述されます。この理由は、Oracle Rdb では、10 進数の範囲に直接マップされない範囲を持つネイティブのバイナリ・データ型が使用されるためです。Oracle の数値では、10 進表現が使用されます。したがって、OCI Services for Oracle Rdb で Oracle Rdb の TINYINT 列が NUMBER(3,0) として表現される場合、その列では数値 111 は格納できますが、数値 999 は格納できません。

4.1.10 Oracle データ・ディクショナリ

OCI Services for Oracle Rdb 用に Oracle Rdb データベースを準備する場合、いくつかのデータベース・オブジェクトをインストールします。これらのデータベース・オブジェクトは、OCI クライアントで使用可能な Oracle データ・ディクショナリを OCI Services for Oracle Rdb がエミュレートするのに役立ちます。これらのオブジェクトは次のとおりです。

- Oracle Rdb メタデータ表に対するビューとして提供される Oracle メタデータ表（またはデータ・ディクショナリ）。リリース 7.1.6 より前の OCI Services for Oracle Rdb では、これらのメタデータ・オブジェクトは、ユーザー表として扱われており、SHOW TABLES コマンドで表示される表のリストに含まれていました。リリース 7.1.6 以上から、これらのオブジェクトは、Oracle Rdb 機能の HIDE_OBJECTS を使用して作成されます。これらのオブジェクトは、SHOW SYSTEM TABLES リストに表示されますが、SHOW TABLES リストには表示されません。以前のリリースからアップグレードした場合、一部のオブジェクトは引き続きユーザー・オブジェクトとして表示され、一部のオブジェクトはシステム・オブジェクトとして表示されます。これは、OCI Services for Oracle Rdb の機能にはまったく影響しません。

詳細は、付録 A を参照してください。

- Oracle アプリケーションを移行するプログラムの便宜を図るために OCI Services for Oracle Rdb によって追加される 4 つのドメイン。次の 4 つのドメインが追加されます。

```
NUMBER AS DOUBLE PRECISION
RDB1LONG AS VARCHAR (65238)
RDB2LONG AS VARCHAR (32634)
RDB4LONG AS VARCHAR (16317)
```

- TO_DATE (DATA, FORMAT, NLS_parameters): データは文字列リテラルで、指定された書式文字列またはデフォルトの書式文字列の使用により、この文字列は DATE VMS データ型に変換されます。Oracle Rdb では、CAST 関数を使用せずに DATE VMS データ型と ANSI 日時データ型を結合することはできません。Oracle レベル 1 言語では、DATE VMS データ型は、数学的演算子とともに使用できるため、可能であれば常に使用してください。
- TO_NUMBER (DATA, FORMAT, NLS_parameters): データは文字列リテラルで、指定された書式文字列またはデフォルトの書式文字列の使用により、この文字列は DOUBLE PRECISION データ型に変換されます。TO_NUMBER 関数は、DOUBLE PRECISION データ型により制限されるため、16 桁以上の精度の整数は正確に表現できません。
- TO_CHAR (DATA, FORMAT, NLS_parameters): データは数値または日付リテラルで、書式設定された文字列が作成されます。TO_CHAR を精度 18 桁のスケールなしの整数とともに使用する場合、書式文字列を指定する必要があります。TO_CHAR をスケールなしの BIGINT データとともに使用する場合も、書式文字列を使用します。TO_CHAR では、書式文字列が指定されない場合、18 桁の数値は日付であると仮定されます。
- USERENV: 入力文字列を 1 つ指定すると、USERENV 関数により現在のセッションの詳細が VARCHAR データ型として提供されます。入力値 TERMINAL および LANGUAGE のみが、重要な情報を返します。入力文字列 LABEL、SESSIONID、USERMODE および ENTRYID は、有効な固定値を返します。
- CHARTOROWID (DATA): データは、行 ID または DB キーに変換される文字列リテラルです。
- ROWIDTOCHAR (DATA): データは、文字列に変換される行 ID または DB キーです。

記載されている関数、ドメインおよびデータ型の詳細は、Oracle ドキュメントを参照してください。

4.1.11 マルチスキーマ・エミュレーション

ほとんどの Oracle Rdb データベースはマルチスキーマ・データベースではなく、すべての Oracle Database はマルチスキーマであるため、OCI Services for Oracle Rdb では、マルチスキーマ・エミュレーションの形式が提供されます。マルチスキーマ・エミュレーションでは、Oracle データ・ディクショナリと、SQL コンパイラに対するフックが使用されます。

マルチスキーマ・エミュレーションは、デフォルトで有効化されます。マルチスキーマ・エミュレーションが不要な場合、OCI Services for Oracle Rdb が提供する ALTER SESSION SET SCHEMA EMULATION RELAXED 文を使用して無効化できます。この文の詳細は、第 5 章を参照してください。

4.2 サーバー環境の初期化

OCI Services for Oracle Rdb データベースは、Oracle SQL/Services によりサービスを提供されます。実行環境を初期化するため、Oracle SQL/Services では、サービスの SQL 初期化ファイルを指定できます。初期化ファイルにより、特定のセッション・パラメータ（ロックのデフォルトやキャラクタ・セットのデフォルトなど）を設定する SQL 文を実行します。

ほとんどの初期化文を次の方法で実行できます。

- サービスに定義された SQL 初期化ファイルで直接実行
- ORA_INIT ストアド・プロシージャを使用して間接的に実行

ORA_INIT ストアド・プロシージャを使用する場合のメリットは、ALTER SESSION 文などのデータ定義言語（DDL）文を IF ブロックで条件的に有効化できることです。Oracle Rdb SQL では、IF ブロックなどの複合文で DDL 文を使用できませんが、ORA_INIT ストアド・プロシージャでは、OCI Services for Oracle Rdb による後続の実行用に DDL やその他の文を格納できます。ORA_INIT で指定した初期化文は、サービスに定義された SQL 初期化ファイルの完了後、指定した順序で実行されます。

たとえば、Oracle Net クライアント・アプリケーションからの接続を対象に完全なサーバー・ロギングを有効化できます。ORA_SESSION を問い合せてクライアント・プログラム名を特定することはできますが、クライアント・プログラム名に応じて完全なサーバー・ロギングを条件的に有効化する問合せを SQL 初期化ファイルに構成することはできません。この理由は、ALTER SESSION LOG FULL 文は DDL 文であり、IF ブロックなどの複合文では使用できないためです。ただし、SQL 初期化ファイルで次のように ORA_INIT ストアド・プロシージャを使用して、同じ効果を得ることができます。

1. クライアント・プログラムが SQL*Plus アプリケーションであるかどうかを判別する SQLPLUS_LOG というストアド・プロシージャを定義します。

ストアド・プロシージャは、次のようになります。

```
create module SQLPLUS_LOG_MODULE language sql
  procedure SQLPLUS_LOG;
begin
  declare :A integer;
  select count(*) into :A from ORA_SESSION
    where (INFO_TYPE = 'PROGRAM' and INFO containing 'SQLPLUS');
  if :A > 0 then call ORA_INIT ('ALTER SESSION LOG FULL'); end if;
end;
end module;
```

2. SQLPLUS_LOG プロシージャをサービスの SQL 初期化ファイルから次のようにコールします。

```
call SQLPLUS_LOG();
```

3. クライアント・プログラムが Oracle Net アプリケーションである場合、OCI Services for Oracle Rdb による後続の実行用に ORA_INIT プロシージャをコールして ALTER SESSION LOG FULL 文を格納できます。

注意: クライアント・プログラム名またはクライアント端末名の間合せは、クライアントのサーバー環境をカスタマイズするための簡単で効果的な方法ですが、セキュリティ目的でこの方法を使用することはお薦めしません。OCI Services for Oracle Rdb では、クライアント・アプリケーションが正確にクライアント・プログラム名またはクライアント端末名を報告しているかどうかを保証できません。

ORA_SESSION 表の詳細は、[A.2 項](#)を参照してください。

4.3 Oracle Forms との互換性

リリース 2.0 より前の Developer/2000 またはリリース 5.0 より前の Oracle Forms を使用する場合、OCI Services for Oracle Rdb との互換性を確保するために Oracle Forms モジュールに次のプロパティを設定する必要があります。

- モジュール・プロパティのセーブポイント・モードを FALSE に設定
- データ・ブロック・プロパティのロック・モードを DELAYED に設定

プロパティを変更するには、プロパティ・パレットを使用するか、サーバーの Oracle SQL/Services のサンプル・ディレクトリに含まれるファイル RDB_PROP.FMB のプロパティ・クラス RDB_PROPERTY_CLASS を使用します。

Developer/2000 リリース 2.0 以上または Oracle Forms リリース 5.0 以上から、これらのプロパティは、Developer/2000 により実行時に自動的に処理される予定です。

4.4 汎用サービスと組み合わせた Oracle SQL/Services の使用

汎用サービスでは、Oracle クライアント・アプリケーションは、任意の数の Oracle Rdb データベースにアタッチできます。Oracle SQL/Services の 1 つまたは少数の汎用サービスにより、低い頻度で使用される多数のデータベースをサポートできます。任意の数のデータベースを処理する 1 つまたは少数の汎用サービスでは、エグゼキュータ・プロセスを事前に起動しておくのが合理的です。これは、エグゼキュータを常に実行しておくために最小限の数のエグゼキュータを設定する方法のかわりになります。汎用サービスの詳細は、[2.6 項](#)を参照してください。

Oracle Network Manager リリース 3.1.0.5.1 以上では、Oracle Rdb データベースの名前を Oracle Rdb サービスの接続文字列に指定できます。接続文字列の RDB_DATABASE フィールドを使用して、汎用サービスのアタッチ先となる 1 つ以上のデータベースを指定します。

OCI 接続文字列がマップされる Oracle SQL/Services サービスが汎用サービスである場合、クライアントでは、次のいずれかを使用して Oracle Rdb データベース名を指定できます。

- 完全な Oracle Rdb SQL ATTACH 文
- @file_spec

指定するファイルには、クライアント接続の SQL 環境をカスタマイズする SQL 文を含めることができます。また、ファイル指定を使用する方法では、クライアント・アプリケーションで複数の ATTACH 文を実行できます。最初の ATTACH 文ではデフォルトの別名 RDB\$DBHANDLE を設定する必要があるため、最初の ATTACH 文に ALIAS 句を含めることはできません。

次のファイルは、3 つの Oracle Rdb データベースにアタッチします。

```
$ type multi.sql
attach 'filename mf_personnel';
attach 'alias a filename a_personnel';
attach 'alias b filename b_personnel';
```

OCI Services for Oracle Rdb では、コマンド・ファイルの実行時に次の構文表記規則を使用します。

- 行中の先行する空白および末尾の空白は無視されます。
- コメントは2つの連続するハイフン (-) で開始します。コメント行は、1行の先頭から開始し、次の新規行に続ける必要があります。
- 各 SQL 文は、SQL EXECUTE IMMEDIATE 文によって動的に準備、実行および解放できる必要があります。
- SQL 文は、複数の行にまたがることはできません。
- SQL 文の末尾のセミコロン (;) は無視されるため、対話型 SQL を使用して SQL ファイルを起動および検証できます。

■ 単純なファイル指定

ATTACH 文は、このファイル指定を基盤として作成されます。次の例は、接続文字列に RDB_DATABASE フィールドが指定された TNSNAMES.ORA ファイルを示しています。

```
RARE_DB = (DESCRIPTION=
            (ADDRESS=
              (PROTOCOL=TCP)
              (HOST=SQS_NODE)
              (PORT=1234)
            )
            (CONNECT_DATA= (SERVICE=OCI_UNIV)
              (RDB_DATABASE=disk: [dev] db.rdb)
            )
          )
```

4.5 RDB_NATCONN71.COM コマンド・ファイルの使用

RDB_NATCONN71.COM コマンド・ファイルでは、OCI Services for Oracle Rdb に使用される Oracle Rdb データベースを準備またはアップグレードすることや、Oracle Rdb データベースから OCI 関連の表およびファンクションを削除することができます。また、コマンド・ファイルにより、準備された Oracle Rdb データベースの暗号化されたパスワードを使用して、ユーザーを追加、変更、削除または表示できます。

4.5.1 データベースの準備

RDB_NATCONN71.COM ファイルの PREPARE コマンドは、エミュレートされた Oracle データ・ディクショナリの初期構成用に設計されています。この操作は、OCI Services for Oracle Rdb 用に準備されていないデータベースに対して行ってください。PREPARE 操作は、OCI Services for Oracle Rdb によってアクセスされる各データベース上で実行する必要があります。

```
$ @SYS$LIBRARY:RDB_NATCONN71 PREPARE database [60 | 6i]
```

Oracle Developer をクライアントとして使用する予定の場合、60 または 6i を指定する必要があります。

注意： PREPARE 操作に置換されたため、以前のリリースの OCI Services for Oracle Rdb で使用されていた SQL 準備スクリプトは使用されなくなりました。

4.5.2 データベースのアップグレード

RDB_NATCONN71.COM コマンド・ファイルの UPGRADE コマンドを使用して、PREPARE 操作で作成した必須メタデータ・オブジェクトをアップグレードします。

```
$ @SYS$LIBRARY:RDB_NATCONN71 UPGRADE database
```

このコマンドは、データベースがすでに以前のリリースで準備されている場合、OCI Services for Oracle Rdb を新規リリースにアップグレードした後に使用する必要があります。

注意：UPGRADE 操作に置換されたため、以前のリリースの OCI Services for Oracle Rdb で使用されていた SQL アップグレード・スクリプトは使用されなくなりました。

4.5.3 OCI Services for Oracle Rdb の削除

PREPARE または UPGRADE 操作の完了後に OCI Services for Oracle Rdb のオブジェクトを削除する必要がある場合、次のコマンドを使用してその定義を削除します。

```
$ @SYS$LIBRARY:RDB_NATCONN71 DROP database
```

このコマンドにより、OCI Services for Oracle Rdb 用にデータベースにインストールされているすべての表、ビュー、ドメイン、ストアド・プロシージャおよび外部プロシージャが削除されます。これらのオブジェクトにユーザー定義の参照が存在する場合、DROP コマンドは失敗します。

4.5.4 ユーザーとパスワードの追加

ほとんどの Oracle ツールでは、ツールが適切にデータベースに接続するため、ユーザー名とパスワードをデータベース (USERS\$ 表) に格納する必要があります。RDB_NATCONN71.COM コマンド・ファイルで使用される ADD_USER コマンドにより、DBA またはデータベースに対する SELECT 権限を持つユーザーは、データベースの USERS\$ 表にユーザーと関連パスワードを追加できます。

```
@SYS$LIBRARY:RDB_NATCONN71 ADD_USER username new_password database
```

ADD_USER 操作には、次の 3 つのパラメータが関連しています。

- 新規ユーザー名
- 新規ユーザー名に関連するパスワード
- アタッチ先のデータベース

<new_password> に値を指定しないと、ユーザーは無効なパスワードで追加されます。ユーザーは、後でパスワードを変更できます。残りのパラメータを指定しない場合、コマンド・ファイルによりそれらのパラメータが要求されます。

注意：ADDUSER キーワードは非推奨であり、将来のリリースではサポートされません。

4.5.5 パスワードの変更

RDB_NATCONN71.COM コマンド・ファイルの MODIFY_USER コマンドを使用して、以前作成されたファイルに指定されたデータベースのリストに含まれる特定のユーザーのパスワードを変更します。

```
$ @SYS$LIBRARY:RDB_NATCONN71 MODIFY_USER username new_password old_password
```

RDB_NATCONN71.COM コマンド・ファイルの MODIFY_USER 操作を実行する前に、システム上のすべての Oracle Rdb データベースのリストを含むデータ・ファイルを作成する必要があります。コマンド・ファイルにより、このファイルが読み取られ、ユーザーがすでに入力されている各データベースでユーザーのパスワードが更新されます。デフォルト指定のファイルは、SYS\$MANAGER:SQLSRV_NATCONN_DBS.DAT です。

論理名 SQLSRV_NATCONN_DBS を定義して、データ・ファイルの場所と名前を上書きできます。ファイルが SYS\$MANAGER に存在しない場合、または論理名が有効なファイルを示していない場合、プログラムでは、現在のディレクトリで SQLSRV_NATCONN_DBS.DAT というファイルが検索されます。

データ・ファイルは、データベース名ごとに 1 行を含むテキスト・ファイルです。このファイルには、完全修飾されたデータベース・ファイル指定が含まれる必要があります。感嘆符 (!) を使用してコメントを指定できます。感嘆符に続く残りの行は無視されます。

MODIFY_USER 操作には、次の 3 つのパラメータが関連しています。

- ユーザー名
- 既存のユーザー名に関連付ける新規パスワード
- ユーザー名の旧パスワード

MODIFY_USER 操作を起動する場合、次の 4 つの使用例があります。最初の 2 つの使用例は、プログラムを実行する現在のユーザーに適用され、権限は必要とされません。次の 2 つの使用例は、システム管理者またはデータベース管理者に適用され、特別な権限が必要とされます。

使用例 1:

```
$ @SYS$LIBRARY:RDB_NATCONN71 MODIFY_USER " " new_password
```

この使用例は、ユーザーの OpenVMS パスワードが変更されているが、新規パスワードがまだユーザーのアクセスするデータベースに格納されていない場合に使用します。コマンド・ファイルが new_password パラメータのみを使用してコールされると、現在のプロセスのユーザー名が仮に使用されます。新規パスワードは、システムのユーザー認可ファイル (UAF) に対してチェックされます。有効である場合、パスワードは、このユーザーがすでに入力されている SQLSRV_NATCONN_DBS データ・ファイルにリストされた各データベースで変更されます。

使用例 2:

```
$ @SYS$LIBRARY:RDB_NATCONN71 MODIFY_USER " " new_password old_password
```

パラメータの組合せを使用して、OpenVMS システムの UAF と、ユーザーがアクセスするデータベースの両方でユーザーのパスワードを更新します。ユーザー名を指定しないと、現在のプロセスのユーザー名が仮に使用されます。旧パスワードは、システムの UAF に対してチェックされます。有効である場合、OpenVMS アカウントのパスワードは、新規パスワードに設定されます。次に、パスワードは、このユーザーがすでに入力されている SQLSRV_NATCONN_DBS データ・ファイルにリストされた各データベースで変更されます。

使用例 3:

```
$ @SYS$LIBRARY:RDB_NATCONN71 MODIFY_USER username new_password old_password
```

この使用例は、システム管理者またはデータベース管理者が、OpenVMS システムの UAF と、ユーザーがアクセスするデータベースの両方で自分以外のユーザーのパスワードを更新する際に使用できます。管理者は、この変更を行うために適切な権限 (BYPASS または SYSPRV など) を保持している必要があります。

旧パスワードが有効な場合、特定のユーザー名に対応する OpenVMS アカウントのパスワードは、新規パスワードを使用して更新されます。次に、パスワードは、このユーザーがすでに入力されている SQLSRV_NATCONN_DBS データ・ファイルにリストされた各データベースで変更されます。

使用例 4:

```
$ @SYS$LIBRARY:RDB_NATCONN71 MODIFY_USER username new_password ""
```

この使用例は、システム管理者またはデータベース管理者が、OpenVMS システムの UAF と、ユーザーがアクセスするデータベースの両方で、旧パスワードを検証せずに自分以外のユーザーのパスワードを更新する際に使用できます。特定のユーザー名に対応する OpenVMS アカウントのパスワードは、新規パスワードを使用して更新されます。管理者は、この使用例でパスワードを変更するために、適切な権限 (BYPASS または SYSPRV など) を保持している必要があります。次に、パスワードは、このユーザーがすでに入力されている SQLSRV_NATCONN_DBS データ・ファイルにリストされた各データベースで変更されます。

4.5.6 ユーザーの削除

RDB_NATCONN71.COM コマンド・ファイルの REMOVE_USER コマンドを使用して、指定したデータベースの USER\$ 表からユーザー名を削除します。

```
$ @SYS$LIBRARY:RDB_NATCONN71 REMOVE_USER username database
```

4.5.7 ユーザーの表示

RDB_NATCONN71.COM コマンド・ファイルの SHOW_USERS コマンドを使用して、指定したデータベースの USER\$ 表のすべてのユーザー名を表示します。

```
$ @SYS$LIBRARY:RDB_NATCONN71 SHOW_USERS database
```

4.6 キャラクタ・セットの定義

OCI Services for Oracle Rdb では、マルチバイト・キャラクタを含む複数のキャラクタ・セットがサポートされます。

表 4-1 に、サーバー・システムでサポートされるキャラクタ・セットを示します。現在のクライアント・システムでサポートされるキャラクタ・セットを確認するには、クライアント固有のドキュメントを参照してください。

表 4-1 に、OCI Services for Oracle Rdb でサポートされるキャラクタ・セットをリストします。

表 4-1 サポートされるキャラクタ・セット

サポートされる言語	Oracle キャラクタ・セット	Oracle Rdb キャラクタ・セット
ポルトガル語 (ブラジル)	.WE8DEC	DEC_MCS
フランス語 (カナダ)		
チェコスロバキア語		
デンマーク語		
オランダ語		
フィンランド語		
フランス語		
ドイツ語		
ギリシャ語		
ハンガリー語		
アイスランド語		
イタリア語		
スペイン語 (メキシコ)		
ノルウェー語		
ポーランド語		
ポルトガル語		
ロシア語		
スロバキア語		
スペイン語		
スウェーデン語		
トルコ語		
DEC-Hanzi	ZHS16CGB2312-80	HANZI
Dec-Kanji	JA16VMS	DEC_KANJI
韓国語	KO16KSC5601	KOREAN
拡張ヨーロッパ言語	WE8ISO8859P1	ISOLATIN1
拡張ヨーロッパ言語	WE8ISO8859P15	ISOLATIN9
日本語 Shift-JIS	JA16SJIS	SHIFT_JIS
ラテン / アラビア語 (ISO)	AR8ISO8859P6	ISOLATINARABIC
ラテン / キリル語 (ISO)	CL8ISO8859P5	ISOLATINCYRILLIC
ラテン / ギリシャ語 (ISO)	EL8ISO8859P7	ISOLATINGREEK
ラテン / ヘブライ語 (ISO)	IW8ISO8859P8	ISOLATINHEBREW
タイ語	TH8TISASCII	THAI
繁体字中国語	ZHT16BIG5	BIG5
Unicode	UTF8	UTF8
GB18030	ZHS32GBI1030	GB18030
ラテン / アラビア語 (8-bit)	AR8MSWIN1256	WIN_ARABIC
ラテン / キリル語 (8-bit)	CL8MSWIN1251	WIN_CYRILLIC
ラテン / ギリシャ語 (8-bit)	EL8MSWIN1253	WIN_GREEK
ラテン / ヘブライ語 (8-bit)	IW8MSWIN1255	WIN_HEBREW
西ヨーロッパ言語 (8-bit)	WE8MSWIN1252	WIN_LATIN1

デフォルトでは、OCI Services for Oracle Rdb では、(NLS_LANG 論理名を使用して US7ASCII として定義された) US7ASCII キャラクタ・セットがサポートされます。

4.6.1 サーバー・システムでのキャラクタ・セットの定義

サーバー・システムでは、NLS_LANG 論理名を使用して、デフォルトの US7ASCII キャラクタ・セット以外のキャラクタ・セットを定義します。NLS_LANG 論理名の書式は、次のとおりです。

```
$ define NLS_LANG "[<language>] [_<territory>].<character_set>"
```

一度セッションが開始したら、そのセッションのキャラクタ・セットは変更できません。

たとえば、西ヨーロッパ言語のすべての文字を提供する DEC_MCS を指定するには、サービスのプロセス初期化ファイルで NLS_LANG 論理名を .WE8DEC に定義します。

例 4-3 では、西ヨーロッパ言語キャラクタ・セットのみを指定します。

例 4-3 西ヨーロッパ言語キャラクタ・セットの指定

```
$ define NLS_LANG ".WE8DEC"
```

例 4-4 では、フランス語と地域、および西ヨーロッパ言語キャラクタ・セットを指定します。

例 4-4 フランス語と地域の指定

```
$ define NLS_LANG "FRENCH_FRANCE.WE8DEC"
```

注意： クライアント・システムでは、言語または地域を指定する必要があります。

別の方法として、次のように SQL 初期化ファイルで Oracle SQL ALTER SESSION 文を使用して、言語と地域を指定することも可能です。

```
ALTER SESSION SET NLS_LANGUAGE=<language> NLS_TERRITORY=<territory>
```

4.6.2 クライアント・システムでのキャラクタ・セットの定義

使用できるクライアント・システムとオペレーティング・システム・プラットフォームは多岐にわたるため、クライアント・システムにキャラクタ・セットを定義する際のすべての使用例をこの項で説明することはできません。クライアント・システムにキャラクタ・セットを指定する方法の詳細は、使用しているプラットフォーム固有のドキュメントを参照してください。

4.6.3 規則および推奨事項

キャラクタ・セットを指定する場合、次の規則と推奨事項に留意してください。

- キャラクタ・セットは、セッションに固有です。
- 特定のセッションに指定できるキャラクタ・セットは1つのみです。
データベースに送信される、またはデータベースからリクエストされるすべての文字データは、定義済みのキャラクタ・セットを使用しているとみなされます。
- US7ASCII 以外のキャラクタ・セットを使用する場合、クライアント・システムとサーバー・システムの両方で適切なキャラクタ・セットを指定します。

クライアント・システムのキャラクタ・セットとサーバーのキャラクタ・セットが一致しない場合、OCI Services for Oracle Rdb では、キャラクタ・セットの変換が試行されます。ただし、変換の結果は期待どおりにならない可能性があります。

4.7 データベース・リンクとしての Oracle Rdb データベースの参照

Oracle SQL/Services リリース 7.1.2 以上から、Oracle SQL を使用してリモート Oracle Rdb データベースへの接続を確立できます。データベース・リンクを定義するには、Oracle SQL CREATE DATABASE LINK 文を使用します。データベース・リンクを作成すると、データ・ディクショナリ表など、Oracle Rdb データベースの 1 つ以上の表を参照できます。Oracle Rdb データベースの表は、相互に結合することや、Oracle Database の表と結合することができます。次の項では、データベース・リンクを定義する方法について説明し、その機能を使用する際の制限事項をリストします。

4.7.1 CREATE DATABASE LINK の例

Oracle Rdb データベースに対するデータベース・リンクを定義するには、Oracle Database サーバーに接続して CREATE DATABASE LINK 文を使用します。次の行は、CREATE DATABASE LINK 文の構文を示しています。

```
CREATE DATABASE LINK <link-name>
  [CONNECT TO username IDENTIFIED BY password]
  USING <connect-name>;
```

コマンドラインで、次のように <link-name> および <connect-name> の名前を指定します。

- <link-name> パラメータには、Oracle SQL/Services サービスのサービス識別子を指定します。サービス名は、8 文字以下で構成されている必要があります。
- <connect-name> パラメータには、TNSNAMES.ORA ファイルの OCI Services for Oracle Rdb 接続文字列を指定します。接続文字列では、アタッチする 1 つ以上のデータベースを指定します。

例 4-5 は、サービス名が my_serv、接続文字列が my_conn である場合の TNSNAMES.ORA ファイルの例です。

例 4-5 TNSNAMES.ORA ファイル

```
my_conn = (DESCRIPTION = (ADDRESS =
  (PROTOCOL = TCP) (HOST = sqs_node) (PORT = 1527) )
  (CONNECT_DATA = (SERVICE = my_serv) )
```

注意： リリース 3.1.0.5.1 より前の Oracle Network Manager を使用した構成では、TNSNAMES.ORA ファイルで SERVICE エントリのかわりに SID (サービス識別子) が記載されていることがあります。次に例を示します。

```
(CONNECT_DATA = (SID = my_serv)
```

例 4-6 は、Oracle Rdb の my_serv データベース・サービスに対する接続を確立するための CREATE DATABASE LINK 文を示しています。

例 4-6 データベース・リンクの定義

```
CREATE DATABASE LINK my_serv
  [CONNECT TO username IDENTIFIED BY password]
  USING 'my_conn';
```

指定したサービス名が不適切な場合、データベース・リンクを使用すると次のエラー・メッセージが返されます。

ORA-2085: データベース・リンク <link_name> は <other_name> に接続します。

このエラー・メッセージで、変数 <other_name> は接続を試行したデータベースのサービス名ですが、この変数は <link_name> に指定された名前とは一致しません。

4.7.2 データベース・リンクの制限事項

この項では、データベース・リンク機能の制限事項について説明します。

- データベース・リンク名は、接続先のサービス名と同じである必要があります。データベース・リンク名は、8文字以下で指定します。
- リリース 7.1.6 以上から、OCI Services for Oracle Rdb では、OpenVMS システム上の XA ゲートウェイを使用して、任意の数の Oracle Rdb データベースを任意の数の Oracle Database と組み合わせて 2 フェーズ・コミット・トランザクションに参加させることができます。この機能を有効化するには、従来どおり ALTER SESSION SET TX_MODE NOWARN_1PC コマンドを発行する必要があります。
また、論理名 RDB\$DDTM_XG_INFO を次のように SYSTEM\$<nodename> に定義する必要があります。

```
$ DEFINE RDB$DDTM_XG_INFO SYSTEM$MYNODE
```

サービスにログオンして障害の発生したトランザクションをリカバリする場合は、次のように論理名 SQLNET_RECO_USER を定義して、リカバリ・プロセスで使用するユーザー名とパスワードを指定する必要があります。

```
$ DEFINE SQLNET_RECO_USER "SCOTT/TIGER"
```

- 標準の DML およびデータベース・リンクを使用して、Oracle Database のデータで Oracle Rdb データベースを更新することはできません。ただし、Oracle Database サーバーで PL/SQL 文を使用して、Oracle Rdb データベースを更新することは可能です。
- 副問合せを含む Oracle SQL UPDATE 文および DELETE 文には、制限があります。Oracle Rdb データベースの UPDATE または DELETE 文で参照されるすべての表は、Oracle Rdb データベースに属している必要があります。

例 4-7 に、有効な UPDATE 文を示します。

例 4-7 有効な Oracle SQL UPDATE 文

```
UPDATE emp@rdb SET sal = sal * 1.1
WHERE deptno=(SELECT deptmp FROM emp@rdb WHERE dname = 'RESEARCH');
```

例 4-8 に、機能しない UPDATE 文を示します。

例 4-8 無効な Oracle SQL UPDATE 文

```
UPDATE emp@rdb SET sal = sal * 1.1
WHERE deptno=(SELECT deptno FROM depts WHERE dname = 'RESEARCH');
```

この文は、Oracle Database から選択することで Oracle Rdb データベースの emp 表を更新しようとしているため、失敗します。例 4-8 の文を実行しようすると、Oracle サーバーにより次のエラーが返されます。

ORA-2025: SQL 文の中の表はすべてリモート・データベースにある必要があります

例 4-9 に、この問題の回避方法を示します。

例 4-9 DML が機能しない場合の PL/SQL の使用方法

```
SQL>
SQL> DECLARE CURSOR acur IS SELECT * FROM dept WHERE dname = 'RESEARCH';
 2   BEGIN
 3     FOR rec IN acur LOOP
 4       UPDATE emp@rdb SET sal = sal * 1.1 WHERE deptno = rec.deptno;
 5     END LOOP;
 6   END;
 7 /
```

PL/SQL procedure successfully completed.

```
SQL>
```

4.8 OCI Services for Oracle Rdb 表での SQL EXPORT/IMPORT の使用

リリース 7.1.0.4 より前の Oracle Rdb では、OCI Services for Oracle Rdb エンティティは EXPORT/IMPORT サイクルを通じて維持されませんでした。旧リリースの Oracle Rdb における EXPORT/IMPORT サイクルの後に OCI Services for Oracle Rdb の機能を維持するには、次の手順を実行します。

1. 現在の USER\$ 表のコピーを保存します。
\$ RMU/UNLOAD database USER\$ USER\$.UNL
2. データベースに DROP プロシージャを実行します。
\$ @SYS\$LIBRARY:RDB_NATCONN71 DROP db_name
3. データベースに EXPORT、DROP および IMPORT 操作を実行します。
4. PREPARE 操作を再度実行します。
\$ @SYS\$LIBRARY:RDB_NATCONN71 PREPARE db_name
5. USER\$ 表の以前のコピーをリストアします。
RMU/LOAD database USER\$ USER\$.UNL

Oracle Rdb リリース 7.1.0.4 以上では、ファンクションを参照するオブジェクトを IMPORT コマンドで作成できるため、OCI サービス・オブジェクトが維持されます。リリース 7.1.0.4 以上の Oracle Rdb を実行している場合、オブジェクトを削除して再作成する必要はありませんが、従来どおり EXPORT/IMPORT サイクル全体を実行する必要があります。

5

SQL ALTER SESSION 文

この章では、Oracle SQL ALTER SESSION 文を使用して、OCI Services for Oracle Rdb の操作の特定部分を制御する方法について説明します。

ALTER SESSION 文

OCI Services for Oracle Rdb では、次の目的で Oracle SQL ALTER SESSION 文を使用できます。

- National Language Support (NLS) パラメータの値の変更
- サーバー・ロギング・レベルの変更
- スキーマ・エミュレーション・モードの変更

OCI Services for Oracle Rdb では、Oracle サーバーでサポートされる ALTER SESSION SET NLS 制御のサブセットに対応しています。また、ALTER SESSION 文では、OCI Services for Oracle Rdb 環境に固有の制御がサポートされます。

