

**Oracle® Database**

管理者リファレンス

10g リリース 2 (10.2) for UNIX Systems

部品番号 : B19278-06

2009 年 5 月

Oracle Database 管理者リファレンス , 10g リリース 2 (10.2) for UNIX Systems

部品番号 : B19278-06

Oracle Database Administrator's Reference, 10g Release 2 (10.2) for UNIX-Based Operating Systems

原本部品番号 : B15658-06

原著者 : Brintha Bennet

原本協力者 : Kevin Flood, Pat Huey, Clara Jaeckel, Emily Murphy, Terri Winters, Ashmita Bose, David Austin, Subhranshu Banerjee, Mark Bauer, Robert Chang, Jonathan Creighton, Sudip Datta, Padmanabhan Ganapathy, Thirumaleshwara Hasandka, Joel Kallman, George Kotsovolos, Richard Long, Rolly Lv, Padmanabhan Manavazhi, Matthew Mckerley, Sreejith Minnanghat, Krishna Mohan, Rajendra Pingte, Hanlin Qian, Janelle Simmons, Roy Swonger, Lyju Vadassery, Douglas Williams

Copyright © 2006, 2009, Oracle and/or its affiliates. All rights reserved.

#### 制限付権利の説明

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントが、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供される場合は、次の Notice 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 USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

このソフトウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアは、危険が伴うアプリケーション（人的傷害を発生させる可能性があるアプリケーションを含む）への用途を目的として開発されていません。このソフトウェアを危険が伴うアプリケーションで使用する場合、このソフトウェア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。このソフトウェアを危険が伴うアプリケーションで使用したこと起因して損害が発生しても、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。

Oracle は Oracle Corporation およびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

このソフトウェアおよびドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても、一切の責任を負いかねます。

---

---

# 目次

はじめに .....	ix
対象読者 .....	x
ドキュメントのアクセシビリティについて .....	x
関連ドキュメント .....	xi
表記規則 .....	xv
コマンド構文 .....	xvi
用語 .....	xvi
ドキュメントへのアクセス .....	xvii
サード・パーティ・ソフトウェア情報 .....	xvii
サポートおよびサービス .....	xviii
<b>1 Oracle Database の管理</b>	
<b>概要</b> .....	1-2
<b>環境変数</b> .....	1-2
Oracle Database の環境変数 .....	1-2
UNIX 環境変数 .....	1-4
共通の環境設定 .....	1-6
システム・タイム・ゾーンの設定 .....	1-7
<b>初期化パラメータ</b> .....	1-7
DB_BLOCK_SIZE 初期化パラメータ .....	1-7
ASM_DISKSTRING 初期化パラメータ .....	1-8
LOG_ARCHIVE_DEST_n 初期化パラメータ .....	1-8
<b>オペレーティング・システムのアカウントとグループ</b> .....	1-9
Oracle ソフトウェア所有者アカウント .....	1-9
OSDBA、OSOPER および Oracle インベントリ・グループ .....	1-9
グループとセキュリティ .....	1-10
外部認証 .....	1-10
orapwd ユーティリティの実行 .....	1-10
パスワード管理 .....	1-11
オペレーティング・システム・アカウントの追加 .....	1-11
Oracle ユーザーのアカウントの構成 .....	1-11
<b>RAW デバイスの使用</b> .....	1-12
RAW デバイスを使用する場合のガイドライン .....	1-12
RAW デバイスの設定 .....	1-13

AIX および Tru64 UNIX システムでの RAW デバイスのデータファイル .....	1-13
Linux システムでの RAW デバイスのサポート .....	1-13
トレース・ファイルおよびアラート・ファイルの使用 .....	1-14
トレース・ファイル .....	1-14
アラート・ファイル .....	1-14

## 2 Oracle ソフトウェアの停止と起動

Oracle プロセスの停止と起動 .....	2-2
Mac OS X での Oracle プロセスの起動 .....	2-2
Oracle Database インスタンスおよび自動ストレージ管理インスタンスの停止と起動 .....	2-2
Oracle CSS デーモンの停止と起動 .....	2-4
Oracle Net Listener の停止と起動 .....	2-4
iSQL*Plus の停止と起動 .....	2-5
Oracle Ultra Search の停止と起動 .....	2-6
Oracle Enterprise Manager Database Control の停止と起動 .....	2-7
Oracle Management Agent の停止と起動 .....	2-8
停止と起動の自動化 .....	2-9
Mac OS X におけるデータベースの停止と起動の自動化 .....	2-9
その他のオペレーティング・システムにおけるデータベースの起動と停止の自動化 .....	2-12

## 3 Oracle Database の構成

Oracle 製品の追加に対する Oracle Database の構成 .....	3-2
スタンドアロン・ツールとしてのコンフィギュレーション・アシスタントの使用 .....	3-2
Oracle Net コンフィギュレーション・アシスタントの使用 .....	3-2
Oracle Database アップグレード・アシスタントの使用 .....	3-3
Oracle Database Configuration Assistant の使用 .....	3-3
新規またはアップグレード済データベースの構成 .....	3-3
実行可能ファイルの再リンク .....	3-4

## 4 SQL\*Plus の管理

コマンドライン SQL*Plus の管理 .....	4-2
設定ファイルの使用 .....	4-2
PRODUCT_USER_PROFILE 表の使用 .....	4-2
Oracle Database サンプル・スキーマの使用 .....	4-3
SQL*Plus のコマンドライン・ヘルプのインストールと削除 .....	4-3
SQL*Plus のコマンドライン・ヘルプのインストール .....	4-3
SQL*Plus のコマンドライン・ヘルプの削除 .....	4-3
コマンドライン SQL*Plus の使用 .....	4-4
SQL*Plus からのシステム・エディタの使用 .....	4-4
SQL*Plus からのオペレーティング・システム・コマンドの実行 .....	4-4
SQL*Plus への割込み .....	4-5
SPOOL コマンドの使用 .....	4-5
SQL*Plus の制限事項 .....	4-5
ウィンドウのサイズ変更 .....	4-5
リターン・コード .....	4-5
パスワードの非表示 .....	4-5

## 5 Oracle Net Services の構成

Oracle Net Services 構成ファイルの保存場所 .....	5-2
Adapters ユーティリティ .....	5-3
Oracle Protocol Support .....	5-4
IPC プロトコル・サポート .....	5-4
TCP/IP プロトコル・サポート .....	5-5
SSL 付き TCP/IP プロトコル・サポート .....	5-5
TCP/IP または SSL 付き TCP/IP 用のリスナーの設定 .....	5-6
Oracle Advanced Security .....	5-6

## 6 Oracle プリコンパイラおよび Oracle Call Interface の使用

Oracle プリコンパイラの概要 .....	6-2
プリコンパイラ構成ファイル .....	6-2
プリコンパイラ実行可能ファイルの再リンク .....	6-2
プリコンパイラの README ファイル .....	6-3
すべてのプリコンパイラに共通の問題 .....	6-3
静的および動的リンク .....	6-3
クライアント共有ライブラリとクライアント静的ライブラリ .....	6-4
クライアント・アプリケーションのビット長サポート .....	6-5
Pro*C/C++ プリコンパイラ .....	6-7
Pro*C/C++ のデモ・プログラム .....	6-7
Pro*C/C++ のユーザー・プログラム .....	6-8
Pro*COBOL プリコンパイラ .....	6-9
Pro*COBOL の環境変数 .....	6-10
Acucorp ACUCOBOL-GT COBOL コンパイラ .....	6-10
Micro Focus Server Express COBOL コンパイラ .....	6-11
Pro*COBOL の Oracle ランタイム・システム .....	6-12
Pro*COBOL のデモ・プログラム .....	6-12
Pro*COBOL のユーザー・プログラム .....	6-13
FORMAT プリコンパイラ・オプション .....	6-14
Pro*FORTRAN プリコンパイラ .....	6-14
Pro*FORTRAN のデモ・プログラム .....	6-14
Pro*FORTRAN のユーザー・プログラム .....	6-15
SQL*Module for ADA .....	6-16
SQL*Module for Ada のデモ・プログラム .....	6-16
SQL*Module for Ada のユーザー・プログラム .....	6-17
OCI と OCCI .....	6-17
OCI と OCCI のデモ・プログラム .....	6-17
OCI と OCCI のユーザー・プログラム .....	6-18
64 ビット・ドライバを使用する Oracle JDBC/OCI プログラム .....	6-19
カスタム Make ファイル .....	6-20
未定義シンボルの修正 .....	6-20
マルチスレッド・アプリケーション .....	6-21
シグナル・ハンドラの使用 .....	6-21
XA 機能 .....	6-23

## 7 SQL\*Loader および PL/SQL のデモ

SQL*Loader のデモ .....	7-2
PL/SQL のデモ .....	7-2
PL/SQL からの 32 ビット外部プロシージャのコール .....	7-5

## 8 Oracle Database のチューニング

チューニングの重要性 .....	8-2
オペレーティング・システムのツール .....	8-2
vmstat .....	8-2
sar .....	8-3
iostat .....	8-4
swap、swapinfo、swapon、lspas .....	8-4
AIX のツール .....	8-5
Base Operation System ツール .....	8-5
Performance Toolbox .....	8-5
System Management Interface Tool .....	8-6
HP-UX のツール .....	8-6
Linux のツール .....	8-7
Solaris のツール .....	8-7
Mac OS X のツール .....	8-7
メモリー管理のチューニング .....	8-8
十分なスワップ領域の割当て .....	8-8
ページングの制御 .....	8-9
Oracle ブロック・サイズの調整 .....	8-10
ディスク I/O のチューニング .....	8-10
自動ストレージ管理の使用 .....	8-10
適切なファイル・システム・タイプを選択 .....	8-10
ディスク・パフォーマンスの監視 .....	8-11
システム・グローバル領域 .....	8-12
SGA サイズの確認 .....	8-13
AIX での共有メモリー .....	8-13
オペレーティング・システムのバッファ・キャッシュのチューニング .....	8-14

## A AIX システムでの Oracle Database の管理

メモリーとページング .....	A-2
バッファ・キャッシュのページング・アクティビティの制御 .....	A-2
AIX ファイル・バッファ・キャッシュのチューニング .....	A-3
十分なページング領域の割当て .....	A-4
ページングの制御 .....	A-4
データベース・ブロック・サイズの設定 .....	A-5
ログ・アーカイブ・バッファのチューニング .....	A-5
I/O バッファおよび SQL*Loader .....	A-5
ディスク I/O の問題 .....	A-6
AIX 論理ボリューム・マネージャ .....	A-6
ジャーナル・ファイル・システムを使用した場合と RAW 論理ボリュームを使用した場合の相違 .....	A-7
非同期 I/O の使用 .....	A-10
I/O スレーブ .....	A-11

DB_FILE_MULTIBLOCK_READ_COUNT パラメータの使用 .....	A-12
後書きの使用 .....	A-12
順次先読みのチューニング .....	A-12
ディスク I/O の歩調合せのチューニング .....	A-13
Oracle Database でのミラー復元 .....	A-13
RAW デバイスのバックアップ .....	A-14
<b>CPU のスケジューリングおよびプロセスの優先順位 .....</b>	<b>A-14</b>
プロセスのランタイム・スライスの変更 .....	A-14
SMP システムでのプロセッサ・バインディングの使用 .....	A-14
環境変数 AIXTHREAD_SCOPE の設定 .....	A-15
ネットワーク情報サービス (NIS) の外部ネーミングのサポート .....	A-15
AIX 5.3 システムでのマルチスレッド同時処理 (SMT) .....	A-15

## B HP-UX システムでの Oracle Database の管理

Oracle インスタンス用の HP-UX 共有メモリー・セグメント .....	B-2
HP-UX SCHED_NOAGE スケジューリング・ポリシー .....	B-2
Oracle Database での SCHED_NOAGE の有効化 .....	B-3
軽量タイマーの実装 .....	B-3
非同期 I/O .....	B-4
MLOCK 権限 .....	B-4
非同期 I/O の実装 .....	B-4
非同期 I/O の検証 .....	B-6
HP-UX 非同期ドライバが Oracle Database に対して設定されているかどうかの検証 .....	B-6
Oracle Database が非同期 I/O を使用しているかどうかの検証 .....	B-6
SGA の非同期フラグ .....	B-7
大規模メモリーの割当てと Oracle Database のチューニング .....	B-7
永続的な専用 SQL 領域とメモリー .....	B-7
デフォルトの大規模仮想メモリー・ページ・サイズ .....	B-8
チューニングに関する推奨事項 .....	B-9
CPU_COUNT 初期化パラメータおよび HP-UX 動的プロセッサ再構成 .....	B-9
ネットワーク情報サービス (NIS) の外部ネーミングのサポート .....	B-10

## C Linux システムでの Oracle Database の管理

拡張バッファ・キャッシュのサポート .....	C-2
SUSE Linux Enterprise Server 9 または Red Hat Enterprise Linux 4 での hugetlbfs の使用 .....	C-3
Red Hat Enterprise Linux AS 3 での hugetlbfs の使用 .....	C-4
SGA アドレス空間の増加 .....	C-5
非同期 I/O サポート .....	C-6
ダイレクト I/O サポート .....	C-6
semtimedop のサポート .....	C-7
高速ネットワークのサポート .....	C-7
マルチスレッド同時処理 (SMT) .....	C-7

## D Mac OS X システムでの Oracle Database の管理

使用可能および使用済のスワップ領域の決定 .....	D-2
----------------------------	-----

<b>E</b>	<b>Solaris システムでの Oracle Database の管理</b>	
	緊密共有メモリー .....	E-2
<b>F</b>	<b>Tru64 UNIX システムでの Oracle Database の管理</b>	
	<b>Oracle Database 指定配置最適化の有効化</b> .....	F-2
	指定配置最適化を実行するための要件 .....	F-2
	Oracle 指定配置最適化の有効化 .....	F-3
	Oracle 指定配置最適化の無効化 .....	F-3
	Oracle 指定配置最適化の使用 .....	F-3
	Oracle 初期化パラメータ .....	F-3
	Tru64 UNIX サブシステム属性 .....	F-4
	RAD に対するプロセスの親和性 .....	F-5
	システムに存在する RAD 数のサブセットへの Oracle Database の実行制限 .....	F-5
	<b>複合 CPU システムのサポート</b> .....	F-6
	<b>Tru64 UNIX でのデータベース統計の収集</b> .....	F-6
	<b>非同期 I/O のチューニング</b> .....	F-7
	aio_task_max_num 属性 .....	F-7
	<b>ダイレクト I/O サポートおよびコンカレント・ダイレクト I/O サポート</b> .....	F-7
	単一インスタンスの要件 .....	F-7
	クラスタ・ファイル・システム .....	F-8
	Tru64 UNIX V5.1B クラスタ・ファイル・システム .....	F-8
	ダイレクト I/O サポートの無効化 .....	F-8
	<b>リアルタイム・クロックへのアクセスの有効化</b> .....	F-9
	<b>RAW デバイスの設定</b> .....	F-10
	<b>Spike 最適化ツール</b> .....	F-12
	Spike の使用 .....	F-12
<b>G</b>	<b>Oracle ODBC Driver の使用</b>	
	サポートされていない機能 .....	G-2
	データ型の実装 .....	G-2
	データ型に関する制限事項 .....	G-3
	SQLDriverConnect 関数の接続文字列の書式 .....	G-3
	プログラムでのロック・タイムアウトの削減 .....	G-5
	ODBC アプリケーションのリンク .....	G-5
	ROWID に関する情報の取得 .....	G-5
	WHERE 句の ROWID .....	G-5
	結果セットの有効化 .....	G-5
	EXEC 構文の有効化 .....	G-11
	サポートされている機能 .....	G-12
	API への準拠 .....	G-12
	ODBC API 関数の実装 .....	G-12
	ODBC SQL 構文の実装 .....	G-13
	データ型の実装 .....	G-13
	<b>Unicode のサポート</b> .....	G-13
	ODBC 環境内での Unicode のサポート .....	G-13
	ODBC API での Unicode のサポート .....	G-14

SQLGetData のパフォーマンス .....	G-14
Unicode のサンプル .....	G-15
パフォーマンスとチューニング .....	G-20
ODBC プログラミングの一般的なガイドライン .....	G-20
データソース構成オプション .....	G-21
DATE および TIMESTAMP データ型 .....	G-22
エラー・メッセージ .....	G-23

## H データベースの制限

データベースの制限 .....	H-2
-----------------	-----

## 索引



---

---

# はじめに

このマニュアルでは、次の各プラットフォームで Oracle Database 10g リリース 2 (10.2) を管理および構成する方法について説明します。

- AIX 5L Based Systems (64-bit)
- Apple Mac OS X (Intel)
- HP Tru64 UNIX
- HP-UX PA-RISC (64-bit)
- HP-UX Itanium
- IBM zSeries Based Linux
- Linux Itanium
- Linux on POWER
- Linux x86
- Linux x86-64
- Solaris Operating System (SPARC 64-bit)
- Solaris Operating System (x86)
- Solaris Operating System (x86-64)

## 対象読者

このマニュアルは、Oracle Database 10g リリース 2 (10.2) の管理および構成を担当する方を対象としています。Oracle Real Application Clusters (RAC) を構成している場合は、『Oracle Database Oracle Clusterware および Oracle Real Application Clusters 管理およびデプロイメント・ガイド』を参照してください。

## ドキュメントのアクセシビリティについて

オラクル社は、障害のあるお客様にもオラクル社の製品、サービスおよびサポート・ドキュメントを簡単にご利用いただけることを目標としています。オラクル社のドキュメントには、ユーザーが障害支援技術を使用して情報を利用できる機能が組み込まれています。HTML 形式のドキュメントで用意されており、障害のあるお客様が簡単にアクセスできるようにマークアップされています。標準規格は改善されつつあります。オラクル社はドキュメントをすべてのお客様がご利用できるように、市場をリードする他の技術ベンダーと積極的に連携して技術的な問題に対応しています。オラクル社のアクセシビリティについての詳細情報は、Oracle Accessibility Program の Web サイト <http://www.oracle.com/accessibility/> を参照してください。

### ドキュメント内のサンプル・コードのアクセシビリティについて

スクリーン・リーダーは、ドキュメント内のサンプル・コードを正確に読めない場合があります。コード表記規則では閉じ括弧だけを行に記述する必要があります。しかし JAWS は括弧だけの行を読まない場合があります。

### 外部 Web サイトのドキュメントのアクセシビリティについて

このドキュメントにはオラクル社およびその関連会社が所有または管理しない Web サイトへのリンクが含まれている場合があります。オラクル社およびその関連会社は、それらの Web サイトのアクセシビリティに関しての評価や言及は行っておりません。

### Oracle サポート・サービスへの TTY リレー・アクセス

AT&T の Customer Assistant を呼び出すには、711 または 1.800.855.2880 にお電話ください。AT&T の Customer Assistant が、お客様と Oracle サポート・サービス (1.800.223.1711) の間で情報を中継します。AT&T のリレー・サービスを利用するための詳細なインストールは、<http://www.consumer.att.com/relay/tty/standard2.html> で参照できます。AT&T の Customer Assistant から Oracle サポート・サービスへの連絡の後、Oracle サポート・サービスの技術者が、Oracle サービス・リクエストのプロセスに従って、技術的な問題を処理し、お客様へのサポートを提供します。

## 関連ドキュメント

各プラットフォームに固有の Oracle Database 10g ドキュメントのリストは、該当する項を参照してください。

- [AIX 5L Based Systems \(64-bit\) のドキュメント](#)
- [HP-UX PA-RISC \(64-bit\) のドキュメント](#)
- [HP-UX Itanium のドキュメント](#)
- [Linux on POWER のドキュメント](#)
- [Linux x86 のドキュメント](#)
- [Linux x86-64 のドキュメント](#)
- [Solaris Operating System \(SPARC 64-bit\) のドキュメント](#)
- [Solaris Operating System \(x86\) のドキュメント](#)
- [Solaris Operating System \(x86-64\) のドキュメント](#)
- [Apple Mac OS X \(Intel\) のドキュメント](#)

### AIX 5L Based Systems (64-bit) のドキュメント

- Oracle Database
  - 『Oracle Database リリース・ノート for AIX 5L Based Systems (64-bit)』
  - 『Oracle Database インストール・ガイド for AIX 5L Based Systems (64-bit)』
  - 『Oracle Database クイック・インストール・ガイド for AIX 5L Based Systems (64-bit)』
  - 『Oracle Database Oracle Clusterware および Oracle Real Application Clusters インストール・ガイド for AIX Based Systems』
  - 『Oracle Database 管理者リファレンス for UNIX Systems』
- Oracle Database Client
  - 『Oracle Database Client インストール・ガイド for AIX 5L Based Systems (64-bit)』
  - 『Oracle Database Client クイック・インストール・ガイド for AIX 5L Based Systems (64-bit)』
- Oracle Database 10g Companion CD
  - 『Oracle Database Companion CD インストール・ガイド for AIX 5L Based Systems (64-bit)』
  - 『Oracle Database Companion CD クイック・インストール・ガイド for AIX 5L Based Systems (64-bit)』

### HP-UX PA-RISC (64-bit) のドキュメント

- Oracle Database
  - 『Oracle Database リリース・ノート for HP-UX PA-RISC (64-bit)』
  - 『Oracle Database インストール・ガイド for HP-UX PA-RISC (64-bit)』
  - 『Oracle Database クイック・インストール・ガイド for HP-UX PA-RISC (64-bit)』
  - 『Oracle Database Oracle Clusterware および Oracle Real Application Clusters インストール・ガイド』
  - 『Oracle Database 管理者リファレンス for UNIX Systems』

- Oracle Database Client
  - 『Oracle Database Client インストール・ガイド for HP-UX PA-RISC (64-bit)』
  - 『Oracle Database Client クイック・インストール・ガイド for HP-UX PA-RISC (64-bit)』
- Oracle Database 10g Companion CD
  - 『Oracle Database Companion CD インストール・ガイド for HP-UX PA-RISC (64-bit)』
  - 『Oracle Database Companion CD クイック・インストール・ガイド for HP-UX PA-RISC (64-bit)』

#### HP-UX Itanium のドキュメント

- Oracle Database
  - 『Oracle Database リリース・ノート for hp-ux Itanium』
  - 『Oracle Database インストール・ガイド for hp-ux Itanium』
  - 『Oracle Database クイック・インストール・ガイド for hp-ux Itanium』
  - 『Oracle Database Oracle Clusterware および Oracle Real Application Clusters インストール・ガイド』
  - 『Oracle Database 管理者リファレンス for UNIX Systems』
- Oracle Database Client
  - 『Oracle Database Client インストール・ガイド for hp-ux Itanium』
  - 『Oracle Database Client クイック・インストール・ガイド for hp-ux Itanium』
- Oracle Database 10g Companion CD
  - 『Oracle Database Companion CD インストール・ガイド for hp-ux Itanium』
  - 『Oracle Database Companion CD クイック・インストール・ガイド for hp-ux Itanium』

#### Linux on POWER のドキュメント

- Oracle Database
  - 『Oracle Database リリース・ノート for Linux on POWER』
  - 『Oracle Database インストール・ガイド for Linux on POWER』
  - 『Oracle Database クイック・インストール・ガイド for Linux on POWER』
  - 『Oracle Database Oracle Clusterware および Oracle Real Application Clusters インストール・ガイド for Linux』
  - 『Oracle Database 管理者リファレンス for UNIX Systems』
- Oracle Database Client
  - 『Oracle Database Client インストール・ガイド for Linux on POWER』
  - 『Oracle Database Client クイック・インストール・ガイド for Linux on POWER』
- Oracle Database Companion CD
  - 『Oracle Database Companion CD インストール・ガイド for Linux on POWER』
  - 『Oracle Database Companion CD クイック・インストール・ガイド for Linux on POWER』

## Linux x86 のドキュメント

- Oracle Database
  - 『Oracle Database リリース・ノート for Linux x86』
  - 『Oracle Database インストール・ガイド for Linux x86』
  - 『Oracle Database クイック・インストール・ガイド for Linux x86』
  - 『Oracle Database Oracle Clusterware および Oracle Real Application Clusters インストール・ガイド for Linux』
  - 『Oracle Database 管理者リファレンス for UNIX Systems』
- Oracle Database Client
  - 『Oracle Database Client インストール・ガイド for Linux x86』
  - 『Oracle Database Client クイック・インストール・ガイド for Linux x86』
- Oracle Database 10g Companion CD
  - 『Oracle Database Companion CD インストール・ガイド for Linux x86』
  - 『Oracle Database Companion CD クイック・インストール・ガイド for Linux x86』

## Linux x86-64 のドキュメント

- Oracle Database
  - 『Oracle Database リリース・ノート for Linux x86-64』
  - 『Oracle Database インストール・ガイド for Linux x86-64』
  - 『Oracle Database クイック・インストール・ガイド for Linux x86-64』
  - 『Oracle Database Oracle Clusterware および Oracle Real Application Clusters インストール・ガイド for Linux』
  - 『Oracle Database 管理者リファレンス for UNIX Systems』
- Oracle Database Client
  - 『Oracle Database Client インストール・ガイド for Linux x86-64』
  - 『Oracle Database Client クイック・インストール・ガイド for Linux x86-64』
- Oracle Database 10g Companion CD
  - 『Oracle Database Companion CD インストール・ガイド for Linux x86-64』
  - 『Oracle Database Companion CD クイック・インストール・ガイド for Linux x86-64』

## Solaris Operating System (SPARC 64-bit) のドキュメント

- Oracle Database
  - 『Oracle Database リリース・ノート for Solaris Operating System (SPARC 64-bit)』
  - 『Oracle Database インストール・ガイド for Solaris Operating System (SPARC 64-bit)』
  - 『Oracle Database クイック・インストール・ガイド for Solaris Operating System (SPARC 64-bit)』
  - 『Oracle Database Oracle Clusterware および Oracle Real Application Clusters インストール・ガイド for Solaris Operating System』
  - 『Oracle Database 管理者リファレンス for UNIX Systems』

- Oracle Database Client
  - 『Oracle Database Client インストール・ガイド for Solaris Operating System (SPARC 64-bit)』
  - 『Oracle Database Client クイック・インストール・ガイド for Solaris Operating System (SPARC 64-bit)』
- Oracle Database 10g Companion CD
  - 『Oracle Database Companion CD インストール・ガイド for Solaris Operating System (SPARC 64-bit)』
  - 『Oracle Database Companion CD クイック・インストール・ガイド for Solaris Operating System (SPARC 64-bit)』

### **Solaris Operating System (x86) のドキュメント**

- Oracle Database
  - 『Oracle Database リリース・ノート for Solaris Operating System (x86)』
  - 『Oracle Database インストール・ガイド for Solaris Operating System (x86)』
  - 『Oracle Database クイック・インストール・ガイド for Solaris Operating System (x86)』
  - 『Oracle Database Oracle Clusterware および Oracle Real Application Clusters インストール・ガイド for Solaris Operating System』
  - 『Oracle Database 管理者リファレンス for UNIX Systems』
- Oracle Database Client
  - 『Oracle Database Client インストール・ガイド for Solaris Operating System (x86)』
  - 『Oracle Database Client クイック・インストール・ガイド for Solaris Operating System (x86)』
- Oracle Database 10g Companion CD
  - 『Oracle Database Companion CD インストール・ガイド for Solaris Operating System (x86)』
  - 『Oracle Database Companion CD クイック・インストール・ガイド for Solaris Operating System (x86)』

### **Solaris Operating System (x86-64) のドキュメント**

- Oracle Database
  - 『Oracle Database リリース・ノート for Solaris Operating System (x86-64)』
  - 『Oracle Database インストール・ガイド for Solaris Operating System (x86-64)』
  - 『Oracle Database クイック・インストール・ガイド for Solaris Operating System (x86-64)』
  - 『Oracle Database Oracle Clusterware および Oracle Real Application Clusters インストール・ガイド for Solaris Operating System』
  - 『Oracle Database 管理者リファレンス for UNIX Systems』
- Oracle Database Client
  - 『Oracle Database Client インストール・ガイド for Solaris Operating System (x86-64)』
  - 『Oracle Database Client クイック・インストール・ガイド for Solaris Operating System (x86-64)』

- Oracle Database 10g Companion CD
  - 『Oracle Database Companion CD インストール・ガイド for Solaris Operating System (x86-64)』
  - 『Oracle Database Companion CD クイック・インストール・ガイド for Solaris Operating System (x86-64)』

#### Apple Mac OS X (Intel) のドキュメント

- Oracle Database
  - 『Oracle Database Release Notes for Apple Mac OS X (Intel)』
  - 『Oracle Database Installation Guide for Apple Mac OS X (Intel)』
  - 『Oracle Database 管理者リファレンス for UNIX Systems』
- Oracle Database Client
  - 『Oracle Database Client Installation Guide for Apple Mac OS X (Intel)』
- Oracle Database 10g Companion CD
  - 『Oracle Database Companion CD Installation Guide for Apple Mac OS X (Intel)』

このマニュアルのリリース時に記載されなかった重要な情報は、各プラットフォームのリリース・ノートを参照してください。Oracle Database のリリース・ノートは定期的に更新されます。最新バージョンは次の URL で OTN-J からダウンロードできます。

<http://www.oracle.com/technology/global/jp/documentation/index.html>

## 表記規則

このマニュアルでは、次の表記規則を使用しています。

規則	意味
太字	太字は、操作に関連する Graphical User Interface 要素、または本文中で定義されている用語および用語集に記載されている用語を示します。
イタリック体	イタリックは、ユーザーが特定の値を指定するプレースホルダ変数を示します。
固定幅フォント	固定幅フォントは、段落内のコマンド、URL、サンプル内のコード、画面に表示されるテキスト、または入力するテキストを示します。