環境

ALTER SESSION 文は次の場所で使用できます。

- Oracle SQL/Services の SQL 初期化ファイル
- コマンドライン (OCI クライアントに SQL コマンドライン・インタフェースがある場合)

書式

ALTER SESSION

```

SET ISOLATION LEVEL {READ COMMITTED | SERIALIZABLE}
SET NLS LANGUAGE=nls_value
SET NLS TERRITORY=nls_value
SET NLS DATE FORMAT=nls_value
SET NLS DATE LANGUAGE=nls_value
SET NLS NUMERIC CHARACTERS=nls_value
SET NLS ISO CURRENCY=nls_value
SET NLS CURRENCY=nls_value
SET NLS SORT=nls_value
SET SCHEMA EMULATION {STRICT | RELAXED}
SET SQLNET STRUCTURED DATE TYPES {ON | OFF}
SET CONSTRAINTS {IMMEDIATE | DEFERRED}
SET SQLNET TIMESTAMP DATE TYPE {ON | OFF}
SET TX MODE NOWARN_1PC

LOG {BRIEF | FULL | OFF}

```

引数

SET ISOLATION LEVEL READ COMMITTED SET ISOLATION LEVEL SERIALIZABLE

あるトランザクションの読取り操作が別の同時実行トランザクションの更新操作から受ける影響の程度を定義します。分離レベルは、読取り / 書込みトランザクションにのみ影響します。読取り専用トランザクションは、スナップショット・ファイルが有効化されている場合は常にスナップショット・ファイルから読取りを行います。

たとえば、ALTER SESSION 文で次のように SET ISOLATION LEVEL 制御を実装できます。

```

ALTER SESSION SET ISOLATION_LEVEL SERIALIZABLE
ALTER SESSION SET ISOLATION_LEVEL READ COMMITTED

```

SET ISOLATION LEVEL 引数は、Oracle Rdb SQL DECLARE TRANSACTION ISOLATION LEVEL 文と同義です。Oracle の分離レベルの詳細は、『Oracle Database SQL 言語リファレンス』を参照してください。Oracle Rdb の分離レベルの詳細は、『Oracle Rdb7 Guide to SQL Programming』を参照してください。

SET NLS keyword = nls_value

NLS パラメータの値を変更します。SET NLS のすべてのキーワードは、構文および意味ともに Oracle SQL 文と同一です。詳細は、『Oracle Database SQL 言語リファレンス』を参照してください。

SET SCHEMA EMULATION RELAXED**SET SCHEMA EMULATION STRICT**

スキーマ・エミュレーション・レイヤーとして、適度なものと厳格なものを選択できます。スキーマ・エミュレーション制御は、OCI Services for Oracle Rdb に固有です。

スキーマ・エミュレーション・レイヤーは、大部分は透過的です。ただし、あくまでもエミュレーション・レイヤーであり、Oracle マルチスキーマ・モデルの正確な実装ではないため、一部の OCI クライアントでは互換性の問題が発生することがあります。このため、OCI Services for Oracle Rdb では、次のように STRICT と RELAXED という 2 つのスキーマ・エミュレーション・モードを提供しています。

- STRICT スキーマ・エミュレーション・モード

これは、デフォルト・モードであり、現在のスキーマとは異なる明示的なスキーマを使用して作成された表およびビューが ORA_OBJECTS 表に記録されます。ORA_OBJECTS 表の各行により、データベース・オブジェクトと、その所属先のスキーマが定義されます。STRICT スキーマ・エミュレーション・モードで現在のスキーマの外部に表またはビューを作成するには、ORA_OBJECTS 表に対する書込みアクセス権を保持している必要があります。Oracle 環境では、現在のスキーマの外部にデータベース・オブジェクトを作成する場合、これらの権限が要求されるためです。

また、STRICT スキーマ・エミュレーション・モードでは、OCI Services for Oracle Rdb により、現在のスキーマ以外のスキーマを含む表およびビューに対する参照が検証されます。指定されたオブジェクトが指定されたスキーマに属していない場合、OCI Services for Oracle Rdb によりエラー状態が生成されます。

デフォルトでは、各表は、作成者であるユーザーの名前が付けられたスキーマに属します。

- RELAXED スキーマ・エミュレーション・モード

OCI Services for Oracle Rdb では、作成された表およびビューが ORA_OBJECTS 表に記録されません。すべての表およびビューは現在のスキーマ内に作成されます。ORA_OBJECTS に対する書込みアクセス権は必要ありません。OCI Services for Oracle Rdb では、スキーマを含む表およびビューに対する参照が検証されません。

デフォルトでは、各表は RDB_SCHEMA スキーマに属します。

SET SQLNET_STRUCTURED_DATE_TYPES ON | YES**SET SQLNET_STRUCTURED_DATE_TYPES OFF | NO**

OCI クライアントに正しいデータ型を返すことができます。次の表に、クライアントに返されるデータ型を示します。

Oracle Rdb データ型	クライアントに返されるデータ型	
	SQLNET_STRUCTURED_DATE_TYPES = ON または YES	SQLNET_STRUCTURED_DATE_TYPES = OFF または NO
DATE	DATE	DATE
TIME	DATE	DATE
TIMESTAMP	TIMESTAMP	DATE
INTERVAL YEAR TO MONTH	INTERVAL YEAR TO MONTH	CHAR
INTERVAL DAY TO SECOND	INTERVAL DAY TO SECOND	CHAR

この引数が ON または YES に設定され、OCI クライアントで構造化データ型が指定されると、クライアントに構造化データ型が返されます。

SET CONSTRAINTS IMMEDIATE**SET CONSTRAINTS DEFERRED**

制約設定を構成します。

IMMEDIATE を指定すると、そのトランザクション中は、DEFERRABLE INITIALLY DEFERRED として定義されたすべての制約が、DEFERRABLE INITIALLY IMMEDIATE と同様に評価されます。

DEFERRED を指定すると、DEFERRABLE INITIALLY DEFERRED として定義されたすべての制約が、制約定義で最初に指定されたとおりに評価されます。

SET SQLNET_TIMESTAMP_DATE_TYPE YES | ON**SET SQLNET_TIMESTAMP_DATE_TYPE NO | OFF**

OpenVMS で TIMESTAMP データ型を使用できます。YES または ON の値を指定すると、Oracle Rdb データ型の TIME または TIMESTAMP が TIMESTAMP 値として返されます。デフォルトの OFF のままにするか、引数を OFF または NO に設定すると、DATE データ型が返されます。

SET TX_MODE NOWARN_1PC

Oracle Database のデータベース・リンクから単一の Oracle Rdb データベースへの 2 フェーズ・コミットを可能にします。この引数により、OCI Services for Oracle Rdb に対して、分散トランザクションを結合し、Oracle Database でトランザクションを調整するよう指示します。XA を使用して複数の Oracle Rdb データベースで 2 フェーズ・コミットを実行する場合も、この引数が必要です。

LOG BRIEF**LOG FULL****LOG OFF**

Oracle SQL/Services ログ・ファイルへの情報のロギングを有効化または無効化します。OCI Services for Oracle Rdb のロギングを使用して、クライアントからサーバーに送信される Oracle SQL 文を確認できます。ALTER SESSION 引数は、OCI Services for Oracle Rdb に固有です。

デフォルト・モードでは、ログオン処理中に完全 (FULL) ロギングを実行します。ロギングは、セッションの残り部分ではデフォルトで OFF になります。ロギングを OFF にすると、OCI Services for Oracle Rdb では、サーバー・ログに処理情報が記録されません。

次のリストで、BRIEF および FULL ロギング・モードについて説明します。

- 簡易 (BRIEF) ロギング・モードでは、問題を診断し、OCI Services for Oracle Rdb によるクライアントとの対話方法を理解するのに必要とされる最も重要な情報のみが記録されません。簡易ロギングでは、次の情報がサーバー・ログに記録されます。
 - クライアントによりリクエストされたすべての SQL 文
 - OCI Services for Oracle Rdb による変更の実行後のクライアント SQL 文
 - サーバー・エラー・メッセージ
 - 行フェッチ数

ユーザーの問題を診断する場合や、クライアント・アプリケーションにより生成される Oracle SQL 文を適切に理解する場合は、一般的に BRIEF ロギングで十分です。

- FULL ログイングでは、OCI Services for Oracle Rdb ログ・ファイルに大量の情報が記録されます。FULL ログイングには、BRIEF ログイングに含まれるすべての情報に加え、次の情報も含まれます。
 - OCI Services for Oracle Rdb により生成および実行されたすべての内部 SQL 文
 - 解析、記述、実行などの OCI プロトコル・イベント
 - OCI Services for Oracle Rdb の内部キャッシュの再ロード
 - ORA_OBJECTS 表と RDB\$RELATIONS 表の同期
 - schema_name.object_name の参照を含むスキーマ・エミュレーション情報
 - OCI Services for Oracle Rdb と Oracle Rdb SQL 間の通信に使用される SQLDA 情報

OCI Services for Oracle Rdb で発生したクライアント側の問題を診断する必要がある場合、FULL ログイングは最も重要な情報ソースとなります。また、問題レポートをオラクル社に提出する必要がある場合、問題レポートに完全なセッション・ログ・ファイルを含める必要があります。

注意： サーバー・ログ・ファイルの形式と内容は、変更される可能性があります。

使用方法

- 現時点では、OCI Services for Oracle Rdb とマルチスキーマの Oracle Rdb データベースを組み合わせて使用することはできません。
- SQL/Services エグゼキュータ・ログ・ファイルに追加情報を記録するために、デバッグ・フラグを使用できます。デバッグ・フラグを有効化するには、SQL Services サービスのプロセス初期化ファイルで論理名 SQLNET_DEBUG_FLAGS を定義します。
- Oracle Rdb では ANSI マルチスキーマ・データベース・モデルがサポートされますが、OCI Services for Oracle Rdb を通じて OCI アプリケーションでアクセスされる Oracle Rdb データベースの大部分は、単一のスキーマ形式内に存在します。また、マルチスキーマ形式内に存在する一部の Oracle Rdb データベースは、標準的な Oracle 環境と互換性のあるスキーマ構成を含まないのが普通です。そのため、OCI Services for Oracle Rdb の提供するスキーマ・エミュレーション・レイヤーには、厳格なものと適度なものがあります。
 - スキーマ・エミュレーション・レイヤーにより、OCI クライアント・アプリケーションは、Oracle Database 内に一般的に存在するスキーマ構成を Oracle Rdb データベースが格納しているかのように、単一スキーマの Oracle Rdb データベースと連携して動作できます。スキーマ・エミュレーション・レイヤーにより、Oracle と類似した仮想スキーマ環境が実現します。
 - また、OCI Services for Oracle Rdb のデータ・ディクショナリにより、事前定義された Oracle スキーマおよびスキーマ・オブジェクトをエミュレートするビューと表が提供されます。ただし、異なるスキーマの異なる 2 つのデータベース・オブジェクトに同じ名前を使用することはできません (Oracle では可能です)。

Oracle 環境に最もよく似た方法で OCI クライアントにこの制限を適用するため、スキーマ・エミュレーション・レイヤーは、データベース内の各オブジェクトに対して現在のスキーマ内にプライベート・シノニムを暗黙的に定義します。Oracle 環境で現在のスキーマにプライベート・シノニムを定義する場合、OCI Services for Oracle Rdb と同じ一意の名前要件が適用されます。

- STRICT スキーマ・エミュレーション・モードを選択した場合、ログオン時または表（ビュー）の削除時に、OCI Services for Oracle Rdb では次のように ORA_OBJECTS 表と RDB\$RELATIONS 表の再同期が試行されます。
 - RDB\$RELATIONS にすでに存在しないデータベース・オブジェクトを ORA_OBJECTS が参照している場合、OCI Services for Oracle Rdb は、ORA_OBJECTS から不要なレコードを削除することで ORA_OBJECTS を訂正しようとします。
 - ORA_OBJECTS に対する書き込みアクセス権を持っていない場合（スキーマ外部のオブジェクトを削除するための Oracle 権限を持たない場合と同じ）、OCI Services for Oracle Rdb では、エラー状態が生成され、現在のトランザクションがロールバックされます。
 - 表またはビューの削除を試みた場合、削除するのに十分な権限は付与されていないため、ロールバックの効果によりその表またはビューはリストアされます。
 - ログオンしようとする、そのログオン試行は失敗します。

この状態が発生するのは、ORA_OBJECTS 表またはデータベース・オブジェクトが OCI Services for Oracle Rdb の外部で操作された場合のみです。この状態は、ORA_OBJECTS 表に対する書き込みアクセス権を持つユーザーがシステムにログオンした時点で自動的に解決されます。

6

管理コマンド

この章では、Oracle SQL/Services の SQLSRV_MANAGE ユーティリティの構文とセマンティクスについて説明します。このユーティリティを使用して、Oracle SQL/Services サーバーとそのコンポーネントを管理します。構文表記規則の詳細は、[6.1 項](#)を参照してください。

このユーティリティの使用方法の詳細は、Oracle SQL/Services Manager GUI のヘルプを参照してください。

SQLSRV_MANAGE コマンドには、管理コマンド、環境コマンドおよびスイッチが含まれます。[6.2 項](#)では、SQLSRV_MANAGE の管理コマンドの動作方法について説明します。

6.1 構文表記規則

SQLSRV_MANAGE ユーティリティでは、その環境コマンドと管理コマンドの両方で次の構文表記規則およびセマンティクスを使用します。

[]	大カッコは、カッコ内の項目を任意に選択することを表します。実際のオプションには大カッコを含めないでください。
{ }	中カッコは、カッコ内の項目から少なくとも1つを選択する必要があることを表します。実際のオプションには中カッコを含めないでください。
	縦線は、示されている項目から1つのみを選択できることを表します。
,	カンマは、示されている項目を任意の数だけ選択できることを表します。コマンドの一部としてカンマを入力し、選択した項目を区切ります。
<>	山カッコで囲まれた部分は、ユーザーが指定する名前です。
::=	2つのコロんと等号が後に続く引数は、引数の定義を表します。
空白と新規行	空白と新規行（改行）は、構文図では意味を持ちません。
キーワード	キーワードでは、大/小文字は区別されません。キーワードは大文字で記載され、下線が引かれています。
...	コマンドの中の水平の省略記号は、例に直接関連しないコマンドの一部が省略されていることを示します。
;	すべての文は、セミコロン (;) で終了する必要があります。ただし、EXIT および HELP コマンドは例外で、セミコロンはオプションです。

次の構文とセマンティクスも使用されます。

<identifier>

<identifier> は、文字で始まり、文字 (a ~ z, A ~ Z)、数字 (0 ~ 9)、ハイフン (-) およびアンダースコア (_) で構成されます。次に例を示します。

```
AARDVARK_1-1101
```

<quoted-string>

<quoted-string> では、一重引用符または二重引用符を使用して、改行文字以外の任意の文字を含めることができます。次に例を示します。

```
'user1'
"Today is 5/6/94"
```

一重引用符で囲まれた文字列には、二重引用符を含めることができます。また、二重引用符で囲まれた文字列には、一重引用符を含めることができます。次に例を示します。

```
'Contestant number three said, "My name is Data"'
"Today's beach report is 'sunny and warm'"
```

文字列の内側にある改行文字は、構文エラー（終了していない引用符付き文字列）とみなされます。

引用符付き文字列は、数字で始まる文字列を表す場合にも役立ちます。次に例を示します。

```
'71_user'
```

<number>

<number> は整数です。これは、プラス記号またはマイナス記号で始めることが可能で、0 ~ 9 の1つ以上の数字で構成されます。この数値は、10進形式または16進形式で表現できます。数値を16進形式で表現するには、数値の前に値0x または0X を付けます。次に例を示します。

```
0x0000088a
```

<version-data-type>

<version-data-type> は、メジャーおよびマイナー・バージョン番号を含むソフトウェア・バージョン番号であり、小数点で区切られた 0～9 の 1 つ以上の数字で構成されます。小数点の左側がメジャー・バージョン番号で、右側がマイナー・バージョン番号です。構文は次のとおりです。

```
n[nnn...].n[nnn...]
```

次に例を示します。

```
7.1
6.10
```

コメント

コメントは、2 つの連続するハイフン (--) で開始し、次の新規行に続きます。次に例を示します。

```
-- This is a comment line.
```

コマンド引数の順序

管理コマンドのコマンド引数の順序は、重要ではありません。2 つ以上の引数を含むコマンドを入力する場合、各引数は、そのコマンドの書式説明に記載されているとおりに並べる必要はありません。

引数のキーワード間でのアンダースコアの使用

コマンドライン・インタフェースでは、各引数のキーワード間のアンダースコアを空白で置換できます。たとえば、コマンドラインでは、アンダースコアで区切られた NETWORK_PORT という 2 つのキーワードを入力せずに、空白で区切られた NETWORK PORT というキーワードを入力できます。SQLSRV_MANAGE では、これら 2 つのキーワードは正しく解析され、エラーは返されません。

SQL 初期化ファイル

SQL 初期化ファイルでは、次の構文表記規則が使用されます。

- 行の先頭の空白と末尾の空白は無視されます。
- 2 つの連続するハイフン (--) で始まるコメントは、行の先頭から開始し、次の改行まで続く必要があります。
- 各 SQL 初期化文は、SQL EXECUTE IMMEDIATE 文によって動的に準備、実行および解放できる必要があります。詳細は、『Oracle Rdb SQL リファレンス・マニュアル』の EXECUTE IMMEDIATE 文を参照してください。
- SQL 文は、複数の行にまたがることはできません。
- SQL 文の末尾のセミコロン (;) は無視されるため、対話型 SQL を使用して SQL 初期化ファイルを起動および検証できます。

次の例は、サンプルの SQL 初期化ファイルを示しています。

```
--
-- This SQL initialization file sets the SQL dialect and default
-- character set for an executor process.
--
SET DIALECT 'SQL89';
SET DEFAULT CHARACTER SET 'KANJI';
```

6.2 SQLSRV_MANAGE コマンドの動作方法

この項では、SQLSRV_MANAGE コマンドの動作方法について説明します。

サーバー構成コマンド

次のコマンドは、サーバー構成においてサーバー、ディスパッチャおよびサービス・オブジェクトに対して動作します。

- ALTER SERVER、CONNECT TO SERVER、CREATE SERVER、DISCONNECT SERVER、DROP SERVER、EXTRACT、RESTART SERVER、SET CONFIG_FILE、SET CONNECTION、SHOW SERVER、SHOW SETTINGS、SHOW VERSION、SHUTDOWN SERVER、START SERVER
- ALTER DISPATCHER、CREATE DISPATCHER、DROP DISPATCHER、SHOW DISPATCHER、SHUTDOWN DISPATCHER、START DISPATCHER
- ALTER SERVICE、CREATE SERVICE、DROP SERVICE、GRANT USE ON SERVICE、KILL EXECUTOR、REVOKE USE ON SERVICE、SHOW CLIENTS FOR SERVICE、SHOW SERVICE、SHUTDOWN SERVICE、START SERVICE

環境に使用するコマンドおよびスイッチ

次のコマンドは、SQLSRV_MANAGE システム管理環境に対して動作します。

- -input スイッチと -output スイッチ
- SHOW CONNECT[ION]、SHOW SETTINGS
- CONNECT [TO] SERVER、DISCONNECT SERVER、SET CONFIG_FILE、SET CONNECTION
- @、CLOSE、EXIT、HELP、OPEN、SET CONFIRM、SET OUTPUT、SET VERIFY

表 6-1 に、Oracle SQL/Services オブジェクトの 3 つの異なるグループを示し、各オブジェクトにコマンド動詞のセットがどのように影響するかについて説明します。

表 6-1 Oracle SQL/Services オブジェクトとコマンドによる各オブジェクトへの影響

オブジェクト	コマンド	説明
ディスパッチャ	ALTER	構成ファイルのディスパッチャ・オブジェクト定義を変更し、実行中のサーバーの選択された属性を動的に変更します。
	CREATE	現在のサーバーにディスパッチャ・オブジェクトを作成し、構成ファイルに定義を追加します。
	DROP	現在のサーバーのアクティブではないディスパッチャに対応するディスパッチャ・オブジェクト定義を構成ファイルから削除します。
	SHOW	ディスパッチャ・オブジェクト定義を表示します。
	SHUT[DOWN]	指定されたディスパッチャ・オブジェクトを停止します。
	START	現在のサーバーに定義されているディスパッチャ・オブジェクトのディスパッチャ・プロセスを起動します。
サーバー	ALTER	構成ファイルのサーバー・オブジェクト定義を変更し、実行中のサーバーの選択された属性を動的に変更します。
	CONNECT	実行中のサーバーに接続します。
	CREATE	構成ファイルとサーバー・オブジェクトを作成します。

表 6-1 Oracle SQL/Services オブジェクトとコマンドによる各オブジェクトへの影響 (続き)

オブジェクト	コマンド	説明
	DISCONNECT	実行中のサーバーとの接続を切断します。
	DROP	サーバー・オブジェクト定義を削除し、アクティブではないサーバーの構成ファイルを削除します。
	EXTRACT	サーバー、ディスパッチャおよびサービスの定義を抽出し、それらを SQL/Services コマンド・スクリプトに書き込みます。
	RESTART	現在のサーバー・オブジェクトで自動的に起動されたすべてのディスパッチャおよびサービスとともにサーバーを再起動します。
	SET CONFIG_FILE	後続のコマンドでサーバーの構成ファイルを変更できるように現在の構成を設定します。
	SET CONNECTION	指定された接続名を使用してサーバー・オブジェクトに対する接続を設定します。
	SHOW	サーバー・オブジェクト定義を表示します。
	SHOW SETTINGS	現在の構成ファイルを表示します。
	SHOW VERSION	SQLSRV_MANAGE 管理クライアントのバージョンを表示します。
	SHUT[DOWN]	現在のサーバー・オブジェクトを停止します。
	START	現在のサーバー・オブジェクトで自動的に起動されたすべてのディスパッチャおよびエグゼキュタ・プロセスとともにサーバーを起動します。
サービス	ALTER	構成ファイルのサービス・オブジェクト定義を変更し、実行中のサービスの選択された属性を動的に変更します。
	CREATE	サービス・オブジェクトを作成し、構成ファイルに定義を追加します。
	DROP	アクティブではないサービスの構成ファイルからサービス・オブジェクト定義を削除します。
	GRANT USE ON	ユーザー名または識別子にサービス・オブジェクトの USE 権限記述子を付与します。
	KILL EXECUTOR	エグゼキュタ・プロセスを中断します。
	REVOKE USE ON	ユーザー名または識別子からサービス・オブジェクトの USE 権限記述子を取り消します。
	SHOW CLIENTS	サービスのアクティブなユーザーを表示します。
	SHOW	すべてのユーザー名および識別子に対応するサービス・オブジェクトの USE 権限記述子を含むサービス・オブジェクト定義を表示します。
	SHUT[DOWN]	指定されたサービス・オブジェクトを停止します。
	START	指定されたサービス・オブジェクトを起動します。

表 6-2 に、SQLSRV_MANAGE の環境コマンドおよびスイッチを示します。

表 6-2 SQLSRV_MANAGE の環境コマンドおよびスイッチ

コマンドまたはスイッチ	説明
-input スイッチ	SQLSRV_MANAGE ユーティリティが入力を読み取る入力ファイルの名前を指定します。
-output スイッチ	SQLSRV_MANAGE ユーティリティが出力を書き込む出力ファイルの名前を指定します。
@	間接コマンド・ファイルを実行します。
CLOSE	出力ファイルを閉じます。
CONNECT [TO] SERVER	実行中のサーバーに接続します。
DISCONNECT SERVER	実行中のサーバーとの接続を切断します。
EXIT	SQLSRV_MANAGE ユーティリティを終了します。
HELP	任意のトピックに関するヘルプを表示します。
OPEN	出力ファイルを開きます。
SET CONFIG_FILE	後続のコマンドでサーバーの構成ファイルを変更できるように現在の構成を設定します。
SET CONFIRM	特定の管理操作の確認を要求します。
SET CONNECTION	サーバーに対する現在の接続を、確立済の接続のグループに含まれる別の接続に変更します。
SET OUTPUT	有効化されると、デフォルト・デバイスに出力を行います。
SET VERIFY	読取り時に、デフォルトの出力デバイスにコマンド・ファイルの入力を表示します。
SHOW CONNECT[ION]	現在のサーバー・オブジェクトに関する情報と、サーバーに対して SQLSRV_MANAGE が保持するアクティブな接続をすべて表示します。
SHOW SETTINGS	検証および出力設定に関する情報を表示します。
SHOW VERSION	SQLSRV_MANAGE 管理クライアントのバージョンを表示します。

-input スイッチ

SQLSRV_MANAGE ユーティリティが入力を読み取る入力ファイルの名前を指定します。

書式

```
-i[[n[put]]] <file-spec>
```

<file-spec> ::= <identifier> または <quoted-string>

引数

<file-spec>

入力ファイル名。ファイル名は、識別子または引用符付き文字列として表します。

使用方法

- -i および -in は、-input コマンドと同義です。
- SQLSRV_MANAGE ユーティリティでは、入力は要求されません。指定のファイルが完全に読み取られると、このユーティリティは終了します。
- SQLSRV プロンプトでは、-input スイッチを入力できません。

例

例 1: SQLSRV_MANAGE ユーティリティが入力を読み取る入力ファイルを指定します。

```
$ sqlsrv_manage := $SYS$SYSTEM:sqlsrv_manage71  
$ sqlsrv_manage -input sqlsrv_create.sqs
```

-output スイッチ

SQLSRV_MANAGE ユーティリティが出力を書き込む出力ファイルの名前を指定します。

書式

```
-o[ut[put]] <file-spec>
```

<file-spec> ::= <identifier> または <quoted-string>

引数

<file-spec>

出力ファイル名。ファイル名は、識別子または引用符付き文字列として表します。

使用方法

- -o および -out は、-output スイッチと同義です。
- SQLSRV_MANAGE ユーティリティでは、CLOSE または OPEN コマンドが実行されるまで指定のファイルにすべての出力が書き込まれます。CLOSE コマンドが発行されると、後続の出力は標準出力に送られます。OPEN コマンドが発行されると、出力は新規出力ファイルに送られます。
- SQLSRV プロンプトでは、-output スイッチを入力できません。

例

例 1: SQLSRV_MANAGE ユーティリティが出力を書き込む出力ファイルを指定します。

```
$ sqlsrv_manage -output out_testfile
```

@ コマンド

SQLSRV_MANAGE 環境で間接コマンド・ファイルを実行します。

書式

```
@ <file-spec>;  
  
<file-spec> ::= <identifier> または <quoted-string>
```

引数

<file-spec>

間接コマンド・ファイル名。ファイル名は、識別子または引用符付き文字列として表します。

使用方法

実行されると、間接コマンド・ファイルが開かれ、構文エラーが発生するかファイル内に文字がなくなるまでそのファイルから入力取得されます。

例

例 1: test_file.sqs という間接スクリプトを実行します。大 / 小文字を区別することが重要な場合、引用符付き文字列を使用します。

```
SQLSRV> @ 'test_file.sqs';
```

ALTER DISPATCHER コマンド

現在のサーバーのみを対象にディスパッチャ・オブジェクト定義を変更します。ディスパッチャ定義の変更は、構成ファイルに格納されます。オフラインでのディスパッチャの変更は、実行中のサーバーに影響しません。オンラインでのディスパッチャの変更は、その変更が動的属性を対象とする場合、実行中のサーバーに影響します。それ以外の場合、ディスパッチャの変更を反映するには、ディスパッチャを停止して再び起動するか、サーバーを再起動する必要があります。

書式

```
ALTER DISPATCHER <disp-name>
->[ AUTOSTART { ON | OFF } ]
->[ MAX_CONNECTIONS <number> ]
->[ IDLE_USER_TIMEOUT <number-in-seconds> ]
->[ MAX_CLIENT_BUFFER_SIZE <number> ]
->[ DUMP_PATH <directory-specification> ]
->[ LOG_PATH <directory-specification> ]
->[ <network-port-spec> ] ...;

<disp-name> ::= <identifier>
<network-port-spec> ::= NETWORK_PORT <transport-spec>
    PROTOCOL { NATIVE | OCI | SQLSERVICES | JDBC }
<transport-spec> ::= { <tcp-spec> | <decnet-spec> | <ipxspx-spec>
    | <sqlnet-spec> }
<tcp-spec> ::= TCPIP [ PORT_ID <number> ]
<decnet-spec> ::= DECNET [ OBJECT { <number> | <identifier>
    | <quoted-string> } ]
<ipxspx-spec> ::= IPXSPX [ <PORT_ID <number> ]
<sqlnet-spec> ::= SQLNET_LISTENER_NAME { <identifier>
    | <quoted-string> }
```

引数

<disp-name>

ディスパッチャ名。ディスパッチャ名は、識別子として表します。

AUTOSTART { ON | OFF }

START SERVER または RESTART SERVER コマンドを発行する場合、ディスパッチャ・オブジェクトを自動的に起動するかどうかを決定します。引数を ON に指定すると、START SERVER または RESTART SERVER コマンドを発行したときに、ディスパッチャ・オブジェクトが自動的に起動されます。デフォルトは ON です。

MAX_CONNECTIONS <number>

ディスパッチャで対応するクライアントからのネットワーク接続の最大数を指定します。接続の最大数は、整数として表します。デフォルトは 100 です。オペレーティング・システム構成、ネットワーク構成および共有サーバー・メモリー以外に上限はありません。

IDLE_USER_TIMEOUT <number-in-seconds>

ディスパッチャがクライアントとの接続を切断する前に、クライアント（ユーザー）がアイドル状態を継続できる時間を秒単位で指定します。<number-in-seconds> の値は、整数として表します。デフォルト値は 0 であり、SHOW DISPATCHER コマンドでは <none> として表示されます。これは、アイドル・タイムアウト値が無制限であることを示します。0 以外の値を指定すると、その値は次に大きい 90 秒の倍数に丸められます。これは動的属性であり、変更するとその内容は即座に反映されます。

MAX_CLIENT_BUFFER_SIZE <number>

許可する最大クライアント・バッファ・サイズを指定します。使用可能な最大クライアント・バッファ・サイズは、32,000 バイトです。クライアント・アプリケーションで最大値より大きいバッファ・サイズを指定すると、Oracle SQL/Services クライアント API により、ディスパッチャに指定された最大サイズにバッファ・サイズが調整されます。

MAX_CLIENT_BUFFER_SIZE 属性に指定可能な最小値（デフォルト値）は、5000 バイトです。

DUMP_PATH <directory-specification>

バグチェック・ダンプ・ファイルのディレクトリ名を指定します。

LOG_PATH <directory-specification>

ログ・ファイルのディレクトリ名を指定します。

LOG_PATH 引数のディレクトリ名のかわりに NOLOG を指定すると、ログ・ファイルは書き込まれません。

<network-port-spec>

ディスパッチャでクライアントとの通信に使用するネットワーク・ポートをリストします。ネットワーク・ポートは、TCP/IP、DECnet、IPX/SPX および Oracle Net のうち、1 つまたは任意の組合せを指定します。TCP/IP のデフォルト・ポート ID は 118、デフォルト DECnet オブジェクトは 81、IPX/SPX のデフォルト・ポート ID は 33969 (0x84b1) です。ネットワーク・ポートを指定しないと、ディスパッチャではデフォルト・ポートが使用されます。

<network-port-spec> 引数は、OCI Services for Oracle Rdb の複数のリスナー名を含めるように繰り返すことができます。ALTER DISPATCHER コマンドで指定できる <network-port-spec> 引数の最大数は、5 です。

この引数により、各ディスパッチャ・ネットワーク・ポートでサポートできるメッセージ・プロトコルも決定されます。ディスパッチャ・ネットワーク・ポートでサポートできるメッセージ・プロトコルは 1 つのみです。ディスパッチャ・ネットワーク・ポートでサポートする予定のクライアントのタイプに一致するメッセージ・プロトコルを指定します。

- NATIVE

Oracle RMU クライアント (Oracle Rdb Performance Monitor、Oracle RMUwin、Parallel Backup Monitor)

- OCI

Oracle Call Interface (OCI) を使用する Oracle クライアント、または Oracle Enterprise Manager クライアント

- SQLSERVICES

Oracle SQL/Services クライアント (Query Performance Tuner、Oracle ODBC Driver for Rdb、Oracle SQL/Services、Oracle Rdb Schema Manager)

- JDBC

JDBC for Oracle Rdb クライアント

注意： 使用する予定の各サービス API がサポートされるディスパッチャ・メッセージ・プロトコルを使用して、ディスパッチャ・ネットワーク・ポートを定義しておく必要があります。たとえば、OCI API をサポートするサービスと、SQLSERVICES API をサポートする別のサービスを定義する場合、OCI ディスパッチャ・メッセージ・プロトコルおよび Oracle Net トランスポートをサポートする少なくとも 1 つのディスパッチャ・ネットワーク・ポートと、SQLSERVICES ディスパッチャ・メッセージ・プロトコルおよび使用可能な任意のトランスポートをサポートする別のディスパッチャ・ネットワーク・ポートをそれぞれ定義する必要があります。

使用方法

- 一般的に、ALTER DISPATCHER 定義に指定された句により、前にディスパッチャ定義に指定された項目が置き換えられます。つまり、ALTER DISPATCHER コマンドに句を指定すると、その句の指定項目は、定義で変更されます。句が指定されない場合、その句に対応する指定項目は変更されません。
- ネットワーク・ポートを変更すると、ネットワーク・ポートの指定全体が置き換えられます。つまり、ALTER DISPATCHER コマンドで既存のリストにネットワーク・ポートを追加できますが、他のネットワーク・ポートの指定もすべて再設定し、それらを構成ファイルに維持する必要があります。
- Oracle Net トランスポート・オプションを使用するには、<transport-spec> 引数に Oracle Net トランスポート・オプションを <sqlnet-spec> として指定し、OCI Services for Oracle Rdb のリスナー名をその <identifier> 引数として指定します。
- LISTENER という語は、キーワード LISTENER_NAME と同義です。
- SQLSRV_MANAGE では、同じポート ID またはオブジェクトを対象にリスニングを行う 2 つ以上のディスパッチャを作成できますが、起動できるのは、一意のポート ID またはオブジェクトが割り当てられた最初のディスパッチャのみです。同じポート ID またはオブジェクトでリスニングを行う 2 番目のディスパッチャを起動しようとしても、指定したどのネットワーク・ポートもリスニングできなければ、起動に失敗します。
- ログ・ファイルに I/O エントリが大量に出力されるという問題が発生しなければ、トラブルシューティング目的でログ・ファイルを作成することをお勧めします。LOG PATH および DUMP PATH 引数のデフォルト・ディレクトリは、SYS\$MANAGER です。
- LOG PATH および DUMP PATH 引数は、SQLSRV_MANAGE コマンド・ユーティリティの使用時には使用できますが、Oracle SQL/Services Manager GUI の使用時には使用できません。sqlsrv_manage サービスが実行中の場合、Oracle SQL/Services Manager GUI を使用すると、ログ・ファイルとバグチェック・ダンプ・ファイルを表示できます。
- 既存の論理名 SQLSRV_DISP_LOGPATH および SQLSRV_DISP_DUMP_PATH に値を割り当てると、それらの値により、LOG PATH および DUMP PATH 引数で指定されたログ・パス値とダンプ・パス値が上書きされます。

例

例 1: アイドル・ユーザー・タイムアウト値を動的に変更します。

```
SQLSRV> ALTER DISPATCHER tcpip_disp IDLE_USER_TIMEOUT 180;
```

例 2: OCI Services for Oracle Rdb プロトコルを使用するようにディスパッチャを変更します。このコマンドにより、このディスパッチャに対応する他のすべてのポートが削除されます。このディスパッチャに以前定義されたネットワーク・ポートが失われることを防ぐため、既存のすべてのネットワーク・ポートを再指定する必要があります。

```
SQLSRV> ALTER DISPATCHER OCI_disp NETWORK_PORT SQLNET LISTENER_NAME "OCI-LISTENER";
%DBS-S-ALTER_RESTART, Restart object to have altered settings take effect
SQLSRV> SHUTDOWN DISPATCHER OCI_disp;
SQLSRV> START DISPATCHER OCI_disp;
```

例 3: ログ・パスおよびダンプ・パスを指定します。

```
SQLSRV> ALTER DISPATCHER SQLSRV_DISP1 LOG_PATH 'USER1:[SQLSRV_TEST1.AAA]'
SQLSRV> DUMP_PATH 'USER1:[SQLSRV_TEST2.BBB]';
%SQLSRV-S-ALTER_RESTART, Restart object to have altered settings take effect
```

ALTER SERVER コマンド

サーバー・オブジェクト定義を変更します。サーバー定義の変更は、構成ファイルに格納されます。オフラインでのサーバーの変更は、実行中のサーバーに影響しません。オンラインでのサーバーの変更は、その変更が動的属性を対象とする場合、実行中のサーバーに影響します。それ以外の場合、変更を反映するには、サーバーを停止して再び起動するか、サーバーを再起動する必要があります。

書式

ALTER SERVER

```

->[ MAX_SHARED_MEMORY_SIZE <number> ]
-->[DUMP_PATH <directory-specification>]
-->[LOG_PATH <directory-specification>]
->[ PROCESS_STARTUP_TIMEOUT <number-in-seconds> ]
->[ PROCESS_SHUTDOWN_TIMEOUT <number-in-seconds> ]
->[ <network-port-spec> ]...;

<network-port-spec> ::= NETWORK_PORT <transport-spec>
<transport-spec> ::= { <tcp-spec> | <decnet-spec> }
<tcp-spec> ::= TCPIP [ PORT_ID <number> ]
<decnet-spec> ::= DECNET [ OBJECT { <number> | <identifier>
| <quoted-string> } ]

```

引数

MAX_SHARED_MEMORY_SIZE <number>

サーバーで使用される最大共有メモリーのサイズを KB 単位で指定します。値を変更すると、その値はモニターの起動時に最大共有メモリー・サイズになります。デフォルト値は、2000KB (2MB) です。Oracle SQL/Services では、モニターの起動時に最大共有メモリー・サイズが割り当てられます。

DUMP_PATH <directory-specification>

バグチェック・ダンプ・ファイルのディレクトリ名を指定します。

LOG_PATH <directory-specification>

ログ・ファイルのディレクトリ名を指定します。

LOG_PATH 引数のディレクトリ名かわりに NOLOG を指定すると、ログ・ファイルは書き込まれません。

PROCESS_STARTUP_TIMEOUT <number-in-seconds>

ディスパッチャまたはエグゼキュータ・プロセスが起動しないと判断し、モニターがアクションを開始してプロセスを終了するまでに待機する時間を秒単位で指定します。デフォルト値は 0 秒であり、プロセス起動タイマー値は未設定です。これは動的属性であり、変更するとその内容は即座に反映されます。詳細は、「使用方法」を参照してください。

PROCESS_SHUTDOWN_TIMEOUT <number-in-seconds>

ディスパッチャまたはエグゼキュータ・プロセスが停止しないと判断し、モニターがアクションを開始してプロセスを終了するまでに待機する時間を秒単位で指定します。デフォルト値は 0 秒であり、プロセス停止タイマー値は未設定です (プロセス停止タイマー値は無制限です)。

これは動的属性であり、変更するとその内容は即座に反映されます。詳細は、「使用方法」を参照してください。

<network-port-spec>

モニターで Oracle SQL/Services 管理クライアント (SQLSRV_MANAGE および Oracle SQL/Services Manager GUI) との通信に使用するネットワーク・ポートをリストします。ネットワーク・ポートの指定は、TCP/IP または DECnet です。TCP/IP のデフォルト・ポート ID は 2199、デフォルト DECnet オブジェクト名は SQLSRV_SERVER です。ネットワーク・ポートを指定しないと、サーバーのモニターではデフォルト・ポートが使用されます。ALTER SERVER コマンドで指定できる <network-port-spec> 引数の最大数は、5 です。ネットワーク・ポートを変更すると、ネットワーク・ポートの指定全体が置き換えられます。

DECnet または TCP/IP は、ALTER SERVER 定義が使用されるノードで使用可能である必要があります。どちらも使用できない場合、サーバーは実行されません。

使用方法

- サーバー定義は、CONNECT [TO] SERVER コマンドを使用してオンラインで変更できます。SET CONFIG_FILE コマンドを使用してその構成ファイルを選択すると、オフラインで変更できます。動的属性をオンラインで変更すると、その内容は即座に反映されます。動的属性以外の属性をオンラインで変更すると、変更した設定を反映するにはサーバーを再起動する必要があるというステータス・メッセージが返されます。管理セッションを完了した後は、Oracle SQL/Services サーバーの全体的な整合性を確保するため、実行中のサーバーを即座に再起動することをお勧めします。(実行中のサーバーを再起動するには、RESTART SERVER コマンドを発行します。)
- 一般的に、ALTER SERVER 定義に指定された句により、前にサーバー定義に指定された項目が置き換えられます。つまり、ALTER SERVER コマンドに句を指定すると、その句の指定項目は、定義で変更されます。句が指定されない場合、その句に対応する指定項目は変更されません。
- プロセス起動タイマーおよび停止タイマーを設定する場合、次のガイドラインに従ってください。
 - 通常、ディスパッチャとエグゼキュータ・プロセスは、合理的な時間間隔で起動および停止します。特別な状況にある場合にのみ、PROCESS_STARTUP_TIMEOUT および PROCESS_SHUTDOWN_TIMEOUT 引数に 0 (ゼロ) 以外の値を指定する必要があります。
 - 負荷の高いシステムの場合、通常は特定の操作が完了するまで時間がかかります。プロセス起動タイマーまたはプロセス停止タイマーを 0 (ゼロ) 以外の値に設定しているときに、明確な理由もなく障害が発生した場合 (他に考えられる原因をチェックしたが、問題を特定できない場合)、PROCESS_STARTUP_TIMEOUT 引数または PROCESS_SHUTDOWN_TIMEOUT 引数により大きな値を設定して、問題が解決するかどうかを確認してください。
- SQLSRV_MANAGE ユーティリティでは、デフォルトの TCP/IP または DECnet ポートを使用してサーバーのモニターに接続しようと試みます。サーバーのネットワーク・ポートを変更する場合、SQLSRV_MANAGE ユーティリティからの接続時に、そのポートを明示的に指定する必要があります。
- ネットワーク・ポートを変更すると、ネットワーク・ポートの指定全体が置き換えられます。つまり、ALTER SERVER コマンドで既存のリストにネットワーク・ポートを追加できますが、他のネットワーク・ポートの指定もすべて再設定し、それらを構成ファイルに維持する必要があります。
- 同じポート ID を複数回指定すると、エラーが返されます。

例

例 1: オンラインでサーバー定義を変更します。

```
SQLSRV> CONNECT SERVER;
Connecting to server ...
Connected
SQLSRV> ALTER SERVER MAX_SHARED_MEMORY_SIZE 4000;
%DBS-S-ALTER_RESTART, Restart object to have altered settings take effect
SQLSRV> RESTART SERVER;
Disconnected from Server
SQLSRV> CONNECT SERVER;
Connecting to server ...
Connected
```

例 2: オフラインでサーバー定義を変更します。

```
SQLSRV> SET CONFIG_FILE 'my_config_file';
SQLSRV> ALTER SERVER MAX_SHARED_MEMORY_SIZE 4000;
SQLSRV> RESTART SERVER;
Connecting to server ...
Connected
```

ALTER SERVICE コマンド

現在のサーバーのみを対象にサービス・オブジェクト定義を変更します。サービス定義の変更は、構成ファイルに格納されます。オフラインでのサービスの変更は、実行中のサーバーに影響しません。オンラインでのサービスの変更は、その変更が動的属性を対象とする場合、実行中のサーバーに影響します。それ以外の場合、サービスの変更を反映するには、サービスを停止して再び起動するか、サーバーを再起動する必要があります。

書式

```
ALTER SERVICE <service-name>
-> [ PROTOCOL { OCI | RMU | SQLSERVICES } ]
-> [ AUTOSTART { ON | OFF } ]
-> [ DEFAULT_CONNECT_USERNAME { <quoted-string> | <identifier> } ]
-> [ DEFAULT_CONNECT_PASSWORD <quoted-string> ]
-> [ REUSE [ SCOPE] [ IS] { SESSION | TRANSACTION } ]
-> [ SQL_VERSION { <version-number> | S[TANDARD] } ]
-> [ PROCESS_INITIALIZATION { <quoted-string> | LOGIN } ]
-> [ ATTACH <quoted-string> ]
-> OWNER { <quoted-string> | <identifier> }
-> [ OWNER PASSWORD <quoted-string> ]
-> [ SCHEMA <quoted-string> ]
-> [ SQL_INIT_FILE <quoted-string> ]
-> [ DATABASE_AUTHORIZATION { [ SERVICE] OWNER
    | [ CONNECT] USERNAME } ]
-> [ APPLICATION_TRANSACTION_USAGE { SERIAL | CONCURRENT } ]
-> [ IDLE_USER_TIMEOUT <number-in-seconds> ]
-> [ IDLE_EXECUTOR_TIMEOUT <number-in-seconds> ]
-> [ MIN_EXECUTORS <number> ]
-> [ MAX_EXECUTORS <number> ]
-> [ CLIENTS_PER_EXECUTOR <number> ];

<service-name> ::= <identifier>
```

引数

<service-name>

サービス名。サービス名は、識別子として表します。

PROTOCOL {OCI | RMU | SQLSERVICES}

各サービスでサポートする Application Program Interface (API) を決定します。サービスでサポートできる API は 1 つのみです。サービスでサポートする予定のクライアントのタイプに一致する API を指定します。

- OCI

Oracle Call Interface (OCI) を使用する Oracle クライアントまたはサード・パーティ・クライアント

- RMU

Oracle RMU クライアント (Oracle Rdb Performance Monitor、Oracle RMUwin、Parallel Backup Monitor)

- SQLSERVICES

Oracle SQL/Services クライアント (Query Performance Tuner、Oracle ODBC Driver for Rdb、Oracle SQL/Services、Oracle Rdb Schema Manager)。この設定がデフォルトです。

注意： 使用する予定の各サービス API がサポートされるディスパッチャ・メッセージ・プロトコルを使用して、ディスパッチャ・ネットワーク・ポートを定義しておく必要があります。たとえば、OCI API をサポートするサービスと、SQLSERVICES API をサポートする別のサービスを定義する場合、OCI ディスパッチャ・メッセージ・プロトコルおよび Oracle Net トランスポートをサポートする少なくとも 1 つのディスパッチャ・ネットワーク・ポートと、SQLSERVICES ディスパッチャ・メッセージ・プロトコルおよび使用可能な任意のトランスポートをサポートする別のディスパッチャ・ネットワーク・ポートをそれぞれ定義する必要があります。

AUTOSTART {ON | OFF}

START SERVER または RESTART SERVER コマンドを発行する場合、サービス・オブジェクトを自動的に起動するかどうかを決定します。引数を ON に指定すると、START SERVER または RESTART SERVER コマンドを発行したときに、サービス・オブジェクトが自動的に起動されます。デフォルトは ON です。

DEFAULT_CONNECT_USERNAME {<quoted-string> | <identifier>}

不明なユーザーによるサービスへの接続を許可する場合に使用するユーザー名を引用符付き文字列または識別子として指定します。この引数は、SQLSERVICES プロトコルをサポートするデータベース・サービスにのみ適用されます。この引数の使用方法の詳細は、[2.7.1 項](#)および [2.8 項](#)を参照してください。これは動的属性であり、変更するとその内容は即座に反映されません。

DEFAULT_CONNECT_PASSWORD <quoted-string>

接続ユーザー名に関連するパスワードを引用符付き文字列として指定します。

REUSE SCOPE IS {SESSION | TRANSACTION}

■ SESSION

セッションで再利用可能なサービスのエグゼキュータは、一度に1つのクライアント・セッションに対するリクエストを処理します。クライアントがサービスに接続すると、セッションが開始され、接続はエグゼキュータ・プロセスにバインドされます。クライアントがサービスから切断されると、セッションは終了し、接続はエグゼキュータ・プロセスからバインド解除されます。この設定がデフォルトです。

■ TRANSACTION

トランザクションで再利用可能なサービスのエグゼキュータは、一度に1つのトランザクションに対するリクエストを処理します。ただし、このエグゼキュータでは、多くの同時クライアント・セッションがサポートされます。暗黙的または明示的にトランザクションを開始する SQL 文がクライアントによって発行されると、トランザクションが起動します。クライアントが SQL COMMIT または ROLLBACK 文を正常に発行すると、トランザクションは終了します。REUSE SCOPE IS TRANSACTION 引数は、SQLSERVICES プロトコルを使用するデータベース・サービスにのみ適用されます。

詳細は、2.6 項を参照してください。

SQL_VERSION {<version-number> | STANDARD}

サービスで使用する SQL のバージョンを指定します。この値は、Oracle Rdb マルチリリース環境では SQL のバージョンを選択するためのバージョン番号データ型 (7.1 など) として表します。Oracle Rdb 単一リリース環境では、SQL の標準バージョンを実行するためのキーワード STANDARD (または S) により表します。インストール情報に記載されているとおり、どちらの値も実行時に Oracle Rdb RDB\$SETVER コマンド・プロシージャの最初のパラメータ引数として使用されます。バージョン番号は n.n パラメータ引数に解決され、STANDARD または S という語は S パラメータ引数に解決されます。値を指定しない場合、デフォルトはキーワード STANDARD です。

PROCESS_INITIALIZATION {<quoted-string> | LOGIN}

プロセス初期化ファイルとして、<quoted-string> で指定した特別なプロセス初期化ファイルか、またはキーワード LOGIN を使用できます。プロセス初期化ファイルとして LOGIN を指定すると、Oracle SQL/Services では、OpenVMS SYS\$GETUAI システム・サービスによって返される AUTHORIZE の LGICMD 修飾子により指定されたファイルが使用されます。このファイルが完全修飾形式で指定されない場合、ファイルは検出されず、エグゼキュータは失敗します。LOGIN としてプロセス初期化を指定する場合、サービス所有者アカウントに LGICMD 修飾子が定義されていることを確認してください。プロセス初期化またはログイン・ファイルを使用すると、このサービスに対応するエグゼキュータ・プロセスの一部の属性を定義する際に役立ちます。引数を指定しない場合、デフォルトではどの初期化ファイルも実行されません。このファイルは、エグゼキュータの起動時に、エグゼキュータごとに1回実行されます。

ATTACH <quoted-string>

SQL ATTACH 文。

SQL ATTACH 文を指定しない場合、特定のデータベースに事前にアタッチされていない汎用サービスを作成します。

SQL ATTACH 文を指定する場合、特定のデータベースに事前にアタッチされたデータベース・サービスを作成します。

この引数は、一重引用符付き文字列であり、動的 SQL で使用されるアタッチ文字列リテラルとまったく同じ形式です。この文字列の FILENAME キーワードは、短縮できません。

ATTACH 文の詳細は、『Oracle Rdb SQL リファレンス・マニュアル』を参照してください。

OWNER {<quoted-string | <identifier>}

サービス所有者のユーザー名を指定します。すべてのサービスには所有者名があります。所有者名は、引用符付き文字列または識別子として指定する必要があり、指定しない場合、エラー・メッセージが返されます。

サービスがデータベース・サービスの場合、エグゼキュータが指定のデータベースにアタッチする際のアクセス・チェックにサービス所有者の権限が使用されます。データベース・サービスの詳細は、[2.6 項](#)を参照してください。

データベース・アクセス認可がサービス所有者別の場合、すべてのデータベース・アクセス操作にサービス所有者の権限が使用されます。データベース・アクセス認可の詳細は、この引数リストで後述される DATABASE_AUTHORIZATION 引数を参照してください。

エグゼキュータは、サービス所有者のアカウントの権限および割当て制限を使用して作成されます。詳細は、[2.10.1 項](#)を参照してください。

OWNER 引数は、SQLSERVICES プロトコルをサポートするデータベース・サービスにのみ適用されます。

OWNER PASSWORD <quoted-string>

サービス所有者のパスワードを指定します。

SCHEMA <quoted-string>

エグゼキュータがマルチスキーマ・データベースにアタッチされる際に使用するデフォルト・スキーマを指定します。

スキーマ名がサービス定義に指定されていない場合、スキーマ名は、デフォルトでサービス所有者のアカウント名（データベース・アクセス認可がサービス所有者の場合）または接続ユーザー名（データベース・アクセス認可が接続ユーザー名の場合）に設定されます（[2.9 項](#)を参照）。

スキーマ引数により、デフォルト値を上書きできます。この引数は、OCI 接続をサポートするサービスで指定された場合、無視されます。

SQL_INIT_FILE <quoted-string>

クライアント接続の SQL 環境をカスタマイズする SQL 文を含んだファイルを指定します。たとえば、SQL 初期化ファイルの使用により、SQL 言語とデフォルト・キャラクタ・セットを設定できます。SQL 初期化ファイルの文は、クライアントがサービスに接続するたびに実行されます。

SQL 初期化ファイルの使用方法の詳細は、[6.1 項](#)を参照してください。

DATABASE_AUTHORIZATION {[SERVICE] OWNER | [CONNECT] USERNAME}

データベースへのアクセスに使用するユーザー名を決定します。デフォルトは、CONNECT USERNAME です。

- SERVICE OWNER

データベース・サービスの場合、データベースへのすべてのアクセスは、サービス所有者ユーザー名を使用して行われます。このオプションは、OCI Services for Oracle Rdb ではサポートされません。

- CONNECT USERNAME

データベースへのアクセスは、クライアント指定のユーザー名、DECnet プロキシ・ユーザー名、または DEFAULT_CONNECT_USERNAME 引数で指定されたユーザー名を使用して行われます。

データベース・アクセス認可の詳細は、[2.7 項](#)および [2.8 項](#)を参照してください。

APPLICATION_TRANSACTION_USAGE {SERIAL | CONCURRENT}

トランザクションで再利用可能なデータベース・サービスにのみ適用されます。一部のアプリケーションは、サービスに対して単一の接続のみを作成してその作業を実行しますが、他のアプリケーションは同じサービスに対して複数の接続を作成します。トランザクションで再利用可能なデータベース・サービスに対して作成された接続は、セッションの継続中は同じエグゼキュータに関連付けられます。

クライアント・アプリケーションにより 1 つのサービスに対する複数の接続が作成され、それらの接続が同じエグゼキュータに割り当てられた場合、ある接続に基づく既存のトランザクションを終了する前にクライアント・アプリケーションで別の接続に基づく新規トランザクションを開始しようとすると、デッドロックが発生します。CONCURRENT キーワードを指定すると、Oracle SQL/Services では、同じノード上の同じクライアント・アプリケーションからの複数の接続が、同じエグゼキュータ・プロセスに割り当てられなくなります。

SERIAL キーワードを指定すると、Oracle SQL/Services では、複数の接続に基づいてクライアント・アプリケーションで同時トランザクションが開始されることはないものとみなされます。Oracle SQL/Services では、最もビジーでないことを基準にエグゼキュータ・プロセスに接続が割り当てられます（エグゼキュータ・プロセスには最少のクライアント接続が割当て済み）。つまり、クライアント・アプリケーションで同じサービスに対して複数の接続が作成され、キーワード SERIAL が指定された場合、最初の接続と同じエグゼキュータ・プロセスに 2 番目の接続が関連付けられるかどうかは、そのエグゼキュータ・プロセスに割り当てられている接続の数と、そのサービスの別のエグゼキュータ・プロセスに割り当てられている接続の数に応じて変わります。

APPLICATION_TRANSACTION_USAGE 引数のデフォルトは、SERIAL です。これは動的属性であり、変更するとその内容は即座に反映されます。

Microsoft Access などの一部のアプリケーションでは、作業を実行するために同じサービスに対して複数の接続が作成されるため、CONCURRENT キーワードを指定する必要があります。CONCURRENT に設定されると、Oracle SQL/Services では、接続に割り当てるエグゼキュータの選択時にクライアントのノード、ユーザー名およびアプリケーション名が考慮され、同じクライアント・アプリケーションからの複数の接続が同じエグゼキュータ・プロセスに割り当てられなくなります。

この引数は、Oracle SQLSERVICES サービスでのみ使用されます。

IDLE_USER_TIMEOUT <number-in-seconds>

サーバーがクライアントとの接続を切断する前に、クライアント（ユーザー）がアイドル状態を継続できる時間を秒単位で指定します。この値は、整数として表します。デフォルト値は 0 であり、SHOW SERVICE コマンドでは <none> として表示されます。これは、アイドル・タイムアウト値が無制限であることを示します。0 以外の値を指定すると、その値は次に大きい 90 秒の倍数に丸められます。これは動的属性であり、変更するとその内容は即座に反映されます。

IDLE_EXECUTOR_TIMEOUT <number-in-seconds>

セッションで再利用可能なサービスのエグゼキュータ・プロセスを削除前にアクティブではない状態（クライアント接続にバウンドしない状態）のまま維持する時間を秒単位で指定します。この値は、整数として表します。デフォルトのタイムアウト値は、1800 秒（30 分）です。これは動的属性であり、変更するとその内容は即座に反映されます。

MIN_EXECUTORS <number>

エグゼキュータ・プロセスの数を減少できる最小値を設定します。これは、START SERVICE または START SERVER コマンドを使用した起動時に開始されるエグゼキュータ・プロセスの数でもあります。この値は、整数として表します。セッションで再利用可能なサービスに対応するエグゼキュータのデフォルトの最小数は、0（ゼロ）です。MIN_EXECUTORS が 0（ゼロ）に設定されたサービスでは、サービスの起動時に開始中という状態が表示されることはありません。この状態は、実行中または失敗として表示されます。これは動的属性であり、変更するとその内容は即座に反映されます。

トランザクションで再利用可能なエグゼキュータを使用する場合、エグゼキュータの最大数の値と等しくなるようにエグゼキュータの最小数の値を設定する必要があります。トランザクションで再利用可能なサービスのデフォルト値は 1 です。

MAX_EXECUTORS <number>

エグゼキュータ・プロセスの数を増加できる最大値を設定します。この値は、整数として表します。エグゼキュータのデフォルトの最大数は1です。これは動的属性であり、変更するとその内容は即座に反映されます。

トランザクションで再利用可能なエグゼキュータを使用する場合、エグゼキュータの最大数の値と等しくなるようにエグゼキュータの最小数の値を設定する必要があります。トランザクションで再利用可能なサービスのデフォルト値は1です。

CLIENTS_PER_EXECUTOR <number>

エグゼキュータごとに許可するクライアントの数を指定します。許可するクライアントの数は、サービスがセッションで再利用可能かトランザクションで再利用可能かに応じて変わります。セッションで再利用可能なサービスの場合、エグゼキュータごとのクライアント数のデフォルトは1であり、1より大きい値は指定できません。トランザクションで再利用可能なサービスの場合、エグゼキュータごとのクライアント数のデフォルトは1ですが、1より大きい値も指定できます。CLIENTS_PER_EXECUTORの値は、整数として表します。これは動的属性であり、変更するとその内容は即座に反映されます。

使用方法

- OCI サービス以外のサービスを作成すると、SYSPRV 権限を保持する権限付きユーザーのみがサービスの使用を認可されます。他の任意のユーザーには権限を付与する必要があります。
- ALTER SERVICE コマンドのパラメータに指定された値により、構成ファイルおよび実行中のサーバーに定義された値が置き換えられます。ただし、実行中のサーバーに対する変更は、即座に反映されず、次の説明にあるような制限を受けます。
 - セッションで再利用可能なサービスに対応するエグゼキュータの最小数の値を減らしても、個々のエグゼキュータが現在のタイムアウト設定を使用してタイムアウトするまで、エグゼキュータ・プロセスの実際の数は減少しません。
 - トランザクションで再利用可能なサービスに対応するエグゼキュータの最小数の値を減らしても、サービスが停止して再起動するまで、エグゼキュータ・プロセスの実際の数は減少しません。
 - エグゼキュータの最大数の値を増やしても、新規に作成されるエグゼキュータ・プロセスの数は、以前の制限値を継承します。
 - エグゼキュータの最小数の値を増やした場合、新規エグゼキュータは、エグゼキュータの新しい最小数がアクティブになるまで作成されません。
 - アイドル・エグゼキュータ・タイムアウト・パラメータの値を変更した場合、新規アイドル・タイムアウト値が使用されるのは、特定のエグゼキュータの次のタイムアウト・サイクル以降です。

例

例 1: トランザクションで再利用可能なデータベース・サービスをオンラインで変更して、エグゼキュータごとのクライアントの数を 20 に増加し、エグゼキュータの最小数と最大数を 10 に増加します。これらの属性は動的属性であるため、サービスを停止して再起動する必要はありません。

```
SQLSRV> CONNECT SERVER;
Connecting to server ...
Connected
SQLSRV> ALTER SERVICE database4
  _SQLSRV> MIN_EXECUTORS 10
  _SQLSRV> MAX_EXECUTORS 10
  _SQLSRV> CLIENTS_PER_EXECUTOR 20;
```

例 2: オンラインでサービスを変更し、SQL_INIT_FILE 属性を変更します。この属性は動的属性ではないため、変更を反映するためにサービスを再起動する必要があります。

```
SQLSRV> CONNECT SERVER;
Connecting to server ...
Connected
SQLSRV> ALTER SERVICE database4
  _SQLSRV> SQL_INIT_FILE 'sql710';
%DBS-S-ALTER_RESTART, Restart object to have altered settings take affect
SQLSRV> SHUTDOWN SERVICE database4;
SQLSRV> START SERVICE database4;
```

CLOSE コマンド

SQLSRV_MANAGE 環境で出力ファイルを閉じます。

書式

CLOSE;

使用方法

出力ファイルを閉じると、出力は標準出力に送られます。

例

例 1: 出力ファイルを閉じます。

```
SQLSRV> CLOSE;
```

CONNECT TO SERVER コマンド

サーバーの管理を開始できるようにオンラインでサーバーに接続します。

書式

```
CONNECT [TO] SERVER [ AS <connect-name> ]
->[ USER { <user-name> USING { <password> } ]
->[ NODE { <quoted-string> | <identifier> } ]
->[ <network-port-spec> ];

<connect-name> ::= <identifier>
<user-name> ::= { <quoted-string> | <identifier> }
<password> ::= <quoted-string>
<network-port-spec> ::= NETWORK_PORT <transport-spec>
<transport-spec> ::= { <tcp-spec> | <decnet-spec> }
<tcp-spec> ::= TCPIP [ PORT_ID <number> ]
<decnet-spec> ::= DECNET [ OBJECT { <number> | <identifier>
| <quoted-string> } ]
```

引数

<connect-name>

接続名。特定のノード上のサーバーに対する接続を一意に識別する識別子です。接続名は、複数のサーバーに同時に接続する際に最も役立ちます。1つのサーバーのみを管理する予定の場合、接続名は不要です。新規接続を作成すると、それは常に現在の接続になります。接続先のサーバー内で管理するサーバーを切り替える場合、SET CONNECTION コマンドを使用し、そのサーバーの接続名を指定します。

接続名は、識別子として表します。

USER <user-name> USING <password>

サーバーを管理する権限を付与されたアカウントのユーザー名とパスワードを指定します。ユーザー名とパスワードは、引用符付き文字列または識別子として表します。

ローカル・ノードでサーバーを管理するのに十分な権限を保持した状態で DECnet または TCP/IP を使用する場合、ローカル・ノードのサーバーに接続する際にユーザー名とパスワードを入力する必要はありません。ユーザー名とパスワードを指定せずにローカル・ノードのサーバーに接続する方法の詳細は、「使用方法」を参照してください。

NODE <quoted-string | identifier>

サーバーの存在するノード。デフォルトでは、ノード名はローカル・ホスト名です。ノード名は、引用符付き文字列または識別子として表します。この名前を使用して、リモート・サーバーに接続できます。

<network-port-spec>

モニターで Oracle SQL/Services 管理クライアント (SQLSRV_MANAGE および Oracle SQL/Services Manager GUI) との通信に使用するネットワーク・ポートをリストします。<network-port-spec> 引数は、TCPIP または DECNET です。デフォルトでは、ネットワーク・ポートの指定は、デフォルト・ポート ID の 2199 を使用して TCP/IP に設定されます。デフォルト DECnet オブジェクト名は、SQLSRV_SERVER です。

使用方法

- オンラインでサーバーを管理するには、事前にサーバーに接続しておく必要があります。オフラインでサーバーを管理するには、サーバーの構成ファイルを選択する必要があります (SET CONFIG_FILE コマンド)。
- CONNECT TO SERVER コマンドを使用してサーバーに対する新規接続を確立すると、その新規接続は現在の接続になります。後続のすべてのオンライン・システム管理コマンドは、現在の接続に対して動作します。SET CONNECT コマンドを使用して、複数のサーバーに対する接続を切り替えます。DISCONNECT コマンドを使用して、サーバーとの接続を切断します。
- ローカル・ユーザーは、ユーザー名またはパスワードを指定することなく DECnet を使用してサーバーに接続できます。TCP/IP を使用してサーバーに接続する場合、ユーザー名とパスワードを省略するには、SYSPRV または BYPASS 権限を保持している必要があります。
- 現在開いている構成ファイルを使用してローカル・サーバーに接続する場合、SQLSRV_MANAGE では、そのサーバーに定義されたすべてのネットワーク・ポートに接続しようと試みます。管理接続を確立するために、ラウンドロビン法でそれぞれ最大 3 回まで各ネットワーク・ポートに接続を試みます。管理接続に使用されるネットワーク・ポートは、接続に成功した最初のポートです。

例

例 1: TCP/IP を使用して権限付きのローカル・ユーザーとしてローカル・ノード上のサーバーに接続します。

```
SQLSRV> CONNECT SERVER;  
Connecting to server ...  
Connected
```

例 2: サーバーに接続します (ユーザー名とパスワードは引用符付き文字列)。

```
SQLSRV> CONNECT SERVER USER 'system' USING 'password';  
Connecting to server ...  
Connected
```

CREATE DISPATCHER コマンド

現在のサーバーを対象にディスパッチャ・オブジェクト定義を作成します。定義は構成ファイルに格納されます。新規ディスパッチャ・オブジェクトは、実行中のサーバーの一部となるようオンラインで起動する必要があります。定義された各ディスパッチャは、ネットワーク・ポートまたはオブジェクトの一意のセットを対象にリスニングする必要があります。