## コマンド構文

UNIX のコマンド構文は、固定幅フォントで表示されます。ドル記号 (\$)、シャープ記号 (#) またはパーセント記号 (%) は、UNIX のコマンド・プロンプトです。これらの記号をコマンドの一部として入力しないでください。このマニュアルでは、コマンド構文に次の表記規則を使用しています。

規則	説明
バックスラッシュ \ \\	バックスラッシュは、UNIX コマンドの行の継続を表す記号です。コマンド例が 1 行に入りきらない場合に使用します。コマンドは、表示どおりにバックスラッシュを付けて入力するか、またはバックスラッシュなしで 1 行に入力します。  dd if=/dev/rdisk/c0t1d0s6 of=/dev/rst0 bs=10b \ count=10000
中カッコ {}	中カッコは、必須の入力項目を表します。  .DEFINE {macro1}
大カッコ []	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。  cvtcrt termname [outfile]
省略記号 ...	省略記号は、同じ項目を任意の数だけ繰り返すことを表します。  CHKVAL fieldname value1 value2 ... valueN
イタリック体	イタリック体は、変数を表します。変数には値を代入します。  <i>library_name</i>
縦線	縦線は、大カッコまたは中カッコ内の複数の選択項目の区切りに使用します。  FILE filesize [K M]

## 用語

このマニュアルでは UNIX オペレーティング・システムの名前を次のように短縮して使用しています。

オペレーティング・システム	短縮名
AIX 5L Based Systems (64-bit)	AIX
Apple Mac OS X (Intel)	Mac OS X
HP-UX PA-RISC (64-bit) HP-UX Itanium	HP-UX <b>注意:</b> 特定のアーキテクチャにおける HP-UX の情報の違いについては、本文中に記載されています。
HP Tru64 UNIX	Tru64 UNIX
Linux Itanium Linux on POWER IBM zSeries Based Linux Linux x86 Linux x86-64	Linux <b>注意:</b> 特定のアーキテクチャにおける Linux の情報の違いについては、本文中に記載されています。
Solaris Operating System (SPARC 64-bit) Solaris Operating System (x86) Solaris Operating System (x86-64)	Solaris <b>注意:</b> 特定のアーキテクチャにおける Solaris の情報の違いについては、本文中に記載されています。

## ドキュメントへのアクセス

Oracle Database 10g リリース 2 (10.2) のドキュメントには、プラットフォーム固有のドキュメントおよび汎用製品のドキュメントがあります。

### プラットフォーム固有のドキュメント

プラットフォーム固有のドキュメントには、Oracle 製品を特定のプラットフォームにインストールして使用する方法が記載されています。この製品のプラットフォーム固有のドキュメントは、製品ディスクに PDF 形式と HTML 形式の両方で収録されています。ディスクに収録されているプラットフォーム固有のドキュメントにアクセスする手順は、次のとおりです。

1. Web ブラウザを使用して、ディスクのトップレベル・ディレクトリにある `welcome.htm` ファイルを開きます。
2. DVD の場合にのみ、該当する製品のリンクを選択します。
3. 「**Documentation**」タブを選択します。

印刷したドキュメントが必要な場合は、PDF ファイルを開いて印刷してください。

### 製品のドキュメント

製品のドキュメントには、各プラットフォームで Oracle 製品を構成、使用または管理する方法が記載されています。Oracle Database 10g 製品の製品ドキュメントは、Oracle Database 10g リリース 2 (10.2) Online Documentation Library に HTML 形式と PDF 形式の両方で用意されています。このライブラリは、次の URL にある Oracle Technology Network Japan (OTN-J) の Web サイトにあります。

<http://www.oracle.com/technology/global/jp/documentation/index.html>

## サード・パーティ・ソフトウェア情報

このプログラムには、HP 社のサード・パーティ・ソフトウェアが含まれています。Oracle プログラム (HP 社のソフトウェアを含む) を使用する権利は、この製品に付随する Oracle プログラム・ライセンスによって許諾されます。これと異なる規定が Oracle プログラム・ライセンス内にある場合でも、HP 社のソフトウェアは現状のままであり、この規定によっていかなる種類の知的財産権保護、保証またはサポートもオラクル社または HP 社から提供されることはありません。

このプログラムには、IBM 社のサード・パーティ・ソフトウェアも含まれています。Oracle プログラム (IBM 社のソフトウェアを含む) を使用する権利は、この製品に付随する Oracle プログラム・ライセンスによって許諾されます。

これと異なる規定が Oracle プログラム・ライセンス内にある場合でも、IBM 社のソフトウェアは現状のままであり、この規定によっていかなる種類の知的財産権保護、保証またはサポートもオラクル社または IBM 社から提供されることはありません。

# サポートおよびサービス

次の各項に、各サービスに接続するための URL を記載します。

## Oracle サポート・サービス

オラクル製品サポートの購入方法、および Oracle サポート・サービスへの連絡方法の詳細は、次の URL を参照してください。

<http://www.oracle.com/lang/jp/support/index.html>

## 製品マニュアル

製品のマニュアルは、次の URL にあります。

<http://www.oracle.com/technology/global/jp/documentation/index.html>

## 研修およびトレーニング

研修に関する情報とスケジュールは、次の URL で入手できます。

[http://education.oracle.com/pls/web\\_prod-plq-dad/db\\_pages.getpage?page\\_id=3](http://education.oracle.com/pls/web_prod-plq-dad/db_pages.getpage?page_id=3)

## その他の情報

オラクル製品やサービスに関するその他の情報については、次の URL から参照してください。

<http://www.oracle.com/lang/jp/index.html>

<http://www.oracle.com/technology/global/jp/index.html>

---

---

**注意：** ドキュメント内に記載されている URL や参照ドキュメントには、Oracle Corporation が提供する英語の情報も含まれています。日本語版の情報については、前述の URL を参照してください。

---

---

---

---

# Oracle Database の管理

この章では、UNIX ベースのオペレーティング・システムで Oracle Database を管理する方法について説明します。次の項目について説明します。

- [概要](#)
- [環境変数](#)
- [初期化パラメータ](#)
- [オペレーティング・システムのアカウントとグループ](#)
- [RAW デバイスの使用](#)
- [トレース・ファイルおよびアラート・ファイルの使用](#)

**関連項目：** Oracle Database の管理に関するプラットフォーム固有の情報は、このマニュアルの付録を参照してください。

## 1.1 概要

Oracle Database を使用するには、Oracle Database の環境変数、パラメータおよびユーザー設定を設定する必要があります。この章では、Oracle Database の各種設定について説明します。

Oracle Database のファイルおよびプログラムでは、疑問符 (?) は環境変数 ORACLE\_HOME の値を表します。たとえば、Oracle Database では、次の SQL 文中の疑問符は、Oracle ホーム・ディレクトリのフルパス名に展開されます。

```
SQL> ALTER TABLESPACE TEMP ADD DATAFILE '?/dbs/temp02.dbf' SIZE 200M
```

同様に、アットマーク (@) 記号は環境変数 ORACLE\_SID を表します。たとえば、ファイルが現行のインスタンスに属していることを示す場合は、次のコマンドを実行します。

```
SQL> ALTER TABLESPACE tablespace_name ADD DATAFILE tempfile@.dbf
```

## 1.2 環境変数

この項では、通常使用される Oracle Database およびオペレーティング・システムの環境変数について説明します。Oracle Database をインストールする前に、これらの環境変数をいくつか定義する必要があります。

環境変数の現在の設定値を表示するには、env コマンドを使用します。たとえば、環境変数 ORACLE\_SID の値を表示するには、次のコマンドを実行します。

```
$ env | grep ORACLE_SID
```

すべての環境変数の現在の設定値を表示するには、env コマンドを次のように実行します。

```
$ env | more
```

### 1.2.1 Oracle Database の環境変数

表 1-1 に、Oracle Database で使用する環境変数を示します。

表 1-1 Oracle Database の環境変数

変数	項目	定義
NLS_LANG	機能	クライアント環境の言語、地域およびキャラクタ・セットを指定します。NLS_LANG で指定するキャラクタ・セットは、端末または端末エミュレータのキャラクタ・セットと一致している必要があります。NLS_LANG で指定されたキャラクタ・セットがデータベースのキャラクタ・セットと異なる場合、そのキャラクタ・セットは自動的に変換されます。この変数の値リストの詳細は、『Oracle Database グローバリゼーション・サポート・ガイド』を参照してください。
	構文	<code>language_territory.characterset</code>
	例	<code>french_france.we8dec</code>
ORA_NLS10	機能	言語、地域、キャラクタ・セットおよび言語の定義ファイルが保存されているディレクトリを指定します。
	構文	<code>directory_path</code>
	例	<code>\$ORACLE_HOME/nls/data</code>

表 1-1 Oracle Database の環境変数 (続き)

変数	項目	定義
ORA_TZFILE	機能	タイム・ゾーン・ファイルのフルパスおよびファイル名を指定します。この環境変数は、データベース内のデータに対して小規模タイム・ゾーン・ファイル (\$ORACLE_HOME/oracore/zoneinfo/timezone.dat) を使用する場合に設定する必要があります。Oracle Database 10g では、デフォルトで大規模タイム・ゾーン・ファイル (\$ORACLE_HOME/oracore/zoneinfo/timezlg.dat) が使用されます。このファイルには、小規模タイム・ゾーン・ファイルよりも多数のタイム・ゾーンに関する情報が記載されています。  情報を共有するデータベースはすべて、同一のタイム・ゾーン・ファイルを使用する必要があります。この環境変数の値を変更した場合は、データベースを停止し、再起動する必要があります。
	構文	<i>directory_path</i>
	例	\$ORACLE_HOME/oracore/zoneinfo/timezlg.dat
ORACLE_BASE	機能	Optimal Flexible Architecture (OFA) に準拠したインストールのディレクトリ構造の基本となるディレクトリを指定します。
	構文	<i>directory_path</i>
	例	/u01/app/oracle
ORACLE_HOME	機能	Oracle ソフトウェアが格納されているディレクトリを指定します。
	構文	<i>directory_path</i>
	例	\$ORACLE_BASE/product/10.2.0/db_1
ORACLE_PATH	機能	SQL*Plus などの Oracle アプリケーションが使用するファイルの検索パスを指定します。ファイルのフルパス名が指定されていない場合やファイルが現行のディレクトリにない場合、Oracle アプリケーションでは、ORACLE_PATH を使用してそのファイルの場所を特定します。
	構文	ディレクトリをコロンで区切ったリスト:  <i>directory1:directory2:directory3</i>
	例	/u01/app/oracle/product/10.2.0/db_1/bin:.  <b>注意:</b> 最後にピリオドを付けることによって、検索パスに現行のディレクトリが追加されます。
ORACLE_SID	機能	Oracle システムの識別子を指定します。
	構文	英字で始まり、数字と英字で構成される文字列を指定します。システム識別子は、8 文字以内で指定することをお勧めします。この環境変数の詳細は、『Oracle Database インストレーション・ガイド』を参照してください。
	例	SAL1
ORACLE_TRACE	機能	インストール時のシェル・スクリプトのトレースを有効にします。この環境変数を T に設定した場合は、ほとんどの Oracle シェル・スクリプトで set -x コマンドが使用されます。これによって、シェル・スクリプトの実行時にコマンドとそれらの引数が印刷されます。他の値を設定した場合、または値を設定しない場合、そのスクリプトでは、set -x コマンドが使用されません。
	構文	T または T 以外
	例	T
ORAENV_ASK	機能	oraenv または coraenv スクリプトで、環境変数 ORACLE_SID の値を入力するためのプロンプトを表示するかどうかを制御します。NO に設定した場合、環境変数 ORACLE_SID の値を入力するためのプロンプトは表示されません。他の値に設定した場合、または値を設定しない場合、スクリプトによって環境変数 ORACLE_SID の値を入力するためのプロンプトが表示されます。
	構文	NO または NO 以外
	例	NO

表 1-1 Oracle Database の環境変数 (続き)

変数	項目	定義
SQLPATH	機能	SQL*Plus での login.sql ファイルの検索先ディレクトリまたはディレクトリのリストを指定します。
	構文	ディレクトリをコロンで区切ったリスト: <i>directory1:directory2:directory3</i>
	例	<i>/home:/home/oracle:/u01/oracle</i>
TNS_ADMIN	機能	Oracle Net Services 構成ファイルが格納されているディレクトリを指定します。
	構文	<i>directory_path</i>
	例	<i>\$ORACLE_HOME/network/admin</i>
TWO_TASK	機能	接続文字列に使用するデフォルトの接続識別子を指定します。この環境変数が設定されている場合は、接続文字列に接続識別子を指定しないでください。たとえば、環境変数 TWO_TASK が sales に設定されている場合は、CONNECT <i>username/password@sales</i> コマンドではなく、CONNECT <i>username/password</i> コマンドを使用してデータベースに接続できます。
	構文	任意の接続識別子
	許容値	ネーミング・メソッドを使用して解決できる有効な接続識別子 (tnsnames.ora ファイルやディレクトリ・サーバーなど)
	例	<i>PRODDB_TCP</i>

**注意：** 競合を防ぐため、Oracle Database サーバーのプロセス名と同じ名前前で環境変数を定義しないでください。Oracle Database サーバーのプロセス名には、ARCH、PMON、DBWR などがあります。

## 1.2.2 UNIX 環境変数

表 1-2 に、Oracle Database で使用する UNIX 環境変数を示します。

表 1-2 Oracle Database で使用する環境変数

変数	項目	定義
ADA_PATH (AIX のみ)	機能	Ada compiler.sm が格納されているディレクトリを指定します。
	構文	<i>directory_path</i>
	例	<i>/usr/lpp/powerada</i>
CLASSPATH	機能	Java アプリケーションで使用します。この環境変数に必要な設定は、Java アプリケーションによって異なります。詳細は、使用している Java アプリケーション製品のドキュメントを参照してください。
	構文	ディレクトリまたはファイルをコロンで区切ったリスト: <i>directory1:directory2:file1:file2</i>
	例	デフォルトの設定はありません。CLASSPATH には、次のディレクトリが含まれている必要があります。 <i>\$ORACLE_HOME/JRE/lib:\$ORACLE_HOME/jlib</i>

表 1-2 Oracle Database で使用する環境変数 (続き)

変数	項目	定義
DISPLAY	機能	X ベースのツールで使います。入出力に使用するディスプレイ・デバイスを指定します。詳細は、使用している X Window System のドキュメントを参照してください。
	構文	<code>hostname:server[.screen]</code> <code>hostname</code> はシステム名 (IP アドレスまたは別名)、 <code>server</code> はサーバーの順次コード番号、 <code>screen</code> は画面の順次コード番号です。使用するモニターが 1 つの場合は、サーバーと画面のどちらにも値 0 を使います (0.0)。 <b>注意:</b> 使用するモニターが 1 つの場合、 <code>screen</code> はオプションです。
	例	135.287.222.12:0.0 bambi:0
DYLD_LIBRARY_PATH (Mac OS X のみ)	機能	共有ライブラリ・ローダーの実行時に共有オブジェクト・ライブラリの検索先となるディレクトリのリストを指定します。この環境変数の詳細は、 <code>dyld</code> の <code>man</code> ページを参照してください。
	構文	ディレクトリをコロンで区切ったリスト: <code>directory1:directory2:directory3</code>
	例	<code>/usr/lib:\$ORACLE_HOME/lib</code>
HOME	機能	ユーザーのホーム・ディレクトリを指定します。
	構文	<code>directory_path</code>
	例	<code>/home/oracle</code>
LANG または LANGUAGE	機能	メッセージなどの出力でオペレーティング・システムが使用する言語およびキャラクタ・セットを指定します。詳細は、オペレーティング・システムのドキュメントを参照してください。 <b>注意:</b> この環境変数は、Mac OS X では使用されません。
	機能	デフォルトのリンカー・オプションを指定します。この環境変数の詳細は、 <code>ld</code> の <code>man</code> ページを参照してください。
LPDEST (Solaris のみ)	機能	デフォルトのプリンタの名前を指定します。
	構文	<code>string</code>
	例	<code>docprinter</code>
LD_LIBRARY_PATH (AIX および Mac OS X 以外のすべてのプラットフォーム)	機能	共有ライブラリ・ローダーの実行時に共有オブジェクト・ライブラリの検索先となるディレクトリのリストを指定します。この環境変数の詳細は、 <code>ld</code> の <code>man</code> ページを参照してください。 HP-UX の場合は、64 ビット共有ライブラリのパスを指定します。
	構文	ディレクトリをコロンで区切ったリスト: <code>directory1:directory2:directory3</code>
	例	<code>/usr/dt/lib:\$ORACLE_HOME/lib</code>
LD_LIBRARY_PATH_64 (SPARC システムのみ)	機能	共有ライブラリ・ローダーの実行時に特定の 64 ビットの共有オブジェクト・ライブラリを検索するディレクトリのリストを指定します。この環境変数の詳細は、 <code>ld</code> の <code>man</code> ページを参照してください。
	構文	コロンで区切ったディレクトリのリスト: <code>directory1:directory2:directory3</code>
	例	<code>/usr/dt/lib:\$ORACLE_HOME/lib64</code>
LIBPATH (AIX のみ)	機能	共有ライブラリ・ローダーの実行時に共有オブジェクト・ライブラリの検索先となるディレクトリのリストを指定します。この環境変数の詳細は、 <code>ld</code> の <code>man</code> ページを参照してください。
	構文	ディレクトリをコロンで区切ったリスト: <code>directory1:directory2:directory3</code>
	例	<code>/usr/dt/lib:\$ORACLE_HOME/lib</code>

表 1-2 Oracle Database で使用する環境変数 (続き)

変数	項目	定義
PATH	機能	シェルで、実行可能プログラムの場所を特定するために使用されます。 \$ORACLE_HOME/bin ディレクトリが含まれている必要があります。
	構文	ディレクトリをコロンで区切ったリスト: <code>directory1:directory2:directory3</code>
	例	<code>/bin:/usr/bin:/usr/local/bin:/usr/bin/X11:\$ORACLE_HOME/bin:\$HOME/bin:.</code> <b>注意:</b> 最後にピリオドを付けることによって、検索パスに現行のディレクトリが追加されます。
PRINTER	機能	デフォルトのプリンタの名前を指定します。
	構文	<code>string</code>
	例	<code>docprinter</code>
SHLIB_PATH (HP-UX の 32 ビットのライブラリのみ)	機能	共有ライブラリ・ローダーの実行時に共有オブジェクト・ライブラリの検索先となるディレクトリのリストを指定します。この環境変数の詳細は、ld の man ページを参照してください。
	構文	ディレクトリをコロンで区切ったリスト: <code>directory1:directory2:directory3</code>
	例	<code>/usr/dt/lib:\$ORACLE_HOME/lib32</code>
TEMP、TMP および TMPDIR	機能	一時ファイル用のデフォルト・ディレクトリを指定します。設定すると、一時ファイルを作成するツールは、指定したデフォルト・ディレクトリの 1 つに一時ファイルを作成します。
	構文	<code>directory_path</code>
	例	<code>/u02/oracle/tmp</code>
XENVIRONMENT	機能	X Window System のリソース定義を含むファイルを指定します。詳細は、使用している X Window System のドキュメントを参照してください。

## 1.2.3 共通の環境設定

この項では、デフォルト・シェルに応じて oraenv または coraenv スクリプトを使用し、共通のオペレーティング・システム環境を設定する方法について説明します。

- Bourne、Bash または Korn シェルの場合は、oraenv コマンドを使用します。
- C シェルの場合は、coraenv コマンドを使用します。

### oraenv および coraenv スクリプト・ファイル

oraenv および coraenv スクリプトは、インストール時に作成されます。この 2 つのスクリプトは、oratab ファイルの内容に基づいて環境変数を設定し、次の機能を提供します。

- データベースの変更をすべてのユーザー・アカウントに反映して更新するための主な方法
- oratab ファイルに指定されているデータベース間で切替えを行うためのメカニズム

開発システムからデータベースに対して頻繁に追加や削除を行ったり、同一システム上にインストールされた複数の異なる Oracle Database 間でユーザーが切替えを行う場合があります。oraenv または coraenv スクリプトを使用すると、ユーザー・アカウントが更新されていることを確認し、データベース間で切替えを行うことができます。

---

**注意:** oraenv または coraenv スクリプトは、Oracle ソフトウェア所有者 (通常は oracle) ユーザーのシェル起動スクリプトからはコールしないでください。これらのスクリプトでは値の入力を促すプロンプトが表示されるため、システムの起動時に dbstart スクリプトが自動的にデータベースを起動できなくなります。

---

oraenv または coraenv スクリプトは通常、ユーザーのシェル起動ファイル（.profile または .login など）からコールされます。このスクリプトは、環境変数 ORACLE\_SID および ORACLE\_HOME を設定し、\$ORACLE\_HOME/bin ディレクトリを環境変数 PATH の設定に含めます。データベース間で切替えを行う場合に、oraenv または coraenv スクリプトを実行して、これらの環境変数を設定できます。

---

**注意：** これらのスクリプトのいずれかを実行するには、適切なコマンドを使用します。

- coraenv スクリプトの場合
 

```
% source /usr/local/bin/coraenv
```
  - oraenv スクリプトの場合
 

```
$ . /usr/local/bin/oraenv
```
- 

### ローカル bin ディレクトリ

oraenv、coraenv および dbhome スクリプトを含むディレクトリは、ローカル bin ディレクトリと呼ばれます。すべてのデータベース・ユーザーは、このディレクトリへの読取りアクセス権が必要です。ローカル bin ディレクトリのパスをユーザーの環境変数 PATH の設定に追加してください。インストール後に root.sh スクリプトを実行すると、ローカル bin ディレクトリのパスを要求するプロンプトが表示されます。指定したディレクトリに、oraenv、coraenv および dbhome スクリプトが自動的にコピーされます。デフォルトのローカル bin ディレクトリは、/usr/local/bin です。root.sh スクリプトを実行しない場合は、手動で oraenv または coraenv スクリプトと dbhome スクリプトを、\$ORACLE\_HOME/bin ディレクトリからローカル bin ディレクトリにコピーできます。

## 1.2.4 システム・タイム・ゾーンの設定

環境変数 TZ は、タイム・ゾーンを設定します。これによって、時間を夏時間に変更したり、別のタイム・ゾーンにすることができます。調整した時刻は、タイムスタンプ・ファイルに使用したり、date コマンドの出力を生成したり、SYSDATE の現在値を取得するために使用します。

個人用の TZ 値は変更しないことをお勧めします。GMT+24 などの異なる TZ 値を使用すると、トランザクションが記録される日付が変更される場合があります。日付が変更されると、SYSDATE を使用する Oracle アプリケーションが影響を受けます。この問題を回避するために、表の順序付けには、日付列ではなく順序番号を使用してください。

## 1.3 初期化パラメータ

次の各項では、Oracle Database 初期化パラメータについて説明します。

- [DB\\_BLOCK\\_SIZE](#) 初期化パラメータ
- [ASM\\_DISKSTRING](#) 初期化パラメータ
- [LOG\\_ARCHIVE\\_DEST\\_n](#) 初期化パラメータ

### 1.3.1 DB\_BLOCK\_SIZE 初期化パラメータ

DB\_BLOCK\_SIZE 初期化パラメータは、データベースの標準ブロック・サイズを指定します。このブロック・サイズは SYSTEM 表領域に使用され、その他の表領域ではデフォルトで使用されます。

DB\_BLOCK\_SIZE に設定できる最大値は、Linux および Solaris の場合は 16KB です。AIX、HP-UX、Tru64 UNIX および Mac OS X の場合は、32KB です。

---

**注意：** DB\_BLOCK\_SIZE 初期化パラメータの値は、データベースの作成後に変更することはできません。

---

## 1.3.2 ASM\_DISKSTRING 初期化パラメータ

---

**注意：** ASM\_DISKSTRING 初期化パラメータをサポートしているのは、自動ストレージ管理インスタンスのみです。

---

ASM\_DISKSTRING 初期化パラメータに値を割り当てるための構文は、次のとおりです。

```
ASM_DISKSTRING = 'path1'[, 'path2', . . .]
```

この構文の *pathn* は、RAW デバイスへのパスです。パスを指定する際は、ワイルドカード文字を使用できます。

表 1-3 に、ASM\_DISKSTRING 初期化パラメータに対するプラットフォーム固有のデフォルト値を示します。

**表 1-3 ASM\_DISKSTRING 初期化パラメータのデフォルト値**

プラットフォーム	デフォルト検索文字列
AIX	/dev/rhdisk*
HP-UX	/dev/rdsk/*
Linux	/dev/raw/*
Solaris	/dev/rdsk/*
Tru64 UNIX	/dev/rdisk/*

**関連項目：** RAW デバイスのパスの指定時に使用できるワイルドカード文字パターンに関するプラットフォーム固有の情報は、`glob(7)` の `man` ページを参照してください。

## 1.3.3 LOG\_ARCHIVE\_DEST\_n 初期化パラメータ

LOG\_ARCHIVE\_DEST\_n 初期化パラメータの ASYNC に設定できる最大値は、次の表に示すように、UNIX のプラットフォームによって異なります。

プラットフォーム	最大値
zSeries Linux	12800
HP-UX および Tru64 UNIX	51200
その他のオペレーティング・システム	102400

## 1.4 オペレーティング・システムのアカウントとグループ

この項では、次の Oracle Database に必要な特殊なオペレーティング・システムのアカウントとグループについて説明します。

- Oracle ソフトウェア所有者アカウント
- OSDBA、OSOPER および Oracle インベントリ・グループ
- グループとセキュリティ
- 外部認証
- orapwd ユーティリティの実行
- パスワード管理
- オペレーティング・システム・アカウントの追加
- Oracle ユーザーのアカウントの構成

### 1.4.1 Oracle ソフトウェア所有者アカウント

Oracle ソフトウェア所有者アカウントは、通常 `oracle` という名前で、Oracle ソフトウェアのインストールに使用します。ソフトウェアを個別に Oracle ホーム・ディレクトリにインストールするたびに、異なる Oracle ソフトウェア所有者アカウントを使用できます。ただし、インストール後の Oracle ホーム・ディレクトリの保守作業では、Oracle ホーム・ディレクトリごとに、ソフトウェアをインストールしたときと同じアカウントを使用する必要があります。

Oracle ソフトウェア所有者は、Oracle インベントリ・グループをプライマリ・グループとして、OSDBA グループをセカンダリ・グループとして所有することをお勧めします。

### 1.4.2 OSDBA、OSOPER および Oracle インベントリ・グループ

表 1-4 に、Oracle Database に必要な、特殊なオペレーティング・システム・グループを示します。

表 1-4 オペレーティング・システム・グループ

グループ	代表的な名前	説明
OSDBA	<code>dba</code>	OSDBA グループのメンバーであるオペレーティング・システム・アカウントには、特殊なデータベース権限があります。このグループのメンバーは、SYSDBA 権限を使用してデータベースに接続できます。Oracle ソフトウェア所有者は、このグループの必須メンバーです。必要に応じて、他のアカウントを追加できます。
OSOPER	<code>oper</code>	OSOPER グループは、オプション・グループです。OSOPER グループのメンバーであるオペレーティング・システム・アカウントには、特殊なデータベース権限があります。このグループのメンバーは、SYSOPER 権限を使用してデータベースに接続できます。
Oracle インベントリ	<code>oinstall</code>	Oracle ソフトウェアをインストールするユーザーはすべて、同じオペレーティング・システム・グループに属している必要があります。このグループは Oracle インベントリ・グループと呼ばれます。インストール時には、このグループが、Oracle ソフトウェア所有者のプライマリ・グループである必要があります。インストール後は、システムにインストールされたすべての Oracle ファイルを、このグループが所有します。

**関連項目：** OSDBA グループと SYSDBA 権限、および OSOPER グループと SYSOPER 権限の詳細は、『Oracle Database 管理者ガイド』および『Oracle Database インストール・ガイド』を参照してください。

Oracle Database では、UNIX オペレーティング・システムの機能をいくつか使用して、ユーザーに安全性の高い環境を提供します。その機能には、ファイル所有権、グループ・アカウント、処理時にそのユーザー ID を変更するプログラム機能が含まれます。

Oracle Database の 2 タスク構造によって、ユーザー・プログラムと oracle プログラム間で作業（およびアドレス領域）を分割し、セキュリティを高めることができます。すべてのデータベース・アクセスは、このシャドウ・プロセスおよび oracle プログラムへの特殊権限によって行うことができます。

**関連項目：** セキュリティ問題の詳細は、『Oracle Database 管理者ガイド』を参照してください。

### 1.4.3 グループとセキュリティ

Oracle プログラムは、セキュリティの観点から 2 つのセットに分けられます。つまり、すべてのユーザー（UNIX 用語では other）が実行できるプログラムと、DBA のみが実行できるプログラムに分けられます。セキュリティを保護するために次の方法で分類することをお勧めします。

- oracle アカウントのプライマリ・グループは、oinstall である必要があります。
- oracle アカウントは、dba グループをセカンダリ・グループとする必要があります。
- SYSDBA 権限が必要なユーザー・アカウントは dba グループに属することができますが、oinstall グループに属することができるユーザー・アカウントは、Oracle ソフトウェア所有者アカウントのみです。たとえば、oracle ユーザーなどです。

### 1.4.4 外部認証

外部認証を使用する場合は、OS\_AUTHENT\_PREFIX 初期化パラメータの値を Oracle ユーザー名の接頭辞として使用する必要があります。このパラメータを明示的に設定しない場合は、UNIX のデフォルト値 ops\$（大 / 小文字区別あり）が使用されます。

オペレーティング・システムと Oracle 認証の両方に同一のユーザー名を使用するには、次のように、この初期化パラメータを NULL 文字列に設定します。

```
OS_AUTHENT_PREFIX=""
```

**関連項目：** 外部認証の詳細は、『Oracle Database 管理者ガイド』を参照してください。

### 1.4.5 orapwd ユーティリティの実行

パスワード・ファイルを使用して、データベースへの接続時に SYSDBA および SYSOPER 権限を使用できるユーザーを識別できます。Oracle Database Configuration Assistant を使用してデータベースを作成すると、新規データベースに対してパスワード・ファイルが作成されます。データベースを手動で作成する場合は、次のようにデータベースのパスワード・ファイルを作成してください。

1. Oracle ソフトウェア所有者でログインします。
2. orapwd ユーティリティを使用して、次のようにパスワード・ファイルを作成します。

```
$ $ORACLE_HOME/bin/orapwd file=filename password=password entries=max_users
```

次の表に、このコマンドで指定する必要がある値を示します。

値	説明
filename	パスワード情報が書き込まれているファイルの名前です。 ファイル名は orapwsid で、フルパス名を指定する必要があります。その内容は暗号化されています。パスワード・ファイルは通常、\$ORACLE_HOME/dbs ディレクトリに作成されます。

値	説明
<code>password</code>	SYS ユーザーのパスワードです。  データベースに接続した後で、ALTER USER 文を使用して SYS ユーザーのパスワードを変更した場合は、データ・ディクショナリに格納されているパスワードとパスワード・ファイルに格納されているパスワードの両方が更新されます。このパラメータの入力は必須です。
<code>max_users</code>	パスワード・ファイルで許可されるエントリの最大数を設定します。これは、SYSDBA 権限または SYSOPER 権限により同時にデータベースに接続できる別々のユーザーの最大数です。

**関連項目：** orapwd ユーティリティの使用方法は、『Oracle Database 管理者ガイド』を参照してください。

## 1.4.6 パスワード管理

Oracle Database Configuration Assistant を使用してデータベースを作成する場合、ユーザーは SYS および SYSTEM アカウントのパスワードを変更する必要があります。デフォルトのパスワード CHANGE\_ON\_INSTALL および MANAGER は使用できません。

Oracle Database Configuration Assistant では、セキュリティ上の理由により、データベースの作成後にほとんどの Oracle ユーザー・アカウントがロックされます。ただし、SYS または SYSTEM の各アカウントはロックされません。ロックされたアカウントを使用するには、ロックを解除してパスワードを変更する必要があります。そのためには、次の方法のいずれかを使用できます。

- Oracle Database Configuration Assistant を使用してパスワードを変更するには、データベース・コンフィギュレーション・アシスタントの「サマリー」ウィンドウで「パスワード管理」をクリックします。
- あるいは、SQL\*Plus を使用して SYS としてデータベースに接続し、次のコマンドを実行してアカウントのロックを解除し、パスワードを再設定します。

```
SQL> ALTER USER username IDENTIFIED BY passwd ACCOUNT UNLOCK;
```

## 1.4.7 オペレーティング・システム・アカウントの追加

オペレーティング・システム・アカウントは、必要に応じて作成します。管理者権限を使用してデータベースに接続するには、ユーザーが OSDBA または OSOPER グループのメンバーである必要があります。

## 1.4.8 Oracle ユーザーのアカウントの構成

oracle ユーザーおよび Oracle ユーザーのオペレーティング・システム・アカウントの起動ファイルを更新し、環境ファイルに適切な環境変数を指定します。

Bourne、Bash または Korn シェルの場合は、環境変数を `.profile` ファイルに追加します。Red Hat Enterprise Linux および Mac OS X の Bash シェルの場合は、`.bash_profile` ファイルに追加します。

C シェルの場合は、環境変数を `.login` ファイルに追加します。

---

**注意：** oraenv または coraenv スクリプトを使用すると、Oracle ユーザーのアカウントが更新されたことを確認できます。

---

## 1.5 RAW デバイスの使用

次の各項では、RAW デバイス（RAW パーティションまたは RAW ボリューム）の使用方法について説明します。

- [RAW デバイスを使用する場合のガイドライン](#)
- [RAW デバイスの設定](#)
- [AIX および Tru64 UNIX システムでの RAW デバイスのデータファイル](#)
- [Linux システムでの RAW デバイスのサポート](#)

**関連項目：** RAW デバイスのチューニングの詳細は、このマニュアルにある各プラットフォームに対応した付録を参照してください。

### 1.5.1 RAW デバイスを使用する場合のガイドライン

RAW デバイス（RAW パーティションまたは RAW ボリューム）には、次のようなデメリットがあります。

- RAW デバイスでは、ファイル・サイズの見込み制限に関する問題を解決できない場合があります。

現在のファイル・サイズ制限を表示するには、次のコマンドを実行します。

- Bourne、Bash または Korn シェルの場合

```
$ ulimit -a
```

- C シェルの場合

```
% limit
```

- 特定のディスク・ドライブに I/O アクティビティが集中していて、Oracle データファイルを別のドライブに移動すればパフォーマンスが向上するような場合、I/O アクティビティの少ないドライブには、受入れに十分なサイズのパーティションまたはボリュームがないことがあります。このため、RAW デバイスを使用している場合は、他のディスク・ドライブにファイルを移動できない場合があります。
- RAW デバイスは、ファイル・システムまたは自動ストレージ管理ディスク・グループに格納されているデータファイルに比べて、管理が難しくなります。

RAW デバイスの使用を決定する際は、次のことに考慮する必要があります。

- RAW ディスク・パーティションの可用性

少なくとも Oracle データファイルと同数の RAW ディスク・パーティションがある場合のみ、Oracle ファイルに RAW パーティションを使用してください。ディスク領域を見直す場合に、RAW ディスク・パーティションがすでに作成済のときは、データファイル・サイズとパーティション・サイズをできるだけ一致させ、無駄な領域をなくすようにします。

また、多数のディスク上の領域を一部のみ使用した場合と少数のディスク上の領域をすべて使用した場合の、パフォーマンスへの影響も考慮する必要があります。

- 論理ボリューム・マネージャ

論理ボリューム・マネージャは、論理レベルでディスク領域を管理するため、RAW デバイスの複雑さを意識する必要がありません。論理ボリュームを使用することによって、RAW パーティションの可用性に基づいて論理ディスクを作成できます。論理ボリューム・マネージャは、次の方法で固定ディスク・リソースを管理します。

- 論理記憶域と物理記憶域との間でデータをマッピングします。
- データを複数のディスクにまたがって不連続に格納、レプリケートおよび動的に拡張できるようにします。

RAC の場合、1 つのシステムに関連付けられているドライブにも、クラスタ内の複数のシステムで共有できるドライブにも、論理ボリュームを使用できます。共有ドライブを使用すると、RAC データベースに関連付けられたすべてのファイルを、これらの共有論理ボリュームに配置できます。

- 動的パフォーマンスのチューニング

ディスク・パフォーマンスを最適化するには、アクティビティの高いディスク・ドライブからアクティビティの低いディスク・ドライブにファイルを移動します。論理ディスク機能を提供しているほとんどのハードウェア・ベンダーからも、チューニングに使用できるグラフィカル・ユーザー・インタフェース (GUI) が提供されています。

- ミラー化およびオンラインでのディスクの交換

データ損害からデータを保護するために、論理ボリュームをミラー化できます。ミラー化された一方のデータに障害が起きた場合は、動的に再同期化できます。ベンダーによっては、ミラー化機能を使用してオンラインでドライブを交換する機能を提供している場合があります。

## 1.5.2 RAW デバイスの設定

RAW デバイスを作成する際は、次の内容を確認してください。

- 所有者が Oracle ソフトウェア所有者ユーザー (oracle) であり、グループが OSDBA グループ (dba) であることを確認します。
- RAW パーティションに作成する Oracle データファイルのサイズは、少なくとも RAW パーティションのサイズよりも Oracle ブロック・サイズ 2 つ分以上、小さいことを確認します。

**関連項目：** RAW デバイスの作成方法の詳細は、オペレーティング・システムのドキュメントを参照してください。

## 1.5.3 AIX および Tru64 UNIX システムでの RAW デバイスのデータファイル

AIX および Tru64 UNIX システムの場合、RAW 論理ボリューム上のデータファイルには、最初の Oracle データ・ブロックにオフセットが存在する場合があります。このオフセットは、論理ボリューム・マネージャに必要です。

オフセット値の指定には、`$ORACLE_HOME/bin/offset` ユーティリティを使用できます。オフセット値の指定は、データファイルを異なるデバイスに転送する場合などに必要です。

**関連項目：** オフセット 0 (ゼロ) を使用できる AIX システムでの RAW 論理ボリュームの作成方法は、[付録 A](#) を参照してください。AIX システムの RAW 論理ボリュームには、オフセット 0 (ゼロ) の使用をお勧めします。

## 1.5.4 Linux システムでの RAW デバイスのサポート

RAW キャラクタ・デバイスとブロック・デバイスの両方を、データベースを作成する際の RAW ボリュームとして使用できます。ブロック・デバイスがサポートされているため、RAW デバイスの最大数に関するカーネル・レベルの制限はなく、必要に応じて追加の RAW ボリュームを構成できます。

## 1.6 トレース・ファイルおよびアラート・ファイルの使用

この項では、操作上の問題を簡単に診断して解決できるように、Oracle Database が作成するトレース・ファイル（ダンプ・ファイル）およびアラート・ファイルについて説明します。この項の内容は、次のとおりです。

- [トレース・ファイル](#)
- [アラート・ファイル](#)

### 1.6.1 トレース・ファイル

各サーバーとバックグラウンド・プロセスでは、トレース・ファイルに情報を書き込みます。プロセスが内部エラーを検出すると、そのエラーに関する情報がトレース・ファイルに書き込まれます。トレース・ファイルのファイル名の形式は `sid_processname_unixpid.trc` です。各項目の説明は次のとおりです。

- `sid` は、インスタンスのシステム識別子です。
- `processname` は、トレース・ファイルを作成した Oracle Database プロセスを識別するための、3～4文字の短縮されたプロセス名（`pmon`、`dbwr`、`ora`、`reco` など）です。
- `unixpid` は、オペレーティング・システム・プロセスの ID 番号です。

次に、トレース・ファイル名のサンプルを示します。

```
$ORACLE_BASE/admin/TEST/bdump/test_lgwr_1237.trc
```

バックグラウンド・プロセス用のトレース・ファイルはすべて、`BACKGROUND_DUMP_DEST` 初期化パラメータで指定した接続先ディレクトリに書き込まれます。この初期化パラメータを設定しない場合は、デフォルトのディレクトリ `$ORACLE_HOME/rdbms/log` が使用されます。

ユーザー・プロセス用のトレース・ファイルはすべて、`USER_DUMP_DEST` 初期化パラメータで指定した接続先ディレクトリに書き込まれます。この初期化パラメータを設定しない場合は、デフォルトのディレクトリ `$ORACLE_HOME/rdbms/log` が使用されます。`MAX_DUMP_FILE` 初期化パラメータを 5000 以上に設定し、トレース・ファイルがエラー情報を格納するのに十分な大きさになるようにします。

### 1.6.2 アラート・ファイル

`alert_sid.log` ファイルには、重要なデータベース・イベント情報やメッセージを格納します。データベース・インスタンスまたはデータベースに影響を与えるイベントは、このファイルに記録されます。このファイルは、データベースに関連付けられ、`BACKGROUND_DUMP_DEST` 初期化パラメータで指定したディレクトリに配置されます。この初期化パラメータを設定しない場合は、デフォルトのディレクトリ `$ORACLE_HOME/rdbms/log` が使用されます。

---

## Oracle ソフトウェアの停止と起動

この章では、Oracle Database プロセスを識別する方法と、プロセスを停止および再起動する基本的な方法について説明します。また、Oracle Database の起動と停止を自動化する方法についても説明します。次の項目について説明します。

- [Oracle プロセスの停止と起動](#)
- [停止と起動の自動化](#)

## 2.1 Oracle プロセスの停止と起動

この項では、Oracle プロセスを停止および起動する方法について説明します。次の項目について説明します。

- [Mac OS X での Oracle プロセスの起動](#)
- [Oracle Database インスタンスおよび自動ストレージ管理インスタンスの停止と起動](#)
- [Oracle CSS デーモンの停止と起動](#)
- [Oracle Net Listener の停止と起動](#)
- [iSQL\\*Plus の停止と起動](#)
- [Oracle Ultra Search の停止と起動](#)
- [Oracle Enterprise Manager Database Control の停止と起動](#)
- [Oracle Management Agent の停止と起動](#)

### 2.1.1 Mac OS X での Oracle プロセスの起動

---

---

**注意：** Oracle Database インスタンスまたは自動ストレージ管理インスタンス、あるいは Oracle Net Listener プロセスを起動する場合は、毎回必ずこの項の指示に従ってください。

---

---

特定のシェル制限が Oracle プロセスの実行に必要な値に設定されていることを確認するには、システムがローカル・システムである場合でも、ssh、rlogin または telnet コマンドを使用して、プロセスを起動するシステムに接続する必要があります。このコマンドの構文は、次のとおりです。

```
$ ssh localhost
```

### 2.1.2 Oracle Database インスタンスおよび自動ストレージ管理インスタンスの停止と起動

この項では、Oracle Database インスタンスおよび自動ストレージ管理インスタンスを停止および起動する方法について説明します。

#### Oracle Database インスタンスまたは自動ストレージ管理インスタンスの停止

---

---

**注意：** 自動ストレージ管理を使用して記憶域を管理している Oracle Database インスタンスをすべて停止するまで、自動ストレージ管理インスタンスは停止しないでください。

---

---

Oracle Database インスタンスまたは自動ストレージ管理インスタンスを停止するには、次の手順を実行します。

1. 停止するインスタンスの SID および Oracle ホーム・ディレクトリを識別する場合は、次のコマンドを実行します。

Solaris の場合

```
$ cat /var/opt/oracle/oratab
```

その他のオペレーティング・システムの場合

```
$ cat /etc/oratab
```

oratab ファイルには、次のような行が含まれています。これによって、システム上の各データベース・インスタンスまたは自動ストレージ管理インスタンスの SID および対応する Oracle ホーム・ディレクトリを識別します。

```
sid:oracle_home_directory:[Y|N]
```

---

**注意：** 自動ストレージ管理インスタンスの SID には、1 文字目にプラス記号 (+) を使用することをお勧めします。

---

2. デフォルト・シェルに応じて oraenv または coraenv スクリプトを実行し、停止するインスタンスの環境変数を設定します。

- Bourne、Bash または Korn シェルの場合

```
$ . /usr/local/bin/oraenv
```

- C シェルの場合

```
% source /usr/local/bin/coraenv
```

プロンプトが表示されたら、インスタンスの SID を指定します。

3. 次のコマンドを実行し、インスタンスを停止します。

```
$ sqlplus /nolog
SQL> CONNECT SYS/sys_password as SYSDBA
SQL> SHUTDOWN NORMAL
```

インスタンスの停止後、SQL\*Plus を終了できます。

## Oracle Database インスタンスまたは自動ストレージ管理インスタンスの再起動

---

**注意：** データベース・インスタンスで記憶域管理に自動ストレージ管理を使用している場合は、データベース・インスタンスを起動する前に、自動ストレージ管理インスタンスを起動する必要があります。

---

Oracle Database インスタンスまたは自動ストレージ管理インスタンスを再起動するには、次の手順を実行します。

1. 必要に応じて前述の手順 1 と 2 を繰り返し、環境変数 ORACLE\_SID および ORACLE\_HOME を設定して、起動するインスタンスの SID および Oracle ホーム・ディレクトリを識別します。
2. 次のコマンドを実行し、インスタンスを起動します。

```
$ sqlplus /nolog
SQL> CONNECT SYS/sys_password as SYSDBA
SQL> STARTUP
```

インスタンスの起動後、SQL\*Plus を終了できます。

## 2.1.3 Oracle CSS デーモンの停止と起動

---

---

**注意：** この項の内容は、Mac OS X には適用されません。

---

---

Oracle Cluster Services Synchronization (CSS) デーモンを停止するには、次のコマンドを実行します。

AIX の場合

```
/etc/init.cssd stop
```

その他のプラットフォームの場合

```
/etc/init.d/init.cssd stop
```

CSS デーモンを起動するには、次のコマンドを実行します。

```
$ORACLE_HOME/bin/localconfig reset
```

このコマンドは、Oracle CSS デーモンを停止して再起動します。

## 2.1.4 Oracle Net Listener の停止と起動

この項では、Oracle Net Listener を停止および起動する方法について説明します。

### Oracle Net Listener の停止

Oracle Net Listener を停止するには、次の手順を実行します。

1. 停止する Oracle Net Listener のリスナー名とホーム・ディレクトリを判別する場合は、次のコマンドを実行します。

Mac OS X の場合

```
$ ps -ax | grep tnslnsr
```

その他のプラットフォームの場合

```
$ ps -ef | grep tnslnsr
```

このコマンドは、システムで稼働中の Oracle Net Listener のリストを表示します。このコマンドの出力は、次のようになります。

```
94248 ?? I 0:00.18 oracle_home1/bin/tnslnsr listenername1 -inherit
94248 ?? I 0:00.18 oracle_home2/bin/tnslnsr listenername2 -inherit
```

このサンプルでは、出力 *listenername1* と *listenername2* がリスナーの名前です。

2. 必要に応じて環境変数 ORACLE\_HOME を設定し、停止するリスナーの Oracle ホーム・ディレクトリを指定します。

- Bourne、Bash または Korn シェルの場合

```
$ ORACLE_HOME=oracle_home1
$ export ORACLE_HOME
```

- C シェルの場合

```
% setenv ORACLE_HOME oracle_home1
```

3. 次のコマンドを実行し、Oracle Net Listener を停止します。

```
$ $ORACLE_HOME/bin/lsnrctl stop listenername
```

---



---

**注意：** リスナー名がデフォルトの LISTENER の場合、このコマンドで名前を指定する必要はありません。

---



---

## Oracle Net Listener の再起動

Oracle Net Listener を起動するには、次の手順を実行します。

1. 必要に応じて環境変数 ORACLE\_HOME を設定し、起動するリスナーの Oracle ホーム・ディレクトリを指定します。

- Bourne、Bash または Korn シェルの場合

```
$ ORACLE_HOME=oracle_home1
$ export ORACLE_HOME
```

- C シェルの場合

```
% setenv ORACLE_HOME oracle_home1
```

2. 次のコマンドを実行し、Oracle Net Listener を再起動します。

```
$ $ORACLE_HOME/bin/lsnrctl start [listenername]
```

リスナー名の指定は、リスナー名がデフォルトのリスナー名 LISTENER と異なる場合にのみ必要となります。リスナー名は、listener.ora ファイルに記述されています。このファイルの内容を表示するには、次のコマンドを実行します。

```
$ more $ORACLE_HOME/network/admin/listener.ora
```

## 2.1.5 iSQL\*Plus の停止と起動

この項では、iSQL\*Plus を停止および起動する方法について説明します。

### iSQL\*Plus の停止

iSQL\*Plus を停止するには、次の手順を実行します。

1. 必要に応じて環境変数 ORACLE\_HOME を設定し、iSQL\*Plus の Oracle ホーム・ディレクトリを指定します。

- Bourne、Bash または Korn シェルの場合

```
$ ORACLE_HOME=oracle_home
$ export ORACLE_HOME
```

- C シェルの場合

```
% setenv ORACLE_HOME oracle_home
```

2. 次のコマンドを実行し、iSQL\*Plus を停止します。

```
$ $ORACLE_HOME/bin/isqlplusctl stop
```

### iSQL\*Plus の起動

iSQL\*Plus を起動するには、次の手順を実行します。

1. 必要に応じて環境変数 ORACLE\_HOME を設定し、起動する iSQL\*Plus インスタンスの Oracle ホーム・ディレクトリを指定します。

- Bourne、Bash または Korn シェルの場合

```
$ ORACLE_HOME=oracle_home
$ export ORACLE_HOME
```

- C シェルの場合

```
% setenv ORACLE_HOME oracle_home
```
2. 次のコマンドを実行し、iSQL\*Plus を起動します。

```
$ $ORACLE_HOME/bin/isqlplusctl start
```

## 2.1.6 Oracle Ultra Search の停止と起動

この項では、Oracle Ultra Search を停止および起動する方法について説明します。

### Oracle Ultra Search の停止

Oracle Ultra Search を停止するには、次の手順を実行します。

1. 必要に応じて環境変数 ORACLE\_HOME を設定し、Oracle Ultra Search の Oracle ホーム・ディレクトリを指定します。
  - Bourne、Bash または Korn シェルの場合

```
$ ORACLE_HOME=oracle_home  
$ export ORACLE_HOME
```
  - C シェルの場合

```
% setenv ORACLE_HOME oracle_home
```
2. 次のコマンドを実行し、Oracle Ultra Search を停止します。

```
$ $ORACLE_HOME/bin/searchctl stop
```

### Oracle Ultra Search の起動

Oracle Ultra Search を起動するには、次の手順を実行します。

1. 必要に応じて環境変数 ORACLE\_HOME を設定し、Oracle Ultra Search の Oracle ホーム・ディレクトリを指定します。
  - Bourne、Bash または Korn シェルの場合

```
$ ORACLE_HOME=oracle_home  
$ export ORACLE_HOME
```
  - C シェルの場合

```
% setenv ORACLE_HOME oracle_home
```
2. 次のコマンドを実行し、Oracle Ultra Search を起動します。

```
$ $ORACLE_HOME/bin/searchctl start
```

## 2.1.7 Oracle Enterprise Manager Database Control の停止と起動

この項では、Oracle Enterprise Manager Database Control を停止および起動する方法について説明します。

---

---

**注意：** この項の内容は、Mac OS X には適用されません。

---

---

### Oracle Enterprise Manager Database Control の停止

Oracle Enterprise Manager Database Control を停止するには、次の手順を実行します。

1. デフォルト・シェルに応じて oraenv または coraenv スクリプトを実行し、停止する Database Control で管理されているデータベースの環境を設定します。

- coraenv スクリプトの場合  

```
% source /usr/local/bin/coraenv
```
- oraenv スクリプトの場合  

```
$ . /usr/local/bin/oraenv
```

2. 次のコマンドを実行し、Database Control を停止します。

```
$ $ORACLE_HOME/bin/emctl stop dbconsole
```

### Oracle Enterprise Manager Database Control の起動

Database Control を起動するには、次の手順を実行します。

1. 環境変数 ORACLE\_SID および ORACLE\_HOME を設定し、起動する Database Control の SID および Oracle ホーム・ディレクトリを識別します。

- Bourne、Bash または Korn シェルの場合  

```
$ ORACLE_HOME=oracle_home  
$ ORACLE_SID=sid  
$ export ORACLE_HOME ORACLE_SID
```
- C シェルの場合  

```
% setenv ORACLE_HOME oracle_home  
% setenv ORACLE_SID sid
```

2. 次のコマンドを実行し、Database Control を起動します。

```
$ $ORACLE_HOME/bin/emctl start dbconsole
```

## 2.1.8 Oracle Management Agent の停止と起動

Oracle Enterprise Manager Grid Control を使用して複数の Oracle 製品を中央から管理する場合は、Oracle Management Agent が各ホスト・システムにインストールされている必要があります。通常、Oracle Management Agent は固有の Oracle ホーム・ディレクトリにインストールされます。

この項では、Oracle Management Agent を停止および起動する方法について説明します。

### Oracle Management Agent の停止

Oracle Management Agent を停止するには、次の手順を実行します。

1. 次のコマンドを実行し、Oracle Management Agent の Oracle ホーム・ディレクトリを判別します。

```
$ ps -ef | grep emagent
```

このコマンドは、Oracle Management Agent プロセスに関する情報を表示します。このコマンドの出力は、次のようになります。

```
94248 ?? I 0:00.18 oracle_home/agent/bin/emagent ...
```

2. 必要に応じて環境変数 ORACLE\_HOME を設定し、Oracle Management Agent の Oracle ホーム・ディレクトリを指定します。

- Bourne、Bash または Korn シェルの場合

```
$ ORACLE_HOME=oracle_home  
$ export ORACLE_HOME
```

- C シェルの場合

```
% setenv ORACLE_HOME oracle_home
```

3. 次のコマンドを実行し、Oracle Management Agent を停止します。

```
$ $ORACLE_HOME/agent/bin/emctl stop agent
```

### Oracle Management Agent の起動

Oracle Management Agent を起動するには、次の手順を実行します。

1. 必要に応じて環境変数 ORACLE\_HOME を設定し、Oracle Management Agent の Oracle ホーム・ディレクトリを指定します。

- Bourne、Bash または Korn シェルの場合

```
$ ORACLE_HOME=oracle_home  
$ export ORACLE_HOME
```

- C シェルの場合

```
% setenv ORACLE_HOME oracle_home
```

2. 次のコマンドを実行し、Oracle Management Agent を起動します。

```
$ $ORACLE_HOME/agent/bin/emctl start agent
```

## 2.2 停止と起動の自動化

Oracle Database は、システムの起動時に自動的に起動し、停止時に自動的に停止するようにシステムを構成することをお勧めします。データベースの起動と停止を自動化することによって、データベースの不正な停止を防ぐことができます。

データベースの起動と停止を自動化するには、`$ORACLE_HOME/bin` ディレクトリにある `dbstart` および `dbshut` スクリプトを使用します。これらのスクリプトは、`oratab` ファイル内の同じエントリを参照します。したがって、同じデータベース・セットに適用されます。たとえば、`dbstart` スクリプトによって、`sid1`、`sid2` および `sid3` を自動的に起動し、`dbshut` スクリプトによって、`sid1` のみを停止することはできません。ただし、`dbstart` スクリプトがまったく使用されていない場合は、`dbshut` スクリプトを使用してデータベース・セットの停止を指定することはできます。これを実行するには、システムの停止ファイルに `dbshut` エントリを追加し、システムの起動ファイルから `dbstart` エントリを除外します。

**関連項目：** システムの起動と停止の手順については、使用しているオペレーティング・システムのドキュメントにある `init` コマンドを参照してください。

- [Mac OS X におけるデータベースの停止と起動の自動化](#)
- [その他のオペレーティング・システムにおけるデータベースの起動と停止の自動化](#)

### 2.2.1 Mac OS X におけるデータベースの停止と起動の自動化

`dbstart` および `dbshut` スクリプトを使用してデータベースの起動と停止を自動化するには、次の手順を実行します。

1. `root` ユーザーでログインします。
2. テキスト・エディタで `oratab` ファイルを開きます。

```
# vi /etc/oratab
```

`oratab` ファイル内のデータベース・エントリは、次の形式で表示されます。

```
SID:ORACLE_HOME:{Y|N}
```

この例の `Y` または `N` は、スクリプトでデータベースの起動または停止を実行するかどうかを指定します。最初に、停止と起動を自動化するデータベースごとに、データベースのインスタンス識別子 (SID) を検索します。これは、最初のフィールドの `SID` で識別されます。次に、最後のフィールドをそれぞれ `Y` に変更します。

---

**注意：** 新規データベース・インスタンスをシステムに追加する場合、これらのインスタンスを自動的に起動するには、必ず `oratab` ファイルでインスタンスのエントリを編集してください。

---

3. 次のコマンドを実行して `/Library/StartupItems/Oracle` ディレクトリを作成し、ディレクトリをこれに変更します。

```
# mkdir -p /Library/StartupItems/Oracle
# cd /Library/StartupItems/Oracle
```

4. テキスト・エディタを使用し、このディレクトリに次のような内容の `Oracle` という起動スクリプトを作成します。

```
#!/bin/bash
```

```
# source the common startup script
```

```
./etc/rc.common
```

```
# Change the value of ORACLE_HOME to specify the correct Oracle home
```

```

# directory for the installation

ORACLE_HOME=/Volumes/u01/app/oracle/product/10.2.0/db_1
#
# change the value of ORACLE to the login name of the
# oracle owner at your site

ORACLE=oracle

PATH=${PATH}:$ORACLE_HOME/bin
export ORACLE_HOME PATH

# Set shell limits for the Oracle Database

ulimit -c unlimited
ulimit -d unlimited
ulimit -s 65536

StartService()
{
    if [ -f $ORACLE_HOME/bin/tnslsnr ] ; then
        ConsoleMessage "Starting Oracle Net"
        su $ORACLE -c "$ORACLE_HOME/bin/lsnrctl start"
    fi
    ConsoleMessage "Starting Oracle Databases"
    su $ORACLE -c "$ORACLE_HOME/bin/dbstart $ORACLE_HOME"
}

StopService()
{
    ConsoleMessage "Stopping Oracle Databases"
    su $ORACLE -c "$ORACLE_HOME/bin/dbshut $ORACLE_HOME"
    if [ -f $ORACLE_HOME/bin/tnslsnr ] ; then
        ConsoleMessage "Stopping Oracle Net"
        su $ORACLE -c "$ORACLE_HOME/bin/lsnrctl stop"
    fi
}

RestartService()
{
    StopService
    StartService
}

RunService "$1"