書式

```
CREATE DISPATCHER <disp-name>
->[ AUTOSTART { ON | OFF } ]
->[ MAX_CONNECTIONS <number> ]
->[ IDLE_USER_TIMEOUT <number-in-seconds> ]
->[ MAX_CLIENT_BUFFER_SIZE <number> ]
->[ DUMP_PATH<directory-specification>]
->[ LOG_PATH<directory-specification>]
->[ <network-port-spec> ] ... ;

<disp-name> ::=<identifier>
<network-port-spec> ::=NETWORK_PORT <transport-spec>
    PROTOCOL { NATIVE | OCI | SQLSERVICES | JDBC }
<transport-spec> ::= { <tcp-spec> | <decnet-spec>
    | <ipxspx-spec> | sqlnet-spec }
<tcp-spec> ::=TCPIP [ PORT_ID <number> ]
<decnet-spec> ::=DECNET [ OBJECT { <number> | <identifier>
    | <quoted-string> } ]
<ipxspx-spec> ::=IPXSPX [ PORT_ID <number> ]
<sqlnet-spec> ::=SQLNET LISTENER_NAME { <identifier>
    | <quoted-string> }
```

引数

<disp-name>

ディスパッチャ名。ディスパッチャ名は、識別子として表します。ディスパッチャ名は、一意である必要があります。

AUTOSTART {ON | OFF}

START SERVER または RESTART SERVER コマンドを発行する場合、ディスパッチャ・オブジェクトを自動的に起動するかどうかを決定します。引数を ON に指定すると、START SERVER または RESTART SERVER コマンドを発行したときに、ディスパッチャ・オブジェクトが自動的に起動されます。デフォルトは ON です。

MAX_CONNECTIONS <number>

ディスパッチャで対応するクライアントからのネットワーク接続の最大数を指定します。接続の最大数は、整数として表します。デフォルトは 100 です。オペレーティング・システム構成、ネットワーク構成および共有サーバー・メモリー以外に上限はありません。

IDLE_USER_TIMEOUT <number-in-seconds>

ディスパッチャがクライアントとの接続を切断する前に、クライアント（ユーザー）がアイドル状態を継続できる時間を秒単位で指定します。この値は、整数として表します。デフォルト値は0であり、SHOW DISPATCHER コマンドでは <none> として表示されます。これは、アイドル・タイムアウト値が無制限であることを示します。0以外の値を指定すると、その値は次に大きい90秒の倍数に丸められます。

MAX_CLIENT_BUFFER_SIZE <number>

許可する最大クライアント・バッファ・サイズを指定します。使用可能な最大クライアント・バッファ・サイズは、32,000 バイトです。クライアント・アプリケーションで最大値より大きいバッファ・サイズを指定すると、Oracle SQL/Services クライアント API により、ディスパッチャに指定された最大サイズにバッファ・サイズが調整されます。MAX_CLIENT_BUFFER_SIZE 属性に指定可能な最小値（デフォルト値）は、5000 バイトです。

DUMP_PATH <directory-specification>

バグチェック・ダンプ・ファイルのディレクトリ名を指定します。

LOG_PATH <directory-specification>

ログ・ファイルのディレクトリ名を指定します。

LOG_PATH 引数のディレクトリ名のかわりに NOLOG を指定すると、ログ・ファイルは書き込まれません。

<network-port-spec>

ディスパッチャでクライアントとの通信に使用するネットワーク・ポートをリストします。ネットワーク・ポートは、TCP/IP、DECnet、IPX/SPX および Oracle Net のうち、1つまたは任意の組合せを指定します。TCP/IP のデフォルト・ポート ID は 118、デフォルト DECnet オブジェクトは 81、IPX/SPX のデフォルト・ポート ID は 33969 (0x84b1) です。ネットワーク・ポートを指定しないと、ディスパッチャではデフォルト・ポートが使用されます。<network-port-spec> 引数は、OCI Services for Oracle Rdb の複数のリスナー名を含めるように繰り返すことができます。CREATE DISPATCHER コマンドで指定できる <network-port-spec> 引数の最大数は5です。

この引数により、各ディスパッチャ・ネットワーク・ポートでサポートできるメッセージ・プロトコルも決定されます。ディスパッチャ・ネットワーク・ポートでサポートできるメッセージ・プロトコルは1つのみです。ディスパッチャ・ネットワーク・ポートでサポートする予定のクライアントのタイプに一致するメッセージ・プロトコルを指定します。

- NATIVE
 - Oracle RMU クライアント (Oracle Rdb Performance Monitor、Oracle RMUwin、Parallel Backup Monitor など)
- OCI
 - Oracle Call Interface (OCI) を使用する Oracle クライアント、または Oracle サーバー・クライアント
- SQLSERVICES
 - Oracle SQL/Services クライアント (Query Performance Tuner、Oracle ODBC Driver for Rdb、Oracle SQL/Services API、Schema Manager)
- JDBC
 - JDBC for Oracle Rdb クライアント

注意：使用する予定の各サービス API がサポートされるディスパッチャ・メッセージ・プロトコルを使用して、ディスパッチャ・ネットワーク・ポートを定義しておく必要があります。たとえば、OCI API をサポートするサービスと、SQLSERVICES API をサポートする別のサービスを定義する場合、OCI ディスパッチャ・メッセージ・プロトコルおよび Oracle Net トランスポートをサポートする少なくとも 1 つのディスパッチャ・ネットワーク・ポートと、SQLSERVICES ディスパッチャ・メッセージ・プロトコルおよび使用可能な任意のトランスポートをサポートする別のディスパッチャ・ネットワーク・ポートをそれぞれ定義する必要があります。

使用方法

- Oracle Net トランスポート・オプションを使用するには、<transport-spec> 引数に Oracle Net トランスポート・オプションを <sqlnet-spec> として指定し、OCI Services for Oracle Rdb のリスナー名をその <identifier> 引数として指定します。
- LISTENER という語は、キーワード LISTENER_NAME と同義です。
- SQLSRV_MANAGE では、同じポート ID またはオブジェクトを対象にリスニングを行う 2 つ以上のディスパッチャを作成できますが、起動できるのは、一意のポート ID またはオブジェクトが割り当てられた最初のディスパッチャのみです。同じポート ID またはオブジェクトでリスニングを行う 2 番目のディスパッチャを起動しようとしても、指定したどのネットワーク・ポートもリスニングできなければ、起動に失敗します。
- ログ・ファイルに I/O エントリが大量に出力されるという問題が発生しなければ、トラブルシューティング目的でログ・ファイルを作成することをお勧めします。LOG PATH および DUMP PATH 引数のデフォルト・ディレクトリは、SYS\$MANAGER です。
- LOG PATH および DUMP PATH 引数は、SQLSRV_MANAGE コマンド・ユーティリティの使用時には使用できますが、Oracle SQL/Services Manager GUI の使用時には使用できません。
- 既存の論理名 SQLSRV_DISP_LOGPATH および SQLSRV_DISP_DUMP_PATH に値を割り当てると、それらの値により、LOG PATH および DUMP PATH 引数で指定されたログ・パス値とダンプ・パス値が上書きされます。

例

例 1: TCP/IP プロトコルを使用するディスパッチャを作成します。

```
SQLSRV> CREATE DISPATCHER tcpip_disp NETWORK_PORT TCP/IP;
SQLSRV> START DISPATCHER tcpip_disp;
```

例 2: SQL*Net プロトコルを使用するディスパッチャを作成します。

```
SQLSRV> CREATE DISPATCHER sqlnet_disp
_SQLSRV> NETWORK_PORT SQLNET LISTENER_NAME LISTENER;
SQLSRV> START DISPATCHER sqlnet_disp;
```

例 3: NOLOG 引数を指定します。次のように LOG PATH 引数のディレクトリ名のかわりに NOLOG を指定すると、ログ・ファイルは書き込まれません。

```
SQLSRV> CREATE DISPATCHER SQLSRV_DISP1 LOG PATH 'NOLOG'
_SQLSRV> DUMP_PATH 'SYS$MANAGER';
```

CREATE SERVER コマンド

サーバー・オブジェクト定義および構成ファイルを作成します。定義は構成ファイルに格納されます。新規サーバーは、オフラインで起動する必要があります。

書式

CREATE SERVER

```

->[ MAX_SHARED_MEMORY_SIZE <number> ]
-->[ LOG_PATH <directory-specification> ]
-->[ DUMP_PATH <directory-specification> ]
->[ PROCESS_STARTUP_TIMEOUT <number-in-seconds> ]
->[ PROCESS_SHUTDOWN_TIMEOUT <number-in-seconds> ]
->[ <network-port-spec> ] ... ;

<network-port-spec> ::= NETWORK_PORT <transport-spec>
<transport-spec> ::= { <tcp-spec> | <decnet-spec> }
<tcp-spec> ::= TCPIP [ PORT_ID <number> ]
<decnet-spec> ::= DECNET [ OBJECT_{ <number> | <identifier>
| <quoted-string> } ]

```

引数

MAX_SHARED_MEMORY_SIZE <number>

サーバーで使用する最大共有メモリーのサイズを KB 単位で設定します。デフォルト値は、2000KB (2MB) です。Oracle SQL/Services リリース 7.0 以上から、サーバーではモニターの起動時に最大共有メモリー・サイズが割り当てられます。

DUMP_PATH <directory-specification>

バグチェック・ダンプ・ファイルのディレクトリ名を指定します。デフォルト・ディレクトリは SYS\$MANAGER です。

LOG_PATH <directory-specification>

ログ・ファイルのディレクトリ名を指定します。デフォルト・ディレクトリは SYS\$MANAGER です。

LOG_PATH 引数のディレクトリ名のかわりに NOLOG を指定すると、ログ・ファイルは書き込まれません。

PROCESS_STARTUP_TIMEOUT <number-in-seconds>

ディスプレイまたはエグゼキュータ・プロセスが起動しないと判断し、モニターがアクションを開始してプロセスを終了するまでに待機する時間を指定します。この引数は、秒単位で表された整数です。デフォルト値は 0 秒であり、プロセス起動タイマー値は未設定です。詳細は、「使用方法」を参照してください。

PROCESS_SHUTDOWN_TIMEOUT <number-in-seconds>

ディスプレイまたはエグゼキュータ・プロセスが停止しないと判断し、モニターがアクションを開始してプロセスを終了するまでに待機する時間を指定します。この引数は、秒単位で表された整数です。デフォルト値は 0 秒であり、プロセス停止タイマー値は未設定です (プロセス停止タイマー値は無制限です)。詳細は、「使用方法」を参照してください。

<network-port-spec>

モニターで Oracle SQL/Services 管理クライアント (SQLSRV_MANAGE および Oracle SQL/Services Manager GUI) との通信に使用するネットワーク・ポートをリストします。ネットワーク・ポートの指定は、TCP/IP または DECnet です。TCP/IP のデフォルト・ポート ID は 2199、デフォルト DECnet オブジェクト名は SQLSRV_SERVER です。ネットワーク・ポートを指定しないと、サーバーのモニターではデフォルト・ポートが使用されます。CREATE SERVER コマンドで指定できる <network-port-spec> 引数の最大数は 5 です。

DECnet または TCP/IP は、サーバー作成定義が指定されるノードで使用可能である必要があります。どちらも使用できない場合、サーバーは起動されません。

使用方法

- CREATE SERVER コマンドは、通常、Oracle SQL/Services のインストール中のみ使用します。インストール・プロシージャでは、SQLSRV_CREATE71.COM プロシージャを使用して、サーバーと、ディスパッチャおよびサービスのデフォルト・セットを含む構成ファイルを作成し、サーバーを起動します。

誤って構成ファイルを削除した場合や、ファイルが破損した場合、バックアップがなければサーバーを再作成する必要があります。まず、元の構成ファイルを削除します (存在している場合)。ただし、Oracle SQL/Services コンポーネントがファイル破損の原因の場合、ソフトウェア問題のレポートとして提出できるように、ファイルのコピーを維持してください。ソフトウェア問題をレポートする方法の詳細は、7.1 項を参照してください。サーバーを再作成する場合、次の 2 つの方法があります。

- SQLSRV_CREATE71.COM プロシージャを実行します。

SYS\$MANAGER:SQLSRV_CREATE71.SQS SQLSRV_MANAGE スクリプトを使用してサーバーを再作成する SYS\$MANAGER:SQLSRV_CREATE71.COM コマンド・プロシージャを実行します。

注意：サーバーを再作成する場合には、この方法をお勧めします。

SQLSRV_CREATE71.SQS ファイルを実行すると、Oracle RMU ディスパッチャおよび Oracle RMU サービス・オブジェクトのみが再作成されます。

- SET CONFIG_FILE コマンドを発行し、存在しない構成ファイルを指定します。この操作を実行すると、今すぐに構成ファイルを作成するかどうかを尋ねられます。YES と回答します。デフォルトは NO です。SET CONFIRM コマンドが OFF に設定されている場合、プロンプトは表示されません。SHOW SETTINGS コマンドでは、現在の設定と、この新規構成ファイルのファイル指定が表示されます。CREATE SERVER コマンドを発行し、この構成ファイルを使用してサーバーを作成します。
- 構成ファイルがすでに存在する状態で CREATE SERVER コマンドを発行すると、エラー・メッセージが表示され、CREATE SERVER コマンドは失敗します。
- SQLSRV_MANAGE ユーティリティでは、デフォルトの TCP/IP または DECnet ポートを使用してサーバーのモニターに接続しようと試みます。サーバーのネットワーク・ポートを変更する場合、SQLSRV_MANAGE ユーティリティからの接続時に、そのポートを明示的に指定する必要があります。

- プロセス起動タイマーおよび停止タイマーを設定する場合、次のガイドラインに従ってください。
 - 通常、ディスパッチャとエグゼキュータ・プロセスは、合理的な時間間隔で起動および停止します。特別な状況にある場合にのみ、PROCESS_STARTUP_TIMEOUT および PROCESS_SHUTDOWN_TIMEOUT 引数に 0 (ゼロ) 以外の値を指定する必要があります。
 - 負荷の高いシステムの場合、通常は特定の操作が完了するまで時間がかかります。プロセス起動タイマーまたはプロセス停止タイマーを 0 (ゼロ) 以外の値に設定しているときに、明確な理由もなく障害が発生した場合 (他に考えられる原因をチェックしたが、問題を特定できない場合)、PROCESS_STARTUP_TIMEOUT 引数または PROCESS_SHUTDOWN_TIMEOUT 引数により大きな値を設定して、問題が解決するかどうかを確認してください。

例

例 1: 現在 Oracle SQL/Services サーバーが存在しないローカル・ノードにサーバー定義を作成します。

```
SQLSRV> SET CONFIG_FILE 'my_config_file';
SQLSRV> CREATE SERVER MAX_SHARED_MEMORY_SIZE 3000;
SQLSRV> START SERVER;
Server started
Connecting to server ...
Connected
```

CREATE SERVICE コマンド

現在のサーバーのみを対象にサービス・オブジェクト定義を作成します。定義は構成ファイルに格納されます。新規サービス・オブジェクトは、実行中のサーバーの一部となるようオンラインで起動する必要があります。

書式

```

CREATE SERVICE <service-name>
->[ PROTOCOL { OCI | RMU | SQLSERVICES } ]
->[ AUTOSTART { ON | OFF } ]
->[ DEFAULT_CONNECT_USERNAME { <quoted-string> | <identifier> } ]
-->[ DEFAULT_CONNECT_PASSWORD <quoted-string> ]
->[ REUSE [ SCOPE ] [ IS ] { SESSION | TRANSACTION } ]
->[ SQL_VERSION { <version-number> | S[TANDARD] } ]
->[ PROCESS_INITIALIZATION { <quoted-string> | LOGIN } ]
->[ ATTACH <quoted-string> ]
-> OWNER { <quoted-string> | <identifier> }
->[ OWNER PASSWORD <quoted-string> ]
->[ SCHEMA <quoted-string> ]
->[ SQL_INIT_FILE <quoted-string> ]
->[ DATABASE_AUTHORIZATION { [ SERVICE ] OWNER
| [ CONNECT ] USERNAME } ]
->[ APPLICATION_TRANSACTION_USAGE
{ SERIAL | CONCURRENT } ]
->[ IDLE_USER_TIMEOUT <number-in-seconds> ]
->[ IDLE_EXECUTOR_TIMEOUT <number-in-seconds> ]
->[ MIN_EXECUTORS <number> ]
->[ MAX_EXECUTORS <number> ]
->[ CLIENTS_PER_EXECUTOR <number> ];

<service-name> ::= <identifier>

```

引数

<service-name>

サービス名。サービス名は、識別子として表します。サービス名は、一意である必要があります。

PROTOCOL {OCI | RMU | SQLSERVICES}

各サービスでサポートする Application Program Interface (API) を決定します。サービスでサポートできる API は 1 つのみです。サービスでサポートする予定のクライアントのタイプに一致する API を指定します。

- OCI

Oracle Call Interface (OCI) を使用する Oracle クライアントまたはサード・パーティ・クライアント

- RMU

Oracle RMU クライアント (Rdb Performance Monitor、RMUwin、Parallel Backup Monitor など)

- SQLSERVICES

Oracle SQL/Services クライアント (Query Performance Tuner、Oracle ODBC Driver for Rdb、Oracle SQL/Services API、Schema Manager)。この設定がデフォルトです。

注意： 使用する予定の各サービス API がサポートされるディスパッチャ・メッセージ・プロトコルを使用して、ディスパッチャ・ネットワーク・ポートを定義しておく必要があります。たとえば、OCI API をサポートするサービスと、SQLSERVICES API をサポートする別のサービスを定義する場合、OCI ディスパッチャ・メッセージ・プロトコルおよび SQL*Net トランスポートをサポートする少なくとも 1 つのディスパッチャ・ネットワーク・ポートと、SQLSERVICES ディスパッチャ・メッセージ・プロトコルおよび使用可能な任意のトランスポートをサポートする別のディスパッチャ・ネットワーク・ポートをそれぞれ定義する必要があります。

AUTOSTART {ON | OFF}

START SERVER または RESTART SERVER コマンドを発行する場合、サービス・オブジェクトを自動的に起動するかどうかを決定します。引数を ON に指定すると、START SERVER または RESTART SERVER コマンドを発行したときに、サービス・オブジェクトが自動的に起動されます。デフォルトは ON です。

DEFAULT_CONNECT_USERNAME {<quoted-string> | <identifier>}

不明なユーザーによるサービスへの接続を許可する場合に使用するユーザー名を引用符付き文字列または識別子として指定します。この引数の使用方法の詳細は、[2.7.1 項](#)および [2.8 項](#)を参照してください。DEFAULT_CONNECT_USERNAME 引数は、SQLSERVICES プロトコルをサポートするデータベース・サービスにのみ適用されます。

DEFAULT_CONNECT_PASSWORD <quoted-string>

接続ユーザー名に関連するパスワードを引用符付き文字列として指定します。

REUSE SCOPE IS {SESSION | TRANSACTION}

■ SESSION

セッションで再利用可能なサービスのエグゼキュータは、一度に1つのクライアント・セッションに対するリクエストを処理します。クライアントがサービスに接続すると、セッションが開始され、接続はエグゼキュータ・プロセスにバインドされます。クライアントがサービスから切断されると、セッションは終了し、接続はエグゼキュータ・プロセスからバインド解除されます。デフォルトはSESSIONです。

■ TRANSACTION

トランザクションで再利用可能なサービスのエグゼキュータは、一度に1つのトランザクションに対するリクエストを処理します。ただし、このエグゼキュータでは、多くの同時クライアント・セッションがサポートされます。暗黙的または明示的にトランザクションを開始するSQL文がクライアントによって発行されると、トランザクションが起動します。クライアントがSQL COMMIT または ROLLBACK 文を正常に発行すると、トランザクションは終了します。REUSE SCOPE IS TRANSACTION 引数は、SQLSERVICES プロトコルをサポートするデータベース・サービスにのみ適用されます。

詳細は、2.6 項を参照してください。

SQL_VERSION {<version-number> | STANDARD}

サービスで使用するSQLのバージョンを指定します。この値は、Oracle Rdb マルチリリース環境ではSQLのバージョンを選択するためのバージョン番号データ型(7.1など)として表します。Oracle Rdb 単一リリース環境では、SQLの標準バージョンを実行するためのキーワードSTANDARD (またはS)により表します。インストール情報に記載されているとおり、どちらの値も実行時にOracle Rdb RDB\$SETVER コマンド・プロシージャの最初のパラメータ引数として使用されます。バージョン番号はn.nパラメータ引数に解決され、STANDARDまたはSという語はSパラメータ引数に解決されます。値を指定しない場合、デフォルトでキーワードSTANDARDが使用されます。

PROCESS_INITIALIZATION {<quoted-string> | LOGIN}

プロセス初期化ファイルとして、<quoted-string> で指定した特別なプロセス初期化ファイルか、またはキーワードLOGINを使用できます。プロセス初期化ファイルとしてLOGINを指定すると、Oracle SQL/Services では、OpenVMS SYS\$GETUAI システム・サービスによって返されるAUTHORIZEのLGICMD修飾子により指定されたファイルが使用されます。このファイルが完全修飾形式で指定されない場合、ファイルは検出されず、エグゼキュータは失敗します。LOGINとしてプロセス初期化を指定する場合、サービス所有者アカウントにLGICMD修飾子が定義されていることを確認してください。プロセス初期化またはログイン・ファイルを使用すると、このサービスに対応するエグゼキュータ・プロセスの一部の属性を定義する際に役立ちます。引数を指定しない場合、デフォルトではどの初期化ファイルも実行されません。このファイルは、エグゼキュータの起動時に、エグゼキュータごとに1回実行されます。

ATTACH <quoted-string>

SQL ATTACH 文。

SQL ATTACH 文を指定しない場合、特定のデータベースに事前にアタッチされていない汎用サービスを作成します。

SQL ATTACH 文を指定する場合、特定のデータベースに事前にアタッチされたデータベース・サービスを作成します。

この引数は、一重引用符付き文字列であり、動的SQLで使用されるアタッチ文字列リテラルとまったく同じ形式です。この文字列のFILENAMEキーワードは、短縮できません。

ATTACH 文の詳細は、『Oracle Rdb SQL リファレンス・マニュアル』を参照してください。

OWNER {<quoted-string | <identifier>}

サービス所有者のユーザー名を指定します。すべてのサービスには所有者名があります。所有者名は、引用符付き文字列または識別子として指定する必要があり、指定しない場合、エラー・メッセージが返されます。

サービスがデータベース・サービスの場合、エグゼキュータが指定のデータベースにアタッチする際のアクセス・チェックにサービス所有者の権限が使用されます。データベース・サービスの詳細は、[2.6 項](#)を参照してください。

データベース・アクセス認可がサービス所有者別の場合、すべてのデータベース・アクセス操作にサービス所有者の権限が使用されます。データベース・アクセス認可の詳細は、`DATABASE_AUTHORIZATION` 引数を参照してください。

エグゼキュータは、サービス所有者のアカウントの権限および割当て制限を使用して作成されます。詳細は、[2.10.1 項](#)を参照してください。

`OWNER` 引数は、`SQLSERVICES` プロトコルをサポートするデータベース・サービスにのみ適用されます。

OWNER PASSWORD <quoted-string>

サービス所有者のパスワードを指定します。

SCHEMA <quoted-string>

エグゼキュータがマルチスキーマ・データベースにアタッチされる際に使用するデフォルト・スキーマを指定します。

スキーマ名がサービス定義に指定されていない場合、スキーマ名は、デフォルトでサービス所有者のアカウント名（データベース・アクセス認可がサービス所有者の場合）または接続ユーザー名（データベース・アクセス認可が接続ユーザー名の場合）に設定されます（[2.9 項](#)を参照）。

スキーマ引数により、デフォルト値を上書きできます。この引数が `OCI Services for Oracle Rdb` に指定された場合、無視されます。

SQL_INIT_FILE <quoted-string>

クライアント接続の SQL 環境をカスタマイズする SQL 文を含んだファイルを指定します。たとえば、SQL 初期化ファイルの使用により、SQL 言語とデフォルト・キャラクタ・セットを設定できます。SQL 初期化ファイルの文は、クライアントがサービスに接続するたびに実行されます。

SQL 初期化ファイルの使用方法の詳細は、[6.1 項](#)を参照してください。

DATABASE_AUTHORIZATION {[SERVICE] OWNER | [CONNECT] USERNAME}

データベースへのアクセスに使用するユーザー名を決定します。デフォルトは、`CONNECT USERNAME` です。

- `SERVICE OWNER`

データベース・サービスの場合、データベースへのすべてのアクセスは、サービス所有者ユーザー名を使用して行われます。このオプションは、`OCI Services for Oracle Rdb` ではサポートされません。

- `CONNECT USERNAME`

データベースへのアクセスは、クライアント指定のユーザー名、`DECnet` プロキシ・ユーザー名、または `DEFAULT_CONNECT_USERNAME` 引数で指定されたユーザー名を使用して行われます。

データベース・アクセス認可の詳細は、[2.7 項](#)および[2.8 項](#)を参照してください。

APPLICATION_TRANSACTION_USAGE {SERIAL | CONCURRENT}

APPLICATION_TRANSACTION_USAGE 引数は、トランザクションで再利用可能なデータベース・サービスにのみ適用されます。一部のアプリケーションは、サービスに対して単一の接続のみを作成してその作業を実行しますが、他のアプリケーションは同じサービスに対して複数の接続を作成します。トランザクションで再利用可能なデータベース・サービスに対して作成された接続は、セッションの継続中は同じエグゼキュータに関連付けられます。詳細は、[2.6.3 項「トランザクションで再利用可能なデータベース・サービス」](#)を参照してください。

クライアント・アプリケーションにより 1 つのサービスに対する複数の接続が作成され、それらの接続が同じエグゼキュータに割り当てられた場合、ある接続に基づく既存のトランザクションを終了する前にクライアント・アプリケーションで別の接続に基づく新規トランザクションを開始しようとすると、デッドロックが発生します。CONCURRENT キーワードを指定すると、Oracle SQL/Services では、同じノード上の同じクライアント・アプリケーションからの複数の接続が、同じエグゼキュータ・プロセスに割り当てられなくなります。

SERIAL キーワードを指定すると、Oracle SQL/Services では、複数の接続に基づいてクライアント・アプリケーションで同時トランザクションが開始されることはないものとみなされます。Oracle SQL/Services では、最もビジーでないことを基準にエグゼキュータ・プロセスに接続が割り当てられます（エグゼキュータ・プロセスには最少のクライアント接続が割当て済）。つまり、クライアント・アプリケーションで同じサービスに対して複数の接続が作成され、キーワード SERIAL が指定された場合、最初の接続と同じエグゼキュータ・プロセスに 2 番目の接続が関連付けられるかどうかは、そのエグゼキュータ・プロセスにすでに割り当てられている接続の数と、そのサービスの別のエグゼキュータ・プロセスに割り当てられている接続の数に応じて変わります。

APPLICATION_TRANSACTION_USAGE 引数のデフォルトは、SERIAL です。

Microsoft Access などの一部のアプリケーションでは、作業を実行するために同じサービスに対して複数の接続が作成されるため、CONCURRENT キーワードを指定する必要があります。CONCURRENT に設定されると、Oracle SQL/Services では、接続に割り当てるエグゼキュータの選択時にクライアントのノード、ユーザー名およびアプリケーション名が考慮され、同じクライアント・アプリケーションからの複数の接続が同じエグゼキュータ・プロセスに割り当てられなくなります。

この引数は、Oracle SQLSERVICES サービスでのみ使用されます。

IDLE_USER_TIMEOUT <number-in-seconds>

サーバーがクライアントとの接続を切断する前に、クライアント（ユーザー）がアイドル状態を継続できる時間を秒単位で指定します。この値は、整数として表します。デフォルト値は 0 であり、SHOW SERVICE コマンドでは <none> として表示されます。これは、アイドル・タイムアウト値が無制限であることを示します。0 以外の値を指定すると、その値は次に大きい 90 秒の倍数に丸められます。

IDLE_EXECUTOR_TIMEOUT <number-in-seconds>

セッションで再利用可能なサービスのエグゼキュータ・プロセスを削除前にアクティブではない状態（クライアント接続にバウンドしない状態）のまま維持する時間を秒単位で指定します。この値は、整数として表します。デフォルトのタイムアウト値は、1800 秒（30 分）です。

MIN_EXECUTORS <number>

エグゼキュータ・プロセスの数を減少できる最小値を設定します。これは、START SERVICE または START SERVER コマンドを使用した起動時に開始されるエグゼキュータ・プロセスの数でもあります。この値は、整数として表します。セッションで再利用可能なサービスに対応するエグゼキュータのデフォルトの最小数は 0（ゼロ）です。MIN_EXECUTORS が 0（ゼロ）に設定されたサービスでは、サービスの起動時に開始中という状態が表示されることはありません。この状態は、実行中または失敗として表示されます。

トランザクションで再利用可能なエグゼキュータを使用する場合、エグゼキュータの最大数の値と等しくなるようにエグゼキュータの最小数の値を設定する必要があります。トランザクションで再利用可能なサービスのデフォルト値は 1 です。

MAX_EXECUTORS <number>

エグゼキュータ・プロセスの数を増加できる最大値を設定します。この値は、整数として表します。エグゼキュータのデフォルトの最大数は1です。

トランザクションで再利用可能なエグゼキュータを使用する場合、エグゼキュータの最大数の値と等しくなるようにエグゼキュータの最小数の値を設定する必要があります。トランザクションで再利用可能なサービスのデフォルト値は1です。

CLIENTS_PER_EXECUTOR <number>

エグゼキュータごとに許可するクライアントの数を指定します。許可するクライアントの数は、サービスがセッションで再利用可能かトランザクションで再利用可能かに応じて変わります。セッションで再利用可能なサービスの場合、エグゼキュータごとのクライアント数のデフォルトは1であり、1より大きい値は指定できません。トランザクションで再利用可能なサービスの場合、エグゼキュータごとのクライアント数のデフォルトは1ですが、1より大きい値も指定できます。CLIENTS_PER_EXECUTORの値は、整数として表します。

使用方法

- OCI サービス以外のサービスを作成すると、SYSPriv 権限を保持する権限付きユーザーのみがサービスの使用を認可されます。GRANT コマンドを使用して、他のユーザーを使用可能にしてください。
- デフォルトの最小数である0（ゼロ）個のエグゼキュータを使用する場合、エグゼキュータのデフォルトの最大数は1です。定義されているエグゼキュータの最小数が0（ゼロ）より大きい場合、エグゼキュータのデフォルトの最大数は、定義されている最小値と等しくなります。たとえば、定義されているエグゼキュータの最小数が5の場合、エグゼキュータのデフォルトの最大数も5です。

例

例 1: V71 という汎用サービスを作成します。

```
SQLSRV> CREATE SERVICE V71 OWNER 'SQLSRV$DEFLT' SQL VERSION 7.1
_SQLSRV> MIN_EXECUTORS 5
_SQLSRV> MAX_EXECUTORS 10;
SQLSRV> START SERVICE V71;
```

DISCONNECT SERVER コマンド

サーバーとの接続を切断します。

書式

```
DISCONNECT SERVER          [ <connect-name> ];  
  
                               <connect-name> ::= <identifier>
```

引数

<connect-name>

接続名。この識別子で、特定のノード上のサーバーに対する接続を一意に識別します。接続名は、識別子として表します。

使用方法

DISCONNECT SERVER コマンドは、CONNECT TO SERVER コマンドと逆の動作をします。このコマンドにより、名前付き接続（接続名が指定されている場合）または現在の接続（接続名が指定されていない場合）が切断されます。

例

例 1: 接続名が eagle であるサーバーとの接続を切断します。

```
SQLSRV> CONNECT TO SERVER AS eagle;  
Connecting to server ...  
Connected  
SQLSRV> DISCONNECT SERVER eagle;
```

DROP コマンド

現在のサーバーで指定されたオブジェクトを削除します。

書式

```
DROP <obj-type> <obj-name>;

<obj-type> ::=DISPATCHER | SERVICE
<obj-name> ::= <identifier>
```

引数

<obj-type>

キーワード DISPATCHER または SERVICE オブジェクト・タイプをそれぞれ使用して、ディスパッチャまたはサービスを指定します。

<obj-name>

削除するオブジェクトの名前。オブジェクト名は、識別子として表します。

使用方法

- オンラインで削除する場合、現在アクティブであるオブジェクトや実行中のオブジェクトは削除できません。つまり、オブジェクトは最初にオンラインで停止しておく必要があります。SHOW CLIENTS コマンドを発行すると、停止および削除する予定のサービスを使用しているクライアント・アプリケーションが存在するかどうかを判別し、そのサービスに接続するクライアントが存在しないことを確認できます。

SQLSRV_MANAGE ユーティリティでは、2 つ以上のノードで同じ構成ファイルが共有されている環境で、ディスパッチャまたはサービスが異なるノードで実行されているときに、ユーザーがそのディスパッチャまたはサービス・オブジェクトをオンラインで削除することを防止できません。このとき、削除されたがまだ実行中のディスパッチャまたはサービスを表示すると、SQLSRV_MANAGE ユーティリティによって次のような警告メッセージが表示されます。

```
SQLSRV> SHOW service <obj_name>;
*****

** This Service has been deleted from the config file. **
** It will not exist after it is shut down.             **
*****
```

- 同じファイルを使用している実行中のサーバーが存在する場合、構成ファイルをオフラインで変更しないことをお勧めします。この場合、SQLSRV_MANAGE ユーティリティなどでは、ディスパッチャまたはサービスが実行されているときに、ユーザーがそのディスパッチャまたはサービス・オブジェクトをオフラインで削除することを防止できません。

オフラインで削除されたサービスまたはディスパッチャを使用しているクライアント・アプリケーションは、サーバー・オブジェクトからの接続が切断されるまで、そのオブジェクトを使用し続けます。ただし、クライアント・アプリケーションによりサーバーとの接続が切断されると、削除されたディスパッチャまたはサービスに再接続されるとはなりません。削除されたオブジェクトを停止する前に、SHOW コマンドを使用すると、前述のリスト項目に示されているような削除済オブジェクトに関するメッセージが表示されます。

- DROP コマンドでは、指定されたオブジェクトが構成ファイルから削除されます。

例

例 1: database_3 サービス・オブジェクトを削除します。

```
SQLSRV> SHUTDOWN SERVICE database_3;  
SQLSRV> DROP SERVICE database_3;
```

例 2: disp_tcpip ディスパッチャ・オブジェクトを削除します。

```
SQLSRV> SHUTDOWN DISPATCHER disp_tcpip;  
SQLSRV> DROP DISPATCHER disp_tcpip;
```

DROP SERVER コマンド

構成ファイルとともに現在のサーバーを削除します。

書式

DROP SERVER:

使用方法

- 現在アクティブであるサーバーは、削除できません。サーバーは最初にオンラインで停止してから、オフラインで削除する必要があります。
- DROP SERVER コマンドは、オフライン操作です。サーバーには接続されません。
- DROP SERVER コマンドでは、構成ファイルが削除されます。

例

例 1: 現在のサーバー・オブジェクトを削除します。

```
SQLSRV> CONNECT SERVER;  
Connecting to server ...  
Connected  
SQLSRV> SHUTDOWN SERVER;  
SQLSRV> DISCONNECT SERVER;  
SQLSRV> SET CONFIG_FILE 'my_config_file';  
SQLSRV> DROP SERVER;  
Configuration file will be deleted, continue? (N) y  
Configuration file deleted
```

EXIT コマンド

SQLSRV_MANAGE 環境を終了します。

書式

`EXIT[;]`

使用方法

- 次の2つの方法で SQLSRV_MANAGE 環境またはユーティリティを終了できます。

- EXIT コマンドを使用して終了
- 最後の入力ソースでファイルの終わり (EOF) に到達したときに終了

SQLSRV_MANAGE ユーティリティを対話的に使用している場合、[Ctrl] キーを押しながら [Z] キーを押すことで SQLSRV_MANAGE ユーティリティを終了できます。

-input ファイル・スイッチを使用してコマンドラインで入力ファイルを指定する場合、そのファイルが最後の入力ソースであり、ファイルの終わり (EOF) に到達すると、SQLSRV_MANAGE は終了します。

- 終端を示すセミコロン (;) の使用は、オプションです。

例

例 1: SQLSRV_MANAGE 環境を終了します。

```
SQLSRV> EXIT
```

EXTRACT コマンド

アクティブな構成ファイルからサーバー・オブジェクト定義を抽出し、それらを SQL/Services コマンド・スクリプト（出力ファイルが定義されている場合）または現在の出力デバイスに書き込みます。スクリプトを使用して、サーバー、ディスパッチャおよびサービスを再作成できます。

書式

```
EXTRACT          keyword [option];
```

引数

keyword [option]

キーワードには、次のいずれかを指定できます。

- **SERVER**
サーバーの定義を抽出します。このキーワードにオプションはありません。
- **DISPATCHER [disp_name]**
disp_name を省略するか、アスタリスク (*) で表すと、すべてのディスパッチャの定義が抽出されます。disp_name を指定すると、そのディスパッチャの定義のみが抽出されます。
- **SERVICE [service_name]**
service_name を省略するか、アスタリスク (*) で表すと、すべてのサービスの定義が抽出されます。service_name を指定すると、そのサービスの定義のみが抽出されます。

使用方法

ファイルに定義を抽出するには、SQLSRV_MANAGE コマンドで **-output** スイッチを使用するか、EXTRACT コマンドを発行する前に OPEN コマンドを発行します。

例

例 1: サービス OCI_AAA の定義を抽出し、セッションに現在定義されている出力デバイスに情報を表示します。

```
$ SQLSRV_MANAGE71
SQLSRV> EXTRACT SERVICE OCI_AAA;
Create Service OCI_AAA
  Owner                'AAA'
  Protocol              OCI
  SQL version          7.1
  Autostart            off
  Process_initialization 'DBD_USER6:[JONES]INIT_OCI_ENG70.COM'
  ATTACH 'filename DBD_USER6:[JONES]mf_personnel'
  Reuse scope is      SESSION
  Database Authorization CONNECT USERNAME
  SQL_init_file       'DBD_USER6:[JONES]init.sql'
  Application Transaction Usage SERIAL
  Idle Executor Timeout 1800
  Min Executors       1
  Max Executors       10
  Clients Per Executor 1
;
Grant use on service OCI_AAA
To 'AAA'
```

```

;
SQLSRV>

```

例 2: ディスパッチャ定義全体を抽出し、それらを出力ファイルに書き込みます。

```

$ SQLSRV_MANAGE71 -OUTPUT A.SQL
SQLSRV> EXTRACT DISP;
SQLSRV> EXIT

$TYPE A.SQL
Create Dispatcher SQLSRV_DISP
  Autostart                on
  Max connects              101
  Idle User Timeout        0
  network_port spxipx port_id 0x84b1      protocol SQLServices
  network_port DECnet object 81          protocol SQLServices
  network_port tcpip  port_id 118        protocol SQLServices
  Log path                  'SYS$MANAGER:'
  Dump path                 'SYS$MANAGER:'
;
Create Dispatcher RMU_DISP
  Autostart                on
  Max connects              100
  Idle User Timeout        0
  network_port tcpip  port_id 1571       protocol Native
  Log path                  'SYS$MANAGER:'
  Dump path                 'SYS$MANAGER:'
;
Create Dispatcher OCI_DISP
  Autostart                on
  Max connects              35
  Idle User Timeout        0
  network_port sqlnet listener oci_listener protocol OCI
  Log path                  'SYS$MANAGER:'
  Dump path                 'SYS$MANAGER:'
;
Create Dispatcher SQLSRV_MANAGE
  Autostart                off
  Max connects              100
  Idle User Timeout        0
  network_port spxipx port_id 0x84b1      protocol SQLServices
  network_port DECnet object 81          protocol SQLServices
  network_port tcpip  port_id 118        protocol SQLServices
  Log path                  'SYS$MANAGER:'
  Dump path                 'SYS$MANAGER:'
;

```

例 3: OPEN コマンドを使用して出力ファイルに定義を抽出します。

```

$ SQLSRV_MANAGE71
SQLSRV> OPEN aaa.sql;
SQLSRV> EXTRACT SERVICE OCI_AAA;
SQLSRV> CLOSE;

```

GRANT USE ON SERVICE コマンド

ユーザーにサービスの USE 権限記述子を付与します。このコマンドを使用して、識別子に USE 権限を付与し、その特定の識別子を保持するユーザーに指定のサービスに対するアクセスを許可します。

書式

```
GRANT USE ON SERVICE <service-name-list> TO <grant-element-list>;

<service-name-list> ::= <service-name> [ , <service-name> ] ...
<service-name> ::= <identifier>
<grant-element-list> ::= <grant-element> [ , <grant-element> ] ...
<grant-element> ::= { PUBLIC | PRIVILEGED_USER
    | [ USER[S] ] <user-name> [ , <user-name> ] ...
    | IDENTIFIER[S] <identifier-name> [ , <identifier-name> ] ...
    | GROUP[S] <group-name> [ , <group-name> ] ... }
<user-name> ::= { <quoted-string> | <identifier> }
<identifier-name> ::= { <quoted-string> | <identifier> }
<group-name> ::= { <quoted-string> | <identifier> }
```

引数

<service-name-list>

GRANT USE ON SERVICE コマンドの操作対象となるサービス名をリストします。サービス名は、識別子として表します。

<grant-element-list>

GRANT USE ON SERVICE コマンドの操作対象となる権限付与要素をリストします。権限付与リスト要素には、キーワード PUBLIC または PRIVILEGED_USER、ユーザー名のリスト、識別子名のリスト、またはグループ名のリストを指定できます。PRIVILEGED_USER は、SYSPRV 権限（デフォルト権限または付与された権限）を保持するユーザーとして定義されます。ユーザー名、識別子名またはグループ名は、引用符付き文字列または識別子として表します。

使用方法

- リリース 7.0 以上では、Oracle SQL/Services により、単一の権限記述子 USE が付与されません。
- 新規ユーザーに対する USE 権限記述子の付与が効果を発揮するのは、権限の変更が完了した後で、ユーザーが Oracle SQL/Services を使用しようと次に試行したときです。たとえば、USE 権限記述子を付与された新規ユーザーは、次の試行時に Oracle SQL/Services を使用できます。
- キーワード IDENTIFIER[S] または GROUP[S] を使用すると、指定した識別子が権限付与済の識別子のリストに追加され、その特定の識別子を保持するユーザーは指定のサービスに対するアクセスを許可されます。キーワード IDENTIFIER[S] または GROUP[S] を省略すると、指定したユーザー名にサービスを使用するためのアクセス権が付与されます。

例

例 1: general サービスの USE 権限記述子を PUBLIC に付与します。

```
SQLSRV> GRANT USE ON SERVICE general TO PUBLIC;
```

例 2: database_2 サービスの USE 権限記述子を fred と wilma に付与します。

```
SQLSRV> GRANT USE ON SERVICE database_2 TO fred,wilma;
```

例 3: SQLSRV_MANAGE システム管理サービスの USE 権限記述子を fred と wilma に付与します。

```
SQLSRV> GRANT USE ON SERVICE sqlsrv_manage TO fred,wilma;
```

例 4: SQLSRV_MANAGE システム管理サービスの USE 権限記述子を識別子 payroll_dba および operator に付与します。

```
SQLSRV> GRANT USE ON SERVICE sqlsrv_manage  
_SQLSRV> TO IDENTIFIERS payroll_dba,operator;
```

HELP コマンド

SQLSRV_MANAGE 環境内のトピックに関するヘルプを表示します。

書式

HELP [<help-keyword>] ... [;]

 <help-keyword> ::= <identifier>

引数

<help-keyword>

ヘルプ・キーワード。ヘルプ・キーワードは、識別子として表します。

使用方法

終端を示すセミコロン (;) の使用は、オプションです。

例

例 1: SQLSRV_MANAGE 環境内のトピックに関するヘルプを表示します。

```
SQLSRV> HELP
```

KILL EXECUTOR コマンド

指定したエグゼキュータを中断します。

書式

```
KILL EXECUTOR      { PID <process-id> | <executor-name> };  
  
                    <process-id> ::= <number>  
                    <executor-name> ::= <identifier>
```

引数

{PID <process-id> | <executor-name>}

プロセス ID またはエグゼキュータ名。プロセス ID は、整数として表します。また、10 進形式または 16 進形式で表現できます。エグゼキュータ名は、識別子として表します。エグゼキュータ名を確認するには、SHOW CLIENTS FULL コマンドを実行します。

使用方法

- プロセス ID は、10 進形式または 16 進形式で表現できます。プロセス ID を 16 進形式で表現するには、プロセス ID 値の前に値 0x または 0X を付けます (0x0000088a など)。
- エグゼキュータはオンライン操作でのみ中断できます。つまり、サーバー上で実行中のエグゼキュータを中断するには、その実行中のサーバーに接続する必要があります (CONNECT TO SERVER コマンド)。

例

例 1: プロセス ID (16 進形式) を指定してエグゼキュータを中断します。

```
SQLSRV> KILL EXECUTOR PID 0x0000072a;
```

例 2: プロセス ID (10 進形式) を指定してエグゼキュータを中断します。

```
SQLSRV> KILL EXECUTOR PID 324693;
```

例 3: 名前を指定してエグゼキュータを中断します。

```
SQLSRV> KILL EXECUTOR generi004000280;
```

OPEN コマンド

SQLSRV_MANAGE 環境で出力ファイルを開きます。SQLSRV_MANAGE による後続の出力は、エラー・メッセージも含めこのファイルに書き込まれます。

書式

```
OPEN <file-spec>;

<file-spec> ::= { <identifier> | <quoted-string> }
```

引数

<file-spec>

出力ファイル名。ファイル名は、識別子または引用符付き文字列として表します。

使用方法

OPEN コマンドにより、指定したファイルが作成され、後続の出力はすべてそのファイルに書き込まれます。OPEN コマンドを入力すると、現在の出力ファイルは暗黙的に閉じられます (出力ファイルがすでに開かれている場合)。

例

例 1: 出力ファイルを開きます。

```
SQLSRV> OPEN test_file;
```

例 2: 出力ファイルを開き、そのファイルにディスパッチャ定義を抽出します。

```
SQLSRV> OPEN ocidisp.lis;
SQLSRV> EXTRACT disp oci_disp;
SQLSRV> CLOSE;
SQLSRV> TY ocidisp.lis;
Create Dispatcher OCI_DISP
Autostart                on
Max connects              35
Idle User Timeout        0
network_port sqlnet listener oci_listener protocol OCI
Log path                  'SYS$MANAGER:'
Dump path                 'SYS$MANAGER:'
```

RESTART SERVER コマンド

現在のサーバーを再起動します。

書式

```
RESTART SERVER [AUTOSTART {ON | OFF}];
```

引数

AUTOSTART {ON | OFF}

RESTART SERVER コマンドの発行時に、他のサーバー・オブジェクト（ディスパッチャおよびサービス）を自動的に再起動するかどうかを決定します。デフォルトは ON です。この引数を ON に指定した場合、各オブジェクトの AUTOSTART 引数の値も ON に設定されていれば、他のサーバー・オブジェクトも自動的に再起動（停止して再び起動）します。他のサーバー・オブジェクトを再起動しない場合、RESTART SERVER コマンドで AUTOSTART 属性の値を OFF に指定します。AUTOSTART OFF 属性設定により、各オブジェクトの AUTOSTART 属性設定は上書きされるため、サーバー・オブジェクトの再起動後に各オブジェクトを個別に起動できません。

使用方法

- サーバーはオンライン操作でのみ再起動できます。つまり、サーバーを再起動するには、そのサーバーに接続する必要があります（CONNECT TO SERVER コマンド）。
- サーバーを再起動するには、RESTART SERVER コマンドを使用してください。デフォルトでは、現在のサーバーのすべてのサーバー・コンポーネント（ディスパッチャおよびサービス）も再起動されます（それらのサーバー・オブジェクトの AUTOSTART 引数が定義上 OFF に指定されていない場合）。

例

例 1: 現在のサーバーを再起動します。

```
SQLSRV> CONNECT TO SERVER;  
Connecting to server ...  
Connected  
SQLSRV> ALTER SERVER MAX_SHARED_MEMORY_SIZE 4000;  
%DBS-S-ALTER_RESTART, Restart object to have altered settings take effect  
SQLSRV> RESTART SERVER;  
Disconnected from Server  
SQLSRV> CONNECT SERVER;  
Connecting to server ...  
Connected
```

REVOKE USE ON SERVICE コマンド

ユーザーに付与されているサービスの USE 権限記述子を取り消します。識別子に付与されている USE を取り消して、その識別子を保持するユーザーごとに指定のサービスに対するアクセス権を削除できます。

書式

```
REVOKE USE ON SERVICE <service-name-list> FROM <grant-element-list>;

<service-name-list> ::= <service-name> [ , <service-name> ] ...
<service-name> ::= <identifier>
<grant-element-list> ::= <grant-element> [ , <grant-element> ] ...
<grant-element> ::= { PUBLIC | PRIVILEGED_USER
| [ USER[S] ] <user-name> [ , <user-name> ] ...
| IDENTIFIER[S] <identifier-name> [ , <identifier-name> ] ...
| GROUP[S] <group-name> [ , <group-name> ] ... }
<user-name> ::= { <quoted-string> | <identifier> }
<identifier-name> ::= { <quoted-string> | <identifier> }
<group-name> ::= { <quoted-string> | <identifier> }
```

引数

<service-name-list>

REVOKE USE ON SERVICE コマンドの操作対象となるサービス名をリストします。サービス名は、識別子として表します。

<grant-element-list>

REVOKE USE ON SERVICE コマンドの操作対象となる権限付与要素をリストします。権限付与リスト要素には、キーワード PUBLIC または PRIVILEGED_USER、ユーザー名のリスト、識別子名のリスト、またはグループ名のリストを指定できます。PRIVILEGED_USER は、SYSPRV 権限（デフォルト権限または付与された権限）を保持するユーザーとして定義されます。ユーザー名、識別子名またはグループ名は、引用符付き文字列または識別子として表します。

使用方法

- リリース 7.0 以上では、Oracle SQL/Services により、単一の権限記述子 USE が取り消されます。
- 既存のユーザーに付与された USE 権限記述子の取消しが効果を発揮するのは、権限の変更が完了した後で、ユーザーが Oracle SQL/Services を使用しようと次に試行したときです。たとえば、USE 権限記述子を取り消されたが、現在 Oracle SQL/Services を使用中のユーザーが Oracle SQL/Services を使用できなくなるのは、一度接続を切断してからサービスに再接続を試みた後です。
- キーワード IDENTIFIER[S] または GROUP[S] を使用すると、指定した識別子が権限付与済の識別子のサービス・リストから削除されます。キーワード IDENTIFIER[S] または GROUP[S] を省略すると、指定したユーザー名が権限付与済のユーザー名のサービス・リストから削除されます。

特定のユーザー名ごとにサービスの使用権を取り消す場合、そのユーザーがサービスの使用権を付与された識別子を保持していれば、そのユーザーは引き続きサービスにアクセスできます。同様に、特定の識別子ごとにサービスの使用権を取り消す場合、その識別子を保持するユーザーは、そのユーザー名にサービスの使用権が付与されていれば、引き続きサービスにアクセスできます。

例

例 1: general サービスの USE 権限記述子を PUBLIC から削除します。

```
SQLSRV> REVOKE USE ON SERVICE general FROM PUBLIC;
```

例 2: database_3 サービスの USE 権限記述子を fred と wilma から削除します。

```
SQLSRV> REVOKE USE ON SERVICE database_3 FROM fred,wilma;
```

例 3: SQLSRV_MANAGE システム管理サービスの USE 権限記述子を fred と wilma から削除します。

```
SQLSRV> REVOKE USE ON SERVICE sqlsrv_manage FROM fred,wilma;
```

例 4: SQLSRV_MANAGE システム管理サービスの USE 権限記述子を識別子名 payroll_dba および operator から削除します。

```
SQLSRV> REVOKE USE ON SERVICE sqlsrv_manage  
_SQLSRV> FROM IDENTIFIERS payroll_dba,operator;
```

SET CONFIGURATION_FILE コマンド

サーバーの起動またはサーバーのオフライン変更に使用するサーバー構成ファイルを選択できます。後続のすべての管理コマンドは、構成ファイルにのみ書き込まれ、実行中のサーバーには影響しません（ただし、GRANT USE および REVOKE USE コマンドと、再起動されたデイスパッチャおよびサービスは除きます）。

書式

```
SET CONFIGURATION_FILE <file-name>;  
  
<file-name> ::= <quoted-string>
```

引数

<file-name>

構成ファイル名。ファイル名は、識別子または引用符付き文字列として表します。

使用方法

- CONFIG_FILE は、キーワード CONFIGURATION_FILE と同義です。
- SQLSRV_MANAGE ユーティリティを起動すると、次のようにデフォルトの構成ファイル名が確定されます。

SYS\$MANAGER:SQLSRV_CONFIG_FILE71.DAT

デフォルトを上書きするには、SQLSRV_CONFIG_FILE71 論理名を設定するか、SET CONFIGURATION_FILE コマンドに異なるファイル名を指定します。
- SHOW SETTINGS コマンドでは、オフライン変更の操作対象となる構成ファイルが表示されます。SHOW SERVER コマンドでは、オンライン変更の操作対象となる構成ファイルが表示されます。
- SET_CONFIG_FILE コマンドを発行し、存在しない構成ファイルを指定すると、今すぐに構成ファイルを作成するかどうかを尋ねられます。デフォルトは NO です。SET CONFIRM コマンドが OFF に設定されている場合、プロンプトは表示されません。SHOW SETTINGS コマンドでは、現在の設定と、この新規構成ファイルのファイル指定が表示されます。CREATE SERVER コマンドを発行すると、この構成ファイルを使用してサーバーが作成されます。
- SET CONFIG_FILE コマンドを使用して構成ファイルを変更する場合、すべての変更はオフラインで実行され、実行中のサーバーには影響しません（GRANT および REVOKE コマンドによる変更を除く）。サーバーの構成ファイルに対する変更は、変更されたオブジェクトを再起動することで実行中のサーバーに適用されます。

例

例 1: 構成ファイルを設定します。

```
SQLSRV> SET CONFIG_FILE 'my_config_file';
```

SET CONFIRM コマンド

SQLSRV_MANAGE 環境で ON に設定すると、デフォルトの出力デバイスに確認プロンプトが表示されます。

書式

```
SET CONFIRM {ON|OFF};
```

引数

{ON|OFF}

確認を ON に設定すると、特定の管理操作の確認をを求める確認プロンプトがデフォルトの出力デバイスに表示されます。確認を OFF に設定すると、確認プロンプトは表示されず、特定の管理操作の確認も求められません。デフォルトは ON です。

使用方法

- SET CONFIRM コマンドを ON (デフォルト) に設定し、確認プロンプトの対象となる SQLSRV_MANAGE コマンドを発行すると、デフォルトの出力デバイスに確認プロンプトが表示されます。たとえば、サーバーを停止して削除し、SET_CONFIG_FILE コマンドを発行して存在しない構成ファイルを指定すると、今すぐに構成ファイルを作成するかどうかを尋ねるプロンプトが表示されます。デフォルトは NO (今すぐに構成ファイルを作成しない) です。SET CONFIRM コマンドを OFF に設定すると、この操作の確認プロンプトは表示されません。
- SHOW SETTINGS コマンドでは、他の項目とともに SET CONFIRM コマンドの現在の設定が表示されます。

例

例 1: デフォルトの出力デバイスに確認プロンプトを表示しません。

```
SQLSRV> SET CONFIRM OFF;
```

SET CONNECTION コマンド

サーバーを管理できるように、指定した接続を現在の接続として確立します。

書式

```
SET CONNECT[ION]    [<connect-name>];  
  
                   <connect-name> ::=<identifier>
```

引数

<connect-name>

接続名。特定のノード上のサーバーに対する接続を一意に識別する識別子です。接続名は、識別子として表します。

使用方法

SET CONNECT コマンドでは、管理対象サーバーに対する接続を切り替えて、単一の SQLSRV_MANAGE セッションで複数のサーバーを管理できます。

サーバーをオンライン管理する場合、最初に CONNECT TO SERVER コマンドを使用してサーバーに接続する必要があります。CONNECT TO SERVER コマンドを使用してサーバーに対する新規接続を確立すると、その新規接続は現在の接続になります。すべてのオンライン・システム管理コマンドは、現在の接続に対して動作します。複数の CONNECT TO SERVER コマンドを発行することで、複数のサーバーに対する接続を確立できます。

その後、SET CONNECT コマンドを使用して、管理するサーバーを選択します。

DISCONNECT SERVER コマンドを使用すると、サーバーとの接続を切断できます。その時点で残っている接続があれば、そのうちの 1 つが現在の接続になります。

例

例1: ノード EAGLE からノード EAGLE および FALCON の 2 つのサーバーを管理します。

```

SQLSRV> CONNECT SERVER AS EAGLE;
Connecting to server ...
Connected
SQLSRV> CONNECT SERVER AS FALCON NODE FALCON
SQLSRV> USER 'dbmgr' USING 'password';
Connecting to server ...
Connected
SQLSRV> SHOW CONNECT;
Active connections:
CURRENT: FALCON
    Service: SQLSRV_MANAGE
    User: dbmgr Node: FALCON Local: No
    Transport: DECNET Object: SQLSRV_SERVER
    Request bufsize: 1024 Response bufsize: 1024
EAGLE
    Service: SQLSRV_MANAGE
    User: <unknown> Node: EAGLE Local: Yes
    Transport: DECNET Object: SQLSRV_SERVER
    Request bufsize: 1024 Response bufsize: 1024

SQLSRV> SHOW SERVICES;

```

Name	State	C l i e n t s			E x e c u t o r s		
		Per-Exec	Max	Active	Min	Max	Running
RMU_SERVICE	RUNNING	1	100	0	4	100	4
GENERIC	RUNNING	1	10	0	2	10	2
SQLSRV_MANAGE	RUNNING	100	0	1	0	0	0

```

SQLSRV> SET CONNECT EAGLE;
SQLSRV> SHOW CONNECT;
Active connections:
FALCON
    Service: SQLSRV_MANAGE
    User: dbmgr Node: FALCON Local: No
    Transport: DECNET Object: SQLSRV_SERVER
    Request bufsize: 1024 Response bufsize: 1024
CURRENT: EAGLE
    Service: SQLSRV_MANAGE
    User: <unknown> Node: EAGLE Local: Yes
    Transport: DECNET Object: SQLSRV_SERVER
    Request bufsize: 1024 Response bufsize: 1024

SQLSRV> SHOW SERVICES;

```

Name	State	C l i e n t s			E x e c u t o r s		
		Per-Exec	Max	Active	Min	Max	Running
V71	RUNNING	1	20	2	5	20	5
RMU_SERVICE	RUNNING	1	100	3	4	100	4
GENERIC	RUNNING	1	50	5	20	50	20
SQLSRV_MANAGE	RUNNING	100	0	1	0	0	0

SET OUTPUT コマンド

SQLSRV_MANAGE 環境で ON に設定すると、出力がデフォルト・デバイスに送られます。

書式

```
SET OUTPUT          {ON|OFF};
```

引数

{ON | OFF}

出力を ON に設定すると、出力はデフォルトの出力デバイスに送られます。出力を OFF に設定すると、出力は抑止されます。

使用方法

なし。

例

例 1: 出力をデフォルト・デバイスに設定します。

```
SQLSRV> SET OUTPUT ON;
```

SET VERIFY コマンド

SQLSRV_MANAGE 環境での読取り時に、デフォルトの出力デバイスにコマンド・ファイルの入力を表示します。

書式

```
SET VERIFY          {ON|OFF};
```

引数

{ON | OFF}

検証を ON に設定すると、デフォルトの出力デバイスにコマンド・ファイルの入力が表示されます。検証を OFF に設定すると、デフォルトの出力デバイスにコマンド・ファイルの入力が表示されません。

使用方法

なし。

例

例 1: 読取り時に、デフォルトの出力デバイスにコマンド・ファイルの入力を表示します。

```
SQLSRV> SET VERIFY ON;
```

SHOW CLIENTS コマンド

構成におけるサービスのアクティブなユーザーを表示します。

書式

```
SHOW CLIENTS      [[FOR] <name-spec> [FULL]];
```

```
<name-spec> ::= { * | [SERVICE] <service-name-list>
                | [USERNAME] <username-list> | [PID] <executor-pid> }
<service-name-list> ::= <identifier> [ , <identifier> ] ...
<user-name-list> ::= <identifier> [ , <identifier> ] ...
<executor-pid> ::= <number>
```

引数

<name-spec>

名前の指定。次の項目を表示できます。

- 1つ以上のサービスに接続されたクライアント
- 特定のエグゼキュータ PID に対応するすべてのクライアント
- サーバーに接続されたすべてのクライアント（特定のユーザー名またはユーザー名のリストを使用）
- 特定のエグゼキュータに接続されたクライアント

デフォルトは * であり、すべてのサービスのすべてのクライアントが表示されます。

FULL

各クライアント情報の完全な説明を表示します。デフォルトでは、各クライアントの簡単な情報（1行の出力）が表示されます。サービス名を指定しないと、SQLSRV_MANAGE では、サービス名でグループ化されたクライアントが表示されます。

使用方法

- すべてのサービスのすべてのクライアントを表示するには、SHOW CLIENTS コマンドを使用して [FOR] [SERVICE] キーワードを指定しないか、アスタリスク (*) を指定します。どちらの方法でも、すべてのサービスのすべてのクライアントが表示されます。次に例を示します。


```
SQLSRV> SHOW CLIENTS;
SQLSRV> SHOW CLIENTS *;
```
- 特定のエグゼキュータ PID に対応するすべてのクライアントを表示するには、SHOW CLIENTS FOR PID コマンドを指定して、エグゼキュータ PID を指定します。
- エグゼキュータ PID は、10 進形式または 16 進形式で表現できます。エグゼキュータ PID を 16 進形式で表現するには、エグゼキュータ PID 値の前に値 0x または 0X を付けます (0x0000088a など)。
- セッションで再利用可能なサービスにより処理されるクライアント接続は、次の 3 つの状態のいずれかを取ります (詳細は 3.2 項を参照)。
 - 実行中 (バインド中) : クライアントは実行中であり、エグゼキュータへのバインド・プロセスが進行中です。
 - 実行中 (バインド済) : クライアントは実行中であり、エグゼキュータにバインド済です。
 - キャンセル中 : クライアント接続の切断プロセスが進行中です。
- トランザクションで再利用可能なデータベース・サービスにより処理されるクライアント接続は、次の 5 つの状態のいずれかを取ります (詳細は 3.2 項を参照)。
 - 実行中 (バインド中) : クライアントは実行中であり、エグゼキュータへのバインド・プロセスが進行中です。
 - 実行中 (バインド済) : クライアントは実行中であり、エグゼキュータにバインド済です。
 - 実行中 (バインド解除済) : クライアントはリクエストを送信しておらず、したがってエグゼキュータにバインドされていませんが、そのエグゼキュータに対する接続は続いています。
 - キャンセル中 (バインド中) : クライアントは、バインドされた接続が解除されるという通知をエグゼキュータに送信中です (この操作はキャンセル操作に先行します)。
 - キャンセル中 : クライアント接続の切断プロセスが進行中です。
- このコマンドでは、管理サービスを使用している管理クライアントも表示されます。SHOW CLIENTS コマンドにより、サーバー・システム管理者は、他のサーバー・システム管理者がサーバーに接続し、管理サービスを使用しているかどうかを確認できます。
- このコマンドでは、エグゼキュータ・ログ・ファイルおよびエラー・ファイルの実際の場合と、エグゼキュータ・ダンプ・ファイルの場所 (作成されている場合) が表示されます。

例

例 1: generic という汎用サービスのクライアントを表示し、簡単な説明を出力します。

```
SQLSRV> SHOW CLIENTS FOR SERVICE generic;
Service: GENERIC
```

Connect	Client	State	Executor	Application
Username	Node		PID	
User1	123.0.0.1	RUNNING BOUND	28c0c4e6	Personnel
User2	121.0.0.1	RUNNING BOUND	30b0a4d5	Personnel

例 2: すべてのサービスのすべてのユーザー名に対応するクライアントを表示し、簡単な説明を出力します。

```
SQLSRV> SHOW CLIENTS;
Service: SS_SERV
```

Connect	Client	State	Executor	Application
Username	Node		PID	
	123.0.0.1	RUNNING BINDING	00000000	
	123.0.0.1	RUNNING BINDING	00000000	

```
Service: MMS
```

Connect	Client	State	Executor	Application
Username	Node		PID	
Rdbuser1	NODE1	RUNNING BOUND	00001045	Personnel

```
Service: SQLSRV_MANAGE
```

Connect	Client	State	Executor	Application
Username	Node		PID	
User1	NODE2	RUNNING BOUND	00000af2	SQLSRV_MANAGE

例 3: generic という汎用サービスのクライアントを表示し、完全な説明を出力します。

```
SQLSRV> SHOW CLIENTS FOR SERVICE generic FULL;
Client Connect Username sqsapim1
```

```
Service:    GENERIC
Application: Personnel
State:     RUNNING BOUND
Node:     12.34.567.89
Executor:  GENERI0050002
Executor PID: 543173877 0X20602cf5
Log File:  SYS$SYSROOT: [SYSMGR] SQS_EAGLES_GENERI0050002.LOG
Dump File: SYS$SYSROOT: [SYSMGR] SQS_EAGLES_GENERI0050002.DMP
```

```
Client Connect Username sqsapim2
```

```
Service:    GENERIC
Application: Personnel
State:     RUNNING BOUND
Node:     LOCAL:.mypc
Executor:  GENERI0080004
Executor PID: 543173877 0X20602cf5
Log File:  SYS$SYSROOT: [SYSMGR] SQS_EAGLES_GENERI0080004.LOG
Dump File: SYS$SYSROOT: [SYSMGR] SQS_EAGLES_GENERI0080004.DMP
```

SHOW CONNECTIONS コマンド

現在のサーバーに関する情報を表示します。

書式

```
SHOW CONNECT[ION];
```

使用方法

- CONNECTS は、キーワード CONNECTIONS と同義です。
- SHOW CONNECT[ION]S コマンドでは、SQLSRV_MANAGE ユーティリティが各サーバーに対して保持するすべてのアクティブな管理接続に関する情報が表示されます。最初に SHOW CONNECT[ION] コマンドを使用して現在の接続を確認してから、追加のサーバー管理コマンドを発行してください。

例

例 1: 現在のサーバーと、他のサーバーに対する接続に関する情報を表示します。

```
SQLSRV> SHOW CONNECT;  
Active connections:  
CURRENT: SQLSRV_MANAGE  
    Service: SQLSRV_MANAGE  
    User: system Node: hawk Local: No  
    Transport: TCP/IP Port-id: 2199  
    Request bufsize: 1024 Response bufsize: 1024  
  
SQLSRV_MANAGE  
    Service: SQLSRV_MANAGE  
    User: system Node: falcon Local: Yes  
    Transport: TCP/IP Port-id: 2199  
    Request bufsize: 1024 Response bufsize: 1024
```