```

---

**注意：** このスクリプトで停止できるのは、パスワードが設定されていない Oracle Net Listener のみです。また、リスナー名がデフォルトの LISTENER ではない場合、stop および start コマンドでリスナー名を指定する必要があります。

---

```
$ORACLE_HOME/bin/lsnrctl {start|stop} listener_name
```

---

5. テキスト・エディタを使用し、このディレクトリに次のような内容の `StartupParameters.plist` という起動項目パラメータ・リスト・ファイルを作成します。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Description</key>
  <string>Oracle Database Startup</string>
  <key>Provides</key>
  <array>
    <string>Oracle Database</string>
  </array>
  <key>Requires</key>
  <array>
    <string>Disks</string>
  </array>
  <key>Uses</key>
  <array>
    <string>Disks</string>
    <string>Network</string>
    <string>NFS</string>
  </array>
  <key>OrderPreference</key>
  <string>Late</string>
</dict>
</plist>
```

---

**注意：** AL32UTF8 は、XMLType データに適した Oracle Database キャラクタ・セットです。このキャラクタ・セットは、IANA に登録された標準 UTF-8 エンコーディングと同等で、有効な XML 文字をすべてサポートしています。

Oracle Database のデータベース・キャラクタ・セット UTF8 (ハイフンなし) を、データベース・キャラクタ・セット AL32UTF8 または文字エンコーディング UTF-8 と混同しないでください。データベース・キャラクタ・セット UTF8 は、AL32UTF8 に置き換えられました。したがって、XML データには UTF8 を使用しないでください。UTF8 は Unicode バージョン 3.1 以前のみをサポートし、有効な XML 文字をすべてサポートしているわけではありません。AL32UTF8 にはこのような制限がありません。

XML データにデータベース・キャラクタ・セット UTF8 を使用すると、致命的エラーやセキュリティ低下の原因になる可能性があります。データベース・キャラクタ・セットでサポートされていない文字が入力ドキュメントの要素名に表示された場合は、置換文字 (通常は「?」) に置き換えられます。その結果、解析が終了して例外が発生します。

---

6. 作成したファイルの所有者、グループおよび権限を次のように変更します。

```
# chown root:wheel *
# chmod 700 *
```

## 2.2.2 その他のオペレーティング・システムにおけるデータベースの起動と停止の自動化

dbstart および dbshut スクリプトを使用してデータベースの起動と停止を自動化するには、次の手順を実行します。

1. root ユーザーでログインします。
2. 使用しているプラットフォームの oratab ファイルを編集します。

ファイルを開くには、次のいずれかのコマンドを使用します。

- Solaris の場合

```
# vi /var/opt/oracle/oratab
```

- AIX、HP-UX、Linux および Tru64 UNIX の場合

```
# vi /etc/oratab
```

oratab ファイル内のデータベース・エントリーは、次の形式で表示されます。

```
SID:ORACLE_HOME:{Y|N|W}
```

この例の値 Y および N は、スクリプトでデータベースの起動または停止を実行するかどうかを指定します。最初に、停止と起動を自動化するデータベースごとに、データベースのインスタンス識別子 (SID) を判別します。これは、最初のフィールドの SID で識別されます。次に、最後のフィールドをそれぞれ Y に変更します。

dbstart を設定すると、自動ストレージ管理インストールを使用している単一インスタンスのデータベースの起動を自動化できます。自動ストレージ管理は Oracle Clusterware によって自動的に起動されます。これは、自動ストレージ管理クラスタのデフォルトの動作です。これを実行する場合は、データベースおよび自動ストレージ管理インストールの oratab エントリーを変更して、3 番目のフィールドに値 W および N をそれぞれ設定する必要があります。これらの値は、自動ストレージ管理インスタンスの起動後にのみ、dbstart がデータベースを自動的に起動するように指定します。

---

**注意：** 新規データベース・インスタンスをシステムに追加する場合、これらのインスタンスを自動的に起動するには、oratab ファイルでインスタンスのエントリーを編集する必要があります。

---

3. 使用しているオペレーティング・システムに応じて、ディレクトリを次のいずれかに変更します。

プラットフォーム	初期化ファイルのディレクトリ
AIX	/etc
Linux および Solaris	/etc/init.d
HP-UX および Tru64 UNIX	/sbin/init.d

4. dbora というファイルを作成し、次の行をこのファイルにコピーします。

---

**注意：** 環境変数 ORACLE\_HOME の値を、各インストールの Oracle ホーム・ディレクトリに変更します。また、環境変数 ORACLE の値を、Oracle ホーム・ディレクトリにインストールされているデータベースの所有者のユーザー名 (通常は oracle) に変更します。

---

```

#!/bin/sh -x
#
# Change the value of ORACLE_HOME to specify the correct Oracle home
# directory for your installation.

ORACLE_HOME=/u01/app/oracle/product/10.2.0/db_1
#
# Change the value of ORACLE to the login name of the
# oracle owner at your site.
#
ORACLE=oracle

PATH=${PATH}:$ORACLE_HOME/bin
HOST=`hostname`
PLATFORM=`uname`
export ORACLE_HOME PATH
#
if [ ! "$2" = "ORA_DB" ] ; then
  if [ "$PLATFORM" = "HP-UX" ] ; then
    remsh $HOST -l $ORACLE -n "$0 $1 ORA_DB"
    exit
  else
    rsh $HOST -l $ORACLE $0 $1 ORA_DB
    exit
  fi
fi
#
case $1 in
'start')
  if [ "$PLATFORM" = "Linux" ] ; then
    touch /var/lock/subsys/dbora
  fi
  $ORACLE_HOME/bin/dbstart $ORACLE_HOME &
  ;;
'stop')
  $ORACLE_HOME/bin/dbshut $ORACLE_HOME &
  ;;
*)
  echo "usage: $0 {start|stop}"
  exit
  ;;
esac
#
exit

```

---

**注意：** このスクリプトで停止できるのは、パスワードが設定されていない Oracle Net Listener のみです。また、リスナー名がデフォルトの LISTENER ではない場合、stop および start コマンドでリスナー名を指定する必要があります。

```
$ORACLE_HOME/bin/lsnrctl {start|stop} listener_name
```

---

5. dbora ファイルのグループを OSDBA グループ（通常は dba）に変更し、その権限を 750 に設定します。

```

# chgrp dba dbora
# chmod 750 dbora

```

6. 次のように、dbora スクリプトへのシンボリック・リンクを、適切な起動レベルのスクリプト・ディレクトリに作成します。

プラットフォーム	シンボリック・リンク・コマンド
AIX	# ln -s /etc/dbora /etc/rc.d/rc2.d/S99dbora # ln -s /etc/dbora /etc/rc.d/rc2.d/K01dbora
HP-UX	# ln -s /sbin/init.d/dbora /sbin/rc3.d/S990dbora # ln -s /sbin/init.d/dbora /sbin/rc3.d/K001dbora
Linux	# ln -s /etc/init.d/dbora /etc/rc.d/rc3.d/K01dbora # ln -s /etc/init.d/dbora /etc/rc.d/rc3.d/S99dbora # ln -s /etc/init.d/dbora /etc/rc.d/rc5.d/K01dbora # ln -s /etc/init.d/dbora /etc/rc.d/rc5.d/S99dbora
Solaris	# ln -s /etc/init.d/dbora /etc/rc3.d/K01dbora # ln -s /etc/init.d/dbora /etc/rc3.d/S99dbora
Tru64 UNIX	# ln -s /sbin/init.d/dbora /sbin/rc3.d/S99dbora # ln -s /sbin/init.d/dbora /sbin/rc3.d/K01dbora

---

## Oracle Database の構成

この章では、Oracle 製品に対する Oracle Database の構成方法について説明します。次の項目について説明します。

- [Oracle 製品の追加に対する Oracle Database の構成](#)
- [スタンドアロン・ツールとしてのコンフィギュレーション・アシスタントの使用](#)
- [実行可能ファイルの再リンク](#)

## 3.1 Oracle 製品の追加に対する Oracle Database の構成

初期インストールの後で Oracle 製品を追加インストールする場合は、次のように、Oracle Database Configuration Assistant を使用して新しい製品用にデータベースを構成します。

1. 必要に応じてデータベースを起動します。
2. 次のコマンドを実行し、Oracle Database Configuration Assistant を起動します。

```
$ $ORACLE_HOME/bin/dbca
```

3. 「データベース・オプションの構成」を選択します。
4. 使用可能なデータベースのリストから、構成するデータベースを選択します。
5. 有効にする製品を選択し、「終了」をクリックします。

## 3.2 スタンドアロン・ツールとしてのコンフィギュレーション・アシスタントの使用

コンフィギュレーション・アシスタントは通常、インストール・セッション中に実行されますが、スタンドアロン・モードでも実行できます。Oracle Universal Installer と同様にすべてのコンフィギュレーション・アシスタントは、応答ファイルを使用して非対話方式で起動できます。

この項では、次の項目について説明します。

- [Oracle Net コンフィギュレーション・アシスタントの使用](#)
- [Oracle Database アップグレード・アシスタントの使用](#)
- [Oracle Database Configuration Assistant の使用](#)
- [新規またはアップグレード済データベースの構成](#)

### 3.2.1 Oracle Net コンフィギュレーション・アシスタントの使用

Oracle Net Server または Oracle Net Client をインストールすると、Oracle Universal Installer によって Oracle Net コンフィギュレーション・アシスタントが自動的に起動されます。

Oracle Database Client のインストールを個別に実行した場合、Oracle Net コンフィギュレーション・アシスタントが、インストール時の選択内容に沿う構成を自動的に作成します。Oracle Universal Installer は Oracle Net コンフィギュレーション・アシスタントを自動的に実行し、クライアント・インストールの \$ORACLE\_HOME/network/admin ディレクトリにあるローカル・ネーミング・ファイル内にネット・サービス名を設定します。

インストールの完了後、Oracle Net コンフィギュレーション・アシスタントを使用して詳細な構成を作成できます。次のコマンドを入力します。

```
$ $ORACLE_HOME/bin/netca
```

---

**注意：** Oracle Database Configuration Assistant を使用してデータベースを作成すると、ネットワーク構成ファイルが自動的に更新され、新しいデータベースに関する情報が追加されます。

---

## 3.2.2 Oracle Database アップグレード・アシスタントの使用

Oracle Database のインストール時に、データベースを以前のリリースから現在のリリースにアップグレードできます。ただし、インストール時にデータベースをアップグレードしない場合、または複数のデータベースをアップグレードする場合は、インストール後に Oracle Database アップグレード・アシスタントを実行できます。

Oracle Database をインストールし、インストール時にデータベースのアップグレードを選択しなかった場合は、マウントする前にデータベースをアップグレードする必要があります。

Oracle Database アップグレード・アシスタントを起動するには、次のコマンドを実行します。

```
$ $ORACLE_HOME/bin/dbua
```

Oracle Database アップグレード・アシスタントで使用できるコマンドライン・オプションに関する情報を取得するには、次のように、`-help` または `-h` コマンドライン引数を使用します。

```
$ $ORACLE_HOME/bin/dbua -help
```

**関連項目：** アップグレードの詳細は、『Oracle Database インストール・ガイド』および『Oracle Database アップグレード・ガイド』を参照してください。

## 3.2.3 Oracle Database Configuration Assistant の使用

Oracle Database Configuration Assistant を使用すると、次のことができます。

- デフォルトの、またはカスタマイズされたデータベースの作成
- Oracle 製品を使用する既存のデータベースの構成
- 自動ストレージ管理のディスク・グループの作成
- 後のデータベース作成時に検査、修正および実行できるシェル・スクリプトと SQL スクリプトのセットの生成

Oracle Database Configuration Assistant を起動するには、次のコマンドを実行します。

```
$ $ORACLE_HOME/bin/dbca
```

Oracle Database Configuration Assistant で使用できるコマンドライン・オプションに関する情報を取得するには、次のように、`-help` または `-h` コマンドライン引数を使用します。

```
$ $ORACLE_HOME/bin/dbca -help
```

## 3.2.4 新規またはアップグレード済データベースの構成

データベースを作成またはアップグレードした後は、`utlrlp.sql` スクリプトを実行することをお勧めします。このスクリプトを実行すると、無効な状態にある、パッケージ、プロシージャ、タイプなどの PL/SQL モジュールがすべて再コンパイルされます。これはオプションの手順ですが、後日ではなく、データベースの作成時に実行することをお勧めします。

`utlrlp.sql` スクリプトを実行するには、次の手順を実行します。

1. ユーザーを `oracle` に切り替えます。
2. `oraenv` または `coraenv` スクリプトを使用して、`utlrlp.sql` スクリプトを実行するデータベースの環境を設定します。
  - Bourne、Bash または Korn シェルの場合
 

```
$ . /usr/local/bin/oraenv
```
  - C シェルの場合
 

```
% source /usr/local/bin/coraenv
```

プロンプトが表示されたら、データベースの SID を指定します。

3. 次のコマンドを実行し、SQL\*Plus を起動します。

```
$ sqlplus "/ AS SYSDBA"
```

4. 必要に応じて次のコマンドを実行し、データベースを起動します。

```
SQL> STARTUP
```

5. utlrlp.sql スクリプトを実行します。

```
SQL> @?/rdbms/admin/utlrlp.sql
```

### 3.3 実行可能ファイルの再リンク

\$ORACLE\_HOME/bin ディレクトリ内の relink シェル・スクリプトを使用して、製品の実行可能ファイルを手動で再リンクできます。製品の実行可能ファイルの再リンクは、オペレーティング・システムのパッチを適用するたびに、またはオペレーティング・システムのアップグレード後に必要となります。

---

**注意：** 実行可能ファイルの再リンク前に、Oracle ホーム・ディレクトリで実行されている実行可能ファイルのうち、再リンクする実行可能ファイルをすべて停止する必要があります。また、Oracle 共有ライブラリにリンクされているアプリケーションも停止してください。

---

relink スクリプトを使用すると、Oracle 製品の実行可能ファイルを再リンクできますが、その方法は Oracle ホーム・ディレクトリにインストールされている製品によって異なります。

製品の実行可能ファイルを再リンクするには、次のコマンドを実行します。

```
$ relink argument
```

このコマンドの *argument* には、表 3-1 に示した値のいずれかを指定します。

**表 3-1 再リンク・スクリプトの引数**

引数	説明
all	インストールされているすべての製品の実行可能ファイル
oracle	Oracle Database の実行可能ファイルのみ
network	net_client、net_server、cman
client	net_client、plssql
client_sharedlib	クライアント共有ライブラリ
interMedia	ctx
ctx	Oracle Text ユーティリティ
precomp	インストールされているすべてのプリコンパイラ
utilities	インストールされているすべてのユーティリティ
oemagent	oemagent  <b>注意：</b> nmo および nmb 実行可能ファイルに正しい権限を付与するには、oemagent を再リンクした後で、root.sh スクリプトを実行する必要があります。
ldap	ldap、oid

---

## SQL\*Plus の管理

この章では、SQL\*Plus の管理方法について説明します。次の項目について説明します。

- [コマンドライン SQL\\*Plus の管理](#)
- [コマンドライン SQL\\*Plus の使用](#)
- [SQL\\*Plus の制限事項](#)

**関連項目：** SQL\*Plus の詳細は、『SQL\*Plus ユーザーズ・ガイドおよびリファレンス』を参照してください。

## 4.1 コマンドライン SQL\*Plus の管理

この項では、コマンドライン SQL\*Plus の管理方法について説明します。例では、疑問符 (?) は環境変数 ORACLE\_HOME の値に置き換えられています。

- 設定ファイルの使用
- PRODUCT\_USER\_PROFILE 表の使用
- Oracle Database サンプル・スキーマの使用
- SQL\*Plus のコマンドライン・ヘルプのインストールと削除

### 4.1.1 設定ファイルの使用

SQL\*Plus を起動すると、最初にサイト・プロファイル設定ファイル glogin.sql が実行され、次にユーザー・プロファイル設定ファイル login.sql が実行されます。

#### サイト・プロファイル・ファイルの使用

グローバルなサイト・プロファイル・ファイルは、`$ORACLE_HOME/sqlplus/admin/glogin.sql` です。この場所にサイト・プロファイル・ファイルがすでに存在する場合は、SQL\*Plus のインストール時にそのファイルが上書きされます。SQL\*Plus を削除すると、そのサイト・プロファイル・ファイルも削除されます。

#### ユーザー・プロファイル・ファイルの使用

ユーザー・プロファイル・ファイルは、`login.sql` です。SQL\*Plus は、最初に現行のディレクトリを検索し、次に環境変数 `SQLPATH` で指定したディレクトリを検索してこのファイルを検出します。この環境変数の値は、ディレクトリをコロンで区切ったリストです。SQL\*Plus では、これらのディレクトリを環境変数 `SQLPATH` にリストされている順序で検索し、`login.sql` ファイルを検出します。

`login.sql` ファイルに設定されているオプションは、`glogin.sql` ファイルに設定されているオプションよりも優先されます。

**関連項目：** プロファイル・ファイルの詳細は、『SQL\*Plus ユーザーズ・ガイドおよびリファレンス』を参照してください。

### 4.1.2 PRODUCT\_USER\_PROFILE 表の使用

Oracle Database には、指定した SQL コマンドや SQL\*Plus コマンドを無効にするための `PRODUCT_USER_PROFILE` 表が用意されています。この表は、事前構成済データベースをインストールするインストール・タイプを選択した場合に、自動的に作成されます。

**関連項目：** インストール・オプションの詳細は、『Oracle Database インストール・ガイド』を参照してください。

`PRODUCT_USER_PROFILE` 表を再作成するには、`SYSTEM` スキーマにある `$ORACLE_HOME/sqlplus/admin/pupbld.sql` スクリプトを実行します。たとえば、次のコマンドを実行します。`SYSTEM_PASSWORD` は、`SYSTEM` ユーザーのパスワードです。

```
$ sqlplus SYSTEM/SYSTEM_PASSWORD
SQL> @?/sqlplus/admin/pupbld.sql
```

また、`$ORACLE_HOME/bin/pupbld` シェル・スクリプトを使用して、手動で `PRODUCT_USER_PROFILE` 表を `SYSTEM` スキーマに再作成することもできます。このスクリプトは、`SYSTEM` パスワードを要求するプロンプトを表示します。プロンプトを表示せずに `pupbld` スクリプトを実行する場合は、環境変数 `SYSTEM_PASS` に `SYSTEM` ユーザー名とパスワードを設定します。

### 4.1.3 Oracle Database サンプル・スキーマの使用

Oracle Database をインストールするか、または Oracle Database Configuration Assistant を使用してデータベースを作成する場合は、Oracle Database サンプル・スキーマをインストールできます。

**関連項目：** Oracle Database サンプル・スキーマのインストールと使用方法は、『Oracle Database サンプル・スキーマ』を参照してください。

### 4.1.4 SQL\*Plus のコマンドライン・ヘルプのインストールと削除

この項では、SQL\*Plus のコマンドライン・ヘルプのインストール方法と削除方法について説明します。

**関連項目：** SQL\*Plus のコマンドライン・ヘルプの詳細は、『SQL\*Plus ユーザーズ・ガイドおよびリファレンス』を参照してください。

#### 4.1.4.1 SQL\*Plus のコマンドライン・ヘルプのインストール

SQL\*Plus のコマンドライン・ヘルプは、次の 3 つの方法でインストールできます。

- 事前構成済データベースのインストールを完了します。

インストールの一部として事前構成済データベースをインストールすると、SQL\*Plus のコマンドライン・ヘルプが SYSTEM スキーマに自動的にインストールされます。

- \$ORACLE\_HOME/bin/helpins シェル・スクリプトを使用して、SYSTEM スキーマにコマンドライン・ヘルプを手動でインストールします。

helpins スクリプトは、SYSTEM パスワードを要求するプロンプトを表示します。プロンプトを表示せずにこのスクリプトを実行する場合は、環境変数 SYSTEM\_PASS に SYSTEM ユーザー名とパスワードを設定します。たとえば、次のように入力します。

- Bourne、Bash または Korn シェルの場合

```
$ SYSTEM_PASS=SYSTEM/system_password; export SYSTEM_PASS
```

- C シェルの場合

```
% setenv SYSTEM_PASS SYSTEM/system_password
```

- \$ORACLE\_HOME/sqlplus/admin/help/helpbld.sql スクリプトを使用して、SYSTEM スキーマにコマンドライン・ヘルプを手動でインストールします。

たとえば、次のコマンドを実行します。system\_password は、SYSTEM ユーザーのパスワードです。

```
$ sqlplus SYSTEM/system_password
SQL> @?/sqlplus/admin/help/helpbld.sql ?/sqlplus/admin/help helpus.sql
```

---

**注意：** helpins シェル・スクリプトおよび helpbld.sql スクリプトは、新しい表を作成する前に、既存のコマンドライン・ヘルプの表を削除します。

---

#### 4.1.4.2 SQL\*Plus のコマンドライン・ヘルプの削除

SYSTEM スキーマから SQL\*Plus のコマンドライン・ヘルプの表を手動で削除するには、\$ORACLE\_HOME/sqlplus/admin/help/helpdrop.sql スクリプトを実行します。このスクリプトには、次のコマンドを実行します。system\_password は、SYSTEM ユーザーのパスワードです。

```
$ sqlplus SYSTEM/system_password
SQL> @?/sqlplus/admin/help/helpdrop.sql
```

## 4.2 コマンドライン SQL\*Plus の使用

この項では、コマンドライン SQL\*Plus の使用方法について説明します。次の項目について説明します。

- SQL\*Plus からのシステム・エディタの使用
- SQL\*Plus からのオペレーティング・システム・コマンドの実行
- SQL\*Plus への割込み
- SPOOL コマンドの使用

### 4.2.1 SQL\*Plus からのシステム・エディタの使用

SQL\*Plus プロンプトで ED または EDIT コマンドを実行すると、ed、emacs、ned、vi などのオペレーティング・システム・エディタが起動します。ただし、環境変数 PATH には、エディタの実行可能ファイルが格納されているディレクトリを指定する必要があります。

エディタを起動すると、現行の SQL バッファがエディタに格納されます。エディタを終了すると、変更された SQL バッファが SQL\*Plus に戻されます。

SQL\*Plus の `_EDITOR` 変数を定義することにより、起動するエディタを指定できます。この変数は、`glogin.sql` サイト・プロファイルまたは `login.sql` ユーザー・プロファイルに定義できます。また、SQL\*Plus セッション時に定義することもできます。たとえば、デフォルト・エディタを vi に設定するには、次のコマンドを実行します。

```
SQL> DEFINE _EDITOR=vi
```

`_EDITOR` 変数を設定しない場合は、環境変数 `EDITOR` または `VISUAL` のいずれかの値が使用されます。両方の環境変数が設定されている場合は、環境変数 `EDITOR` の値が使用されます。

`_EDITOR`、`EDITOR` および `VISUAL` のいずれも指定されていない場合、デフォルト・エディタは ed になります。

エディタを起動すると、SQL\*Plus は一時ファイル `afiedt.buf` を使用してエディタにテキストを渡します。SET EDITFILE コマンドを使用すると、別のファイル名を指定できます。たとえば、次のように入力します。

```
SQL> SET EDITFILE /tmp/myfile.sql
```

SQL\*Plus は、一時ファイルを削除しません。

### 4.2.2 SQL\*Plus からのオペレーティング・システム・コマンドの実行

SQL\*Plus プロンプトの後の最初の文字として HOST コマンドまたは感嘆符 (!) を使用すると、後続の文字がサブシェルに渡されます。オペレーティング・システム・コマンドを実行するとき使用するシェルは、環境変数 SHELL によって設定されます。デフォルト・シェルは Bourne シェルです。シェルが実行できない場合は、エラー・メッセージが表示されます。

SQL\*Plus に戻るには、exit コマンドを実行するか、[Ctrl] キーを押しながら [D] キーを押します。

たとえば、1 つのコマンドを実行するには、次のコマンド構文を使用します。

```
SQL> ! command
```

この例の `command` は、実行するオペレーティング・システム・コマンドです。

SQL\*Plus から複数のオペレーティング・システム・コマンドを実行するには、HOST または ! コマンドを実行してから [Enter] キーを押します。オペレーティング・システム・プロンプトに戻ります。

### 4.2.3 SQL\*Plus への割込み

SQL\*Plus の実行中に、[Ctrl] キーを押しながら [C] キーを押すと、スクロール中のレコード表示を停止し、SQL 文を終了できます。

### 4.2.4 SPOOL コマンドの使用

SPOOL コマンドで生成されるファイルのデフォルトのファイル名拡張子は、.lst です。この拡張子を変更するには、ピリオド (.) を含めたスプール・ファイル名を指定します。たとえば、次のように入力します。

```
SQL> SPOOL query.txt
```

## 4.3 SQL\*Plus の制限事項

この項では、次の SQL\*Plus の制限事項を説明します。

- ウィンドウのサイズ変更
- リターン・コード
- パスワードの非表示

### 4.3.1 ウィンドウのサイズ変更

SQL\*Plus のシステム変数 LINESIZE および PAGESIZE のデフォルト値では、ウィンドウのサイズは自動的に調整されません。

### 4.3.2 リターン・コード

オペレーティング・システムのリターン・コードは 1 バイトですが、Oracle エラー・コードを返すには、1 バイトでは不十分です。リターン・コードの範囲は、0 ~ 255 です。

### 4.3.3 パスワードの非表示

環境変数 SYSTEM\_PASS に SYSTEM ユーザーのユーザー名とパスワードを設定した場合は、ps コマンドの出力にこの情報が表示される場合があります。権限のないアクセスを防ぐため、SQL\*Plus によってプロンプトが表示された場合のみ SYSTEM パスワードを入力してください。

スクリプトを自動的に実行する場合は、パスワードの格納が不要な認証方法の使用を考慮してください。たとえば、Oracle Database への外部認証ログインなどがあります。セキュリティの低い環境では、スクリプト・ファイルにオペレーティング・システム・パイプを使用して、パスワードを SQL\*Plus に渡すことを検討してください。たとえば、次のように入力します。

```
$ echo system_password | sqlplus SYSTEM @MYSCRIPT
```

あるいは、次のコマンドを実行します。

```
$ sqlplus <<EOF
SYSTEM/system_password
SELECT ...
EXIT
EOF
```

この例の `system_password` は、SYSTEM ユーザーのパスワードです。



---

## Oracle Net Services の構成

この章では、Oracle Net Services を構成する方法について説明します。次の項目について説明します。

- [Oracle Net Services 構成ファイルの保存場所](#)
- [Adapters ユーティリティ](#)
- [Oracle Protocol Support](#)
- [TCP/IP または SSL 付き TCP/IP 用のリスナーの設定](#)
- [Oracle Advanced Security](#)

**関連項目：** Oracle Net Services の詳細は、『Oracle Database Net Services 管理者ガイド』を参照してください。

## 5.1 Oracle Net Services 構成ファイルの保存場所

Oracle Net Services 構成ファイルは、多くの場合、`$ORACLE_HOME/network/admin` ディレクトリにあります。ファイルのタイプに応じて、Oracle Net は異なる検索順序でファイルを検出します。

`sqlnet.ora` および `ldap.ora` の各ファイルの検索順序は、次のとおりです。

1. 環境変数 `TNS_ADMIN` で指定したディレクトリ（この環境変数が設定されている場合）
2. `$ORACLE_HOME/network/admin` ディレクトリ

`cman.ora`、`listener.ora` および `tnsnames.ora` の各ファイルの検索順序は、次のとおりです。

1. 環境変数 `TNS_ADMIN` で指定したディレクトリ（この環境変数が設定されている場合）
2. 次のいずれかのディレクトリ
  - Solaris の場合  
    `/var/opt/oracle`
  - その他のプラットフォームの場合  
    `/etc`

3. `$ORACLE_HOME/network/admin` ディレクトリ

一部のシステム・レベルの構成ファイルでは、対応するユーザー・レベルの構成ファイル（ユーザーのホーム・ディレクトリに格納されている）を作成できます。ユーザー・レベルのファイルの設定によって、システム・レベルのファイルの設定が上書きされます。次の表に、システム・レベルの構成ファイルとそれに対応するユーザー・レベルの構成ファイルを示します。

システム・レベルの構成ファイル	ユーザー・レベルの構成ファイル
<code>sqlnet.ora</code>	<code>\$HOME/.sqlnet.ora</code>
<code>tnsnames.ora</code>	<code>\$HOME/.tnsnames.ora</code>

### サンプル構成ファイル

`$ORACLE_HOME/network/admin/samples` ディレクトリには、`cman.ora`、`listener.ora`、`sqlnet.ora`、`tnsnames.ora` の各構成ファイルのサンプルが含まれています。

---

**注意：** `cman.ora` ファイルがインストールされるのは、カスタム・インストールの一部として **Connection Manager** を選択した場合のみです。

---

## 5.2 Adapters ユーティリティ

使用しているシステムで Oracle Database がサポートしているトランスポート・プロトコル、ネーミング・メソッドおよび Oracle Advanced Security のオプションを表示する場合は、adapters ユーティリティを使用します。adapters ユーティリティを使用するには、次のコマンドを実行します。

```
$ cd $ORACLE_HOME/bin
$ adapters ./oracle
```

Oracle Database Client システムの adapters ユーティリティでは、次のような出力が表示されます。

Oracle Net transport protocols linked with ./oracle are

```
IPC
BEQ
TCP/IP
SSL
RAW
```

Oracle Net naming methods linked with ./oracle are:

```
Local Naming (tnsnames.ora)
Oracle Directory Naming
Oracle Host Naming
NIS Naming
```

Oracle Advanced Security options linked with ./oracle are:

```
RC4 40-bit encryption
RC4 128-bit encryption
RC4 256-bit encryption
DES40 40-bit encryption
DES 56-bit encryption
3DES 112-bit encryption
3DES 168-bit encryption
AES 128-bit encryption
AES 192-bit encryption
SHA crypto-checksumming (for FIPS)
SHA-1 crypto-checksumming
Kerberos v5 authentication
RADIUS authentication
ENTRUST authentication
```

**関連項目：** adapters ユーティリティの詳細は、『Oracle Database Net Services 管理者ガイド』を参照してください。

## 5.3 Oracle Protocol Support

Oracle Protocol Support は、Oracle Net のコンポーネントの 1 つです。次の要素で構成されています。

- [IPC プロトコル・サポート](#)
- [TCP/IP プロトコル・サポート](#)
- [SSL 付き TCP/IP プロトコル・サポート](#)

IPC、TCP/IP および Secure Sockets Layer (SSL) 付き TCP/IP の各プロトコル・サポートには、それぞれアドレス指定が必要です。このアドレスは、Oracle Net Services 構成ファイルおよび DISPATCHER 初期化パラメータで使用されます。次の各項では、各プロトコル・サポートのアドレス指定について説明します。

### 関連項目：

- HP-UX (PA-RISC) および Tru64 UNIX システムで DCE がインストールされている場合は、DCE を Oracle Net プロトコルとして使用できます。DCE プロトコル・サポートの構成の詳細は、『Oracle Database Advanced Security 管理者ガイド』を参照してください。
- Oracle Protocol Support の詳細は、『Oracle Database Net Services 管理者ガイド』を参照してください。

### 5.3.1 IPC プロトコル・サポート

IPC プロトコル・サポートは、クライアント・プログラムと Oracle Database が同じシステムにインストールされている場合にのみ使用できます。このプロトコル・サポートには、リスナーが必要です。このプロトコル・サポートは、すべてのクライアント・ツールおよび oracle 実行可能ファイルにインストールおよびリンクされます。

IPC プロトコル・サポートには、次の書式のアドレス指定が必要です。

```
(ADDRESS = (PROTOCOL=IPC) (KEY=key))
```

次の表に、このアドレス指定で使用されるパラメータを示します。

パラメータ	説明
PROTOCOL	使用するプロトコルを表します。この値は IPC です。大 / 小文字は区別されません。
KEY	同じシステムで IPC KEY として使用されている他の名前とは異なる、一意の名前を表します。

次に、IPC プロトコル・アドレスのサンプルを示します。

```
(ADDRESS= (PROTOCOL=IPC) (KEY=EXTPROC))
```

### 5.3.2 TCP/IP プロトコル・サポート

TCP/IP は、ネットワークを介したクライアント / サーバー通信に使用される標準的な通信プロトコルです。TCP/IP プロトコル・サポートを使用すると、クライアント・プログラムと Oracle Database が同じシステムまたは別のシステムのどちらかにインストールされていても、それらの間で通信を行うことができます。使用しているシステムに TCP/IP プロトコルをインストールした場合、TCP/IP プロトコル・サポートがすべてのクライアント・ツールおよび oracle 実行可能ファイルにインストールおよびリンクされます。

TCP/IP プロトコル・サポートには、次の書式のアドレス指定が必要です。

```
(ADDRESS = (PROTOCOL=TCP) (HOST=hostname) (PORT=port))
```

次の表に、このアドレス指定で使用されるパラメータを示します。

パラメータ	説明
PROTOCOL	使用するプロトコル・サポートを表します。この値は TCP です。大 / 小文字は区別されません。
HOST	ホスト名またはホストの IP アドレスを表します。
PORT	TCP/IP ポートを表します。このポートには、番号または /etc/services ファイルでこのポートにマップされた別名を指定します。推奨値は 1521 です。

次に、TCP/IP プロトコル・アドレスのサンプルを示します。

```
(ADDRESS= (PROTOCOL=TCP) (HOST=MADRID) (PORT=1521))
```

### 5.3.3 SSL 付き TCP/IP プロトコル・サポート

SSL 付き TCP/IP プロトコル・サポートを使用すると、クライアントの Oracle アプリケーションは、TCP/IP および SSL を介したリモートの Oracle Database インスタンスと通信できます。SSL 付き TCP/IP を使用するには、Oracle Advanced Security をインストールする必要があります。

SSL 付き TCP/IP プロトコル・サポートには、次の書式のアドレス指定が必要です。

```
(ADDRESS = (PROTOCOL=TCPS) (HOST=hostname) (PORT=port))
```

次の表に、このアドレス指定で使用されるパラメータを示します。

パラメータ	説明
PROTOCOL	使用するプロトコルを表します。この値は TCPS です。大 / 小文字は区別されません。
HOST	ホスト名またはホストの IP アドレスを表します。
PORT	SSL 付き TCP/IP ポートを表します。このポートには、番号または /etc/services ファイルでこのポートにマップされた別名を指定します。推奨値は 2484 です。

次に、SSL 付き TCP/IP プロトコル・アドレスのサンプルを示します。

```
(ADDRESS= (PROTOCOL=TCPS) (HOST=MADRID) (PORT=2484))
```

## 5.4 TCP/IP または SSL 付き TCP/IP 用のリスナーの設定

リスナー用のポートは、ネットワーク上の各 Oracle Net Services ノードの `/etc/services` ファイルで予約することをお勧めします。デフォルトのポートは 1521 です。エントリーにはリスナー名とポート番号がリストされます。たとえば、次のように入力します。

```
oraclelistener 1521/tcp
```

この例の `oraclelistener` は、`listener.ora` ファイルに定義されているリスナーの名前です。複数のリスナーを起動する場合は、複数のポートを予約してください。

SSL を使用する場合は、SSL 付き TCP/IP 用のポートを `/etc/services` ファイルに定義する必要があります。推奨値は 2484 です。たとえば、次のように入力します。

```
oraclelistenersssl 2484/tcps
```

この例の `oraclelistenersssl` は、`listener.ora` ファイルに定義されているリスナーの名前です。複数のリスナーを起動する場合は、複数のポートを予約してください。

## 5.5 Oracle Advanced Security

Oracle Advanced Security をインストールすると、3 つの `.bak` ファイル (`naect.o.bak`、`naect.o.bak` および `naedhs.o.bak`) が作成されます。これらのファイルは、`$ORACLE_HOME/lib` ディレクトリに格納されます。これらのファイルは、Oracle Advanced Security を削除するときの再リンクに必要です。これらのファイルを削除しないでください。

---

---

## Oracle プリコンパイラおよび Oracle Call Interface の使用

この章では、Oracle プリコンパイラおよび Oracle Call Interface の使用方法について説明します。次の項目について説明します。

- Oracle プリコンパイラの概要
- クライアント・アプリケーションのビット長サポート
- Pro\*C/C++ プリコンパイラ
- Pro\*COBOL プリコンパイラ
- Pro\*FORTRAN プリコンパイラ
- SQL\*Module for ADA
- OCI と OCCI
- 64 ビット・ドライバを使用する Oracle JDBC/OCI プログラム
- カスタム Make ファイル
- 未定義シンボルの修正
- マルチスレッド・アプリケーション
- シグナル・ハンドラの使用
- XA 機能

---

---

**注意：** この章で説明するデモンストレーションを使用するには、Oracle Database 10g Companion CD に収録されている Oracle Database Examples をインストールしてください。

---

---

## 6.1 Oracle プリコンパイラの概要

Oracle プリコンパイラは、Oracle Database の SQL 文を高水準言語で記述されたプログラムと組み合わせるためのアプリケーション開発ツールです。Oracle プリコンパイラは、ANSI SQL と互換性があり、Oracle Database やその他の ANSI SQL データベース管理システムで実行するオープンでカスタマイズされたアプリケーションを開発するために使用します。

次の項目について説明します。

- [プリコンパイラ構成ファイル](#)
- [プリコンパイラ実行可能ファイルの再リンク](#)
- [プリコンパイラの README ファイル](#)
- [すべてのプリコンパイラに共通の問題](#)
- [静的および動的リンク](#)
- [クライアント共有ライブラリとクライアント静的ライブラリ](#)

### 6.1.1 プリコンパイラ構成ファイル

Oracle プリコンパイラの構成ファイルは、`$ORACLE_HOME/precomp/admin` ディレクトリにあります。表 6-1 に、各プリコンパイラの構成ファイルの名前を示します。

表 6-1 Oracle プリコンパイラのシステム構成ファイル

製品	構成ファイル
Pro*C/C++	pcscfg.cfg
Pro*COBOL (AIX、HP-UX、Solaris、Tru64 UNIX および zSeries Linux)	pcbcfg.cfg
Pro*FORTRAN (AIX、HP-UX、Solaris および Tru64 UNIX)	pccfor.cfg
Object Type Translator	ottcfg.cfg
SQL*Module for Ada (AIX)	pmscfg.cfg

### 6.1.2 プリコンパイラ実行可能ファイルの再リンク

すべてのプリコンパイラ実行可能ファイルを再リンクするには、Make ファイル `$ORACLE_HOME/precomp/lib/ins_precomp.mk` を使用します。特定のプリコンパイラ実行可能ファイルを手動で再リンクするには、次のコマンドを入力します。

```
$ make -f ins_precomp.mk relink exename = executable_name
```

このコマンドを実行すると、最初に、新しい実行可能ファイルが `$ORACLE_HOME/precomp/lib` ディレクトリに作成され、次にそのファイルが `$ORACLE_HOME/bin` ディレクトリに移動されます。

この例では、`executable` を表 6-2 で示した製品の実行可能ファイルで置き換えます。

表 6-2 Oracle プリコンパイラの実行可能ファイル

製品	実行可能ファイル
Pro*C/C++	proc
Pro*COBOL (AIX、HP-UX、Linux on POWER、Tru64 UNIX、Solaris SPARC (64 ビット) および zSeries Linux)	procob または rtsora
Pro*COBOL 32 ビット (AIX、HP-UX PA-RISC、HP-UX Itanium、Solaris SPARC (64 ビット) および zSeries Linux)	procob32 または rtsora32
Pro*FORTRAN (AIX、HP-UX、Linux on POWER、Solaris および Tru64 UNIX)	profor (AIX および Solaris の場合は 32 ビット)

表 6-2 Oracle プリコンパイラの実行可能ファイル (続き)

製品	実行可能ファイル
Pro*FORTRAN 32 ビット (HP-UX および Linux on POWER)	profor32
SQL*Module for Ada (AIX)	modada

### 6.1.3 プリコンパイラの README ファイル

表 6-3 に、プリコンパイラの README ファイルの場所を示します。README ファイルには、前回のリリース以降にプリコンパイラに加えられた変更が記載されています。

表 6-3 プリコンパイラの README ファイルの保存場所

プリコンパイラ	README ファイル
Pro*C/C++	\$ORACLE_HOME/precomp/doc/proc2/readme.doc
Pro*COBOL	\$ORACLE_HOME/precomp/doc/procob2/readme.doc
Pro*FORTRAN	\$ORACLE_HOME/precomp/doc/pro1x/readme.txt

### 6.1.4 すべてのプリコンパイラに共通の問題

次の問題は、すべてのプリコンパイラに共通しています。

- 大文字から小文字への変換

C 言語以外では、コンパイラによって、大文字の関数やサブプログラム名が小文字に変換されます。この結果、ユーザーが存在しない旨のエラー・メッセージが表示されることがあります。このエラー・メッセージが表示された場合は、オプション・ファイル内の関数またはサブプログラム名の大 / 小文字が、IAPXTB 表の文字と一致しているかどうかを確認してください。

- ベンダー提供のデバッガ・プログラム

プリコンパイラとベンダー提供のデバッガに互換性がない場合があります。デバッガを使用して動作したプログラムが、デバッガを使用しないときも同様に動作するとはかぎりません。

- IRECLLEN および ORECLLEN の各パラメータの値

IRECLLEN および ORECLLEN の各パラメータには、最大値がありません。

### 6.1.5 静的および動的リンク

Oracle ライブラリは、プリコンパイラや OCI または OCCI アプリケーションと静的または動的にリンクできます。静的リンクの場合、アプリケーション全体のライブラリおよびオブジェクトは、1 つの実行可能プログラムにリンクされます。その結果、アプリケーションの実行可能ファイルが非常に大きくなる場合があります。

動的リンクの場合、実行コードの一部が実行可能プログラムに格納され、残りの部分はアプリケーションの実行時に動的にリンクされるライブラリに格納されます。実行時にリンクされるライブラリを、動的ライブラリまたは共有ライブラリと呼びます。動的リンクには、次のようないくつかのメリットがあります。

- ディスク領域要件が少なくなります。複数のアプリケーションで、または同一アプリケーションのコールで、同じ動的ライブラリを使用できます。
- メイン・メモリー要件が少なくなります。同一の動的ライブラリ・イメージをメイン・メモリーに一度ロードしておくことで、複数のアプリケーションでそれを共有できます。

## 6.1.6 クライアント共有ライブラリとクライアント静的ライブラリ

クライアント共有ライブラリとクライアント静的ライブラリは、`$ORACLE_HOME/lib` ディレクトリまたは `$ORACLE_HOME/lib32` ディレクトリにあります。オラクル社が提供している Make ファイル `demo_product.mk` を使用してアプリケーションをリンクする場合は、デフォルトでクライアント共有ライブラリがリンクされます。

共有ライブラリ・パスの環境変数設定にクライアント共有ライブラリを含むディレクトリが指定されていない場合、実行可能ファイルの起動時に、次のようなエラー・メッセージのいずれかが表示される場合があります。

```
Cannot load library libclntsh.a
Can't open shared library: ../libclntsh.sl.10.1
libclntsh.so.10.1: can't open file: errno=2
can't open library: ../libclntsh.dylib.10.1
Cannot map libclntsh.so
```

このエラーを防ぐには、共有ライブラリ・パスの環境変数を設定し、適切なディレクトリを指定します。次の表に、環境変数名のサンプル設定を示します。使用しているプラットフォームで 32 ビットおよび 64 ビットのアプリケーションがどちらもサポートされている場合、実行するアプリケーションに応じて、正しいディレクトリを指定してください。

プラットフォーム	環境変数	サンプル設定
AIX (32 ビット・アプリケーション)	LIBPATH	<code>\$ORACLE_HOME/lib32</code>
AIX (64 ビット・アプリケーション)	LIBPATH	<code>\$ORACLE_HOME/lib</code>
HP-UX (32 ビット・アプリケーション)	SHLIB_PATH	<code>\$ORACLE_HOME/lib32</code>
HP-UX (64 ビット・アプリケーション)、Linux および Tru64 UNIX	LD_LIBRARY_PATH	<code>\$ORACLE_HOME/lib</code>
Mac OS X	DYLD_LIBRARY_PATH	<code>\$ORACLE_HOME/lib</code> (64 ビット・ライブラリ) <code>\$ORACLE_HOME/lib32</code> (32 ビット・ライブラリ)
zSeries Linux (31 ビット・アプリケーション)、Linux on POWER (32 ビット・アプリケーション) および Solaris (32 ビット・アプリケーション)	LD_LIBRARY_PATH	<code>\$ORACLE_HOME/lib32</code>
Linux on POWER (64 ビット・アプリケーション) および Solaris (32 ビット・アプリケーション)	LD_LIBRARY_PATH_64	<code>\$ORACLE_HOME/lib</code>

クライアント共有ライブラリは、インストール時に自動的に作成されます。クライアント共有ライブラリを再作成する場合は、次の手順を実行します。

1. クライアント共有ライブラリを使用するすべてのクライアント・アプリケーションを終了します。この中には、SQL\*Plus や Oracle Recovery Manager などの Oracle Client アプリケーションもすべて含まれます。
2. oracle ユーザーでログインし、次のコマンドを実行します。

```
$ $ORACLE_HOME/bin/genclntsh
```

## 非スレッド化クライアント共有ライブラリ

**注意:** この項の内容は、HP-UX PA-RISC システムに適用されます。

HP-UX PA-RISC では、非スレッド化クライアント共有ライブラリを使用できます。ただし、このライブラリは、スレッドを使用する OCI アプリケーションやスレッドに依存する OCI アプリケーションでは使用できません。

スレッドを使用しないアプリケーションにこのライブラリを使用するには、次のいずれかのコマンドを実行して OCI アプリケーションを作成します。

- 32 ビット・アプリケーションの場合
 

```
$ make -f demo_rdbms32.mk build_nopthread EXE=oci02 OBJS=oci02.o
```
- 64 ビット・アプリケーションの場合
 

```
$ make -f demo_rdbms.mk build_nopthread EXE=oci02 OBJS=oci02.o
```

## 6.2 クライアント・アプリケーションのビット長サポート

次の表に、クライアント・アプリケーションについてサポートされるビット長（31 ビット、32 ビットまたは 64 ビット）を示します。

クライアント・アプリケーションのタイプ	サポートされるプラットフォーム
32 ビットのみ	Linux x86 および Solaris x86
64 ビットのみ	Tru64 UNIX および Linux Itanium
32 ビットおよび 64 ビット	Linux on POWER、Solaris SPARC、Linux x86-64 および Mac OS X
31 ビットおよび 64 ビット	zSeries Linux

AIX、HP-UX、Linux on POWER、Solaris SPARC および zSeries Linux では、Oracle Database 10g リリース 2 (10.2) とともに提供されているデモンストレーションおよびクライアント・アプリケーションはすべて、64 ビット・モードでリンクおよび実行します。AIX、Linux on POWER、Solaris SPARC および HP-UX では、同じ Oracle ホーム・ディレクトリに 32 ビットおよび 64 ビットのクライアント・アプリケーションを作成できます。同様に、zSeries Linux では、同じ Oracle ホーム・ディレクトリに 31 ビットおよび 64 ビットのクライアント・アプリケーションを作成できます。

次の表に、32 ビットおよび 64 ビットのクライアント共有ライブラリを示します。

プラットフォーム	32 ビット (または 31 ビット) のクライアント共有ライブラリ	64 ビットのクライアント共有ライブラリ
AIX	\$ORACLE_HOME/lib32/libclntsh.a \$ORACLE_HOME/lib32/libclntsh.so	\$ORACLE_HOME/lib/libclntsh.a \$ORACLE_HOME/lib/libclntsh.so
HP-UX PA-RISC	\$ORACLE_HOME/lib32/libclntsh.sl	\$ORACLE_HOME/lib/libclntsh.sl
HP-UX Itanium、Solaris、Linux on POWER、Linux x86-64 および zSeries Linux	\$ORACLE_HOME/lib32/libclntsh.so <b>注意:</b> zSeries Linux では、\$ORACLE_HOME/lib32 ディレクトリに 31 ビットのライブラリが含まれます。	\$ORACLE_HOME/lib/libclntsh.so
Mac OS X	\$ORACLE_HOME/lib32/libclntsh.dylib	\$ORACLE_HOME/lib/libclntsh.dylib

異なるワード・サイズが混在しているインストールを実装するには、次の手順を実行します。

1. 次のコマンドを実行して、32 ビットおよび 64 ビットのクライアント共有ライブラリを作成します。  

```
$ $ORACLE_HOME/bin/genclntsh
```
2. 使用しているプラットフォームに応じて、次のいずれかの環境変数に、必要な 32 ビットおよび 64 ビットのクライアント共有ライブラリのパスを入力します。

プラットフォーム	環境変数
AIX	LIBPATH
HP-UX (32 ビットのクライアント・アプリケーション)	SHLIB_PATH
HP-UX、Linux on POWER、Linux x86、Linux x86-64、Solaris および Tru64 UNIX	LD_LIBRARY_PATH
Mac OS X	DYLD_LIBRARY_PATH

### 32 ビットの Pro\*C および OCI カスタマ・アプリケーションの作成

32 ビットおよび 64 ビットの Pro\*C および Oracle Call Interface (OCI) カスタマ・アプリケーションをサポートするオペレーティング・システムでは、次のファイルに 32 ビット Pro\*C および OCI アプリケーションの作成方法が記載されています。

作成内容..	参照先の Make ファイル..
32 ビットの Pro*C アプリケーションの作成 (Linux on POWER を除くすべてのプラットフォーム)	\$ORACLE_HOME/precomp/demo/proc/demo_proc32.mk
32 ビットの OCI アプリケーションの作成 (Linux on POWER を除くすべてのプラットフォーム)	\$ORACLE_HOME/rdbms/demo/demo_rdbms32.mk
32 ビットの Pro*C アプリケーションの作成 (Linux on POWER の場合)	\$ORACLE_HOME/precomp/demo/proc/demo_proc.mk
32 ビットの OCI アプリケーションの作成 (Linux on POWER の場合)	\$ORACLE_HOME/rdbms/demo/demo_rdbms.mk

**注意：** Linux on POWER で、前の表で指定した make ファイルを使用して COMPILER\_MODE=32 を設定すると、GCC を持つ 32 ビット・オブジェクトが作成されます。

COMPILER=VAC. を設定すると、オペレーティング・システムで IBM XL Compiler を起動できます。

### 32 ビットの実行可能ファイルとライブラリ

**注意：** この項の内容は、Mac OS X、AIX、HP-UX、Linux on POWER、Solaris SPARC および zSeries Linux の各プラットフォームに適用されます。

32 ビットおよび 64 ビットのアプリケーションをサポートするプラットフォームでは、\$ORACLE\_HOME/bin ディレクトリに 32 ビットおよび 64 ビットの実行可能ファイルが格納されています。さらに、次のディレクトリに 32 ビットのライブラリが格納されています。

- \$ORACLE\_HOME/lib32
- \$ORACLE\_HOME/rdbms/lib32
- \$ORACLE\_HOME/hs/lib32
- \$ORACLE\_HOME/network/lib32
- \$ORACLE\_HOME/precomp/lib32

## 6.3 Pro\*C/C++ プリコンパイラ

Pro\*C/C++ プリコンパイラを使用する場合は、事前にオペレーティング・システムの適切なバージョンのコンパイラが正しくインストールされていることを確認してください。

### 関連項目：

- サポートされるコンパイラのバージョンの詳細は、『Oracle Database インストレーション・ガイド』を参照してください。
- Pro\*C/C++ プリコンパイラおよびインタフェース機能の詳細は、『Pro\*C/C++ プログラマーズ・ガイド』を参照してください。

### 6.3.1 Pro\*C/C++ のデモ・プログラム

デモ・プログラムは、Pro\*C/C++ プリコンパイラの機能を紹介するために用意されています。デモ・プログラムには、C、C++ および Object プログラムの 3 種類があります。デモ・プログラムはすべて、\$ORACLE\_HOME/precomp/demo/proc ディレクトリにあります。デフォルトでは、すべてのプログラムがクライアント共有ライブラリに動的にリンクされます。

デモ・プログラムを実行するには、\$ORACLE\_HOME/sqlplus/demo/demobld.sql スクリプトで作成したデモンストレーション表が SCOTT スキーマにあり、そのパスワードが TIGER である必要があります。

---

**注意：** デモンストレーションを作成する前に、SCOTT アカウントのロックを解除してパスワードを設定する必要があります。

---

デモ・プログラムを作成するには、\$ORACLE\_HOME/precomp/demo/proc/ ディレクトリにある Make ファイル demo\_proc.mk を使用します。たとえば、sample1 のデモ・プログラムをプリコンパイル、コンパイルおよびリンクするには、次のコマンドを実行します。

---

**注意：** AIX システムの場合は、デモ・プログラムが正しくコンパイルされるようにするために、次の例で make コマンドに -r オプションを指定してください。たとえば、次のように入力します。

---

```
$ make -r -f demo_proc.mk sample1
```

---

```
$ make -f demo_proc.mk sample1
```

Pro\*C/C++ の C デモ・プログラムをすべて作成するには、次のコマンドを実行します。

```
$ make -f demo_proc.mk samples
```

Pro\*C/C++ の C++ デモ・プログラムをすべて作成するには、次のコマンドを実行します。

```
$ make -f demo_proc.mk cppsamples
```

Pro\*C/C++ の Object デモ・プログラムをすべて作成するには、次のコマンドを実行します。

```
$ make -f demo_proc.mk object_samples
```

一部のデモ・プログラムでは、`$ORACLE_HOME/precomp/demo/sql` ディレクトリ内にある SQL スクリプトを実行する必要があります。このスクリプトを実行しないと、実行を要求するメッセージが表示されます。

デモ・プログラムを作成し、それに対応する SQL スクリプトを実行するには、「make マクロ引数 `RUNSQL=run`」をコマンドラインに追加します。たとえば、`sample9` デモ・プログラムを作成し、必要な `$ORACLE_HOME/precomp/demo/sql/sample9.sql` スクリプトを実行するには、次のコマンドを実行します。

```
$ make -f demo_proc.mk sample9 RUNSQL=run
```

Object デモ・プログラムのすべてを作成し、必要な SQL スクリプトすべてを実行するには、次のコマンドを実行します。

```
$ make -f demo_proc.mk object_samples RUNSQL=run
```

## 6.3.2 Pro\*C/C++ のユーザー・プログラム

Make ファイル `$ORACLE_HOME/precomp/demo/proc/demo_proc.mk` を使用して、ユーザー・プログラムを作成できます。この Make ファイルは、32 ビットまたは 64 ビットのユーザー・プログラムを作成します。Make ファイル `demo_proc32.mk` を使用して、32 ビット (zSeries Linux の場合は 31 ビット) のユーザー・プログラムを作成することもできます。次の表に、Pro\*C/C++ により 32 ビット (または 31 ビット) および 64 ビットのユーザー・プログラムを作成するための Make ファイルを示します。

プラットフォーム	64 ビット Make ファイル	32 ビット Make ファイル
AIX、HP-UX、Solaris SPARC、zSeries Linux および Mac OS X	<code>demo_proc.mk</code>	<code>demo_proc32.mk</code> <b>注意:</b> zSeries Linux の場合、この Make ファイルは 31 ビットのユーザー・プログラムを作成します。
Linux x86 および Solaris	該当なし	<code>demo_proc.mk</code>
Linux Itanium および Tru64 UNIX	<code>demo_proc.mk</code>	該当なし
Linux on POWER	<code>demo_proc.mk</code>	<code>demo_proc.mk</code> <b>注意:</b> <code>export COMPILER_MODE=32</code>

**関連項目:** ユーザー・プログラムの作成方法は、その Make ファイルを参照してください。

**注意:** AIX システムの場合は、プログラムが正しくコンパイルされるようにするために、次の例の `make` コマンドに `-r` オプションを指定してください。

Make ファイル `demo_proc.mk` を使用してプログラムを作成する場合は、次のようなコマンドを実行します。

```
$ make -f demo_proc.mk target OBJS="objfile1 objfile2 ..." EXE=exename
```

各項目の意味は次のとおりです。

- `target` は、使用する Make ファイルのターゲットです。
- `objfilen` は、プログラムをリンクするためのオブジェクト・ファイルです。
- `exename` は、実行可能プログラムです。

たとえば、Pro\*C/C++ ソース・ファイル myprog.pc からプログラム myprog を作成する場合は、作成する実行可能ファイルのソースとタイプに応じて、次のいずれかのコマンドを実行します。

- C ソースの場合、クライアント共有ライブラリに動的にリンクさせるには、次のコマンドを実行します。

```
$ make -f demo_proc.mk build OBJS=myprog.o EXE=myprog
```

- C ソースの場合、クライアント共有ライブラリに静的にリンクさせるには、次のコマンドを実行します。

```
$ make -f demo_proc.mk build_static OBJS=myprog.o EXE=myprog
```

- C++ ソースの場合、クライアント共有ライブラリに動的にリンクさせるには、次のコマンドを実行します。

```
$ make -f demo_proc.mk cppbuild OBJS=myprog.o EXE=myprog
```

- C++ ソースの場合、クライアント共有ライブラリに静的にリンクさせるには、次のコマンドを実行します。

```
$ make -f demo_proc.mk cppbuild_static OBJS=myprog.o EXE=myprog
```

## 6.4 Pro\*COBOL プリコンパイラ

---

**注意：** Linux Itanium、Solaris x86 および Mac OS X では、Pro\*COBOL はサポートされていません。Linux on POWER の場合、Pro\*COBOL は 64 ビットでのみサポートされます。

---

表 6-4 に、Pro\*COBOL プリコンパイラのネーミング規則を示します。

**表 6-4 Pro\*COBOL のネーミング規則**

項目	ネーミング規則
実行可能ファイル	procob または procob32
デモンストレーション・ディレクトリ	procob2
Make ファイル	demo_procob.mk または demo_procob_32.mk

Pro\*COBOL では、静的リンク、動的リンクまたは動的読取りプログラムをサポートしています。動的リンク・プログラムは、クライアント共有ライブラリを使用します。動的読取りプログラムは、\$ORACLE\_HOME/bin ディレクトリにある rtsora (32 ビットの COBOL コンパイラの場合は、rtsora32) 実行可能ファイルを使用します。

## 6.4.1 Pro\*COBOL の環境変数

この項では、Pro\*COBOL で必要な環境変数について説明します。

### 6.4.1.1 Acucorp ACUCOBOL-GT COBOL コンパイラ

---



---

**注意：** Linux on POWER では、Acucorp ACUCOBOL はサポートされていません。

---



---

Acucorp ACUCOBOL-GT COBOL コンパイラを使用するには、`A_TERMCAP`、`A_TERM`、`PATH` および `LD_LIBRARY_PATH` 環境変数を設定する必要があります。環境変数 `LD_LIBRARY_PATH` に正しいディレクトリが設定されていない場合は、プログラムのコンパイルまたは実行時に、次のようなエラー・メッセージが表示されます。

```
runcbl: error while loading shared libraries: libclntsh.so:
cannot open shared object file: No such file or directory
```

#### A\_TERMCAP および A\_TERM

`a_termcap` ファイルの場所を指定するには環境変数 `A_TERMCAP` を設定し、そのファイルからサポートされる端末を指定するには環境変数 `A_TERM` を設定します。たとえば、次のように入力します。

- Bourne、Bash または Korn シェルの場合

```
$ A_TERMCAP=/opt/COBOL/etc/a_termcap
$ A_TERM=vt100
$ export A_TERMCAP A_TERM
```

- C シェルの場合

```
% setenv A_TERMCAP /opt/COBOL/etc/a_termcap
% setenv A_TERM vt100
```

#### PATH

`/opt/COBOL/bin` ディレクトリを含むように環境変数 `PATH` を設定します (zSeries Linux システムで 31 ビット・コンパイラを使用している場合は、`/opt/COBOL31/bin` ディレクトリ)。

- Bourne、Bash または Korn シェルの場合

```
$ PATH=/opt/COBOL/bin:$PATH
$ export PATH
```

- C シェルの場合

```
% setenv PATH opt/COBOL/bin:${PATH}
```

#### LD\_LIBRARY\_PATH

---



---

**注意：** AIX での `LIBPATH` 変数は、`LD_LIBRARY_PATH` 変数と同等です。AIX では、次のコマンドで `LD_LIBRARY_PATH` 変数ではなく、`LIBPATH` 変数を使用する必要があります。

---



---

環境変数 `LD_LIBRARY_PATH` には、コンパイラ・ライブラリがインストールされているディレクトリを設定します。たとえば、コンパイラ・ライブラリが `/opt/COBOL/lib` ディレクトリにインストールされている場合は、次のコマンドを実行します。

- Bourne、Bash または Korn シェルの場合

```
$ LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/opt/COBOL/lib
$ export LD_LIBRARY_PATH
```

- C シェルの場合

```
% setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:/opt/COBOL/lib
```

### 6.4.1.2 Micro Focus Server Express COBOL コンパイラ

Micro Focus Server Express COBOL コンパイラを使用するには、環境変数 COBDIR と PATH および共有ライブラリ・パス環境変数を設定する必要があります。

**関連項目：** 共有ライブラリ・パス環境変数の詳細は、6-4 ページの「[クライアント共有ライブラリとクライアント静的ライブラリ](#)」の項を参照してください。

共有ライブラリ・パス環境変数に \$COBDIR/coblib ディレクトリが設定されていない場合は、プログラムのコンパイルまたは実行時に、次のようなエラー・メッセージが表示されます。

- Linux の場合

```
rtsora: error while loading shared libraries: libcobrts_t.so:
cannot open shared object file: No such file or directory
```

- Tru64 UNIX の場合

```
356835:rtsora: /sbin/loader: Fatal Error:
Cannot map library libcobrts64_t.so.2
```

- HP-UX PA-RISC および Solaris SPARC の場合

```
ld.so.1: rts32: fatal: libfhutil.so.2.0: Can't open file: errno=2
```

- AIX の場合

```
ld: rts32: fatal: libfhutil.so: Can't open file: errno=2
```

- HP-UX Itanium の場合

```
/usr/lib/hpux64/dld.so: Unable to find library 'libcobrts64_t.so.2'.
Killed
```

#### COBDIR

環境変数 COBDIR には、コンパイラがインストールされているディレクトリを設定します。たとえば、コンパイラが /opt/lib/cobol ディレクトリにインストールされている場合は、次のコマンドを実行します。

- Bourne、Bash または Korn シェルの場合

```
$ COBDIR=/opt/lib/cobol
$ export COBDIR
```

- C シェルの場合

```
% setenv COBDIR /opt/lib/cobol
```

#### PATH

\$COBDIR/bin ディレクトリを含むように環境変数 PATH を設定します。

- Bourne、Bash または Korn シェルの場合

```
$ PATH=$COBDIR/bin:$PATH
$ export PATH
```

- C シェルの場合

```
% setenv PATH ${COBDIR}/bin:${PATH}
```

**共有ライブラリ・パス**

環境変数 LIBPATH、LD\_LIBRARY\_PATH または SHLIB\_PATH には、コンパイラ・ライブラリがインストールされているディレクトリを設定します。たとえば、プラットフォームで環境変数 LD\_LIBRARY\_PATH が使用されており、コンパイラ・ライブラリが \$COBDIR/coblib ディレクトリにインストールされている場合は、次のコマンドを実行します。

- Bourne、Bash または Korn シェルの場合
 

```
$ LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:$COBDIR/coblib
$ export LD_LIBRARY_PATH
```
- C シェルの場合
 

```
% setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:$COBDIR/coblib
```

**6.4.2 Pro\*COBOL の Oracle ランタイム・システム**

Oracle では、動的読取り Pro\*COBOL プログラムを実行するために、rtsora (64 ビット・システムの 32 ビット COBOL コンパイラの場合は rtsora32) という専用のランタイム・システムを用意しています。動的読取り Pro\*COBOL プログラムを実行するには、cobrun ランタイム・システムのかわりに、rtsora (または rtsora32) ランタイム・システムを使用します。cobrun を使用して Pro\*COBOL プログラムを実行すると、次のようなエラー・メッセージが表示されます。

```
$ cobrun sample1.gnt
Load error : file 'SQLADR'
error code: 173, pc=0, call=1, seg=0
173      Called program file not found in drive/directory
```

**6.4.3 Pro\*COBOL のデモ・プログラム**

デモ・プログラムは、Pro\*COBOL プリコンパイラの機能を紹介するために用意されています。デモ・プログラムは、\$ORACLE\_HOME/precomp/demo/procob2 ディレクトリにあります。デフォルトでは、すべてのプログラムがクライアント共有ライブラリに動的にリンクされます。

デモ・プログラムを実行するには、\$ORACLE\_HOME/sqlplus/demo/demobld.sql スクリプトで作成したデモンストレーション表が SCOTT スキーマにあり、そのパスワードが TIGER である必要があります。

---

**注意：** デモンストレーションを作成する前に、SCOTT アカウントのロックを解除してパスワードを設定する必要があります。

---

次の Make ファイルを使用してデモ・プログラムを作成します。

```
$ORACLE_HOME/precomp/demo/procob2/demo_procob.mk
```

sample1 という Pro\*COBOL 用のデモ・プログラムをプリコンパイル、コンパイルおよびリンクするには、次のコマンドを実行します。

```
$ make -f demo_procob.mk sample1
```

Pro\*COBOL デモ・プログラムを作成するには、次のコマンドを実行します。

```
$ make -f demo_procob.mk samples
```

rtsora ランタイム・システムで使用する動的読取りプログラム sample1.gnt を作成し、実行するには、次のコマンドを実行します。

```
$ make -f demo_procob.mk sample1.gnt
$ rtsora sample1.gnt
```

一部のデモ・プログラムでは、`$ORACLE_HOME/precomp/demo/sql` ディレクトリ内にある SQL スクリプトを実行する必要があります。このスクリプトを実行しないと、実行を要求するメッセージが表示されます。

デモ・プログラムを作成し、それに対応する SQL スクリプトを実行するには、「make マクロ引数 `RUNSQL=run`」をコマンドに追加します。たとえば、`sample9` デモ・プログラムを作成し、必要な `$ORACLE_HOME/precomp/demo/sql/sample9.sql` スクリプトを実行するには、次のコマンドを実行します。

```
$ make -f demo_procob.mk sample9 RUNSQL=run
```

Pro\*COBOL デモ・プログラムを作成し、必要な SQL スクリプトすべてを実行するには、次のコマンドを実行します。

```
$ make -f demo_procob.mk samples RUNSQL=run
```

## 6.4.4 Pro\*COBOL のユーザー・プログラム

Make ファイル `$ORACLE_HOME/precomp/demo/procob2/demo_procob.mk` を使用して、ユーザー・プログラムを作成できます。この Make ファイルは、32 ビットまたは 64 ビットのユーザー・プログラムを作成します。Make ファイル `demo_procob_32.mk` を使用して、32 ビット（または 31 ビット）のユーザー・プログラムを作成することもできます。次の表に、Pro\*COBOL により 32 ビット（または 31 ビット）および 64 ビットのユーザー・プログラムを作成するための Make ファイルを示します。

プラットフォーム	64 ビット Make ファイル	32 ビット Make ファイル
AIX、HP-UX、Solaris SPARC および zSeries Linux	<code>demo_procob.mk</code>	<code>demo_procob_32.mk</code> <b>注意：</b> zSeries Linux の場合、この Make ファイルは 31 ビットのユーザー・プログラムを作成します。
Linux x86	該当なし	<code>demo_procob.mk</code>
Tru64 UNIX	<code>demo_procob.mk</code>	該当なし
Linux on POWER	<code>demo_procob.mk</code>	<code>demo_procob.mk</code> <b>注意：</b> <code>export COMPILER_MODE=32</code>

**関連項目：** ユーザー・プログラムの作成方法は、その Make ファイルを参照してください。

Make ファイル `demo_procob.mk` を使用してプログラムを作成する場合は、次のようなコマンドを実行します。

```
$ make -f demo_procob.mk target COBS="cobfile1 cobfile2 ..." EXE=exename
```

各項目の意味は次のとおりです。

- `target` は、使用する Make ファイルのターゲットです。
- `cobfilen` は、プログラムの COBOL ソース・ファイルです。
- `exename` は、実行可能プログラムです。

たとえば、プログラム `myprog` を作成する場合は、作成する実行可能ファイルのソースとタイプに応じて、次のいずれかのコマンドを実行します。

- COBOL ソースの場合、クライアント共有ライブラリに動的にリンクさせるには、次のコマンドを実行します。

```
$ make -f demo_procob.mk build COBS=myprog.cob EXE=myprog
```

- COBOL ソースの場合、静的にリンクさせるには、次のコマンドを実行します。  
\$ make -f demo\_procob.mk build\_static COBS=myprog.cob EXE=myprog
- COBOL ソースの場合、rtsora (32 ビットの COBOL コンパイラの場合は rtsora32) で使用するための動的読取りプログラムを作成するには、次のコマンドを実行します。  
\$ make -f demo\_procob.mk myprog.gnt

## 6.4.5 FORMAT プリコンパイラ・オプション

FORMAT プリコンパイラ・オプションは、COBOL の入力行の形式を指定します。デフォルト値の ANSI を指定した場合、列 1～6 はオプションの順序番号、列 7 はコメントまたは継続行を示します。段落名は列 8～11 で開始され、列 12～72 が文となります。

値 TERMINAL を指定した場合、列 1～6 は削除され、列 7 が左端の列になります。

## 6.5 Pro\*FORTRAN プリコンパイラ

---

**注意：** Linux または Mac OS X では、Pro\*FORTRAN はサポートされていません。

---

Pro\*FORTRAN プリコンパイラを使用する場合は、事前に適切なバージョンのコンパイラがインストールされていることを確認してください。この項では、次の項目について説明します。

- [Pro\\*FORTRAN のデモ・プログラム](#)
- [Pro\\*FORTRAN のユーザー・プログラム](#)

### 関連項目：

- サポートされるコンパイラのバージョンの詳細は、『Oracle Database インストール・ガイド』を参照してください。
- Pro\*FORTRAN プリコンパイラおよびインタフェース機能の詳細は、『Pro\*FORTRAN Supplement to the Oracle Precompilers Guide』を参照してください。

### 6.5.1 Pro\*FORTRAN のデモ・プログラム

デモ・プログラムは、Pro\*FORTRAN プリコンパイラの機能を紹介するために用意されています。デモ・プログラムはすべて、\$ORACLE\_HOME/precomp/demo/profor ディレクトリにあります。デフォルトでは、すべてのプログラムがクライアント共有ライブラリに動的にリンクされます。

デモ・プログラムを実行するには、\$ORACLE\_HOME/sqlplus/demo/demobld.sql スクリプトで作成したデモンストレーション表が SCOTT スキーマにあり、そのパスワードが TIGER である必要があります。

---

**注意：** デモンストレーションを作成する前に、SCOTT アカウントのロックを解除してパスワードを設定する必要があります。

---

デモ・プログラムを作成するには、\$ORACLE\_HOME/precomp/demo/profor ディレクトリにある Make ファイル demo\_profor.mk を使用します。たとえば、sample1 のデモ・プログラムをプリコンパイル、コンパイルおよびリンクするには、次のコマンドを実行します。

```
$ make -f demo_profor.mk sample1
```

Pro\*FORTRAN のデモ・プログラムを作成するには、次のコマンドを実行します。

```
$ make -f demo_profor.mk samples
```

一部のデモ・プログラムでは、\$ORACLE\_HOME/precomp/demo/sql ディレクトリ内にある SQL スクリプトを実行する必要があります。このスクリプトを実行しないと、実行を要求するメッセージが表示されます。

デモ・プログラムを作成し、それに対応する SQL スクリプトを実行するには、「make マクロ引数 RUNSQL=run」をコマンドラインに追加します。たとえば、sample11 デモ・プログラムを作成し、必要な \$ORACLE\_HOME/precomp/demo/sql/sample11.sql スクリプトを実行するには、次のコマンドを実行します。

```
$ make -f demo_profor.mk sample11 RUNSQL=run
```

Pro\*FORTRAN のデモ・プログラムを作成し、必要な SQL スクリプトすべてを実行するには、次のコマンドを実行します。

```
$ make -f demo_profor.mk samples RUNSQL=run
```

## 6.5.2 Pro\*FORTRAN のユーザー・プログラム

Make ファイル \$ORACLE\_HOME/precomp/demo/profor/demo\_profor.mk を使用して、ユーザー・プログラムを作成できます。この Make ファイルは、32 ビットまたは 64 ビットのユーザー・プログラムを作成します。Make ファイル demo\_profor\_32.mk を使用して、32 ビットのユーザー・プログラムを作成することもできます。次の表に、Pro\*FORTRAN により 32 ビットおよび 64 ビットのユーザー・プログラムを作成するための Make ファイルを示します。

プラットフォーム	64 ビット Make ファイル	32 ビット Make ファイル
AIX、HP-UX および Solaris SPARC	demo_profor.mk	demo_profor_32.mk
Tru64 UNIX	demo_profor.mk	該当なし

**関連項目：** ユーザー・プログラムの作成方法は、その Make ファイルを参照してください。

Make ファイル demo\_proc.mk を使用してプログラムを作成する場合は、次のようなコマンドを実行します。

```
$ make -f demo_profor.mk target FORS="forfile1 forfile2 ..." EXE=exename
```

各項目の意味は次のとおりです。

- `target` は、使用する Make ファイルのターゲットです。
- `forfilen` は、プログラムの FORTRAN ソース・ファイルです。
- `exename` は、実行可能プログラムです。

たとえば、Pro\*FORTRAN ソース・ファイル myprog.pfo からプログラム myprog を作成する場合は、作成する実行可能ファイルのタイプに応じて、次のいずれかのコマンドを実行します。

- 実行可能ファイルの場合、クライアント共有ライブラリに動的にリンクさせるには、次のコマンドを実行します。

```
$ make -f demo_profor.mk build FORS=myprog.f EXE=myprog
```

- 実行可能ファイルの場合、クライアント共有ライブラリに静的にリンクさせるには、次のコマンドを実行します。

```
$ make -f demo_profor.mk build_static FORS=myprog.f EXE=myprog
```

## 6.6 SQL\*Module for ADA

---



---

**注意：** この項の内容は、AIX プラットフォームに適用されます。

---



---

SQL\*Module for Ada を使用する場合は、事前に適切なバージョンのコンパイラがインストールされていることを確認してください。

**関連項目：**

- 必要なコンパイラのバージョンの詳細は、『Oracle Database インストール・ガイド』を参照してください。
- SQL\*Module for Ada の詳細は、『Oracle SQL\*Module for Ada Programmer's Guide』を参照してください。

### 6.6.1 SQL\*Module for Ada のデモ・プログラム

デモ・プログラムは、SQL\*Module for Ada の機能を紹介するために用意されています。デモ・プログラムはすべて、`$ORACLE_HOME/precomp/demo/modada` ディレクトリにあります。デフォルトでは、すべてのプログラムがクライアント共有ライブラリに動的にリンクされます。

`chl_drv` デモ・プログラムを実行するには、`$ORACLE_HOME/sqlplus/demo/demobld.sql` スクリプトで作成したデモンストレーション表が SCOTT スキーマにあり、そのパスワードが TIGER である必要があります。

---



---

**注意：** デモンストレーションを作成する前に、SCOTT アカウントのロックを解除してパスワードを設定する必要があります。

---



---

`demcalsp` および `demohost` の各デモ・プログラムでは、大学のサンプル・データベースが MODTEST スキーマに存在している必要があります。適切な `make` コマンドを使用して、MODTEST スキーマを作成し、大学のサンプル・データベースをロードできます。

SQL\*Module for Ada デモ・プログラムを作成して、MODTEST ユーザーの作成に必要な SQL スクリプトを実行し、大学のサンプル・データベースを作成するには、次のコマンドを実行します。

```
$ make -f demo_modada.mk all RUNSQL=run
```

デモ・プログラム (`demohost`) を 1 つのみ作成して、MODTEST ユーザーの作成に必要な SQL スクリプトを実行し、大学のサンプル・データベースを作成するには、次のコマンドを実行します。

```
$ make -f demo_modada.mk makeuser loaddb demohost RUNSQL=run
```

SQL\*Module for Ada デモ・プログラムを作成し、大学のサンプル・データベースを再作成しない場合は、次のコマンドを実行します。

```
$ make -f demo_modada.mk samples
```

デモ・プログラム (`demohost`) を 1 つのみ作成し、大学のサンプル・データベースを再作成しない場合は、次のコマンドを実行します。

```
$ make -f demo_modada.mk demohost
```

プログラムを実行するには、Oracle Net の接続文字列を定義するか、または適切な表が存在しているデータベースに接続できる `INST1_ALIAS` という別名を定義する必要があります。

## 6.6.2 SQL\*Module for Ada のユーザー・プログラム

Make ファイル `$ORACLE_HOME/precomp/demo/modada/demo_modada.mk` を使用して、ユーザー・プログラムを作成できます。Make ファイル `demo_modada.mk` を使用してユーザー・プログラムを作成する場合は、次のようなコマンドを実行します。

```
$ make -f demo_modada.mk ada OBJS="module1 module2 ..." \
EXE=exename MODARGS=SQL_Module_arguments
```

各項目の意味は次のとおりです。

- `modulen` は、コンパイル済の Ada オブジェクトです。
- `exename` は、実行可能プログラムです。
- `SQL_Module_arguments` は、SQL\*Module に渡されるコマンドライン引数です。

**関連項目：** SQL\*Module for Ada の詳細は、『Oracle SQL\*Module for Ada Programmer's Guide』を参照してください。

## 6.7 OCI と OCCI

Oracle Call Interface (OCI) または Oracle C++ Call Interface (OCCI) を使用する前に、適切なバージョンの C または C++ がインストール済であることを確認してください。

**関連項目：**

- サポートされるコンパイラのバージョンの詳細は、『Oracle Database インストラクション・ガイド』を参照してください。
- OCI と OCCI の詳細は、『Oracle Call Interface プログラマーズ・ガイド』または『Oracle C++ Call Interface プログラマーズ・ガイド』を参照してください。

### 6.7.1 OCI と OCCI のデモ・プログラム

デモ・プログラムは、ソフトウェアで提供される OCI と OCCI の機能を紹介するために用意されています。デモ・プログラムには、C および C++ プログラムの 2 種類があります。デモ・プログラムは、`$ORACLE_HOME/rdbms/demo` ディレクトリにあります。デフォルトでは、すべてのプログラムがクライアント共有ライブラリに動的にリンクされます。

デモ・プログラムを実行するには、`$ORACLE_HOME/sqlplus/demo/demobld.sql` スクリプトで作成したデモンストレーション表が SCOTT スキーマにあり、そのパスワードが TIGER であることが必要です。

---

**注意：** デモンストレーションを作成する前に、SCOTT アカウントのロックを解除してパスワードを設定する必要があります。

---

デモ・プログラムを作成するには、`$ORACLE_HOME/rdbms/demo` ディレクトリにある Make ファイル `demo_rdbms.mk` を使用します。たとえば、`cdemo1` のデモ・プログラムをコンパイルおよびリンクするには、次のコマンドを実行します。

```
$ make -f demo_rdbms.mk cdemo1
```

OCI の C デモ・プログラムを作成するには、次のコマンドを実行します。

```
$ make -f demo_rdbms.mk demos
```

OCCI の C++ デモ・プログラムを作成するには、次のコマンドを実行します。

```
$ make -f demo_rdbms.mk occidemos
```

## 6.7.2 OCI と OCCI のユーザー・プログラム

Make ファイル `$ORACLE_HOME/rdbms/demo/demo_rdbms.mk` を使用して、ユーザー・プログラムを作成できます。この Make ファイルは、32 ビットまたは 64 ビットのユーザー・プログラムを作成します。Make ファイル `demo_rdbms32.mk` を使用して、32 ビットのユーザー・プログラムを作成することもできます。次の表に、Pro\*FORTRAN により 32 ビットおよび 64 ビットのユーザー・プログラムを作成するための Make ファイルを示します。

プラットフォーム	64 ビット Make ファイル	32 ビット Make ファイル
AIX、HP-UX、Solaris SPARC、zSeries Linux および Mac OS X	<code>demo_rdbms.mk</code>	<code>demo_rdbms32.mk</code> <b>注意:</b> zSeries Linux の場合、この Make ファイルは 31 ビットのユーザー・プログラムを作成します。
Linux x86 および Solaris x86	該当なし	<code>demo_rdbms.mk</code>
Linux Itanium および Tru64 UNIX	<code>demo_rdbms.mk</code>	該当なし
Linux on POWER	<code>demo_rdbms.mk</code>	<code>demo_rdbms.mk</code> <b>注意:</b> <code>export COMPILER_MODE=32</code>

**関連項目:** ユーザー・プログラムの作成方法は、その Make ファイルを参照してください。

Make ファイル `demo_rdbms.mk` を使用してプログラムを作成する場合は、次のようなコマンドを実行します。

```
$ make -f demo_rdbms.mk target OBJS="objfile1 objfile2 ..." EXE=exename
```

この例の各項目の意味は、次のとおりです。

- `target` は、使用する Make ファイルのターゲットです。
- `objfilen` は、プログラムをリンクするためのオブジェクト・ファイルです。
- `exename` は、実行可能プログラムです。

たとえば、C ソース・ファイル `myprog.c` からプログラム `myprog` を作成する場合は、作成する実行可能ファイルのタイプに応じて、次のいずれかのコマンドを実行します。

- C ソースの場合、クライアント共有ライブラリに動的にリンクさせるには、次のコマンドを実行します。

```
$ make -f demo_rdbms.mk build OBJS=myprog.o EXE=myprog
```

- C ソースの場合、静的にリンクさせるには、次のコマンドを実行します。

```
$ make -f demo_rdbms.mk build_static OBJS=myprog.o EXE=myprog
```

たとえば、C++ ソース・ファイル `myprog.cpp` からプログラム `myprog` を作成する場合は、作成する実行可能ファイルのタイプに応じて、次のいずれかのコマンドを実行します。

- C++ ソースの場合、クライアント共有ライブラリに動的にリンクさせるには、次のコマンドを実行します。

```
$ make -f demo_rdbms.mk buildc++ OBJS=myprog.o EXE=myprog
```

- C++ ソースの場合、静的にリンクさせるには、次のコマンドを実行します。

```
$ make -f demo_rdbms.mk buildc++_static OBJS=myprog.o EXE=myprog
```

## 6.8 64 ビット・ドライバを使用する Oracle JDBC/OCI プログラム

---



---

### 注意:

- この項の内容は、Mac OS X、AIX、HP-UX、Linux on POWER、Solaris SPARC および zSeries Linux の各プラットフォームに適用されます。
  - この項で説明する指示と Make ファイルを使用して、64 ビット・ドライバを使用する JDBC/OCI ユーザー・プログラムを作成できます。
- 
- 

64 ビット・ドライバを使用する JDBC/OCI デモ・プログラムを実行するには、次の手順を実行します。

1. 次の各ファイルについて、環境変数 CLASSPATH の先頭に \$ORACLE\_HOME/jdbc/lib/ojdbc14.jar を追加します。

```
jdbc/demo/samples/jdbcoci/Makefile
jdbc/demo/samples/generic/Inheritance/Inheritance1/Makefile
jdbc/demo/samples/generic/Inheritance/Inheritance2/Makefile
jdbc/demo/samples/generic/Inheritance/Inheritance3/Makefile
jdbc/demo/samples/generic/JavaObject1/Makefile
jdbc/demo/samples/generic/NestedCollection/Makefile
```

2. \$ORACLE\_HOME/jdbc/demo/samples/generic/Makefile ファイル内で、JAVA 変数と JAVAC 変数を変更し、JDK の位置および -d64 フラグを次のように指定します。

```
JAVA=${ORACLE_HOME}/java/bin/java -d64
JAVAC=${ORACLE_HOME}/java/bin/javac -d64
```

Linux on POWER 上で JAVA 変数と JAVAC 変数を変更し、64 ビット JDK をインストールした JDK の位置を次のように指定します。

```
JAVA=/opt/IBMJava2-ppc64-142/bin/java
JAVAC=/opt/IBMJava2-ppc64-142/bin/javac
```

3. jdbc/demo/samples/generic/Makefile ファイル内で、JAVA と JAVAC が定義されている箇所を除き、JDK14\_HOME/bin/javac の出現箇所をすべて JAVAC で置き換え、JDK14\_HOME/bin/java の出現箇所をすべて JAVA で置き換えます。
4. \$ORACLE\_HOME/lib ディレクトリを含むように、環境変数 LD\_LIBRARY\_PATH\_64 を設定します。

---



---

**注意:** AIX での LIBPATH 変数は、LD\_LIBRARY\_PATH\_64 変数と同等です。AIX では、LD\_LIBRARY\_PATH\_64 変数ではなく、LIBPATH 変数を使用する必要があります。

Linux on POWER では、LD\_LIBRARY\_PATH\_64 変数ではなく LD\_LIBRARY\_PATH 変数を設定する必要があります。

Mac OS X では、LD\_LIBRARY\_PATH\_64 変数ではなく DYLD\_LIBRARY\_PATH 変数を設定する必要があります。

---



---

## 6.9 カスタム Make ファイル

この章の製品別の項で説明したように、ユーザー・プログラムを作成する場合は、オラクル社がソフトウェアとともに提供する Make ファイル `demo_product.mk` を使用してください。この Make ファイルを変更する場合、またはカスタム Make ファイルを使用する場合は、次の制限事項に注意してください。

- Oracle ライブラリの順番は変更しないでください。リンク中にすべてのシンボルが解決されるように、Oracle ライブラリはリンク・ラインに 2 回以上追加されます。  
AIX を除いて、Oracle ライブラリの順番がすべてのプラットフォームで重要である理由は、次のとおりです。
  - Oracle ライブラリは相互に参照し合います。たとえば、ライブラリ A の関数はライブラリ B の関数をコールし、ライブラリ B の関数はライブラリ A の関数をコールします。
  - HP-UX、Mac OS X および Tru64 UNIX の各リンカーは、1 パス・リンカーです。AIX、Linux および Solaris の各リンカーは、2 パス・リンカーです。
- リンク・ラインにライブラリを追加する場合は、リンク・ラインの最初または最後に追加します。Oracle ライブラリ間にユーザー・ライブラリを入れないでください。
- `nmake` または `GNU make` などの `make` ユーティリティを使用する場合は、マクロおよび接頭辞の処理について、オペレーティング・システムで提供されている `make` ユーティリティとの違いに注意してください。Oracle の Make ファイルは、`make` ユーティリティによってテストおよびサポートされます。
- Oracle ライブラリの名前および内容は、リリース間で変更されることがあります。必要なライブラリを判断するには、現行のリリースで提供されている Make ファイル `demo_product.mk` を必ず使用してください。

## 6.10 未定義シンボルの修正

オラクル社が提供している `symfind` ユーティリティを使用すると、シンボルが定義されているライブラリまたはオブジェクト・ファイルの場所を確認する際に役立ちます。プログラムのリンク時に、未定義シンボルは一般的なエラーの 1 つとみなされ、次のようなエラー・メッセージが生成されます。

```
$ make -f demo_proc.mk sample1
Undefined                          first referenced
   symbol                            in file
sqlcex                               sample1.o
sqlglm                               sample1.o
ld: fatal: Symbol referencing errors. No output written to sample1
```

このエラーは、参照するシンボルの定義をリンカーが検出できなかった場合に発生します。このエラー・メッセージが表示された場合は、シンボルが定義されているライブラリまたはオブジェクト・ファイルがリンク・ラインにあるかどうか、およびリンカーが検索しているファイルのディレクトリが正しいかどうかを確認します。

次の例は、`symfind` ユーティリティの出力です。このユーティリティを `sqlcex` シンボルの検索に使用しています。

```
$ symfind sqlcex

SymFind - Find Symbol <sqlcex> in <*>.a, .o, .so
-----
Command:          /u01/app/oracle/product/10.2.0/bin/symfind sqlcex
Local Directory:  /u01/app/oracle/product/10.2.0
Output File:      (none)
Note:             I do not traverse symbolic links
                  Use '-v' option to show any symbolic links
```



例 6-1 に、シグナル・ルーチンおよび受取りルーチンの設定方法を示します。

#### 例 6-1 シグナル・ルーチンおよび受取りルーチン

```
/* user side interrupt set */
word osnsui( /*_ word *handlep, void (*astp), char * ctx, _*/)
/*
** osnsui: Operating System dependent Network Set User-side Interrupt. Add an
** interrupt handling procedure astp. Whenever a user interrupt (such as a ^C)
** occurs, call astp with argument ctx. Put in *handlep handle for this
** handler so that it may be cleared with osncui. Note that there may be many
** handlers; each should be cleared using osncui. An error code is returned if
** an error occurs.
*/

/* user side interrupt clear */
word osncui( /*_ word handle _*/ );
/*
** osncui: Operating System dependent Clear User-side Interrupt. Clear the
** specified handler. The argument is the handle obtained from osnsui. An error
** code is returned if an error occurs.
*/
```

例 6-2 に、アプリケーション・プログラムでの osnsui() および osncui() ルーチンの使用方法を示します。

#### 例 6-2 osnsui() および osncui() ルーチンのテンプレート

```
/*
** User interrupt handler template.
*/
void sig_handler()
{
...
}

main(argc, argv)
int arc;
char **argv;
{

    int handle, err;
    ...