SHOW DISPATCHER コマンド

ディスパッチャ・オブジェクトの静的定義と、現在のサーバーでの稼働状態を表示します。

書式

```
SHOW DISPATCHER [ <dispatcher-spec> ] ;
```

```
<dispatcher-spec> ::= { * | <dispatcher-name-list> }
```

```
dispatcher-name-list ::= <identifier> [ , <identifier> ] ...
```

引数

<dispatcher-spec>

ディスパッチャ・オブジェクトの指定。1つ以上のディスパッチャ・オブジェクト名を指定するか、アスタリスク (*) で指定します。* を指定すると、すべてのディスパッチャ・オブジェクト名に関する情報が表示されます。

使用方法

- ディスパッチャ・オブジェクトを指定しないと、すべてのディスパッチャ・オブジェクトに関する情報が表示されます。
- ディスパッチャ状態およびネットワーク・ポート状態は、次の3つの状態のいずれかを取ります。
 - 実行中: ディスパッチャまたはディスパッチャ・ネットワーク・ポートは実行中です。
 - 非アクティブ: ディスパッチャまたはディスパッチャ・ネットワーク・ポートは停止しています。
 - 不明: 管理クライアントがオンラインでサーバーに接続していないため、ディスパッチャとそのネットワーク・ポートの状態を特定できません。管理クライアントでは、SET CONFIG_FILE コマンドを使用してサーバーをオフラインで管理しています。CONNECT TO SERVER コマンドを使用してオンラインでサーバーに接続し、ディスパッチャ状態とネットワーク・ポート状態を確認してください。
- 実行中のサーバーとその構成ファイル間の属性に差異が存在する場合、SHOW DISPATCHER コマンドでは出力表示の最後にその差異が示されます。また、サーバーの再起動時に、構成ファイルに含まれるサーバーのディスパッチャ定義と一致するよう実行中のサーバーのディスパッチャが更新されることが示されます。
- このコマンドでは、ディスパッチャ・ログ・ファイルおよびエラー・ファイルの実際の場所と、ディスパッチャ・ダンプ・ファイルの場所（作成されている場合）が表示されます。

例

例 1: sqlsrv_disp デイスペッチャの情報を表示します。

```
SQLSRV> CONNECT TO SERVER NODE hawk USER system USING password;
Connecting to server ...
Connected
SQLSRV> SHOW DISPATCHER sqlsrv_disp;
Dispatcher SQLSRV_DISP
  State:                RUNNING
  Autostart:            on
  Max connects:         100 clients
  Idle user Timeout:   <none>
  Max client buffer size: 5000 bytes
  Network Ports:
    SPX/IPX port      0x84b1          (State) (Protocol)
    DECnet object    81              Running  SQL/Services
    TCP/IP port      118              Running  SQL/Services
    SQL*Net listener FUBAR          Running  SQL/Services
  Log Path:            USER1:[SQLSRV_TEST1]
  Dump path:          USER1:[SQLSRV_TEST1]
  Log File:           USER1:[SQLSRV_TEST1]:SQS_EAGLE_SQLSRV_DIS100380.LOG
  Dump File:          USER1:[SQLSRV_TEST1]:SQS_EAGLE_SQLSRV_DIS1003.DMP
```

例 2: デイスペッチャの再起動後に確定されるログ・パスとダンプ・パスの変更された値を表示します。

```
SQLSRV> SHOW DISP SQLSRV_DISP;
Dispatcher SQLSRV_DISP
.
.
.
Log Path:            USER1:[SQLSRV_TEST1]
Dump path:          USER1:[SQLSRV_TEST1]
.
.
.

** This Dispatcher will be updated as follows when it is restarted **
Log path:           USER1:[SQLSRV_TEST1.AAA]
Dump path:          USER1:[SQLSRV_TEST2.BBB]
```

SHOW SERVER コマンド

サーバー・オブジェクトの静的定義と、その稼働状態を表示します。

書式

SHOW SERVER;

使用方法

- サーバー・ネットワーク・ポート状態は、次の3つの状態のいずれかを取ります。
 - 実行中:サーバー・ネットワーク・ポートは実行中です。
 - 非アクティブ:サーバー・ネットワーク・ポートは停止しています。
 - 不明:管理クライアントがオンラインでサーバーに接続していないため、サーバー・ネットワーク・ポートの状態を特定できません。管理クライアントでは、**SET CONFIG_FILE** コマンドを使用してサーバーをオフラインで管理しています。**CONNECT TO SERVER** コマンドを使用してオンラインでサーバーに接続し、ネットワーク・ポート状態を確認してください。
- 実行中のサーバーとその構成ファイル間の属性に差異が存在する場合、**SHOW SERVER** コマンドでは出力表示の最後にその差異が示されます。また、サーバーの再起動時に、構成ファイルに含まれるサーバーの定義と一致するよう実行中のサーバーが更新されることが示されます。
- サーバー・ログ・ファイルおよびエラー・ファイルの実際の場合と、サーバー・ダンプ・ファイルの場合（作成されている場合）が表示されます。

例

例 1: 構成ファイルに定義されたサーバーの情報を表示します。

```
SQLSRV> SHOW SERVER;
Server Version:          7.1
Server Platform:        Digital OpenVMS Alpha
Max Shared Mem Size:    2000 Kb
Config file:            SYS$SYSROOT: [SYSMGR] SQLSRV_CONFIG_FILE71.DAT;1
Log path:               SYS$MANAGER:
Dump path:              SYS$MANAGER:
Proc start time:        <none>
Proc shut time:         <none>
Network Ports:
  (State)  (Protocol)
  DECnet object  SQLSRV_SERVER      Running  Native
  TCP/IP port   2199                            Running  Native
Current shared memory usage:
Allocation unit:        65536 bytes
Total memory:          2031616 bytes ( 31 units)
Free memory:           1769472 bytes ( 27 units)
Partly allocated:      196608 bytes ( 3 units)
Log File:              SYS$SYSROOT: [SYSMGR] SQS_CRANES_SQLSRV_MON_0071.LOG;
Dump File:             SYS$SYSROOT: [SYSMGR] SQS_CRANES_SQLSRV_71.DMP;
```

SHOW SERVICE コマンド

サービス・オブジェクトの静的定義、または構成ファイルに現在定義されているすべてのサービス・オブジェクトを表示します。

書式

```
SHOW SERVICE[S]    [<service-spec> ] [FULL];

<service-spec> ::= { * | <service-name-list>
<service-name -list> ::= { <identifier> [ , <identifier> ] ... }
```

引数

<service-spec>

サービス・オブジェクトの指定。1つ以上のサービス・オブジェクト名を指定するか、アスタリスク (*) で指定します。1つ以上のサービス・オブジェクト名を指定すると、それらの名前付きサービス・オブジェクトの情報のみが表示されます。* を指定すると、すべてのサービス・オブジェクト名に関する情報が表示されます。

FULL

各サービス情報の完全な説明を表示します。デフォルトでは、各サービスの簡単な情報（1行の出力）が表示されます。

使用方法

- SERVICE は、キーワード SERVICES と同義です。
- サービス・オブジェクトを指定しないと、すべてのサービス・オブジェクト名に関する情報が表示されます。
- SHOW SERVICE * コマンドと SHOW SERVICE コマンドでは、両方とも現在定義されているすべてのサービスが表示されます。
- サービス状態は、次の 5 つの状態のいずれかを取ります。
 - 開始中: サービスは起動中です。

MIN_EXECUTORS が 0 (ゼロ) に設定されたサービスでは、サービスの起動時に開始中という状態が表示されることはありません。この状態は、実行中または失敗として表示されます。
 - 失敗: サービスは起動に失敗しました。
 - 実行中: サービスは実行中です。
 - 非アクティブ: サービスは停止しています。
 - 不明: 管理クライアントがオンラインでサーバーに接続していないため、サービスの状態を特定できません。管理クライアントでは、SET CONFIG_FILE コマンドを使用してサーバーをオフラインで管理しています。CONNECT TO SERVER コマンドを使用してオンラインでサーバーに接続し、サービス状態を確認してください。
- 実行中のサービスとその構成ファイル間の属性に差異が存在する場合、SHOW SERVICE コマンドでは出力表示の最後にその差異が示されます。また、サービスの再起動時に、構成ファイルに含まれるサーバーのサービス定義と一致するよう実行中のサーバーのサービスが更新されることが示されます。
- FULL を指定すると、指定のサービスに対するアクセス権を付与されたユーザー名と識別子のリストが表示されます。

例

例 1: 構成に定義されたサービスと、簡単な説明を表示します。

```
SQLSRV> SHOW SERVICES;
```

Name	State	Clients			Executors		
		Per-Exec	Max	Active	Min	Max	Running
SQLSRV_MANAGE	UNKNOWN	100	100	1	1	1	0
GENERIC	UNKNOWN	1	10	0	2	10	1
RMJ_SERVICE	UNKNOWN	1	100	0	0	100	0

例 2: 構成に定義された payroll サービスと、完全な説明を表示します。

```
SQLSRV> SHOW SERVICE payroll FULL;
```

```
Service PAYROLL
  State:                RUNNING
  Owner:                PAYROLLACCNT
  Protocol:             SQL/Services
  Default Connect Username: <not specified>
  SQL version:         7.1
  Autostart:           on
  Process init:        <not specified>
  Attach:              ATTACH 'FILENAME PAYROLL_DISK:PAYROLL_DB'
  Schema:              <not specified>
  Reuse:               SESSION
  Database Authorization: CONNECT USERNAME
  dbsrc file:          <not specified>
  SQL init file:       <not specified>
  Appl Transaction Usage: SERIAL
  Idle User Timeout:   <none>
  Idle Exec Timeout:   1800 seconds
  Min Executors:       5
  Max Executors:       10
  Clients Per Executor: 1
  Active Clients:      0
```

```
Access to service PAYROLL
```

```
Granted to users:
  PRIVILEGED_USER 'PAYROLLACCNT'
Granted to identifiers:
  'PAYROLL_DBA' 'PAYROLLDEPT'
```

SHOW SETTINGS コマンド

現在の SQLSRV_MANAGE 設定に関する情報を表示します。

書式

```
SHOW SETTINGS;
```

使用方法

サーバーの起動後に、SHOW SETTINGS コマンドを使用して、SQLSRV_MANAGE 環境の現在の設定を確認します。これらの SQLSRV_MANAGE 環境設定は、独自の用途に基づいて変更できます。

例

例 1: SQLSRV_MANAGE 環境の現在の設定に関する情報を表示します。

```
SQLSRV> CONNECT SERVER;  
Connecting to server ...  
Connected  
SQLSRV> SHOW SETTINGS;  
Settings:  
  version:    v7.1  
  verify:    off  
  output:    on  
  config-file:  SYS$SYSROOT: [SYSMGR] SQLSRV_CONFIG_FILE.DAT;1  
  confirm:    on
```

SHOW VERSION コマンド

SQLSRV_MANAGE 管理クライアントのバージョンを表示します。

書式

SHOW VERSION;

使用方法

SHOW VERSION コマンドを使用して、SQLSRV_MANAGE 管理クライアントのバージョンを確認します。

例

例 1: SQLSRV_MANAGE 管理クライアントのバージョンを表示します。

```
SQLSRV> SHOW VERSION;  
Version:          v7.1-6
```

SHUTDOWN DISPATCHER コマンド

指定されたディスパッチャを停止します。

書式

```
SHUTDOWN DISPATCHER          <dispatcher-name>;  
  
                                <dispatcher-name> ::= <identifier>
```

引数

<dispatcher-name>

ディスパッチャ・オブジェクト名を指定します。ディスパッチャ・オブジェクト名は、識別子として表します。

使用方法

- SHOW CLIENTS コマンドを使用して、停止対象のディスパッチャが提供するネットワーク・トランスポートを使用してサービスに接続しているクライアントが存在しないことを確認します。
- ディスパッチャはオンライン操作でのみ停止できます。つまり、サーバーで定義されている実行中のディスパッチャを停止するには、その実行中のサーバーに接続する必要があります (CONNECT TO SERVER コマンド)。
- 起動に失敗したディスパッチャは、失敗状態のまま残るため、停止する必要があります。問題を訂正し (通常は引数の値が間違っていて指定されています)、ディスパッチャを再起動してください。

例

例 1: disp_tcpip というディスパッチャを停止します。

```
SQLSRV> CONNECT SERVER;  
Connecting to server ...  
Connected  
SQLSRV> SHOW CLIENTS;  
SQLSRV> SHUTDOWN DISPATCHER disp_tcpip;
```

SHUTDOWN SERVER コマンド

現在のサーバーを停止します。

書式

SHUTDOWN SERVER;

使用方法

- SHOW CONNECT[ION] コマンドを使用して、停止対象のサーバーが間違っていないことを確認します。
- サーバーはオンライン操作でのみ停止できます。つまり、サーバーを停止するには、実行中のサーバーに接続する必要があります (CONNECT TO SERVER コマンド)。

例

例 1: 現在のサーバーを停止します。

```
SQLSRV> CONNECT SERVER;  
Connecting to server ...  
Connected  
SQLSRV> SHOW CONNECT;  
Active connections:  
CURRENT: SQLSRV_MANAGE  
         Service: SQLSRV_MANAGE  
         User: run_username Node: EAGLE Local: Yes  
         Transport: DECNET Object: SQLSRV_SERVER  
         Request bufsize: 1024 Response bufsize: 1024  
SQLSRV> SHUTDOWN SERVER;  
Disconnected from Server
```

SHUTDOWN SERVICE コマンド

指定されたサービスを停止します。

書式

```
SHUTDOWN SERVICE          <service-name>;  
  
                             <service-name> ::= <identifier>
```

引数

<service-name>

サービス・オブジェクト名を指定します。サービス・オブジェクト名は、識別子として表します。

使用方法

- SHOW CLIENTS コマンドを使用して、停止対象のサービスに接続しているクライアントが存在しないことを確認します。
- サービスはオンライン操作でのみ停止できます。つまり、サーバーで定義されている実行中のサービスを停止するには、その実行中のサーバーに接続する必要があります (CONNECT TO SERVER コマンド)。
- 起動に失敗したサービスは、失敗状態のまま残るため、停止する必要があります。問題を訂正し (通常は引数の値が間違っ指定されています)、サービスを再起動してください。

例

例 1: generic という汎用サービスを停止します。

```
SQLSRV> CONNECT SERVER;  
Connecting to server ...  
Connected  
SQLSRV> SHUTDOWN SERVICE generic;
```

START DISPATCHER コマンド

現在のサーバーに指定された名前で定義されているディスパッチャ・オブジェクトのディスパッチャ・プロセスを起動します。

書式

```
START_DISPATCHER <disp-name>;  
  
<disp-name> ::= <identifier>
```

引数

<disp-name>

ディスパッチャ名を指定します。ディスパッチャ名は、識別子として表します。

使用方法

ディスパッチャはオンライン操作でのみ起動できます。つまり、サーバーで定義されているディスパッチャを起動するには、その実行中のサーバーに接続する必要があります (CONNECT TO SERVER コマンド)。

例

例 1: tcpip_disp ディスパッチャを起動します。

```
SQLSRV> CONNECT SERVER;  
Connecting to server ...  
Connected  
SQLSRV> START DISPATCHER tcpip_disp;
```

START SERVER コマンド

現在のサーバーを起動し、オプションで現在のサーバーに定義されているディスパッチャおよびサービス・オブジェクトを起動します (AUTOSTART が ON に設定されている場合)。AUTOCONNECT が ON に設定されている場合は、起動後にサーバーに接続します。

書式

START SERVER

```
->[USER { <user-name> } USING { <password> }]
```

```
->[ AUTOCONNECT { ON | OFF } ]
```

```
->[ AUTOSTART { ON | OFF } ];
```

```
<user-name> ::= { <quoted-string> | <identifier> }
```

```
<password> ::= { <quoted-string> | <identifier> }
```

引数

USER {<user-name> USING <password>}

起動後のサーバーに接続するためのユーザー名とパスワードを指定します。ユーザー名とパスワードは、引用符付き文字列または識別子として表します。DECnet が稼働するサーバーを起動する場合、または TCP/IP が稼働するサーバーを起動し、ユーザーが SYSPRV または BYPASS 権限を保持している場合、ユーザー名とパスワードを指定する必要はありません。

AUTOCONNECT {ON | OFF}

START SERVER コマンドの発行後に、サーバーに自動的に接続するかどうかを決定します。引数を ON に指定すると、START SERVER コマンドの発行後に SQLSRV_MANAGE は自動的にサーバーに接続します。OFF の値を指定すると、サーバーは起動しますが、起動後のサーバーへの接続は試行されません。デフォルトは ON です。

AUTOSTART {ON | OFF}

START SERVER コマンドの発行時に、他のすべてのサーバー・オブジェクト (ディスパッチャおよびサービス) を自動的に起動するかどうかを決定します。この引数を ON (デフォルト) に指定した場合、各オブジェクトの AUTOSTART 引数の値も ON に設定されていれば、他のすべてのサーバー・オブジェクトも自動的に起動します。他のすべてのサーバー・オブジェクトを起動しない場合、START SERVER コマンドで AUTOSTART 属性の値を OFF に指定します。AUTOSTART OFF 属性設定により、各オブジェクトの AUTOSTART 属性設定は上書きされるため、サーバー・オブジェクトの起動後に各オブジェクトを個別に起動できます。デフォルトは ON です。

使用方法

- サーバーはオフライン操作でのみ起動できます。つまり、SET CONFIG_FILE コマンドを使用して、起動するサーバーの構成ファイルを選択するか、デフォルトを使用する必要があります。
- ON に指定された AUTOCONNECT 引数付きでサーバーが起動されると、SQLSRV_MANAGE は、サーバーに定義されたすべてのネットワーク・ポートに対して接続を試みます。接続を確立するために、ラウンドロビン法でそれぞれ最大3回まで各ネットワーク・ポートに接続を試みます。
- SETPRV 権限またはすべての権限を保持している必要があります。
- SQLSRV_MANAGE を起動すると、次のようにデフォルトの構成ファイル名が確定されます。
 - デフォルトの構成ファイルは、次のとおりです。
SYS\$MANAGER:SQLSRV_CONFIG_FILE71.DAT
 - デフォルトを上書きするには、SQLSRV_CONFIG_FILE71 論理名を設定するか、SET CONFIGURATION_FILE コマンドに異なるファイル名を指定します。

例

例 1: 現在のサーバーを起動します。

```
SQLSRV> SET CONFIG_FILE 'my_config_file';
SQLSRV> START SERVER;
Server started
Connecting to server ...
Connected
```

START SERVICE コマンド

現在のサーバーに定義されている指定のサービス・オブジェクトを起動します。

書式

```
START SERVICE      <service-name>;  
  
                    <service-name> ::= <identifier>
```

引数

<service-name>

サービス名を指定します。サービス名は、識別子として表します。

使用方法

サービスはオンライン操作でのみ起動できます。つまり、サーバーで定義されているサービスを起動するには、その実行中のサーバーに接続する必要があります (CONNECT TO SERVER コマンド)。

例

例 1: V71 という汎用サービスを起動します。

```
SQLSRV> CONNECT SERVER;  
Connecting to server ...  
Connected  
SQLSRV> START SERVICE v71;
```

ロギングとトラブルシューティング

この章では、SQL/Services および OCI Services for Oracle Rdb のロギングを有効化する方法と、より一般的ないくつかのエラーを識別および修正する方法について説明します。

Oracle SQL/Services サーバー・システムで発生する可能性のある問題は、次の方法で特定できます。

- ログ・ファイルの調査
- 異なるタイプの問題の調査

次の項では、生成されるログ・ファイル、各タイプのログ・ファイルに含まれる内容、発生する可能性のある様々なエラー状態、および特定の問題の識別方法について説明します。

7.1 問題のレポート

Oracle SQL/Services または OCI Services for Oracle Rdb の使用中にエラーが発生し、そのエラーが Oracle 製品の問題に起因することが確実である場合、技術上の支援を受けるために Oracle サポート・サービスに連絡してください。

再現可能な問題が発生した場合、できるかぎり詳細な情報を報告することが重要です。完全ロギングを有効化し、現在のセッションに関する詳細な情報を収集してください。問題レポートには次の項目を含めます。

- Oracle SQL/Services モニター・ログ・ファイル、ディスパッチャ・ログ・ファイル、適切なエグゼキュータ・ログ・ファイル、および関連するクライアント・ログ・ファイルのコピー
- Oracle SQL/Services 構成ファイルのコピー
- 生成されたバグチェック・ダンプ・ファイルのコピー

7.2 エラー・メッセージ

指定する言語により、Oracle エラー・メッセージとボイラープレート・テキスト（日付に使用する月日の名前を含む）に使用される言語が決定されます。デフォルトでは、Oracle Rdb エラー・メッセージは英語で表示されます（特別に翻訳されたエラー・メッセージ・ファイルを使用し、そのファイルを Rdb データベースに定義しない場合）。Oracle エラー・メッセージ・ファイルは、表 4-1 にリストされたすべてのサポート対象言語で提供されています。

Oracle Rdb エラー・メッセージでは、データベースへのアタッチの際に指定された言語が、そのアタッチの継続中に使用されます。

ソート、照合および比較は、Oracle Rdb セマンティクスに基づいて行われます。詳細は、『Oracle Rdb SQL リファレンス・マニュアル』を参照してください。

サーバー・エラー・メッセージ・ファイルには、すべてのサーバー・エラーと、各エラーおよび実行可能なユーザー・アクションの説明が含まれます。サーバー・エラー・メッセージ・ファイルの PostScript およびテキスト・バージョンは、次のディレクトリ内にあります。

- SYS\$HELP:SQLSRV_MESSAGES71.PS (Oracle SQL/Services サーバー・エラー・メッセージの PostScript ファイル)
- SYS\$HELP:SQLSRV_MESSAGES71.TXT

7.3 サーバーのログ・ファイル

この項では、サーバーでロギングを有効および無効にする方法と、ログ・ファイルを使用する方法について説明します。

サーバー側には、次のようないくつかの種類のログ・ファイルがあります。

- Oracle SQL/Services モニター・ログ・ファイル
- Oracle SQL/Services ディスパッチャ・ログ・ファイル
- Oracle SQL/Services エグゼキュータ・ログ・ファイル

Oracle SQL/Services では、次の表記規則を使用してサーバー・コンポーネントのログ・ファイル名を生成します (nodename はノード名、component-id はサーバー・コンポーネント、version はバージョン番号です)。

- SCSNODE SYSGEN パラメータが設定されている場合
sqs_<nodename><component-id><instance><version>.log
- SCSNODE SYSGEN パラメータが空白の場合
sqs_<component-id><instance><version>.log

7.3.1 Oracle SQL/Services モニター・ログ・ファイル

Oracle SQL/Services では、次の情報がモニター・ログ・ファイルに記録されます。

- ディスパッチャおよびエグゼキュータ・プロセスの起動と停止の情報メッセージ
- ディスパッチャおよびエグゼキュータ・プロセスの障害エラー・メッセージ (コンポーネント・ログ・ファイルの名前と場所を含む)
- Oracle SQL/Services システム管理クライアントに対する Oracle SQL/Services の認証および認可の失敗
- モニター・プロセスのバグチェック・ダンプの名前と場所 (モニターでリカバリ不可能なエラーが検出された場合)

次のコマンドを使用してモニター・ログ・ファイルをリストします。

```
$ DIRECTORY SYS$MANAGER:SQS*MON*.LOG
```

次に例を示します。

```
SYS$MANAGER:SQS_NODE1_SQLSRV_MON_0071.LOG
```

7.3.2 Oracle SQL/Services ディスパッチャ・ログ・ファイル

Oracle SQL/Services では、次の情報がディスパッチャ・ログ・ファイルに記録されます。

- Oracle SQL/Services、OCI クライアントおよび Oracle RMU クライアントに対する Oracle SQL/Services の認証および認可の失敗
- エグゼキュータ・プロセスの障害に起因するサーバー側のクライアント・ネットワーク・リンクの切断
- クライアント側のクライアント・ネットワーク・リンクの障害
- ディスパッチャ・プロセスのバグチェック・ダンプの名前と場所 (ディスパッチャでリカバリ不可能なエラーが検出された場合)

次のコマンドを使用してディスパッチャ・ログ・ファイルの場所とファイル名を表示します。

```
SQLSRV> SHOW DISPATCHER <dispatcher_specification>
```

次に例を示します。

```
SQLSRV> SHOW DISPATCHER SQLSRV_DISP
```

```
Dispatcher SQLSRV_DISP
State: RUNNING
Autostart: on
Max connects: 100 clients
Idle User Timeout: <none>
Max client buffer size: 5000 bytes
Network Ports: (State) (Protocol)
DEcnet object 81 Running SQL/Services
TCP/IP port 119 Running SQL/Services
Log path: SYS$MANAGER:
Dump path: SYS$MANAGER:
Log File: SYS$SYSROOT:[SYSMGR]SQS_NODE1_SQLSRV_DIS00371.LOG;
Dump File: SYS$SYSROOT:[SYSMGR]SQS_NODE1_SQLSRV_DIS003.DMP;
```

ログ・ファイルの場所とファイル名は、Log File: で始まる行に示されます。

次のコマンドを使用してログ・ファイルをリストします。

```
$ DIRECTORY <directory_name>:<filename>
```

次に例を示します。

```
$ DIRECTORY SYS$SYSROOT:[SYSMGR]SQS*DIS*.LOG
```

```
SYS$SYSROOT:[SYSMGR]SQS_NODE1_SQLSRV_DIS00371.LOG
```

SQLNET または OCI クライアントが Oracle SQL/Services との接続の切断に成功するたびに、次のエントリがディスパッチャ・ログ・ファイルに書き込まれることがあります。

```
-----EVENT BEGIN:  EVENT_LOG at Tue Oct  8 2004 18:40:01.055-----
%SQLSRV-I-EVENT_LOG, event logged at line 1674 in file COM_TNS.C;1
%SQLSRV-E-TNSFAILURE, SQL*Net TNS nsrecv() service has failed
%SQLSRV-E-TNSEXTENDED, SQL*Net TNS error codes: primary (12537) secondary (0)
-----EVENT END   :  EVENT_LOG at Tue Oct  8 2004 18:40:01.172-----
```

```
-----EVENT BEGIN:  EVENT_LOG at Tue Oct  8 2004 18:40:01.519-----
%SQLSRV-I-EVENT_LOG, event logged at line 284 in file DISP.C;1
%SQLSRV-W-EXCEPTION_RAISE, Exception raised: DBS_TNSFAILED
%SQLSRV-I-CONNECTNAME, Connect : CONNECT_0000001
%SQLSRV-I-CONNECTSTATE, Connect state: 4
%SQLSRV-I-USERNAME, User name: dtmtest1
%SQLSRV-I-NODENAME, Node : unknown
%SQLSRV-I-SERVICENAME, Service : NATCONN_OCI
-----EVENT END   :  EVENT_LOG at Tue Oct  8 2004 18:40:01.527-----
```

これらのエントリは無視できます。

7.3.3 Oracle SQL/Services エグゼキュータ・ログ・ファイル

Oracle SQL/Services では、次の情報がエグゼキュータ・ログ・ファイルに記録されます。

- エグゼキュータ・プロセスの起動エラー
- データベース認可が接続ユーザーに設定されたデータベース・サービスの Oracle SQL/Services クライアントに対する Oracle Rdb の認証および認可の失敗
- Oracle Rdb および SQL エラー・メッセージ
- エグゼキュータ・プロセスのバグチェック・ダンプの名前と場所（エグゼキュータでリカバリ不可能なエラーが検出された場合）
- OCI サービスの場合、ALTER SESSION LOG BRIEF または ALTER SESSION LOG FULL コマンド、あるいは SQLNET_DEBUG_FLAGS 論理名で指定された OCI Services for Oracle Rdb のログ・メッセージ

エグゼキュータ・ログ・ファイルは、サービス所有者アカウントのデフォルト・ディレクトリに作成されます。たとえば、次のコマンドを使用すると、デフォルト・ディレクトリが SYS\$SYSDEVICE:[SQLSRV\$DEFLT] である SQLSRV\$DEFLT というサービス所有者アカウントで GENERIC というサービスのエグゼキュータ・ログ・ファイルをリストできます。

```
$ DIRECTORY SYS$SYSDEVICE:[SQLSRV$DEFLT] SQS*GENERI*.LOG
SYS$SYSDEVICE:[SQLSRV$DEFLT] SQS_NODE1_GENERI004000171.LOG
```

7.3.4 OCI Services for Oracle Rdb におけるエグゼキュータ・ログ・ファイルのデバッグ・フラグの使用

デバッグ・フラグを使用して、SQL/Services エグゼキュータ・ログ・ファイルに追加情報を記録できます。この機能を有効化するには、SQL/Services サービスのプロセス初期化ファイルで論理名 SQLNET_DEBUG_FLAGS を定義します。次のデバッグ・フラグ（大 / 小文字の区別あり）が現在サポートされています。

- H
ルーチンが実行されるたびに、ルーチン・ヘッダー情報が記録されます。
- T
ヘッダー情報が記録されるたびに、現在の時刻が記録されます。このフラグは、H フラグに依存します。H フラグを指定しない場合、時刻情報は記録されません。

たとえば、次の行をプロセス初期化ファイルに挿入すると、ヘッダー情報が記録されます。

```
$ DEFINE SQLNET_DEBUG_FLAGS "H"
```

次のような情報がエグゼキュータ・ログ・ファイルに出力されます。この情報は、オラクル社に問題をレポートするときに特に役立ちます。次に例を示します。

```
... Server entered ...
GTZI_SERVER STARTUP UP -> Wed Aug 8 13:34:33 2004
*** GTZGPRD ***
*** GTAGPRD ***
NLS initialization ...
* GTWILAN: Setting LANGUAGE *
Language is AMERICAN_AMERICA.US7ASCII
```

次の行をプロセス初期化ファイルに挿入すると、ヘッダー情報と現在の時刻が記録されます。

```
$ DEFINE SQLNET_DEBUG_FLAGS "HT"
```

次のような情報がエグゼキュータ・ログ・ファイルに出力されます。

```
... Server entered ...
Current Time = Wed Aug 8 13:34:33 2004
```

```

GTZI_SERVER START UP -> Wed Aug  8 13:34:33 2004

Current Time = Wed Aug  8 13:34:33 2004

*** GTZGPRD ***
Current Time = Wed Aug  8 13:34:33 2004

*** GTAGPRD ***
Current Time = Wed Aug  8 13:34:33 2004

NLS initialization ...
Current Time = Wed Aug  8 13:34:34 2004

* GTWILAN: Setting LANGUAGE *
Current Time = Wed Aug  8 13:34:34 2004
Language is AMERICAN_AMERICA.US7ASCII
Current Time = Wed Aug  8 13:34:34 2004

```

タイミングまたはパフォーマンスの問題をデバッグする場合を除き、T デバッグ・フラグを使用することは避けてください。T フラグを使用すると、すべてのコマンドに対して余分なタイムスタンプが追加されるため、ログ・ファイルが非常に大きくなる可能性があります。

7.3.5 SQL および Oracle Rdb でのロギングの有効化

RDMS\$DEBUG_FLAGS 論理名を定義して、SQL および Oracle Rdb でロギングを有効化できます。このロギングも、エグゼキュータ・ログ・ファイルに書き込まれます。

詳細は、『Oracle Rdb7 データベース・パフォーマンスおよびチューニングのためのガイド』を参照してください。

7.3.6 SQL/Services でのロギングの無効化

ディスパッチャ・レベルまたはエグゼキュータ・レベルでロギングを無効化できます。

ディスパッチャ・ロギングの場合、次のように CREATE または ALTER コマンドを使用します。

```

SQLSRV> CREATE DISPATCHER SQLSRV_DISP1 LOG PATH 'NOLOG'
_SQLSRV> DUMP PATH 'SYS$MANAGER';

```

または

```

SQLSRV> ALTER DISPATCHER SQLSRV_DISP1 LOG PATH 'NOLOG'
_SQLSRV> DUMP PATH 'SYS$MANAGER';
%SQLSRV-S-ALTER_RESTART

```

変更した設定を反映するため、オブジェクトを再起動します。

エグゼキュータ・ロギングの場合、次のようにします。

システム・レベルで SQLSRV_EXEC_LOG 論理名を定義します。次に例を示します。

```

$DEFINE/SYSTEM SQLSRV_EXEC_LOG NOLOG

```

7.3.7 ディスパッチャ・ログ・ファイルのシステム・ディスクからの移動

SQL/Services のリリース 7.1.5.5 以上から、ディスパッチャの CREATE または ALTER コマンドを使用してシステム・ディスクからディスパッチャ・ログ・ファイルを移動できます。次に例を示します。

```

SQLSRV> ALTER DISPATCHER SQLSRV_DISP1 LOG PATH 'USER1: [SQLSRV_TEST1.AAA]'
_SQLSRV> DUMP PATH 'USER1: [SQLSRV_TEST2.BBB]';
%SQLSRV-S-ALTER_RESTART

```

変更した設定を反映するため、オブジェクトを再起動します。

7.4 OCI Services for Oracle Rdb の完全ロギングの有効化

OCI Services for Oracle Rdb の完全ロギングを有効化するには、次の手順を実行します。

1. エグゼキュータのロギングを有効化する必要があります。
2. サービス所有者のログイン・ディレクトリに SQL.INI ファイルを作成します。ファイルには次の行を含める必要があります。

```
ALTER SESSION LOG FULL;
```