    /* Set up the user interrupt handler */
    if (err = osnsui(&handle, sig_handler, (char *) 0))
    {
        /* If the return value is nonzero, then an error has occurred
        Take appropriate action for the error. */
        ...
    }

    ...

    /* Clear the interrupt handler */
    if (err = osncui(handle))
    {
        /* If the return value is nonzero, then an error has occurred
        Take appropriate action for the error. */
        ...
    }
    ...
}
}
```

## 6.13 XA 機能

Oracle XA は、X/Open Distributed Transaction Processing (DTP) XA インタフェースの Oracle 実装です。XA 標準には、トランザクション内の共有リソースへのアクセスを制御するリソース・マネージャ間や、トランザクションを監視および解決するトランザクション・サービス間の双方向インタフェースが規定されています。

Oracle Call Interface には、XA 機能があります。TP モニター XA アプリケーションを作成するときは、TP モニター・ライブラリ (シンボル `ax_reg` および `ax_unreg` を定義するライブラリ) が、リンク・ラインで Oracle Client 共有ライブラリより前に設定されていることを確認してください。このリンク制限は、XA の動的登録 (Oracle XA スイッチ `xaoswd`) を使用する場合に必要です。

Oracle Database の XA コールは、クライアント共有ライブラリ (プラットフォームに応じて `libclntsh.a`、`libclntsh.sl`、`libclntsh.so` または `libclntsh.dylib`) およびクライアント静的ライブラリ (`libclntst10.a`) の両方で定義されています。これらのライブラリは、`$ORACLE_HOME/lib` ディレクトリにあります。



---

---

## SQL\*Loader および PL/SQL のデモ

この章では、Oracle Database とともに使用できる SQL\*Loader と PL/SQL の各デモ・プログラムを作成および実行する方法について説明します。次の項目について説明します。

- [SQL\\*Loader のデモ](#)
- [PL/SQL のデモ](#)
- [PL/SQL からの 32 ビット外部プロシージャのコール](#)

---

---

**注意：** この章で説明するデモンストレーションを使用するには、Oracle Database 10g Companion CD に収録されている Oracle Database Examples をインストールする必要があります。

また、デモンストレーションを作成する前に、SCOTT アカウントのロックを解除してパスワードを設定する必要があります。

---

---

## 7.1 SQL\*Loader のデモ

SQL\*Loader のデモを実行する場合は、`ulcase.sh` ファイルを実行します。デモを個別に実行する場合は、ファイル内に含まれている情報を読んで、実行方法を確認してください。

## 7.2 PL/SQL のデモ

PL/SQL には多数のデモ・プログラムが含まれています。これらのプログラムを使用する前に、データベース・オブジェクトを作成し、サンプル・データをロードする必要があります。オブジェクトを作成してサンプル・データをロードするには、次の手順を実行します。

1. ディレクトリを PL/SQL デモ・ディレクトリに変更します。
 

```
$ cd $ORACLE_HOME/plsql/demo
```
2. SQL\*Plus を起動し、SCOTT/TIGER として接続します。
 

```
$ sqlplus SCOTT/TIGER
```
3. 次のコマンドを実行し、オブジェクトを作成してサンプル・データをロードします。
 

```
SQL> @exampbld.sql
SQL> @examplod.sql
```

---

**注意：** デモは、十分な権限を持つ Oracle ユーザーとして作成してください。デモは、作成時と同じ Oracle ユーザーで実行してください。

---

### PL/SQL カーネル・デモ

次の PL/SQL カーネル・デモは、ソフトウェアとともに使用できます。

- `examp1.sql` ~ `examp8.sql`
- `examp11.sql` ~ `examp14.sql`
- `sample1.sql` ~ `sample4.sql`
- `extproc.sql`

PL/SQL カーネル・デモ `exampn.sql` または `samplen.sql` をコンパイルして実行するには、次の手順に従ってください。

1. SQL\*Plus を起動し、SCOTT/TIGER として接続します。
 

```
$ cd $ORACLE_HOME/plsql/demo
$ sqlplus SCOTT/TIGER
```
2. 次のようなコマンドを実行してデモを実行します。 `demo_name` はデモ名です。
 

```
SQL> @demo_name
```

`extproc.sql` デモを実行するには、次の手順を実行します。

1. 必要に応じて、次のように外部プロシージャのエントリを `tnsnames.ora` ファイルに追加します。

```
EXTPROC_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS=(PROTOCOL = IPC) ( KEY = EXTPROC))
    )
    (CONNECT_DATA =
      (SID = PLSExtProc)
    )
  )
```

- 必要に応じて、次のように外部プロシージャのエントリを listener.ora ファイルに追加します。

---

**注意：** listener.ora ファイルの SID\_NAME に指定する値と、tnsnames.ora ファイルの SID に指定する値は、一致している必要があります。

---

- HP-UX、Linux、Solaris および Tru64 UNIX の場合

```
SID_LIST_LISTENER =
(SID_LIST =
(SID_DESC=
(SID_NAME=PLSExtProc)
(ORACLE_HOME=oracle_home_path)
(ENVS=EXTPROC_DLLS=oracle_home_path/plsql/demo/extproc.so,
LD_LIBRARY_PATH=oracle_home_path/plsql/demo)
(PROGRAM=extproc)
)
)
)
```

- AIX の場合

```
SID_LIST_LISTENER =
(SID_LIST =
(SID_DESC=
(SID_NAME=PLSExtProc)
(ORACLE_HOME=oracle_home_path)
(ENVS=EXTPROC_DLLS=oracle_home_path/plsql/demo/extproc.so,
LIBPATH=oracle_home_path/plsql/demo)
(PROGRAM=extproc)
)
)
)
```

- Mac OS X の場合

```
SID_LIST_LISTENER =
(SID_LIST =
(SID_DESC=
(SID_NAME=PLSExtProc)
(ORACLE_HOME=oracle_home_path)
(ENVS=EXTPROC_DLLS=oracle_home_path/plsql/demo/extproc.dylib,
DYLD_LIBRARY_PATH=oracle_home_path/plsql/demo)
(PROGRAM=extproc)
)
)
)
```

- ディレクトリを \$ORACLE\_HOME/plsql/demo に変更します。
- 次のコマンドを実行して extproc.so 共有ライブラリを作成し、必要なデータベース・オブジェクトを構築してサンプル・データをロードします。

---

**注意：** Mac OS X システムでは、次の各例において共有ライブラリ名 extproc.dylib を使用してください。

---

```
$ make -f demo_plsql.mk extproc.so exampbld examplod
```

データベース・オブジェクトが構築済でサンプル・データがロードされている場合は、次のコマンドを実行します。

```
$ make -f demo_plsql.mk extproc.so
```

5. SQL\*Plus から、次のコマンドを実行します。

```
SQL> CONNECT SYSTEM/SYSTEM_password
SQL> GRANT CREATE LIBRARY TO SCOTT;
SQL> CONNECT SCOTT/TIGER
SQL> CREATE OR REPLACE LIBRARY demolib IS
  2  'oracle_home_path/plsql/demo/extproc.so';
  3  /
```

6. デモを起動するには、次のコマンドを実行します。

```
SQL> @extproc
```

### PL/SQL プリコンパイラ・デモ

---

---

**注意：** この項で示す make コマンドは、必要なデータベース・オブジェクトを作成し、サンプル・データを SCOTT スキーマにロードします。

---

---

次のプリコンパイラ・デモを利用できます。

- examp9.pc
- examp10.pc
- sample5.pc
- sample6.pc

PL/SQL プリコンパイラ・デモを作成するには、`$ORACLE_HOME/lib` ディレクトリを含むようにライブラリ・パス環境変数を設定し、次のコマンドを実行します。

```
$ cd $ORACLE_HOME/plsql/demo
$ make -f demo_plsql.mk demos
```

デモを1つのみ作成する場合は、make コマンドにそのデモ名を引数として指定します。たとえば、examp9 デモを作成するには、次のコマンドを実行します。

```
$ make -f demo_plsql.mk examp9
```

examp9 デモを起動するには、次のコマンドを実行します。

```
$ ./examp9
```

## 7.3 PL/SQL からの 32 ビット外部プロシージャのコール

---

**注意：** この項の内容は、AIX、HP-UX、Solaris SPARC、zSeries Linux および Apple Mac OS X (Intel) に適用されます。

---

64 ビットの外部プロシージャ実行可能ファイル (extproc) および 32 ビットの外部プロシージャ実行可能ファイル (extproc32) は、\$ORACLE\_HOME/bin ディレクトリにインストールされています。デフォルトでは、実行可能ファイル extproc を使用して、AIX、HP-UX、Solaris SPARC および zSeries Linux の各システムで、64 ビット外部プロシージャを実行できます。32 ビットの外部プロシージャを使用可能にするには、次の手順を実行します。

1. listener.ora ファイルに PROGRAM パラメータの値を次のように設定します。  
(PROGRAM=extproc32)
2. 使用しているプラットフォームに応じて、次のいずれかの環境変数に \$ORACLE\_HOME/lib32 ディレクトリを指定します。

プラットフォーム	環境変数
AIX	LIBPATH
HP-UX	SHLIB_PATH
Linux on POWER、zSeries Linux および Solaris	LD_LIBRARY_PATH
Apple Mac OS X (Intel)	DYLD_LIBRARY_PATH

3. リスナーを再起動します。

---

**注意：** 32 ビットまたは 64 ビットのいずれかの外部プロシージャを実行できるようにリスナーを構成できますが、同時に両方のプロシージャを実行することはできません。ただし、32 ビットと 64 ビットの外部プロシージャを両方ともサポートする必要がある場合は、2 つのリスナーを構成できます。

---



---

---

## Oracle Database のチューニング

この章では、Oracle Database のチューニング方法について説明します。次の項目について説明します。

- チューニングの重要性
- オペレーティング・システムのツール
- メモリー管理のチューニング
- ディスク I/O のチューニング
- ディスク・パフォーマンスの監視
- システム・グローバル領域
- オペレーティング・システムのバッファ・キャッシュのチューニング

## 8.1 チューニングの重要性

Oracle Database は、高度に最適化できるソフトウェア製品です。チューニングを頻繁に行うことで、システム・パフォーマンスが最適化され、データのボトルネックの発生を防ぐことができます。

データベースのチューニングを始める前に、8-2 ページの「オペレーティング・システムのツール」で説明するツールを使用して、通常の動作を監視する必要があります。

## 8.2 オペレーティング・システムのツール

データベースのパフォーマンスを評価し、データベース要件を決定できるオペレーティング・システム・ツールがいくつかあります。これらのツールは、Oracle プロセスの統計に加えて、システム全体の CPU 使用率、割込み、スワッピング、ページング、コンテキストのスイッチング、I/O についての統計情報も提供します。

この項では、次に示す共通のツールについて説明します。

- [vmstat](#)
- [sar](#)
- [iostat](#)
- [swap](#)、[swapinfo](#)、[swapon](#)、[lspcs](#)
- [AIX のツール](#)
- [HP-UX のツール](#)
- [Linux のツール](#)
- [Solaris のツール](#)
- [Mac OS X のツール](#)

**関連項目：** これらのツールの詳細は、オペレーティング・システムのドキュメントおよび man ページを参照してください。

### 8.2.1 vmstat

---

**注意：** Mac OS X では、`vm_stat` コマンドを使用すると、仮想メモリ情報が表示されます。このコマンドの使用方法は、`vm_stat` の man ページを参照してください。

---

プロセス、仮想メモリ、ディスク、トラップおよび CPU アクティビティを表示するときは、`vmstat` コマンドを使用します。表示内容はコマンドで切り替えます。CPU アクティビティのサマリーを 5 秒間隔で 6 回表示する場合は、次のいずれかのコマンドを実行します。

- HP-UX および Solaris の場合
 

```
$ vmstat -S 5 6
```
- AIX、Linux および Tru64 UNIX の場合
 

```
$ vmstat 5 6
```

次に、HP-UX でこのコマンドを実行した場合のサンプル出力を示します。

```
procs      memory          page            disk            faults          cpu
 r  b  w    swap  free  si  so  pi  po  fr  de  sr  f0  s0  s1  s3    in  sy    cs  us  sy  id
0  0  0    1892  5864   0   0  0  0  0  0  0  0  0  0  0   90  74   24  0  0  99
0  0  0    85356  8372   0   0  0  0  0  0  0  0  0  0  0   46  25   21  0  0  100
0  0  0    85356  8372   0   0  0  0  0  0  0  0  0  0  0   47  20   18  0  0  100
0  0  0    85356  8372   0   0  0  0  0  0  0  0  0  0  2   53  22   20  0  0  100
```

```
0 0 0 85356 8372 0 0 0 0 0 0 0 0 0 0 87 23 21 0 0 100
0 0 0 85356 8372 0 0 0 0 0 0 0 0 0 0 48 41 23 0 0 100
```

procs 列の下にある w サブ列は、スワップ・アウトされてディスクに書き込まれたプロセスの数を示します。値が 0 (ゼロ) 以外の場合は、スワッピングが発生してシステムがメモリー不足になっています。

page 列の下にある si 列および so 列は、それぞれ 1 秒当たりのスワップ・インとスワップ・アウトの回数を示します。スワップ・インとスワップ・アウトは、常に 0 (ゼロ) にしてください。

page 列の下にある sr 列は、スキャン率を示します。利用できるメモリーが不足すると、スキャン率が高くなります。

page 列の下にある pi 列および po 列は、それぞれ 1 秒当たりのページインとページアウトの回数を示します。ページインおよびページアウトの回数は通常、増加します。利用できるメモリーが十分にあるシステムでも、常に多少のページングは行われます。

---

**注意:** vmstat コマンドの出力は、プラットフォーム間で異なります。

---

**関連項目:** 出力の解釈については、man ページを参照してください。

## 8.2.2 sar

オペレーティング・システムのアクティビティ・カウンタの累計を表示するときは、sar (system activity reporter) コマンドを使用します。表示内容はコマンドで切り替えます。

---

**注意:** Tru64 UNIX システムでは、sar コマンドは、UNIX SVID2 互換サブセットの OSFSVID で利用できます。

---

HP-UX システムでは、次のコマンドは、I/O アクティビティのサマリーを 10 秒間隔で 10 回表示します。

```
$ sar -b 10 10
```

次に、このコマンドを実行した場合の出力例を示します。

```
13:32:45 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
13:32:55      0      14      100       3      10       69       0       0
13:33:05      0      12      100       4       4       5       0       0
13:33:15      0       1      100       0       0       0       0       0
13:33:25      0       1      100       0       0       0       0       0
13:33:35      0      17      100       5       6       7       0       0
13:33:45      0       1      100       0       0       0       0       0
13:33:55      0       9      100       2       8      80       0       0
13:34:05      0      10      100       4       4       5       0       0
13:34:15      0       7      100       2       2       0       0       0
13:34:25      0       0      100       0       0      100       0       0

Average      0       7      100       2       4      41       0       0
```

sar 出力は、ある時点におけるシステムの I/O アクティビティのスナップショットを提供します。複数のオプションを使用して時間隔を指定すると、出力の読取りができなくなることがあります。時間隔を 4 以下に指定すると、sar アクティビティ自体が出力に影響を与えることがあります。

**関連項目:** sar の詳細は、man ページを参照してください。

## 8.2.3 iostat

端末およびディスクのアクティビティを表示するときは、`iostat` コマンドを使用します。表示内容は、コマンドで切り替えます。`iostat` コマンドの出力には、ディスク要求キューは表示されず、ビジー状態のディスクが表示されます。この情報は、I/O 負荷のバランスを調整する場合に役立ちます。

次のコマンドは、端末およびディスク・アクティビティを 5 秒間隔で 5 回表示します。

```
$ iostat 5 5
```

次に、Solaris でこのコマンドを実行した場合のサンプル出力を示します。

```
tty          fd0          sd0          sd1          sd3          cpu
tin tout Kps tps serv  Kps tps serv  Kps tps serv  Kps tps serv  us sy wt id
0    1    0  0  0    0  0  0  31    0  0  18    3  0  42    0  0  0  99
0   16    0  0  0    0  0  0   0    0  0  0    1  0  14    0  0  0  100
0   16    0  0  0    0  0  0   0    0  0  0    0  0  0    0  0  0  100
0   16    0  0  0    0  0  0   0    0  0  0    0  0  0    0  0  0  100
0   16    0  0  0    0  0  0   2    0  14   12  2  47    0  0  1  98
```

大きなディスク要求キューを調べるときは、`iostat` コマンドを使用します。要求キューは、特定のディスク・デバイスに対する I/O 要求が実行されるまでにかかる時間を示します。要求キューが発生する原因は、特定のディスクに対する I/O 要求のボリュームが大きいこと、または I/O の平均シーク時間が長いことです。ディスク要求キューは、0 (ゼロ) またはそれに近い値であることが理想的です。

## 8.2.4 swap、swapinfo、swapon、lsp

---

**関連項目：** [Mac OS X システムのスワップ領域の詳細は、D-2 ページの「使用可能および使用済のスワップ領域の決定」を参照してください。](#)

---

スワップ領域の使用量に関する情報を表示するときは、`swap`、`swapinfo`、`swapon` または `lsp` の各コマンドを使用します。スワップ領域が不足するとプロセスの応答が停止し、Out of Memory エラーでプロセスが生成できなくなることがあります。次の表は、プラットフォームごとの適切なコマンドの一覧です。

プラットフォーム	コマンド
AIX	<code>lsp -a</code>
HP-UX	<code>swapinfo -m</code>
Linux および Tru64 UNIX	<code>swapon -s</code>
Solaris	<code>swap -l</code> および <code>swap -s</code>

次の例は、Solaris で `swap -l` コマンドを実行した場合のサンプル出力です。

```
swapfile      dev      swaplo blocks      free
/dev/dsk/c0t3d0s1  32,25    8      197592      162136
```

## 8.2.5 AIX のツール

次の各項では、AIX システムで利用できるツールについて説明します。

- [Base Operation System ツール](#)
- [Performance Toolbox](#)
- [System Management Interface Tool](#)

**関連項目：** これらのツールの詳細は、AIX のオペレーティング・システムのドキュメントおよび man ページを参照してください。

### 8.2.5.1 Base Operation System ツール

AIX の Base Operation System (BOS) は、UNIX システムに由来組み込まれていたパフォーマンス・ツールや、AIX の実装固有の機能を管理するパフォーマンス・ツールで構成されています。次の表は、最も重要な BOS ツールの一覧です。

ツール	機能
lsattr	デバイスの属性を表示します。
lslv	論理ボリューム、つまり物理ボリュームに対する論理ボリュームの割当てに関する情報を表示します。
netstat	ネットワーク関連のデータ構造の内容を表示します。
nfsstat	ネットワーク・ファイル・システム (NFS) とリモート・プロシージャ・コール (RPC) のアクティビティに関する統計を表示します。
nice	プロセスの初期優先順位を変更します。
no	ネットワーク・オプションを表示または設定します。
ps	1 つまたは複数のプロセスのステータスを表示します。
reorgvg	ボリューム・グループ内の物理パーティション割当てを再編成します。
time	経過した実行時間、ユーザーの CPU 処理時間およびシステムの CPU 処理時間を表示します。
trace	選択したシステム・イベントを記録および報告します。
vmo	Virtual Memory Manager のチューニング可能なパラメータを管理します。

### 8.2.5.2 Performance Toolbox

AIX Performance Toolbox (PTX) には、システム・アクティビティをローカルおよびリモートで監視およびチューニングするためのツールがいくつか含まれています。PTX は、PTX Manager と PTX Agent という 2 つのコンポーネントで主に構成されています。PTX Manager は、xmperf ユーティリティを使用して、構成内の様々なシステムからデータを収集し、表示します。PTX Agent は、xmserd デーモンを使用して、データを収集し PTX Manager に転送します。PTX Agent は、Performance Aide for AIX と呼ばれる製品として個別に利用することもできます。

PTX と Performance Aide の両方には、次の表に示す監視およびチューニング・ツールがあります。

ツール	説明
fdpr	特定の作業負荷にあわせて実行可能プログラムを最適化します。
filemon	トレース機能を使用して、ファイル・システムのアクティビティを監視および報告します。
fileplace	論理ボリュームまたは物理ボリューム内の特定のファイルについて、ブロックの配置を表示します。

ツール	説明
lockstat	カーネル・ロックの競合に関する統計を表示します。
lvedit	ボリューム・グループ内の論理ボリュームを対話方式で配置します。
netpmon	トレース機能を使用して、ネットワーク I/O およびネットワーク関連の CPU 使用率を報告します。
rmss	様々なサイズのメモリーでシステムをシミュレートし、パフォーマンスをテストします。
svmon	仮想メモリーの使用量に関する情報を取得し、分析します。
syscalls	システム・コールを記録し、カウントします。
tprof	trace 機能を使用して、モジュールとソース・コード文の各レベルで CPU 使用率を報告します。
BigFoot	プロセスのメモリー・アクセス・パターンを報告します。
stem	サブルーチン・レベルのエントリを許可し、既存の実行可能ファイルのインストルメント処理を終了します。

**関連項目：**

- 各ツールの詳細は、『Performance Toolbox for AIX Guide and Reference』を参照してください。
- 一部のツールの構文は、『AIX 5L Performance Management Guide』を参照してください。

**8.2.5.3 System Management Interface Tool**

AIX System Management Interface Tool (SMIT) は、様々なシステム管理およびパフォーマンス・ツールに対して、メニュー方式のインタフェースを提供します。SMIT を使用すると、実行するジョブを中心に様々なツールにナビゲートできます。

**8.2.6 HP-UX のツール**

次のパフォーマンス分析ツールは、HP-UX システムで使用できます。

- GlancePlus/UX

この HP-UX ユーティリティは、システムのアクティビティを測定するためのオンライン診断ツールです。GlancePlus は、システム・リソースの使用状況を表示します。つまり、システムの I/O、CPU およびメモリー使用量に関する動的な情報を一連の画面に表示します。このユーティリティを使用して、プロセスごとのリソースの使用状況を監視できます。

- HP PAK

HP Programmer's Analysis Kit (HP PAK) は次のツールで構成されています。

- Puma

このツールは、プログラムの実行時にパフォーマンス統計を収集します。収集した統計をグラフィカルに表示し、分析できるようにします。

- Thread Trace Visualizer (TTV)

このツールは、インストルメント処理スレッド・ライブラリ libpthread\_tr.sl によって作成されるトレース・ファイルをグラフィカルに表示します。このツールを使用すると、スレッドの相互作用を表示したり、スレッドがブロックされてリソースを待機している箇所を調べることができます。

HP PAK は、HP Fortran 77、HP Fortran 90、HP C、HP C++、HP ANSI C++ および HP Pascal の各コンパイラに組み込まれています。

次の表は、HP-UX でパフォーマンス・チューニングの追加に使用できるパフォーマンス・チューニング・ツールの一覧です。

ツール	機能
caliper (Itanium のみ)	高速の動的インスツルメント処理とともに、キャッシュ・ミス、Translation Look-aside Buffer (TLB)、命令サイクルなどのシステム分析タスク用にランタイム・アプリケーション・データを収集します。C、C++、Fortran およびアセンブリ・アプリケーション用の動的パフォーマンス測定ツールです。
gprof	プログラムの実行プロファイルを作成します。
monitor	プログラム・カウンタを監視し、特定の関数をコールします。
netfmt	ネットワークを監視します。
netstat	ネットワーク・パフォーマンスに関する統計を報告します。
nfsstat	ネットワーク・ファイル・システム (NFS) とリモート・プロシージャ・コール (RPC) のアクティビティに関する統計を表示します。
nettl	ロギングとトレースによって、ネットワーク・イベントまたはパケットを取得します。
prof	C プログラムの実行プロファイルを作成してプログラムのパフォーマンス統計を表示し、そのプログラムが実行時間の大半を消費している箇所を示します。
profil	プログラム・カウンタ情報をバッファにコピーします。
top	システムの上位プロセスを表示し、その情報を定期的に更新します。

## 8.2.7 Linux のツール

Linux システムでは、top、free および cat /proc/meminfo コマンドを使用して、スワップ領域、メモリーおよびバッファの使用量を表示します。

## 8.2.8 Solaris のツール

Solaris システムでは、mpstat コマンドを使用して、マルチプロセッサ・システムのプロセッサごとの統計を表示します。表の各行は、1 プロセッサのアクティビティを示します。1 行目は、起動してからのすべてのアクティビティをまとめて表示します。後続の各行は、時間隔でのアクティビティをまとめて表示します。特に指定しないかぎり、すべての値は 1 秒当たりのイベント数です。引数は、統計および反復回数の時間隔を示します。

次の例は、Solaris で mpstat コマンドを実行した場合のサンプル出力です。