3. SQL INIT パラメータがこのファイルを示すようにサービスを変更します。

```
SQLSRV> ALTER SERVICE FOO SQL_INIT_FILE "<pathname>:SQL.INI";
```

4. サービスを再起動します。

```
SQLSRV> SHUTDOWN SERVICE FOO;
```

```
SQLSRV> START SERVICE FOO;
```

7.5 SQL/Services API ログ・ファイルの調査

問題が発生した場合、クライアント側とサーバー側のそれぞれで生成されたログ・ファイルを調査することで、問題の特定を試みることができます。

クライアント側のロギングには、最大で次の 4 つの種類があります。

- クライアントおよびドライバ・ロギング
- Winsock ロギング
- Oracle Net ロギング
- ODBC トレース

7.5.1 クライアントおよびドライバ・ロギング

クライアントおよびドライバ・ロギングは、Oracle SQL/Services と Oracle ODBC Driver for Oracle Rdb の両方で使用されます。

Oracle SQL/Services クライアント・ロギングを有効化するには、sqlsrv_associate ルーチンのパラメータを使用するか、Windows 95 以上のクライアントでは sqsapiw.ini ファイルまたは sqsapi32.ini ファイルを使用します。Oracle SQL/Services クライアント API により、アプリケーションのデフォルト・ディレクトリに clientxx.log というログ・ファイルが作成されます。clientxx.log ファイルには、Oracle SQL/Services へのコール、API サービス、データ値およびメッセージ・プロトコルが記録されます。

clientxx.log ファイルをチェックして、サーバーに渡される SQL 文、またはサーバーから返されるエラー・メッセージを確認します。詳細は、『Guide to Using the Oracle SQL/Services Client API』を参照してください。

Oracle ODBC Driver for Rdb を使用する場合、oraodbc.ini ファイル（バージョン 2 ドライバの場合は rdbodbc.ini）のクライアント・ロギングおよびドライバ・ロギングのデフォルト設定を変更して、ODBC ロギングを有効化できます。デフォルト値は 0（ゼロ）で、ロギングは無効です。すべての有効な値は、ドライバに付属する sqrdh.hlp ヘルプ・ファイルに記載されています。

クライアント・ロギングでは、Oracle SQL/Services API を通じてクライアントからサーバーに渡される Oracle SQL コールが記録されます。情報は、clientxx.log というログ・ファイルに書き込まれます（xx は 01 ～ 99 の値）。番号は、自動的にリセットされないため、99 を超えないようにしてください。

ドライバ・ロギングでは、情報をサーバーに渡すプロセス中に実行されるドライバ・コールが記録されます。情報は、アプリケーション・ディレクトリに存在する rdbodbc.log ファイルに書き込まれます。

7.5.2 Winsock ロギング

Windows 95 以上で Winsock トランスポートを使用する場合、このロギング・オプションはネットワーク問題の診断に役立ちます。このオプションを有効化するには、`sqsap32.ini` ファイルの `";Winsock Logging = 1"` という行のコメントを解除します。このロギングにより、問合せが渡されるときに実行される Winsock またはネットワーク・コールがトレースされます。ログ・ファイルの名前は `sqsap32.log` であり、アプリケーションの作業ディレクトリ内に配置されます。

7.5.3 Oracle Net ロギング

OCI サービスを使用する場合、Oracle 環境の `SQLNET.ORA` ファイルで `TRACE_LEVEL_CLIENT` または `TRACE_LEVEL_SERVER` を定義して、Oracle Net ロギングを有効化できます。

コンポーネント構成ファイルを使用してトレース・パラメータを設定するには、次の手順を実行します。

1. トレースするコンポーネントを実行または起動します。

次のトレース・パラメータをコンポーネント構成ファイル（クライアントまたはサーバーの場合 `SQLNET.ORA`、リスナーの場合 `LISTENER.ORA`）に設定します。

```
TRACE_LEVEL_<CLIENT | LISTENER | SERVER> = (0 | 4 | 10 | 16)
TRACE_DIRECTORY_<CLIENT | LISTENER | SERVER> = <directory_name>
TRACE_FULL_<CLIENT | LISTENER | SERVER> = <directory_name>
```

次の表で、トレース・レベルごとの出力について説明します。

トレース・レベル	出力
0 または OFF	デフォルト。トレース出力なし。
4 または USER	ユーザー・トレース情報。
10 または ADMIN	管理トレース情報。
16 または SUPPORT	Oracle サポート・サービス・トレース情報。

デフォルト・ディレクトリは、OpenVMS のプロセス所有者のログイン・ディレクトリ、または Microsoft プラットフォームの実行可能イメージの現行ディレクトリです。デフォルトのファイル名は、`SQLNET.LOG` および `SQLNET.TRC` です。Oracle Net 構成を作成または変更する場合、Oracle Network Manager を通じてファイル名とディレクトリの場所の両方を変更できます。この操作は、クライアント・プロファイル・オブジェクトのロギングおよびトレース・タブを編集することで、任意のクライアント・プロセスを対象に実行できます。

2. コンポーネントの実行中に構成ファイルを変更した場合、そのコンポーネントを起動または再起動してパラメータの変更を有効化します。

7.5.4 ODBC トレース

ODBC トレースを有効化するには、ODBC 管理者のアプリケーションで `TRACING` タブを選択します。デフォルトのファイル名は、`SQL.LOG` です。ファイルの作成場所を指定する必要があります。「ただちにトレースを開始」をクリックしてトレースを有効化します。このログには、ODBC ドライバが Oracle SQL/Services に送信する SQL が記録されます。

7.6 プロセス障害

Oracle SQL/Services では、失敗したプロセスのタイプに応じて異なる方法でプロセス障害が処理されます。

7.6.1 モニター・プロセス障害

Oracle SQL/Services は、モニター・プロセスに障害が発生してもリカバリを試みません。モニター・プロセスに障害が発生すると、サーバー構成のすべてのプロセスが停止します。そのため、モニター・プロセス障害によってシステムが一貫性のない状態に置かれることはありません。

7.6.2 ディスパッチャ・プロセス障害

Oracle SQL/Services は、障害の発生したディスパッチャ・プロセスを自動的に再起動しません。ただし、サーバーの整合性に影響を与える可能性のある重要な操作中に障害が発生した場合を除き、サーバーは稼働を続けます。したがって、障害の発生したディスパッチャ・プロセスは、手動で再起動する必要があります。

7.6.3 エグゼキュータ・プロセス障害

Oracle SQL/Services は、サーバーの整合性に影響を与える可能性のある重要な操作中に障害が発生した場合を除き、障害の発生したエグゼキュータ・プロセスを自動的に再起動しようと試みます。ただし、`SQLSRV$MAX_EXECUTOR_FAILURES` 論理名が定義されていない状態で、エグゼキュータ・プロセスの起動時に 3 回以上障害が発生すると、Oracle SQL/Services はエグゼキュータを停止します。サービスのアクティブなエグゼキュータが他に存在しない場合、サービスも停止されて失敗とマークされます。

`SQLSRV$MAX_EXECUTOR_FAILURES` 論理名を定義して、障害の最大数をデフォルトの 2 から正の整数値に変更できます。このようにして、定期的なデータベース・メンテナンス中にエグゼキュータとサービスを停止する頻度を制御できます。

7.7 異なるタイプの問題の調査

システム管理者は、状況に応じて異なるタイプの問題を多数調査する必要があります。次に、一般的なエラー状態のセットと、特定の問題の追跡および識別に役立つ各エラーのガイドラインを示します。

7.7.1 ネットワーク・トランスポートの問題

新規ユーザーや新規サーバー構成に関連してときどき発生する問題は、サーバーにまったく接続できないという問題です。この状況の場合、クライアント・アプリケーションは、**Oracle SQL/Services API** ルーチンおよび **OCI Services for Oracle Rdb** ルーチンからネットワーク・エラーを受信します。

このタイプのエラーが発生した場合、最初に確認することは、選択したトランスポートをサポートするディスパッチャが実行中であるかどうかと、指定したネットワーク・ポートまたはオブジェクトがアクティブであるかどうかです。マルチリリース環境で代替ネットワーク・ポートまたはオブジェクトを使用している場合、クライアントで正しいネットワーク・ポートまたはオブジェクトを指定していることを確認します。**SQLSRV_MANAGE** ユーティリティの **SHOW SERVER** コマンドにより、すべてのネットワーク・ポート番号を表示できます。クライアント・アプリケーションで正しいポートを使用していることを確認してください。ディスパッチャが正しく動作していると考えられる場合、**TCP/IP Ping** ユーティリティなどのトランスポート固有のツールを使用して、クライアントとサーバー・ノード間の接続性を確認します。

ディスパッチャが稼働していないか、選択したネットワーク・ポートまたはオブジェクトがアクティブではない場合、ディスパッチャ・ログを確認して問題の原因を特定します。ディスパッチャ・プロセスが完全に停止すると、**Oracle SQL/Services** により、モニター・ログにディスパッチャ・ログ・ファイルの名前と場所が書き込まれます。ディスパッチャ・ログ・ファイルをチェックして、ディスパッチャの障害発生時にバグチェック・ダンプが生成されたかどうかを確認します。

7.7.2 ユーザー認証および認可の問題

認証および認可エラーは、新規ユーザーに関連して発生する可能性のある別のクラスの問題です。この場合、サーバーは正しく動作しています。ただし、ユーザーは、サーバーに接続できないか、サーバーにより提供される特定のサービスに接続できません。

最初に、ディスパッチャ・ログ・ファイルをチェックしてエラーの原因を特定する必要があります。異なるトランスポートに対して複数のディスパッチャを構成している場合、必ず適切なディスパッチャ・ログをチェックしてください。

たとえば、**SQLSRV** エグゼキュータの場合、認証または認可の問題を解決するには、状況に応じてネットワーク・アクセス権を付与するか、ユーザーのアカウントに **SQLSRV\$CLIENT** 識別子を付与する必要があります。または、特定のサービスに対するアクセス権を新規ユーザーに付与する必要があります。

OCI Services for Oracle Rdb エグゼキュータで問題が発生する可能性があるのは、サービス所有者がデータベースのアタッチ権限を保持していない場合 (**SELECT**)、接続ユーザーが **USERS** 表に含まれていない場合、またはユーザーがデータベースまたは表にアクセスするのに必要な **Oracle Rdb** 権限を保持していない場合です。

これらのタイプのエラーは、すべてディスパッチャ・ログ・ファイルに記録されます。

データベース認可が接続ユーザーに設定されたデータベース・サービスにユーザーが接続する場合、そのユーザーのアカウントにデータベースへのアクセス権を付与する必要があります。ユーザーにデータベースへのアクセス権を付与しないと、**Oracle Rdb** により、権限不足 (-1008) エラーが返されます。エラーのテキストは、**Oracle SQL/Services** によりクライアント・アプリケーションに返され、エグゼキュータ・ログ・ファイルに書き込まれます。

7.7.3 サービス起動時のエグゼキュータの障害

新規サービスを作成して最初に起動したときに、エラーが発生することがあります。エグゼキュータ・プロセスが正しく起動しない場合、常に Oracle SQL/Services によってエグゼキュータ・ログ・ファイルの名前がモニター・ログに書き込まれます。エグゼキュータ・ログで、エラーの原因を特定できます。

たとえば、起動後に新規サービスに障害が発生する理由を特定するには、モニター・ログの内容を表示して、障害の発生したサービスのエグゼキュータのログ・ファイル名を確認します。次に、いずれかのエグゼキュータ・ログ・ファイルの内容を表示して、障害の原因を特定します。原因としては、無効な SQL ATTACH 文を入力した、データベース・ファイル名を間違えて入力した、または、データベース・サービスのサービス所有者アカウントにデータベースへのアタッチ権限を付与していないなどが考えられます。このような問題は、すべてサービスの SQL ATTACH 文の失敗につながります。

7.7.4 クライアント接続時のエグゼキュータの問題

サービスは正常に起動するものの、新規クライアント接続用に作成されたエグゼキュータ・プロセスが起動時に失敗することがあります。この問題は、サービスの MIN_EXECUTORS 属性の値が 0 (ゼロ) に設定されている場合に発生する可能性があります。この場合、サービスは正常に起動されますが、新規クライアント接続用に作成されたエグゼキュータは起動時に失敗するため、サービスは最終的に失敗状態に変化します。この問題は、データベースがサービスの起動後に変更された場合にも発生する可能性があります。たとえば、データベースにアタッチする権限が、サービスの起動後にデータベース・サービスのサービス所有者アカウントから取り消され、最小数のエグゼキュータが作成されている場合、サービス用に作成された新規エグゼキュータは、サービスの ATTACH 文を実行しようとして失敗します。

ユーザーがサービスに接続しようとして、新規接続用に作成されたエグゼキュータが起動時に失敗した場合、モニターによってエグゼキュータ障害イベントがエグゼキュータ・ログの名前とともにモニター・ログに記録されます。その後、ディスパッチャによりディスパッチャ・ログにサマリー・エラー・メッセージが記録され、エグゼキュータ障害 (-2035) エラー・コードがエグゼキュータ起動エラー・メッセージとともにクライアント・アプリケーションに返されます。以前に失敗状態に移行したサービスにユーザーが接続しようとする、ディスパッチャによりディスパッチャのイベントが記録され、エグゼキュータ障害 (-2035) エラー・コードがサービス障害エラー・メッセージとともにクライアント・アプリケーションに返されます。

この種の問題を調査するには、最初にディスパッチャ・ログをチェックして、新規クライアント接続が拒否された理由を特定します。次に、モニター・ログを確認して、サービスのエグゼキュータの障害を詳細に説明したエントリを見つけます。最後に、エグゼキュータ・ログをチェックして、障害の原因を特定します。

7.7.5 クライアント・リクエスト実行時のエグゼキュータの問題

ほとんどのユーザーがサービスに正常にアクセスできるのに、ある特定のユーザーのエグゼキュータに障害が発生することがあります。この場合、ディスパッチャによってエグゼキュータ障害 (-2035) エラーがクライアント・アプリケーションに返され、モニターによりエグゼキュータ障害イベントがエグゼキュータ・ログの名前とともにモニター・ログに記録されます。最初にモニター・ログをチェックして障害の発生したエグゼキュータのログ・ファイル名を確認し、次にエグゼキュータのログをチェックして障害の原因を特定します。たとえば、特定のユーザーによりアクセスされるデータが、障害の発生したディスク上に存在している可能性があります。または、Oracle SQL/Services、Oracle Rdb または SQL に内部エラーが発生した可能性があります。この場合、エグゼキュータ・ログ・ファイルをチェックして、これらのコンポーネントのいずれかによりバグチェック・ダンプ・ファイルが生成されているかどうかを確認します。

7.7.6 内部エラーを原因とするサーバーの障害

ごくまれに、サーバー全体が停止することがあります。たとえば、コンポーネントに内部エラーが発生したか、重要な操作の実行中に障害が発生した可能性があります。この場合、サーバー構成の整合性が追加の影響を受けないように、サーバー全体が停止します。

Oracle SQL/Services モニター・プロセスでは、Oracle SQL/Services サーバー構成のすべてのプロセスが管理されます。したがって、モニター・ログ・ファイルから調査を開始するのが最適です。この場合、モニターでは常にバグチェック・ダンプが生成されます。ただし、ディスパッチャまたはエグゼキュータ・プロセスの以前の障害がエラーの原因である可能性もあります。したがって、次の手順では、現在のサーバー障害の前に障害が発生していたディスパッチャおよびエグゼキュータのログ・ファイルを確認します。これらのログ・ファイルをチェックして、Oracle SQL/Services、Oracle Rdb および SQL バグチェック・ダンプに対する参照を調査します。

問題の特定中にバグチェック・ダンプ・ファイルに対する参照を見つけた場合、オラクル社に問題レポート・フォームを提出する方法の詳細を [7.1 項](#) で参照してください。バグチェック・ダンプ・ファイルは、CREATE SERVER、ALTER SERVER、CREATE DISPATCHER または ALTER DISPATCHER コマンドを使用して別の場所を指定しないかぎり、デフォルトで SYS\$MANAGER に送られます。ただし、唯一の例外として、エグゼキュータのバグチェック・ダンプ・ファイルは、データベース認可に応じて接続ユーザーまたはサービス所有者のデフォルト・ディレクトリに書き込まれます。

7.7.7 Oracle SQL/Services のインストール後に機能しなくなるクライアントからの接続

マルチリリース環境で Oracle SQL/Services クライアント API をインストールすると、デフォルト以外のネットワーク・ポート番号に変更するかどうかを尋ねられます。デフォルト以外のポート番号は、特定のリリースの SQL/Services にアクセスするクライアントごとに、sqsap32.ini ファイルに指定する必要があります。各クライアントでは、次の項目を指定してファイルを変更する必要があります。

- 接続先のノード
- 使用するネットワーク・トランスポート
- ネットワークでリスニングするポート

たとえば、ノード NODE1 でポート 119 用に TCPIP を構成する場合、sqsap32.ini のエントリは次のようになります。

```
[NODE1]
TCPIPPortNumber=119
```

7.7.8 ネットワーク・エラー

プライマリ SQLSRV_NETERR エラーを受信した場合、ネットワーク・エラーのドキュメントを確認して、セカンダリ・エラー・メッセージで参照されているネットワーク・エラーを調査します。たとえば、表 7-1 および表 7-2 には、それぞれ DECnet と TCP/IP のプラットフォーム固有のエラー情報が含まれます。また、ネットワーク・エラーの結果として返されるセカンダリ・エラー・コードの詳細を、プラットフォーム固有の独自のドキュメントで確認する必要があります。

DECnet エラー・コードに関する情報は、表 7-1 にリストされた場所で確認できます。

表 7-1 DECnet のエラー・コード・ファイル

オペレーティング・システム	ファイル指定	説明
OpenVMS	SYS\$LIBRARY:SSDEF.H	システム・サービスにより返される DECnet のステータス・コード定義
Windows 95 以上	skernmo.h	DECnet エラー・コード (PATHWORKS SDK を参照)

TCP/IP エラー・コードに関する情報は、表 7-2 にリストされた場所で確認できます。

表 7-2 TCP/IP のエラー・コード・ファイル

オペレーティング・システム	ファイル指定	説明
OpenVMS	SYS\$LIBRARY:ERRNO.H	システム・サービスにより返される TCP/IP のステータス・コード定義
Windows 95 以上	winsock.h	TCP/IP エラー・コード (Microsoft Windows 95 SDK または Microsoft C++ を参照)

7.8 OCI クライアント・アプリケーションに返されるエラー・メッセージ

この項では、OCI クライアント・アプリケーションに返される発生頻度の高いエラー・メッセージについて説明します。その他のエラー・メッセージは、『Oracle Database エラー・メッセージ』を参照してください。

7.8.1 ログオン・エラー

ERROR: 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。

原因: 無効なログオン情報が入力されました。この問題は、USER\$ 表にユーザー名とパスワードのエントリが存在しない場合、またはプログラム検証が有効化されており、対応するエントリが ORA_VALID_PROGRAMS 表に存在しない場合にも発生する可能性があります。

処置: 正しい情報を入力して再度ログインします。詳細は、Oracle SQL/Services エグゼキュータ・ログ・ファイルを参照してください。

7.8.2 データベース設定エラー

ERROR: ORA-00904: 列名が無効です。

データベースが OCI Services for Oracle Rdb 用に正しく設定されていません。

詳細は、Oracle SQL/Services エグゼキュータ・ログ・ファイルを参照してください。

<... エグゼキュータ・ログの完全なファイル指定 ...>

原因: データベースが OCI Services for Oracle Rdb 用に正しく設定されていません。

処置: このエラーの詳細は、Oracle SQL/Services エグゼキュータ・ログ・ファイルを参照してください。データベースの準備に役立つ手順は、『Oracle Rdb Oracle SQL/Services インストール・ガイド』の 4.1 項を参照してください。

7.8.3 SQL 初期化ファイル・エラー

ERROR: ORA-00900: SQL 文が無効です。

SQL 初期化ファイルの実行エラーです。

詳細は、Oracle SQL/Services エグゼキュータ・ログ・ファイルを参照してください。

<... エグゼキュータ・ログの完全なファイル指定 ...>

原因: SQL 初期化ファイルの実行時エラーです。

処置: 詳細は、Oracle SQL/Services エグゼキュータ・ログ・ファイルを参照してください。

7.8.4 Rdb データベースへのアタッチ・エラーまたは Oracle SQL/Services データベース・サービスのエラー

ERROR: ORA-03113: 通信チャンネルで end-of-file が検出されました

このエラーが返される原因は様々です。詳細は、Oracle SQL/Services エグゼキュータ・ログ・ファイル、モニター・ログ・ファイルおよびディスパッチャ・ログ・ファイルを参照してください。各ログ・ファイルの調査方法の詳細は、7.3 項を参照してください。次の説明では、このエラーの一般的な原因をいくつか示します。実際の原因についてはログ・ファイルを確認してください。

原因: Oracle SQL/Services サービスが起動していないか、使用できません。

処置: サービスが起動しているかどうかをチェックし、起動していない場合はサービスを起動します。サービスに障害が発生した場合、障害の発生したエグゼキュータのログに対する参照を Oracle SQL/Services モニター・ログ・ファイルで確認します。その後、エグゼキュータ・ログ・ファイルで詳細情報を確認します。

原因: リクエストされた Oracle SQL/Services サービス名が無効です。

処置: Oracle SQL/Services ディスパッチャ・ログ・ファイルを調査して、リクエストされたサービスに関するエラー・エントリが存在しないかどうかを確認します。リクエストされた Oracle SQL/Services サービス名が無効である場合、対応する Oracle Net 構成ファイル TNSNAMES.ORA の Oracle SQL/Services サービス名を変更します。

原因: プロトコルが OCI に設定されていません。

処置: ディスパッチャを変更し、PROTOCOL OCI を指定します。次に、変更を反映するために Oracle SQL/Services ディスパッチャを停止して再起動します。

7.9 Oracle SQL/Services サーバーまたは OCI ディスパッチャが使用できない場合のエラー

7.9.1 ERROR: ORA-12203: TNS: 接続先に接続できません。

原因: Oracle SQL/Services および OCI_DISP ディスパッチャが起動していません。

処置: Oracle SQL/Services および OCI_DISP ディスパッチャを起動します。

原因: OCI_DISP ディスパッチャに障害が発生しました。

処置: ディスパッチャに障害が発生した場合、障害の発生したディスパッチャのログに対する参照を Oracle SQL/Services モニター・ログ・ファイルで確認します。その後、ディスパッチャ・ログ・ファイルで障害の原因を確認します。

ディスパッチャ・ログ・ファイルに次のエントリが含まれる場合、ディスパッチャでは、Oracle SQL/Services のディスパッチャ指定で 사용되는リスナー名の定義を検出できません。

```
%SQLSRV-E-TNSFAILURE, Oracle SQL*Net TNS nlpagas() service has failed
%SQLSRV-E-ERROR_TEXT, Error text: listener
```

Oracle Net 構成ファイル LISTENER.ORA を確認してください。このファイルの場所は、各インストール環境に応じて変わります。Oracle Net 構成ファイルを含むサブディレクトリは、次の場所に格納されています。

```
SYS$MANAGER:SQLSRV_SQLNET $m$ .DAT
```

m は、Oracle SQL/Services のリリース番号です。

このファイルに格納された場所の下の [.NETWORK.ADMIN] サブディレクトリに、LISTENER.ORA ファイルが含まれます。

7.9.2 Oracle Net サービス名が未定義の場合のエラー

ERROR: ORA-12154: TNS: サービス名を解決できませんでした。

原因: Oracle Net サービス名が適切に定義されていない可能性があります。

処置: Oracle Net 構成ファイルをチェックして、Oracle Net サービス名が適切に定義されているかどうかを確認します。ファイル・ベースの Oracle Net 構成を使用している場合は、TNSNAMES.ORA ファイルを確認します。このファイルの場所は、次のようにクライアント・システムのインストール場所に応じて変わります。

- OpenVMS システムの場合、構成ファイルの場所を含むサブディレクトリは、次の場所に格納されています。

```
SYS$MANAGER:SQLSRV_SQLNET $m$ .DAT
```

m は、Oracle SQL/Services のリリース番号です。

このファイルに格納された場所の下の [.NETWORK.ADMIN] サブディレクトリに、TNSNAMES.ORA ファイルが含まれます。

- Microsoft Windows システムの場合、次の場所に格納されています。

```
<Oracle installation directory>%network%admin
```

次に例を示します。

```
c:%orawin%network%admin
```

注意：状況によっては、Oracle SQL/Services により返されたエラー・コードを Oracle クライアントで解析できないことがあります。したがって、正確な内容を理解するため、常に Oracle SQL/Services のログ・ファイルを確認する必要があります。

表とリレーション

この付録には、次の情報が含まれます。

- Oracle データ・ディクショナリをエミュレートする Oracle メタデータ・ビュー
- ORA_SESSION 表に格納された現在のセッション情報
- 31 文字の表名に関する推奨事項
- Oracle Rdb データベースにアプリケーション・データを格納するのに必要な表およびビューの定義

A.1 Oracle メタデータ・オブジェクト

OCI Services for Oracle Rdb のインストール後、使用する各 Oracle Rdb データベースを準備する必要があります。PREPARE プロセスで次の操作を行います。

- セッションの管理を容易にするために OCI Services for Oracle Rdb によって使用されるプロシージャおよびファンクションのセットを作成します。
- Oracle データ・ディクショナリのエミュレートに使用されるリレーションおよびビューの大規模なセットを作成します。

これらのリレーションとビューにより、OCI クライアント・アプリケーションで Oracle メタデータを選択する方法が決まります。Oracle メタデータのすべての要素が Oracle Rdb データベースに存在するわけではないため、リレーションとビューのセットもそこに含まれるデータも完全ではありません。

ビュー、ファンクションおよびプロシージャの定義を表示するには、対話型 SQL を起動して SHOW VIEW、SHOW PROCEDURE または SHOW FUNCTION の各文を発行します。その後、ビューのどの列が固定値で、どの列に重要な情報があるかを確認できます。

次の表に、Oracle メタデータをエミュレートするために作成されるデータベース・オブジェクトを示します。

注意： アスタリスク (*) の付いたオブジェクトは、Oracle の順序機能をエミュレートするために、Oracle Rdb リリース 7.0 のデータベース専用に作成されます。Oracle Rdb リリース 7.1 では、順序はネイティブな形式で実装されます。

表 A-1 OCI Services for Oracle Rdb のオブジェクト

表

FRM50_BUFFER	GLOBAL_NAME	ORA_ALL_USERS
ORA_COMM_TRANS	ORA_OBJECTS	*ORA_SEQUENCES
ORA_VALID_PROGRAMS	PRODUCT_PROFILE	ROLE_SYS_PRIVS
ROLE_TAB_PRIVS	ROSLFDESC	SESSION_ROLES
SNAP\$	SQL_CAPABILITY	TOOL_ACCESS
TOOL_LRAW	TOOL_MODULE	USER\$
USER_ROLE_PRIVS	USER_SYS_PRIVS	USER_TABLESPACES
V\$OPTION	VG_COLOR	VG_IMAGE

一時表

DUAL	ORA_DBMS_OUTPUT	ORA_SESSION
V\$NLS_PARAMETERS	V\$SESSION	V\$SQLAREA
V\$VERSION		

ビュー

ACCESSIBLE_COLUMNS	ALL_ALL_TABLES	ALL_ARGUMENTS
ALL_CATALOG	ALL_COLL_TYPES	ALL_COL_COMMENTS
ALL_COL_PRIVS	ALL_CONSTRAINTS	ALL_CONS_COLUMNS
ALL_DB_LINKS	ALL_INDEXES	ALL_IND_COLUMNS
ALL_OBJECTS	ALL_OBJECT_TABLES	ALL_SEQUENCES
ALL_SNAPSHOTS	ALL_SYNONYMS	ALL_TABLES
ALL_TAB_COLUMNS	ALL_TAB_COMMENTS	ALL_TAB_PRIVS
ALL_TRIGGERS	ALL_TRIGGER_COLS	ALL_TYPES
ALL_USERS	ALL_VIEWS	CAT
DBA_CATALOG	DBA_COL_COMMENTS	DBA_COL_PRIVS
DBA_CONSTRAINTS	DBA_CONS_COLUMNS	DBA_DB_LINKS
DBA_OBJECTS	DBA_SNAPSHOTS	DBA_SYS_PRIVS
DBA_TABLES	DBA_TAB_COLUMNS	DBA_TAB_PRIVS
DBA_USERS	NLS_DATABASE_PARAMETERS	NLS_INSTANCE_PARAMETERS
NLS_SESSION_PARAMETERS	PRODUCT_COMPONENT_VERSION	PRODUCT_PRIVS
PRODUCT_USER_PROFILE	USER_ARGUMENTS	USER_CATALOG
USER_COL_COMMENTS	USER_COL_PRIVS	USER_CONSTRAINTS
USER_CONS_COLUMNS	USER_DB_LINKS	USER_INDEXES
USER_IND_COLUMNS	USER_OBJECTS	USER_SEQUENCES
USER_SNAPSHOTS	USER_SYNONYMS	USER_TABLES
USER_TAB_COLUMNS	USER_TAB_COMMENTS	USER_TAB_PRIVS
USER_TRIGGERS	USER_TRIGGER_COLS	USER_USERS
USER_VIEWS	V\$PARAMETER	

プロシージャ

ORA_APPLINFO_READ_CLIENT_INFO	ORA_APPLINFO_READ_MODULE	ORA_APPLINFO_SET_ACTION
ORA_APPLINFO_SET_CLIENT_INFO	ORA_APPLINFO_SET_MODULE	ORA_CREATE_DBLINK
ORA_CREATE_ERROR	ORA_CREATE_OBJECT	*ORA_CREATE_SEQUENCE
ORA_CREATE_TRANS	ORA_CREATE_USER	ORA_CVT_ROWID_TO_CHAR
ORA_CVT_TO_CHAR	ORA_CVT_TO_DATE	ORA_CVT_TO_NUMBER
ORA_CVT_TO_ROWID	ORA_DATE_ROUND	ORA_DATE_TRUNC
ORA_DELETE_ERROR	ORA_DELETE_OBJECT	ORA_DELETE_PHANTOMS
ORA_DELETE_TRANS	ORA_DROP_PASSWORD	*ORA_DROP_SEQUENCE
ORA_DROP_USER	ORA_ENCRYPT_PASSWORD	ORA_GETSYI
ORA_INIT	ORA_LOGIN	ORA_LOGIN2
ORA_LOGIN3	ORA_LOGON	ORA_OUTPUT_DISABLE
ORA_OUTPUT_ENABLE	ORA_OUTPUT_GET_LINE	ORA_OUTPUT_GET_LINES
ORA_OUTPUT_NEW_LINE	ORA_OUTPUT_PUT	ORA_OUTPUT_PUT_LINE
ORA_SET-NLS	ORA_SET_PASSWORD	ORA_TEST_BIT
ORA_TO_CHAR	V\$NLS_SET_FUNC	V\$NLS_UPDATE

ドメイン

NUMBER	ORA_LONG	ORA_MAXCHAR
ORA_OBJECT_NAME	RDB1LONG	RDB2LONG
RDB4LONG	RDBLONGRAW	

ファンクション

CHARTOROWID	*CURRVAL	*NEXTVAL
ORASTATE	ORA_DDL_RDB	ORA_NUMBER
ORA_OWNER	ROUN2	ROWIDTOCHAR
TEST_BIT	TO_CHAR	TO_DATE
TO_NUMBER	TRUN2	USEREN2
USERENV		

モジュール		
ORA_APPLINFO	ORA_CONVERT	ORA_LOG
ORA_MISC1	ORA_MISC2	ORA_MISC3
ORA_OUTPUT	*ORA_SEQUENCE	ORA_SESS
ORA_TRANS	ORA_USER	

A.2 現在のセッション情報

現在のセッションに固有の情報を含む複数の表があります。これらの表は、すべて Oracle Rdb のグローバル一時表であり、その内容は各セッションに固有です。（一時表の詳細は、『Oracle Rdb SQL リファレンス・マニュアル』を参照してください。）

V\$NLS_PARAMETERS 表には、すべての NLS パラメータの現在の設定に関する情報が含まれます。PARAMETER 列には NLS_PARAMETER の名前が含まれ、VALUE 列にはセッションでの現在の値が含まれます。

V\$VERSION 表には、基礎となるすべてのソフトウェア要素のバージョン番号が含まれます。単一行の BANNER には、次の要素のバージョン番号が含まれます。

- Oracle Rdb OCI サーバー
- Oracle Rdb SQL
- CORE
- TNS フォーム
- NLSRTL
- メタデータ・ビュー
- Oracle 互換リリース

V\$SESSSION 表には、USER# (USERID)、USERNAME、SCHEMA# (USER# と同じ)、SCHEMANAME (USERNAME と同じ)、OSUSER (USERNAME と同じ)、PROCESS (クライアントから送信されるプロセス ID)、MACHINE (クライアントから送信されるマシン名)、TERMINAL (クライアントから送信される端末名)、PROGRAM (クライアントから送信されるクライアント・プログラム名) などの情報が含まれます。

表 A-2 に、ORA_SESSION 表に格納される情報のタイプを示します。

表 A-2 ORA_SESSION 表

INFO_TYPE 列の文字列値	INFO 列の対応する文字列値
ERROR_MSG	エラー・メッセージ・テキスト。ORA_CREATE_ERROR のコールにより挿入され、エラーの SELECT 文により取得されます。
DEFAULT_SCHEMA	セッションのデフォルト・スキーマ。
INIT_STATEMENTn	ORA_INIT ストアド・プロシージャにより格納される初期化文。変数 n は、初期化文のゼロ・ベースの順序番号です。
DB_LINK	別名。ORA_CREATE_DBLINK のコールにより挿入されます。

A.3 31 文字のオブジェクト名の処理

OCI Services for Oracle Rdb では、31 文字のオブジェクト名がサポートされます。ただし、すべてのクライアント・アプリケーションで 31 文字の名前がサポートされるかどうかは不明であるため、30 文字以下のオブジェクト名を使用することをお勧めします。

30 文字を超える名前が存在するかどうかを確認するには、次の問合せを使用します。

オブジェクト	問合せ
制約	<pre>SELECT RDB\$CONSTRAINT_NAME FROM RDB\$RELATION_CONSTRAINTS WHERE CHARACTER_LENGTH(TRIM(RDB\$CONSTRAINT_NAME)) > 30;</pre>
フィールド	<pre>SELECT RDB\$FIELD_NAME FROM RDB\$RELATION_FIELDS WHERE CHARACTER_LENGTH(TRIM(RDB\$FIELD_NAME)) > 30;</pre>
索引	<pre>SELECT RDB\$INDEX_NAME FROM RDB\$INDICES WHERE CHARACTER_LENGTH(TRIM(RDB\$INDEX_NAME)) > 30;</pre>
モジュール	<pre>SELECT RDB\$MODULE_NAME FROM RDB\$MODULES WHERE CHARACTER_LENGTH(TRIM(RDB\$MODULE_NAME)) > 30;</pre>
ルーチン	<pre>SELECT RDB\$ROUTINE_NAME FROM RDB\$ROUTINES WHERE CHARACTER_LENGTH(TRIM(RDB\$ROUTINE_NAME)) > 30;</pre>
表	<pre>SELECT RDB\$RELATION_NAME FROM RDB\$RELATIONS WHERE CHARACTER_LENGTH(TRIM(RDB\$RELATION_NAME)) > 30;</pre>
トリガー	<pre>SELECT RDB\$TRIGGER_NAME FROM RDB\$TRIGGERS WHERE CHARACTER_LENGTH(TRIM(RDB\$TRIGGER_NAME)) > 30;</pre>

31 文字の名前をサポートするため、OCI Services for Oracle Rdb に付属する Oracle データ・ディクショナリ (メタデータ表) では、ドメイン ORA_OBJECT_NAME が VARCHAR(30) データ型ではなく VARCHAR(31) として定義されています。31 文字の長さの名前は、Oracle メタデータ・ビューの使用時に正しく表示されない可能性があります。

記号

@

- 間接コマンド・ファイル, 6-9, 6-44
 - SQLSRV_MANAGE, 6-9, 6-44
 - SQLSRV 管理, 6-9
- コマンド, 6-9, 6-44

数字

- 10 進表現, 4-4

A

- ALTER DISPATCHER コマンド, 6-10
- ALTER SERVER コマンド, 6-14
- ALTER SERVICE コマンド, 6-17
- ALTER SESSION 文, 5-2
 - LOG BRIEF を使用したロギング, 4-4
 - LOG 句, 5-4
 - NLS パラメータの変更, 5-2
 - SCHEMA EMULATION 句, 4-6
 - SET ISOLATION LEVEL 句, 5-2
 - SET NLS 句, 5-3
 - SET SCHEMA EMULATION 句, 5-3
 - キャラクタ・セットの指定, 4-13
 - 使用環境, 5-2
 - 書式, 5-2
 - データ書式設定, 4-3
 - 日付および数値データの書式設定, 4-3
- AR8ISO8859P6 キャラクタ・セット, 4-12
- AR8MSWIN1256 キャラクタ・セット, 4-12
- ATTACH 文, 4-7
- AUTHORIZE ユーティリティ, 2-33

B

- BIGINT(2) データ型, 4-4

C

- CAST 関数
 - DATE ANSI 形式との一致, 4-3
 - TO_DATE オブジェクトによる処理, 4-5
- CHAR データ型, 4-4
- CL8ISO8859P5 キャラクタ・セット, 4-12
- CL8MSWIN1251 キャラクタ・セット, 4-12
- CLOSE コマンド, 6-24

- CONNECT TO SERVER コマンド, 6-25
- CREATE DATABASE LINK 文
 - 例, 4-14
- CREATE DISPATCHER コマンド, 6-27
- CREATE SERVER コマンド, 6-30
- CREATE SERVICE コマンド, 6-33

D

- DATE ANSI 形式, 4-3
- DATE VMS データ型
 - TO_DATE オブジェクトによる処理, 4-5
- DATE VMS 日付, 4-3
- DATE データ型
 - DATE VMS 日付との類似, 4-3
- DDL
 - Oracle の動作を模倣する Oracle Rdb, 4-2
 - SQL 初期化ファイルの文の有効化, 4-6
- DECnet ソフトウェア
 - エラー・コード, 7-13
- Developer/2000
 - Oracle Forms の互換性, 4-7
- DISCONNECT SERVER コマンド, 6-39
- DOUBLE PRECISION データ型, 4-5
- DROP SERVER コマンド, 6-42
- DROP コマンド, 6-40

E

- EL8ISO8859P7 キャラクタ・セット, 4-12
- EL8MSWIN1253 キャラクタ・セット, 4-12
- ENQLM アカウントの割当て制限, 2-33
- EXIT コマンド, 6-43
- EXTRACT コマンド, 6-44

G

- GRANT USE ON SERVICE コマンド, 6-46

H

- HELP コマンド, 6-48

I

- input スイッチ, 6-7
- INTEGER データ型, 4-4
- IW8ISO8859P8 キャラクタ・セット, 4-12

IW8MSWIN1255 キャラクタ・セット, 4-12

J

JA16SJIS キャラクタ・セット, 4-12

JA16VMS キャラクタ・セット, 4-12

JTQUOTA アカウントの割当て制限, 2-33

K

KILL EXECUTOR コマンド, 6-49

KO16KSC5601 キャラクタ・セット, 4-12

L

LIST OF BYTE VARYING 列, 4-4

LOG 句

ALTER SESSION 文, 5-4

M

Microsoft Windows

ネットワーク・エラー・コード, 7-13

MODIFY コマンド

AUTHORIZE ユーティリティ, 2-33

N

NATCONN.COM, 「Rdb_NATCONN.COM」を参照

NETMBX 権限, 2-33

NLS_LANG 論理名, 4-13

NLS パラメータ

ALTER SESSION 文での変更, 5-3

NUMBER データ型, 4-4

O

OCI Services for Oracle Rdb

キャラクタ・セットの変換, 4-13

処理機能, 1-11, 4-2

接続文字列, 4-14

OCI Services for Oracle Rdb のログイン, 7-7

OCI アプリケーション

Oracle または Rdb に対して実行, 1-9

Oracle メタデータの選択, A-2

エミュレートされた Oracle データ・ディクショナリ
の使用, 4-5

カーソル管理, 4-2

新規および既存, 1-10

メッセージ・マッピング, 4-2

OCI テクノロジ, 1-10

クライアント・アプリケーションの Rdb への接続,
1-10

OCI でのメッセージ・マッピング, 4-2

ODBC ログイン, 7-7

OpenVMS オペレーティング・システム

ステータス・コード定義, 7-13

OPEN コマンド, 6-50

ORA_INIT スタアド・プロシージャ

DDL と他の文の格納, 4-6

SQL 初期化文の実行, 4-6

ORA-2025 エラー・メッセージ, 4-15

ORA-2085 エラー・メッセージ, 4-14

Oracle

DATABASE LINK 句, 4-14

SQL UPDATE 文, 4-15

エラー・メッセージ, 7-2

Oracle Call Interface (OCI), 1-10

Oracle Forms

OCI Services for Oracle Rdb との互換性, 4-7

Oracle Network Manager

エラーおよびトレース・ファイル名の変更, 7-8

接続文字列としての Rdb データベース名の指定, 4-7

Oracle Rdb

現行ユーザー名, 2-31

システム・ユーザー名, 2-30

セッション・ユーザー名, 2-30

Oracle Rdb のシステム・ユーザー名, 2-30

Oracle SQL ALTER SESSION 文

「ALTER SESSION 文」を参照

Oracle SQL/Services

オブジェクト

影響 (コマンド), 6-4

環境スイッチ, 6-6

初期化ファイル, 4-6

データベース・サービスの推奨, 2-19

マネージャ・クライアント GUI

定義, 1-4

Oracle システム識別子 (SID)

データベース・リンクでの使用, 4-14

Oracle データ・ディクショナリ

31 文字の表名のサポート, A-6

TO_CHAR オブジェクト, 4-5

TO_DATE オブジェクト, 4-5

TO_NUMBER オブジェクト, 4-5

USERENV 関数, 4-5

エミュレーション, 4-5, A-2

クライアントの間合せ, 1-10

マルチスキーマ・エミュレーション, 4-6

Oracle レベル 1 言語

DATE VMS データ型, 4-5

SQL 文の解析, 4-3

-output スイッチ, 6-8

P

PAGFILCNT システム・パラメータ, 2-33

PGFLQUOTA システム・パラメータ, 2-33

PL/SQL 文

DML が機能しない場合の使用例, 4-16

データベース・リンクを通じた使用, 4-15

Pro* コンパイラ

クライアント・アプリケーションによるデータベース
へのアクセス, 1-10

R

Rdb

Oracle のデータベース表の参照, 4-14

RDB\$DBHANDLE 別名

ATTACH 文での設定, 4-7

RDB\$DDTM_XG_INFO, 2-37

RDB_NATCONN.COM, 4-8

DROP, 4-9

MODIFY_USER, 4-10

NEW_USER, 4-9

PREPARE, 4-8
REMOVE_USER, 4-11
SHOW_USERS, 4-11
UPGRADE, 4-9
RESTART SERVER コマンド, 6-51
REVOKE USE ON SERVICE コマンド, 6-52

S

SET CONFIGURATION_FILE コマンド, 6-54
SET CONFIRM コマンド, 6-55
SET CONNECTION コマンド, 6-56
SET ISOLATION LEVEL
 READ COMMITTED 句, 5-2
 SERIALIZABLE 句, 5-2
SET NLS 句
 ALTER SESSION 文, 5-3
SET OUTPUT コマンド, 6-58
SET SCHEMA EMULATION
 RELAXED 句, 5-3
 STRICT 句, 5-3
SET VERIFY コマンド, 6-59
SHOW CLIENTS コマンド, 6-60
SHOW CONNECTS コマンド, 6-63
SHOW DISPATCHER コマンド, 6-64
SHOW SERVER コマンド, 6-66
SHOW SERVICE コマンド, 6-67
SHOW SETTINGS コマンド, 6-69
SHOW VERSION コマンド, 6-70
SHOW コマンド
 サーバー・アクティビティの監視, 3-2
SHUTDOWN DISPATCHER コマンド, 6-71
SHUTDOWN SERVER コマンド, 6-72
SHUTDOWN SERVICE コマンド, 6-73
SID パラメータ
 データベース・リンクでの使用, 4-14
SQL ALTER SESSION 文
 「ALTER SESSION 文」を参照
SQL*Net
 Oracle クライアントの Rdb サーバーへの接続, 1-9
SQL/Services の開始, 2-2
SQLNET_BLOB, 2-36
SQLNET_BUGCHECK_FILE, 2-36
SQLNET_DEBUG_FLAGS, 2-36
SQLNET_DOMAIN, 2-36
SQLNET_RECO_USER, 2-36
SQLNET_STRUCTURED_DATE_TYPES, 2-37
SQLNET_TIMESTAMP_DATE_TYPE, 2-37
SQLNET_VALIDATE_PROGRAM, 2-37
SQLSRV\$LOG_CONNECTIONS, 2-37
SQLSRV\$MAX_EXECUTOR_FAILURES, 2-36
SQLSRV_DISP_DUMPPATH, 2-36
SQLSRV_DISP_LOGPATH, 2-36
SQLSRV_EXEC_LOG, 2-36
SQLSRV_MANAGE クライアント
 定義, 1-4
SQLSRV_MANAGE ユーティリティ
 @ 間接コマンド・ファイル, 6-9, 6-44
 エグゼキュータの中断, 6-49
 オフライン管理, 1-7
 オンライン管理, 1-7
 システム管理, 1-6
 実行, 1-6

終了, 6-43
出力ファイルの指定, 6-8
出力ファイルを閉じる, 6-24
出力ファイルを開く, 6-50
入力ファイルの指定, 6-7
必要な権限, 1-5
ヘルプの表示, 6-48

SQLSRV 管理ユーティリティ
 @ 間接コマンド・ファイル, 6-9
SQL 初期化ファイル, 2-36, 4-6
 構文表記規則, 4-8
SQL 初期化ファイルの使用, 2-36
SQL 文
 ALTER SESSION を使用したキャラクタ・セットの指
 定, 4-13
 OCI Services for Oracle Rdb による変更, 4-3
 Oracle セマンティクスのエミュレート, 1-9
 Oracle または Rdb データベースに対して実行, 1-11
 SQL 初期化ファイル, 4-6
 カーソル・セマンティクス, 4-2
 解析, 4-3
 クライアントとサーバー間の転送のロギング, 5-4
 失敗の解析, 4-3
START DISPATCHER コマンド, 6-74
START SERVER コマンド, 6-75
START SERVICE コマンド, 6-77
SYSGEN ユーティリティ
 使用, 2-33

T

TCP/IP ソフトウェア
 エラー・コード, 7-13
TH8TISASCII キャラクタ・セット, 4-12
TINYINT 列, 4-4
TMPMBX 権限, 2-33
TNSNAMES.ORA ファイル, 4-14
 例, 4-14
TO_CHAR 関数
 Oracle データ・ディクショナリのエミュレーション,
 4-5
 書式文字列の指定, 4-5
 処理, 4-3
TO_DATE 関数
 Oracle データ・ディクショナリのエミュレーション,
 4-5
 処理, 4-3
 文の解析, 4-3
TO_NUMBER 関数
 Oracle データ・ディクショナリのエミュレーション,
 4-5
 処理, 4-3

U

UPDATE 文
 データベース・リンクを通じた使用例, 4-15
 無効な使用と回避方法, 4-15
US7ASCII キャラクタ・セット, 4-13
USERENV 関数, 4-5
USE 権限
 取消し, 6-52
 付与, 6-46

UTF8 キャラクタ・セット, 4-12

V

VARCHAR データ型, 4-4, 4-5

VIRTUALPAGECNT システム・パラメータ, 2-33

W

.WE8DEC キャラクタ・セット, 4-12

WE8ISO8859P15 キャラクタ・セット, 4-12

WE8ISO8859P1 キャラクタ・セット, 4-12

WE8MSWIN1252 キャラクタ・セット, 4-12

Windows 95

ネットワーク・エラー・コード, 7-13

Windows NT Alpha

ネットワーク・エラー・コード, 7-13

Windows NT X86

ネットワーク・エラー・コード, 7-13

Winsock ログイン, 7-8

Z

ZHS16CGB2312-80 キャラクタ・セット, 4-12

ZHS32GBI1030 キャラクタ・セット, 4-12

ZHT16BIG5 キャラクタ・セット, 4-12

あ

アーキテクチャ

クライアント / サーバー, 1-3

サーバー・システム, 1-2

アプリケーション

汎用サービスを使用した Rdb とのタッチ, 4-7

複数の ATTACH 文の実行, 4-7

アプリケーション開発

共通の SQL プログラム, 1-11

い

移植可能データ型, 4-2

インストール

Oracle データ・ディクショナリのエミュレート用に
作成されるリレーションおよびビュー, A-2

え

エグゼキュータ

中断, 6-49

定義, 1-2

プロセス構成, 2-29

プロセスの特性, 2-29

ログ・ファイル, 7-5

エグゼキュータの障害と問題, 7-11

エグゼキュータ・プロセス

構成メカニズム, 2-29

事前起動, 4-7

エグゼキュータ・プロセスの構成, 2-29

エミュレーション

Oracle データ・ディクショナリ, 4-5

マルチスキーマ・データベース, 4-6

エラー・コード・ファイル

DECnet の場所, 7-13

TCP/IP の場所, 7-13

エラー処理

文の解析, 4-3

エラー・メッセージ

ALTER SESSION 文でのログイン, 5-4

OCI クライアント・アプリケーションに返される,
7-13

オラクル社への問題のレポート, 7-2

キャラクタ・セットの指定時, 7-2

エラー・ログ・ファイル

エグゼキュータ, 7-5

ディスパッチャ, 7-3

モニター, 7-3

お

オブジェクト

DROP SERVER コマンド, 6-42

DROP コマンド, 6-40 ~ 6-41

影響 (コマンド), 6-4

削除, 6-40 ~ 6-41

オフライン・システム管理, 1-7

オペレーティング・システムのプロセス・ユーザー名,
2-29

オンライン・システム管理, 1-7

か

カーソル

SQL セマンティクス, 4-2

管理, 4-2

解析

SQL 文, 4-3

外部ファンクション

使用, 2-33

外部ファンクションの使用, 2-33

確認プロンプト

表示, 6-55

確認プロンプトの表示, 6-55

環境コマンド, 6-1, 6-6

環境スイッチ, 6-1, 6-6

管理コマンド, 6-4

き

記憶域, 2-33

起動

サーバー, 6-75 ~ 6-76

サービス, 6-77

ディスパッチャ, 6-74

キャラクタ・セット

ALTER SESSION 文を使用した指定, 4-13

US7ASCII, 4-13

エラー・メッセージ, 7-2

規則および推奨事項, 4-13

サーバーでの定義, 4-12

デフォルト, 4-12

西ヨーロッパ言語を指定する例, 4-13

フランス語と地域を指定する例, 4-13

マルチバイト, 4-12

共有メモリー・サイズ, 2-3

く

- クライアント
 - SHOW CLIENTS コマンド, 6-60 ~ 6-62
 - 識別および認証, 2-24
 - 定義, 1-2
 - 表示, 6-60 ~ 6-62
- クライアント・アプリケーション
 - OCI Services for Oracle Rdb との対話, 1-10
 - Rdb または Oracle に対して実行, 1-9
 - 返されるエラー・メッセージ, 7-13
 - 問合せ, 4-7
 - プログラミング, 1-11
- クライアント / サーバー処理, 1-9
- クライアント・システム
 - エラーおよびトレース・メッセージ, 7-8
 - キャラクタ・セットの定義, 4-12, 4-13
 - キャラクタ・セットを指定する際の規則および推奨事項, 4-13
 - サーバー・システムとのキャラクタ・セットの互換性, 4-13
- クライアント接続
 - 監視, 3-2
- クライアント接続状態
 - セッションで再利用可能なサービス, 3-2
 - トランザクションで再利用可能なデータベース・サービス, 3-3
- クライアント接続の監視, 3-2
- クライアントの識別, 2-24
- クライアントの認証, 2-24

け

- 決定
 - サービスへのアクセス権, 2-22
 - データベース・アクセス認可, 2-20
 - データへのアクセス, 2-23
 - デフォルトの接続ユーザー名, 2-22
- 権限
 - GRANT USE ON SERVICE コマンド, 6-46 ~ 6-47
 - REVOKE USE ON SERVICE コマンド, 6-52 ~ 6-53
 - システム管理に必要, 1-5
 - 取消し, 6-52 ~ 6-53
 - 付与, 6-46 ~ 6-47
- 権限の取消し, 6-52
- 言語
 - サポートされるキャラクタ・セット, 4-12

こ

- 構成
 - OCI による多様な組合せの実現, 1-10
 - コピー, 2-8
 - サーバーの削除, 2-8
 - サーバーの作成, 2-5
 - サーバーの変更, 2-7
 - サービスの再起動, 2-11
 - サービスの削除, 2-13
 - サービスの作成, 2-10
 - サービスの停止, 2-11
 - サービスの変更, 2-12
 - ディスパッチャの再起動, 2-9
 - ディスパッチャの削除, 2-10

- ディスパッチャの作成, 2-8
- ディスパッチャの停止, 2-9
- ディスパッチャの変更, 2-9
- 構成のコピー, 2-8
- 構成ファイル
 - SET CONFIGURATION_FILE コマンド, 6-54
 - 定義, 1-4
 - 表示, 6-69
- コマンド
 - 管理コマンド, 6-1
- コマンド・ファイル出力の表示, 6-59
- コマンド・ファイル入力の検証, 6-59

さ

- サーバー
 - ALTER SESSION 文を使用した情報の受渡し, 4-3
 - CONNECT TO SERVER コマンド, 6-25 ~ 6-26
 - DISCONNECT SERVER コマンド, 6-39
 - RESTART SERVER コマンド, 6-51
 - SET CONNECTION コマンド, 6-56 ~ 6-57
 - SHOW SERVER コマンド, 6-66
 - SHUTDOWN SERVER コマンド, 6-72
 - START SERVER コマンド, 6-75 ~ 6-76
 - アーキテクチャ, 1-3
 - アクティビティの監視, 3-2
 - オペレーティング・システムの独立, 1-10
 - 管理コマンド, 6-1
 - 管理方法, 1-5
 - 管理ユーティリティ, 1-5
 - 起動, 6-75 ~ 6-76
 - 計画, 2-2
 - 構成ファイル, 6-54
 - 再起動, 6-51
 - 削除, 2-8, 6-42
 - 作成, 6-30 ~ 6-32
 - システム, 1-2
 - システム管理, 2-5
 - システム要件, 2-2
 - 接続, 6-25 ~ 6-26
 - 接続の切断, 6-39
 - 属性, 2-7
 - 定義
 - ALTER SERVER コマンド, 6-14 ~ 6-16
 - CREATE SERVER コマンド, 6-30 ~ 6-32
 - 停止, 6-72
 - デフォルト設定, 2-7
 - 表示, 6-66
 - 変更, 6-14 ~ 6-16
- サーバー・オブジェクト定義
 - 抽出, 6-44
- サーバー側のソリューション, 1-10
- サーバー・システム
 - キャラクタ・セットの定義, 4-12
 - キャラクタ・セットの定義の例, 4-13
 - キャラクタ・セットを指定する際の規則および推奨事項, 4-13
- サーバーとの接続の切断, 6-39
- サーバーの計画, 2-2
- サーバーへの接続, 6-25
- サービス
 - GRANT USE ON SERVICE コマンド, 6-46
 - SHOW SERVICE コマンド, 6-67 ~ 6-68

SHUTDOWN SERVICE コマンド, 6-73
START SERVICE コマンド, 6-77
アクセス権
制限, 2-22
データ, 2-23
認可, 2-25, 2-26
アクセス権の付与, 2-22
起動, 6-77
サービス・タイプの選択, 2-19
再起動, 2-11
削除, 2-13
作成, 6-33 ~ 6-38
システム管理, 2-10
所有者, 2-34
OpenVMS, 2-32, 2-33, 2-34
データベース・アクセス認可, 2-20, 2-21
セッション再利用, 2-15
セッションで再利用可能 (データベース), 2-17
セッションで再利用可能 (汎用), 2-16
属性, 2-12
定義, 1-2
ALTER SERVICE コマンド, 6-17 ~ 6-23
CREATE SERVICE コマンド, 6-33 ~ 6-38
停止, 6-73
データベース, 2-22
データベース・アクセス認可, 2-20
デフォルト設定, 2-12
デフォルトの接続ユーザー名, 2-22
トランザクション再利用, 2-15
トランザクションで再利用可能 (データベース),
2-17, 2-18
汎用, 2-14, 2-22
表示, 6-67 ~ 6-68
変更, 6-17 ~ 6-23
サービス所有者
データベース・アクセス認可, 2-20, 2-21, 2-32
サービス所有者ユーザー名
選択, 2-32
サービス所有者ユーザー名の選択, 2-32
サービス・タイプの選択, 2-19
サービスへのアクセス権の制限, 2-22
再起動
サーバー, 6-51
サービス, 2-11
ディスパッチャ, 2-9
削除
オブジェクト, 6-40 ~ 6-41
サーバー, 2-8, 6-42
サービス, 2-13
ディスパッチャ, 2-10
データベース, 4-9
ユーザー名, 4-11
作成
サーバー, 2-5, 6-30 ~ 6-32
サービス, 2-10, 6-33 ~ 6-38
ディスパッチャ, 2-8, 6-27 ~ 6-29

し

システム管理
Oracle SQL/Services Manager GUI ユーティリティ,
1-6
SQLSRV_MANAGE ユーティリティ, 1-6

オフライン, 1-7
オンライン, 1-7
共有メモリー・サイズの設定, 2-3
構成のコピー, 2-8
サーバー, 2-5
サーバー・アクティビティの監視, 3-2
サーバー・システム要件, 2-2
サーバーの管理方法, 1-5
サーバーの計画, 2-2
サービス, 2-10
設定
セキュリティ, 2-24
ディスパッチャ, 2-13
ディスパッチャ, 2-8
必要な権限, 1-5
終了
SQLSRV_MANAGE 環境, 6-43
出力
-output スイッチ, 6-8
SET OUTPUT コマンド, 6-58
出力ファイルの指定, 6-8
ファイル
クローズ, 6-24
出力の送信, 6-58
出力ファイルを閉じる, 6-24
出力ファイルを開く, 6-50
障害リカバリ, 7-9
初期化ファイル
SQL 実行環境の初期化, 4-6
汎用サービス設定時の構文表記規則, 4-8
書式
データ, 4-3
書式文字列
TO_CHAR 関数での指定, 4-5

す

数値データ
書式設定, 4-3
数値データ型, 4-4

せ

静的な表の内容, A-2
セキュリティ
各層の動作, 2-27
クライアント識別および認証, 2-24
サーバー, 2-27
サービス・アクセス認可, 2-25
データベースおよびデータ・アクセス認可, 2-26
メカニズム, 2-24
セッション
キャラクタ・セット指定の推奨事項, 4-13
セッション再利用サービス, 2-15
セッション情報, A-5
セッションで再利用可能なデータベース・サービス,
2-17
セッションで再利用可能な汎用サービス, 2-16
接続
SET CONNECTION コマンド, 6-56
SHOW CONNECTS コマンド, 6-63
現在の接続として確立, 6-56

接続文字列

- Rdb データベース名の指定, 4-7
- Rdb データベース名を指定する例, 4-14

接続ユーザー名, 2-34

- OpenVMS, 2-32, 2-34
- データベース・アクセス認可, 2-20, 2-21, 2-32
- デフォルト, 2-22

設定

- SET VERIFY コマンド, 6-59
- SHOW SETTINGS コマンド, 6-69
- 確認, 6-55
- 共有メモリー・サイズ, 2-3
- 検証, 6-59, 6-59
- 構成ファイル, 6-54
- 出力, 6-58
- セキュリティ, 2-24
- 接続, 6-56 ~ 6-57
- ディスパッチャ, 2-13
- 表示, 6-69
- ユーザー名の設定に使用されるメカニズム, 2-29

た

ダンプ・ファイル

- 「バグチェック・ダンプ・ファイル」を参照

ち

中断

- 指定したエグゼキュータ, 6-49

て

停止

- サーバー, 6-72
- サービス, 2-11, 6-73
- ディスパッチャ, 2-9, 6-71

ディスパッチャ

- SHOW DISPATCHER コマンド, 6-64 ~ 6-65
- SHUTDOWN DISPATCHER コマンド, 6-71
- START DISPATCHER コマンド, 6-74
- 再起動, 2-9
- 削除, 2-10
- 作成, 6-27 ~ 6-29
- システム管理, 2-8
- 設定, 2-13
- 属性, 2-9
- 定義
 - ALTER DISPATCHER コマンド, 6-10 ~ 6-13
 - CREATE DISPATCHER コマンド, 6-27 ~ 6-29
- デフォルト設定, 2-9
- トランスポート選択, 2-13
- 変更, 6-10 ~ 6-13
- ログ・ファイル, 7-3

データ型

- 10進表現, 4-4
- DOUBLE PRECISION, 4-5
- OCI Services for Oracle Rdb の説明, 4-4
- 変換, 4-2
- データ書式設定, 4-3
 - ALTER SESSION 文での情報のロギング, 5-4
- データベース
 - OCI アプリケーションでのアクセス, 1-10

- データベース・リンクを使用した接続, 4-14
- 複数のクライアント・アプリケーションによる Rdb とのアタッチ, 4-7

マルチスキーマ, 4-6

リモート・アクセス, 4-14

- データベース・アクセス認可, 2-26
- サービス所有者, 2-20, 2-21, 2-34
 - OpenVMS, 2-32, 2-33, 2-34
- 接続ユーザー名, 2-20, 2-21, 2-34
 - OpenVMS, 2-32, 2-34

汎用サービス, 2-32

データベース・オブジェクト

- Oracle データ・ディクショナリのエミュレーション用にインストール, 4-5

データベース・サービス, 2-14, 2-22

- 推奨, 2-19
- セッションで再利用可能, 2-17
- データベース・アクセス認可の設定, 2-21, 2-34
 - OpenVMS, 2-32, 2-33
- トランザクションで再利用可能, 2-17

データベース認可, 2-26

データベースのアップグレード, 4-9

データベースの準備, 4-8

データベース・リンク

- SQL UPDATE 文の例, 4-15

制限事項, 4-15

定義例, 4-14

デバッグ・フラグ

- エグゼキュータ・ログ・ファイル, 7-5

デフォルト設定

- サーバー, 2-7
- サーバー・オブジェクト, 2-7
- サービス, 2-12
- ディスパッチャ, 2-9
- ディスパッチャ・オブジェクト, 2-9

と

トランザクション再利用サービス, 2-15

- トランザクションで再利用可能なデータベース・サービス, 2-17

トランスポート選択

- ディスパッチャ, 2-13

に

入力

- input スイッチ, 6-7
- 入力ファイルの指定
 - SQLSRV_MANAGE, 6-7

認可

- サービス・アクセス, 2-25
- データベースおよびデータ・アクセス, 2-20, 2-26

ね

ネイティブ・データ型, 4-2

ネイティブのバイナリ・データ型, 4-4

ネットワーク

- DECnet エラー・コード, 7-13
- TCP/IP エラー・コード, 7-13
- ネットワーク・エラー, 7-13
- ネットワーク・トランスポートの問題, 7-10

は

バージョン

SHOW VERSION コマンド, 6-70

バグチェック・ダンプ・ファイル, 7-5

はじめに, xiii

パスワードの変更, 4-10

汎用サービス, 2-14, 2-22

Rdb データベースの指定, 4-7

SQL 初期化ファイルの構文表記規則, 4-8

サポート, 4-7

セッションで再利用可能, 2-16

データベース・アクセス認可の設定, 2-20, 2-32, 2-34

ひ

日付リテラル

書式設定, 4-3

ビュー

インストール済, A-1

表

インストール済, A-1

表示

クライアント, 6-60 ~ 6-62

サーバー, 6-66

サービス, 6-67 ~ 6-68

接続, 6-63

設定, 6-69

ディスパッチャ, 6-64 ~ 6-65

バージョン, 6-70

ユーザー名, 4-11

表名

31 文字のサポート, A-6

ふ

ファイル

ODBC ロギング, 7-7

SQL 初期化, 2-36

Winsock ロギング, 7-8

エグゼキュータ・ログ・ファイル, 7-5

ディスパッチャ・ログ・ファイル, 7-3

入力, 6-7

バグチェック・ダンプ・ファイル, 7-5

モニター・ログ・ファイル, 7-3

不明なユーザー

認可, 2-22, 6-18, 6-34

不明なユーザーの認可, 2-22, 6-18, 6-34

付与

権限, 6-46

サービスへのアクセス権, 2-22

プロセス障害

エグゼキュータ, 7-9

ディスパッチャ, 7-9

モニター, 7-9

プロパティ・クラス

RDB_PROPERTY_CLASS を使用した Oracle Forms の変更, 4-7

分離レベル

ALTER SESSION 文での設定, 5-2

へ

変更

サーバー, 2-7, 6-14 ~ 6-16

サービス, 2-12, 6-17 ~ 6-23

ディスパッチャ, 2-9, 6-10 ~ 6-13

ま

マネージャ・クライアント GUI ユーティリティ, 1-4, 1-6

システム管理, 1-6

マルチスキーマ・エミュレーション, 4-6

適度なまたは厳格なエミュレーションの設定, 5-3

マルチスキーマ・データベース, 4-6

マルチバイト・キャラクタ・セット, 4-12

め

メカニズム

エグゼキュータ・プロセスの構成に使用, 2-29

セキュリティ, 2-24

ユーザー名の設定に使用, 2-29

メタデータ

OCI アプリケーションによる選択, A-2

Oracle データ・ディクショナリからの取得, 1-10

Oracle データ・ディクショナリのエミュレーション, 4-5

メモリー

サイズの設定, 2-3

も

モニター

サーバー・アクティビティ, 3-2

定義, 1-4

ログ・ファイル, 7-3

問題

ログ・ファイルのチェック, 7-5

問題の特定

ログ・ファイルのチェック, 7-5

問題のレポート, 7-2

ゆ

ユーザーとパスワードの追加, 4-9

ユーザー認証および認可の問題, 7-10

ユーザー名

Oracle Rdb システム, 2-30

Oracle Rdb セッション, 2-30

Oracle Rdb の現行ユーザー, 2-31

オペレーティング・システムのプロセス, 2-29

接続ユーザー名, 2-20, 2-21, 2-32

ユーザー名の表示, 4-11

ユーティリティ

システム管理, 1-5

り

リカバリ, 7-9

ろ

ロギング

SQL 初期化ファイルでの有効化, 4-6
無効化, 7-6

ロギングの無効化, 7-6

ログ・ファイル, 7-3

エグゼキュータ, 7-5

エラー, 7-3, 7-5

ディスパッチャ, 7-3

モニター, 7-3

問題の特定, 7-5

論理名, 2-36