```
CPU minf mjf xcal intr ithr csw icsw migr smtx srw syscl usr sys wt idl
  0  0  0  1    71  21  23   0  0  0  0  55  0  0  0  99
  2  0  0  1    71  21  22   0  0  0  0  54  0  0  0  99
CPU minf mjf xcal intr ithr csw icsw migr smtx srw syscl usr sys wt idl
  0  0  0  0    61  16  25   0  0  0  0  57  0  0  0  100
  2  1  0  0    72  16  24   0  0  0  0  59  0  0  0  100
```

## 8.2.9 Mac OS X のツール

次のパフォーマンス・チューニング・ツールを追加して使用できます。

- top コマンドを使用して、実行中のプロセスおよびメモリー使用量を表示します。
- Shark や BigTop などの Apple Computer Hardware Understanding Developer (CHUD) ツールを使用して、システム・アクティビティの監視およびアプリケーションのチューニングをします。

**関連項目：** CHUD ツールの詳細は、次のサイトを参照してください。

<http://developer.apple.com/documentation/Performance/Conceptual/PerformanceOverview/Introduction/Introduction.html/>

## 8.3 メモリー管理のチューニング

メモリー・チューニング・プロセスでは、最初にページングおよびスワッピング領域を測定して、使用可能なメモリー量を確認します。システムのメモリー使用量の確認後、Oracle バッファ・キャッシュをチューニングします。

Oracle バッファ・マネージャによって、アクセス頻度の最も高いデータをキャッシュに長く保存できます。バッファ・マネージャを監視し、バッファ・キャッシュをチューニングすると、Oracle Database のパフォーマンスが大幅に向上することがあります。各システムの Oracle Database バッファ・サイズの最適値は、システム全体の負荷や他のアプリケーションと比較した場合の Oracle Database の優先順位によって異なります。

この項では、次の項目について説明します。

- [十分なスワップ領域の割当て](#)
- [ページングの制御](#)
- [Oracle ブロック・サイズの調整](#)

### 8.3.1 十分なスワップ領域の割当て

スワッピングは、オペレーティング・システムのオーバーヘッドに大きく影響するため、最小限に抑える必要があります。スワッピングが行われているかどうかを調べるには、`sar` コマンドまたは `vmstat` コマンドを使用します。この 2 つのコマンドで使用するオプションについては、`man` ページを参照してください。

システムでスワッピングが行われている場合は、メモリーを節約するために、次の処理を行います。

- 必要以上にシステム・デーモン・プロセスまたはアプリケーション・プロセスを実行しないようにします。
- データベース・バッファの数を減らし、一部のメモリーを解放します。
- オペレーティング・システム・ファイル・バッファの数を減らします（特に RAW デバイスを使用する場合）。

---

**注意：** Mac OS X システムでは、スワップ領域は動的に割り当てられます。オペレーティング・システムでスワップ領域の追加が必要な場合は、`/private/var/vm` ディレクトリに追加スワップ・ファイルが作成されます。このディレクトリが含まれるファイル・システムに、追加スワップ・ファイルに対応できるだけの十分な空きディスク領域があることを確認してください。スワップ領域の割当て方法の詳細は、D-2 ページの「[使用可能および使用済のスワップ領域の決定](#)」を参照してください。

---

スワップ領域の使用量を確認するには、使用しているプラットフォームに応じて、次のいずれかのコマンドを実行します。

プラットフォーム	コマンド
AIX	<code>lspv -a</code>
HP-UX	<code>swapinfo -m</code>
Linux	<code>swapon -s</code>
Solaris	<code>swap -l</code> および <code>swap -s</code>
Tru64 UNIX	<code>swapon -s</code>

スワップ領域をシステムに追加するには、使用しているプラットフォームに応じて、次のいずれかのコマンドを実行します。

プラットフォーム	コマンド
AIX	chps または mkps
HP-UX	swapon
Linux	swapon -a
Solaris	swap -a
Tru64 UNIX	swapon -a

スワップ領域は、物理メモリーの2～4倍に設定してください。スワップ領域の使用量を監視し、必要に応じて値を大きくしてください。

**関連項目：** これらのコマンドの詳細は、オペレーティング・システムのドキュメントを参照してください。

### 8.3.2 ページングの制御

プログラムを実行するためにプログラム全体をメモリーに格納しておく必要はないため、ページングはスワッピングほど深刻な問題ではありません。少量のページアウトでは、システムのパフォーマンスにほとんど影響はありません。

大量のページングを検出するには、高速応答時またはアイドル時の測定値と、低速応答時の測定値を比較します。

ページングを監視するには、vmstat (Mac OS X では vm\_stat) または sar コマンドを使用します。

**関連項目：** プラットフォームの監視結果の解釈については、man ページまたはオペレーティング・システムのドキュメントを参照してください。

次の表は、これらのコマンドの出力から得られる重要な列の一覧です。

プラットフォーム	列	機能
Solaris	vf1t/s	アドレス変換ページ・フォルトの数を示します。アドレス変換フォルトは、プロセスが、メモリー内にない有効ページを参照したときに発生します。
Solaris	rclm/s	ページアウト・アクティビティによって再生され、空きリストに追加された有効ページ数を示します。この値は0 (ゼロ) である必要があります。
HP-UX	at	アドレス変換ページ・フォルトの数を示します。アドレス変換フォルトは、プロセスが、メモリー内にない有効ページを参照したときに発生します。
HP-UX	re	ページアウト・アクティビティによって再生され、空きリストに追加された有効ページ数を示します。この値は0 (ゼロ) である必要があります。

システムで大量のページアウト・アクティビティが常に発生している場合は、次の方法で解決してください。

- メモリーを増設します。
- 一部の作業を別のシステムに移します。
- システム・グローバル領域 (SGA) で使用するメモリーを少なく設定します。

### 8.3.3 Oracle ブロック・サイズの調整

読取り操作時には、オペレーティング・システムのブロック全体がディスクから読み取られます。データベースのブロック・サイズが、オペレーティング・システムのファイル・システムのブロック・サイズより小さい場合は、I/O 帯域幅の効率が悪くなります。Oracle Database のブロック・サイズをファイル・システムのブロック・サイズの倍数になるように設定すると、パフォーマンスを最大 5 パーセント向上させることができます。

データベースのブロック・サイズは、DB\_BLOCK\_SIZE 初期化パラメータで設定します。ただし、このパラメータの値を変更するには、データベースを再作成する必要があります。

DB\_BLOCK\_SIZE パラメータの現在の設定値を調べるには、SQL\*Plus の SHOW PARAMETER DB\_BLOCK\_SIZE コマンドを実行します。

## 8.4 ディスク I/O のチューニング

使用可能なディスク全体で I/O を均等に分散して、ディスクへのアクセス時間が短くなるようにしてください。小規模なデータベースや RAID を使用しないデータベースでは、それぞれのデータファイルと表領域を使用可能なディスク間に分散してください。

### 8.4.1 自動ストレージ管理の使用

データベース記憶域に自動ストレージ管理を使用すると、すべてのデータベース I/O が、自動ストレージ管理ディスク・グループ内の使用可能なすべてのディスク・デバイス間に分散されます。自動ストレージ管理では、RAW デバイスを管理する煩わしさがなく、RAW デバイス I/O のパフォーマンスが向上します。

自動ストレージ管理を使用することで、ディスク I/O を手動でチューニングする必要がなくなります。

### 8.4.2 適切なファイル・システム・タイプの選択

使用するオペレーティング・システムに応じて、いくつかのファイル・システム・タイプから選択できます。ファイル・システム・タイプごとにそれぞれ特性が異なります。このことが、データベースのパフォーマンスに大きな影響を与えます。次の表に、一般的なファイル・システム・タイプを示します。

ファイル・システム	プラットフォーム	説明
S5	HP-UX および Solaris	UNIX System V ファイル・システム
UFS	AIX、HP-UX、Mac OS X、Solaris、Tru64 UNIX	Unified ファイル・システム (BSD UNIX から派生) <b>注意:</b> Mac OS X システムでは、ソフトウェアおよびデータベース・ファイルのいずれにも UFS ファイルの使用はお勧めしません。
VxFS	AIX、HP-UX および Solaris	VERITAS ファイル・システム
なし	すべて	RAW デバイス (ファイル・システムなし)
ext2/ext3	Linux	Linux 用拡張ファイル・システム
OCFS	Linux	Oracle Cluster ファイル・システム
AdvFS	Tru64 UNIX	Advanced ファイル・システム
CFS	Tru64 UNIX	クラスタ・ファイル・システム
JFS/JFS2	AIX	ジャーナル・ファイル・システム

ファイル・システム	プラットフォーム	説明
HFS Plus、HFSX	Mac OS X	HFS Plus は、Mac OS X により使用される標準の階層ファイル・システムです。HFSX は HFS Plus の拡張で、大 / 小文字を区別したファイル名を使用できます。
GPFS	AIX	一般的なパラレル・ファイル・システム

ファイル・システムとアプリケーションには、必ずしも互換性があるとはかぎりません。たとえば、Unified ファイル・システムの各実装間でさえ、互換性の比較は容易ではありません。選択したファイル・システムによって、パフォーマンスに最大 20 パーセントの開きが出る場合があります。ファイル・システムを使用する場合は、次のことを行ってください。

- ハードディスクがクリーンで断片化されないように、ファイル・システムのパーティションを新しく作成します。
- データベース・ファイルに対してファイル・システムを使用する前に、パーティションでファイル・システム・チェックを行います。
- ディスク I/O をできるだけ均等に分散します。
- 論理ボリューム・マネージャまたは RAID デバイスを使用していない場合、データファイルとは異なるファイル・システムにログ・ファイルを格納することを検討してください。

## 8.5 ディスク・パフォーマンスの監視

次の項では、ディスク・パフォーマンスの監視方法について説明します。

### Mac OS X でのディスク・パフォーマンスの監視

ディスク・パフォーマンスを監視するには、`iostat` コマンドおよび `sar` コマンドを使用します。これらのコマンドの使用方法は、`man` ページを参照してください。

### その他のオペレーティング・システムでのディスク・パフォーマンスの監視

ディスク・パフォーマンスを監視するには、`sar -b` および `sar -u` コマンドを使用します。

次の表に、`sar -b` コマンド出力の列をいくつか示します。これらの列は、ディスク・パフォーマンスの分析に重要です。

列	説明
<code>bread/s</code> 、 <code>bwrit/s</code>	1 秒ごとに読み取られるブロック数と書き込まれるブロック数 (ファイル・システム・データベースに重要)
<code>pread/s</code> 、 <code>pwrit/s</code>	1 秒ごとに読み取られるパーティション数と書き込まれるパーティション数 (RAW パーティション・データベース・システムに重要)

ディスク・パフォーマンスの分析に重要な `sar -u` 列の 1 つに、`%wio` があります。これによって、ブロックされた I/O で待機する CPU 時間の割合がわかります。

---

**注意：** 一部の Linux では、`sar -u` コマンドの出力結果に `%wio` 列が表示されません。詳細な I/O 統計を表示するには、`iostat -x` コマンドを使用できます。

---

キー・インジケータは次のとおりです。

- `bread`、`bwrit`、`pread` および `pwrit` 列の値の合計は、ディスク I/O サブシステムのアクティビティのレベルを示します。合計値が大きいほど、I/O サブシステムはビジーになります。物理ドライブの数が多いほど、合計のしきい値が高くなる可能性があります。デフォルトの最適値は、ドライブ 2 個の場合は 40 以下、ドライブ 4～8 個の場合は 60 以下です。
- `%rcache` 列の値は 91 以上、`%wcache` 列の値は 61 以上である必要があります。それ以外の場合は、システムがディスク I/O バウンドになる可能性があります。
- `%wio` 列の値が常に 21 以上の場合、システムは I/O バウンドになります。

## 8.6 システム・グローバル領域

SGA とは、共有メモリーに格納されている Oracle 構造体のことです。この構造体には、静的データ構造体、ロックおよびデータ・バッファが含まれています。各 Oracle プロセスが SGA 全体をアドレス指定するためには、十分な共有メモリーが必要です。

1 つの共有メモリー・セグメントの最大サイズは、`shmmax` (Tru64 UNIX では `shm_max`) カーネル・パラメータで指定します。

次の表に、このパラメータの推奨値をプラットフォームごとに示します。

プラットフォーム	推奨値
AIX	該当なし
HP-UX	システムに搭載された物理メモリーのサイズ。 <b>関連項目:</b> HP-UX の <code>shmmax</code> パラメータの詳細は、B-2 ページの「Oracle インスタンス用の HP-UX 共有メモリー・セグメント」を参照してください。
Linux	システムに搭載された物理メモリーの半分のサイズ。
Mac OS X	システムに搭載された物理メモリーの半分のサイズ。
Solaris および Tru64 UNIX	4294967295 または 4GB マイナス 16MB。 <b>注意:</b> Oracle Database インスタンスを起動する場合は、 <code>shm_max</code> パラメータの値を 16MB 以上にする必要があります。システムで Oracle9i Database と Oracle Database 10g の両方のインスタンスが動作している場合は、このパラメータの値を 2GB マイナス 16MB に設定する必要があります。Solaris の場合、この値は、64 ビットのシステムでは 4GB より大きくすることも可能です。

SGA のサイズが共有メモリー・セグメントの最大サイズ (`shmmax` または `shm_max`) を超える場合、Oracle Database では、要求された SGA サイズになるように、連続したセグメントが連結されます。`shmseg` カーネル・パラメータ (Tru64 UNIX では `shm_seg`) には、任意のプロセスで連結できるセグメントの最大数を指定します。SGA のサイズを制御するには、次の初期化パラメータを設定します。

- `DB_CACHE_SIZE`
- `DB_BLOCK_SIZE`
- `JAVA_POOL_SIZE`
- `LARGE_POOL_SIZE`
- `LOG_BUFFERS`
- `SHARED_POOL_SIZE`

または、SGA サイズを自動的にチューニングできるように、`SGA_TARGET` 初期化パラメータを設定します。

これらのパラメータの値は、十分注意して設定してください。値を大きく設定しすぎると、物理メモリーに対する共有メモリーの割合が大きくなりすぎます。そのため、パフォーマンスが低下します。

共有サーバーで構成されている Oracle Database では、SHARED\_POOL\_SIZE 初期化パラメータの値を大きく設定するか、LARGE\_POOL\_SIZE 初期化パラメータを使用したカスタム構成が必要です。Oracle Universal Installer を使用してデータベースをインストールした場合、SHARED\_POOL\_SIZE パラメータの値は、Oracle Database Configuration Assistant によって自動的に設定されます。ただし、データベースを手動で作成した場合は、パラメータ・ファイルで SHARED\_POOL\_SIZE パラメータの値を同時ユーザーごとに 1KB ずつ増やしてください。

## 8.6.1 SGA サイズの確認

次のいずれかの方法で、SGA サイズを確認できます。

- 次の SQL\*Plus コマンドを実行して、実行中のデータベースの SGA サイズを表示します。

```
SQL> SHOW SGA
```

結果はバイト単位で表示されます。

- データベース・インスタンスを起動すると、SGA がシステム・グローバル領域の合計ヘッダーの横に表示されます。
- Mac OS X 以外のシステムでは、oracle ユーザーとして ipcs コマンドを実行します。

## 8.6.2 AIX での共有メモリー

---

**注意：** この項の内容は、AIX のみに適用されます。

---

共有メモリーとしてプロセス間で共通の仮想メモリー・リソースが使用されます。各プロセスは、パフォーマンスを向上させるために、表やキャッシュ・エントリなどの共通の仮想メモリー変換リソース・セットを介して仮想メモリー・セグメントを共有します。

ページングを回避し、I/O のオーバーヘッドを減らすために共有メモリーを確保できます。これを実行するには、LOCK\_SGA パラメータを true に設定します。AIX 5L の場合、基礎となるハードウェアでラージ・ページ機能がサポートされている場合、同じパラメータによってその機能がアクティブになります。

確保されたメモリーを Oracle Database で利用できるようにするには、次のコマンドを実行します。

```
$ /usr/sbin/vmo -r -o v_pinshm=1
```

確保されたメモリーに利用できる実メモリーの最大量を設定するには、次のようなコマンドを実行します。percent\_of\_real\_memory は、設定する実メモリーの最大量（パーセント）です。

```
$ /usr/sbin/vmo -r -o maxpin%=percent_of_real_memory
```

maxpin% オプションを使用しているときは、確保されたメモリーの量が Oracle SGA のサイズに対して、システムの実メモリーの 3 パーセント以上であることが重要です。これにより、確保可能な空きメモリーをカーネルが使用できます。たとえば、物理メモリーが 2GB であり、SGA を 400MB (RAM の 20 パーセント) 確保する場合は、次のコマンドを実行します。

```
$ /usr/sbin/vmo -r -o maxpin%=23
```

システムの操作時に確保されたメモリーの使用量を監視するには、svmon コマンドを使用します。Oracle Database は、LOCK\_SGA パラメータが true に設定されている場合にのみメモリーを確保しようとします。

**AIX POWER4-Based Systems および POWER5-Based Systems でのラージ・ページ機能**

POWER4 または POWER5 システム上でサイズがそれぞれ 16MB のラージ・ページを 10 件オンにして予約するには、次のコマンドを実行します。

```
$ /usr/sbin/vmo -r -o lgpg_regions=10 -o lgpg_size=16777216
```

このコマンドは、bosboot を提案し、変更を有効にするには再起動する必要があることを示す警告を表示します。

SGA 全体を格納できるラージ・ページを指定することをお勧めします。LOCK\_SGA パラメータが true に設定されている場合、Oracle Database インスタンスではラージ・ページを割り当てようとします。SGA のサイズが確保可能なメモリー・サイズまたはラージ・ページのサイズを超える場合は、これらのサイズを超える SGA 部分が通常の共有メモリーに割り当てられます。

**関連項目：** 確保されたメモリーやラージ・ページの有効化およびチューニングについては、AIX のドキュメントを参照してください。

## 8.7 オペレーティング・システムのバッファ・キャッシュのチューニング

RAW デバイスを最大限に活用するには、Oracle Database バッファ・キャッシュのサイズを調整します。メモリーに制限がある場合は、オペレーティング・システムのバッファ・キャッシュも調整します。

オペレーティング・システムのバッファ・キャッシュには、メモリーからディスクまたはディスクからメモリーへの転送中に、メモリー内のデータ・ブロックが保持されます。

Oracle Database バッファ・キャッシュは、Oracle Database バッファを格納するためのメモリー内の領域です。Oracle Database では RAW デバイスを使用できるため、オペレーティング・システムのバッファ・キャッシュは使用しません。

RAW デバイスを使用する場合は、Oracle Database バッファ・キャッシュのサイズを大きくします。システムで使用できるメモリーに制限がある場合は、それに応じてオペレーティング・システムのバッファ・キャッシュのサイズを小さくします。

調整するバッファ・キャッシュを判断するには、sar コマンドを使用します。

**関連項目：** sar コマンドの詳細は、Tru64 UNIX の man ページを参照してください。

---

**注意：** Tru64 UNIX の場合は、オペレーティング・システムのバッファ・キャッシュを小さくしないでください。これは、オペレーティング・システムでは、ファイル・システム I/O のバッファリングに必要なメモリー量が自動的にサイズ変更されるためです。オペレーティング・システムのバッファ・キャッシュを制限すると、パフォーマンスの問題が発生する可能性があります。

---

---

# AIX システムでの Oracle Database の管理

この付録では、AIX システムで Oracle Database を管理する方法について説明します。次の項目について説明します。

- [メモリーとページング](#)
- [ディスク I/O の問題](#)
- [CPU のスケジューリングおよびプロセスの優先順位](#)
- [環境変数 AIXTHREAD\\_SCOPE の設定](#)
- [ネットワーク情報サービス \(NIS\) の外部ネーミングのサポート](#)
- [AIX 5.3 システムでのマルチスレッド同時処理 \(SMT\)](#)

## A.1 メモリーとページング

メモリーの競合は、プロセスに必要なメモリー量が、利用できる容量よりも大きくなったときに発生します。このようなメモリー不足に対処するために、メモリーとディスクとの間でプログラムやデータのページングが行われます。

この項では、次の項目について説明します。

- バッファ・キャッシュのページング・アクティビティの制御
- AIX ファイル・バッファ・キャッシュのチューニング
- 十分なページング領域の割当て
- ページングの制御
- データベース・ブロック・サイズの設定
- ログ・アーカイブ・バッファのチューニング
- I/O バッファおよび SQL\*Loader

### A.1.1 バッファ・キャッシュのページング・アクティビティの制御

ページング・アクティビティが過剰に行われると、パフォーマンスが大幅に低下します。ジャーナル・ファイル・システム (JFS および JFS2) 上に作成されたデータベース・ファイルでは、このページングが問題になることがあります。この状況では、大量の SGA データ・バッファ内に、参照頻度の最も高いデータを含む、似たようなジャーナル・ファイル・システムのバッファが存在する可能性があります。AIX ファイル・バッファ・キャッシュ・マネージャの動作は、パフォーマンスに大きく影響します。この動作によっては、I/O ボトルネックが発生し、システム全体のスループットが低下する可能性があります。

バッファ・キャッシュ・ページング・アクティビティをチューニングできますが、回数を絞って十分注意して行う必要があります。/usr/sbin/vmo コマンドを使用して、次の表の AIX システム・パラメータをチューニングします。

パラメータ	説明
minfree	空きリスト・サイズの最小値。バッファ内の空きリストの容量がこのサイズを下回ると、ページ・スティーリングによって空きリストが補充されます。
maxfree	空きリスト・サイズの最大値。バッファ内の空きリストの容量がこのサイズを上回ると、ページ・スティーリングによる空きリストの補充が中止されます。
minperm	ファイル I/O の永続バッファ・ページの最小数。
maxperm	ファイル I/O の永続バッファ・ページの最大数。

**関連項目：** AIX システム・パラメータの詳細は、『AIX 5L Performance Management Guide』を参照してください。

## A.1.2 AIX ファイル・バッファ・キャッシュのチューニング

AIX ファイル・バッファ・キャッシュの目的は、ジャーナル・ファイル・システム使用時のディスクへのアクセス頻度を減少させることです。このキャッシュが小さすぎると、ディスク使用率が増加して、1つ以上のディスクが一杯になることがあります。このキャッシュが大きすぎると、メモリーが無駄になります。

AIX ファイル・バッファ・キャッシュは、`minperm` および `maxperm` パラメータの調整によって構成できます。一般に、`sar -b` コマンドで指定するバッファ・ヒット率が低い (90 パーセント未満) の場合は、`minperm` パラメータ値を増やします。バッファ・ヒット率を高くする必要がない場合は、`minperm` パラメータ値を減らし、利用できる物理メモリーを増やします。AIX ファイル・バッファ・キャッシュのサイズの増加については、AIX のドキュメントを参照してください。

パフォーマンスの向上は、マルチプログラミングの量や作業負荷の I/O 特性によって変化するため、簡単には数値化できません。

### `minperm` および `maxperm` パラメータのチューニング

AIX では、ファイルに使用されるページ・フレームと、計算可能 (作業用またはプログラム・テキスト) セグメントに使用されるページ・フレームとの比率を、大まかに制御できます。次のガイドラインに従って、`minperm` および `maxperm` の値を調整してください。

- 実メモリー内でのファイル・ページの占有率が `minperm` 値を下回る場合は、再ページ率に関係なく、**Virtual Memory Manager (VMM)** のページ置換アルゴリズムによって、ファイル・ページおよび計算可能ページが取得されます。
- 実メモリー内でのファイル・ページの占有率が `maxperm` 値を上回る場合は、VMM のページ置換アルゴリズムによって、ファイル・ページおよび計算可能ページが取得されます。
- 実メモリー内でのファイル・ページの占有率がパラメータ値 `minperm` と `maxperm` の間にある場合は、VMM によって、通常はファイル・ページのみが取得されます。ただし、ファイル・ページの再ページ率が計算可能ページの再ページ率よりも大きい場合は、計算可能ページも同様に取得されます。

デフォルト値を計算するには、次のアルゴリズムを使用します。

- `minperm` (ページ数) = ((ページ・フレームの数) - 1024) × 0.2
- `maxperm` (ページ数) = ((ページ・フレームの数) - 1024) × 0.8

`minperm` パラメータの値を総ページ・フレーム数の 5 パーセントに変更し、`maxperm` パラメータの値を総ページ・フレーム数の 20 パーセントに変更するには、次のコマンドを使用します。

```
# /usr/sbin/vmo -o minperm%=5 -o maxperm%=20
```

デフォルト値は、それぞれ 20 パーセントと 80 パーセントです。

新しいデータベース接続をオープンする際の即時応答を最適化するには、`minfree` パラメータを調節して、システムが空きリストにページを追加せずにアプリケーションをメモリーにロードできるように、十分な空きページを確保します。プロセスの実メモリー・サイズ (常駐セット・サイズ、作業用セット) を判断するには、次のコマンドを使用します。

```
$ ps v process_id
```

この値または 8 フレームのいずれか大きい値を `minfree` パラメータに設定します。

データベース・ファイルが RAW デバイス上にある場合、またはダイレクト I/O を使用している場合は、`minperm` および `maxperm` パラメータを低い値に設定できます。たとえば、それぞれ 5 パーセントと 20 パーセントに設定します。これは、RAW デバイスまたはダイレクト I/O では、AIX ファイル・バッファ・キャッシュが使用されないためです。メモリーは、Oracle システム・グローバル領域など、他の目的に有効に利用できます。

### A.1.3 十分なページング領域の割当て

ページング領域（スワップ領域）が十分に割り当てられていないと、システムの応答が停止したり、応答時間が非常に遅くなります。AIX では、RAW ディスク・パーティションにページング領域を動的に追加できます。設定するページング領域の大きさは、実装されている物理メモリーの量およびアプリケーションのページング領域要件によって異なります。ページング領域の使用量を監視するには、`lspgs` コマンドを使用します。システムのページング・アクティビティを監視するには、`vmstat` コマンドを使用します。ページング領域を増やすには、`smittpgsp` コマンドを使用します。

ページング領域が事前に割り当てられる場合は、ページング領域を RAM の量よりも大きい値に設定することをお勧めします。AIX では、ページング領域は必要になるまで割り当てられません。システムでは、実メモリーが不足した場合のみスワップ領域が使用されます。メモリーのサイズを正しく設定した場合は、ページングが行われなため、ページング領域を小さくできます。要求されるページ数が大幅に増減しない場合は、ページング領域が小さくてもシステムは適切に動作します。ページングが大幅に増加する可能性がある場合は、その最大ページ数を処理できるページング領域が必要になります。

一般に、ページング領域の初期設定は、32GB を上限として、RAM の半分のサイズに 4GB を加えた値です。`lspgs -a` コマンドを使用してページング領域を監視し、その結果に従ってページング領域のサイズを増減します。`lspgs -a` コマンドによって出力される使用率 (%Used) は、適正なシステムの場合は通常 25 パーセント未満になります。メモリー・サイズが適切に割り当てられている場合、ページング領域はほとんど必要ありません。また、スワッピングが過剰に発生する場合は、システムに対して RAM サイズが小さすぎる可能性があります。

---

---

**注意：** ページング領域のサイズは小さくしないでください。サイズを小さくしたことによって領域が不足すると、アクティブなプロセスが終了します。一方ページング領域が大きすぎたとしても、マイナスの影響はほとんどありません。

---

---

### A.1.4 ページングの制御

過剰なページングが頻繁に発生する場合は、実メモリーが不足していることを意味します。通常は、次のように対処してください。

- ページングが頻繁に発生しないようにします。または、システムに高速の拡張記憶域を装備し、メモリーと拡張記憶域間のページングが、SGA とディスク間のデータの読取り / 書き込み速度よりも十分に速くなるようにします。
- 制限されたメモリー・リソースを、システム・パフォーマンスが最も向上する場所に割り当てます。場合によっては、メモリー・リソース要件とその影響のバランスを調整するために、この処理を繰り返し行う必要があります。
- メモリーが不足している場合は、メモリーを必要とするシステム内のプロセスと要素を優先順に並べたリストを作成します。次に、パフォーマンスが最も向上する場所にメモリーを割り当てます。優先順リストの例を次に示します。
  1. OS および RDBMS のカーネル
  2. ユーザー・プロセスおよびアプリケーション・プロセス
  3. REDO ログ・バッファ
  4. PGA および共有プール
  5. データベース・ブロック・バッファ・キャッシュ

たとえば、Oracle Database の動的パフォーマンス表およびビューを問い合わせた結果、共有プールおよびデータベース・バッファ・キャッシュにメモリーを追加する必要があるとします。この場合、制限された予備メモリーは、データベース・ブロック・バッファ・キャッシュではなく共有プールに割り当てるとパフォーマンスが向上します。

次の AIX コマンドを実行すると、ページングの状況および統計が表示されます。

- `vmstat -s`
- `vmstat interval [repeats]`
- `sar -r interval [repeats]`

## A.1.5 データベース・ブロック・サイズの設定

Oracle Database のブロック・サイズを設定することによって、I/O スループットを改善できます。AIX では、`DB_BLOCK_SIZE` 初期化パラメータの値を 2 ~ 32KB に設定できます。デフォルト値は 4KB です。Oracle Database がジャーナル・ファイル・システムにインストールされている場合は、そのブロック・サイズをファイル・システムのブロック・サイズ (JFS では 4KB、GPFS では 16KB ~ 1MB) の倍数にする必要があります。データベースが RAW パーティション上にある場合は、Oracle Database のブロック・サイズをオペレーティング・システムの物理ブロック・サイズ (AIX では 512 バイト) の倍数にします。

Oracle Database のブロック・サイズは、オンライン・トランザクション処理 (OLTP) または複合作業負荷の環境では小さめ (2 または 4KB) に設定し、意思決定支援システム (DSS) 作業負荷環境では大きめ (8、16 または 32KB) に設定することをお勧めします。

## A.1.6 ログ・アーカイブ・バッファのチューニング

トランザクションが長い場合やトランザクションの数が多い場合は特に、`LOG_BUFFER` サイズを増やすことで、データベースのアーカイブ速度を向上させることができます。ログ・ファイル I/O アクティビティおよびシステム・スループットを監視して、最適な `LOG_BUFFER` サイズを判断します。`LOG_BUFFER` パラメータをチューニングするときは、通常データベース・アクティビティの全体的なパフォーマンスが低下しないように注意してください。

---

---

**注意：** `LOG_ARCHIVE_BUFFER_SIZE` パラメータは、Oracle8i Database で廃止されました。

---

---

## A.1.7 I/O バッファおよび SQL\*Loader

SQL\*Loader ディレクト・パス・オプションを使用しながらデータを並行してロードする場合など、データを高速にロードするときは、CPU 時間の大半が I/O 完了の待ち時間として使用されます。バッファの数を増やすことにより、CPU 使用率が最大になり、スループット全体も向上します。

選択するバッファの数 (SQL\*Loader の `BUFFERS` パラメータで設定) は、使用可能なメモリー量や CPU 使用率を最大化する程度によって異なります。

パフォーマンスの向上は、CPU 使用率やデータのロード時に使用する並列度によって変化します。

**関連項目：** `BUFFERS` パラメータのファイル処理オプション文字列の調整および SQL\*Loader ユーティリティについては、『Oracle Database ユーティリティ』を参照してください。

### インポート・ユーティリティ用の BUFFER パラメータ

インポート・ユーティリティ用の `BUFFER` パラメータには、高速ネットワークのパフォーマンスを最適化するために、大きい値を設定する必要があります。たとえば、IBM RS/6000 Scalable POWERparallel Systems (SP) スイッチを使用する場合は、`BUFFER` パラメータの値を 1MB 以上に設定する必要があります。

## A.2 ディスク I/O の問題

ディスク I/O の競合は、メモリー管理が良好でない場合（その結果として発生するページングおよびスワッピングを含む）や、ディスク間の表領域とファイルの配分が適切でない場合に発生します。

filemon、sar、iostat などの AIX ユーティリティや、その他のパフォーマンス・ツールを使用して、I/O 負荷が高いディスクを識別し、すべてのディスク・ドライブに I/O 負荷を均等に分散します。

### A.2.1 AIX 論理ボリューム・マネージャ

AIX 論理ボリューム・マネージャ (LVM) は、複数のディスクにデータをストライプ化して、ディスクの競合を軽減できます。ストライプ化の主な目的は、大容量の順次ファイルに対する読取り / 書込みのパフォーマンスを向上させることです。LVM のストライプ機能を効果的に使用すると、ディスク間に I/O を均等に分散できるため、全体的なパフォーマンスが向上します。

---

**注意：** 自動ストレージ管理ディスク・グループには論理ボリュームを追加しないでください。自動ストレージ管理は、ディスク・グループに RAW ディスク・デバイスが追加された場合に最も効果を発揮します。自動ストレージ管理を使用している場合、LVM によるストライプ化は行わないでください。自動ストレージ管理は、ストライプ化およびミラー化を実装します。

---

#### ストライプ化された論理ボリュームの設計

ストライプ化された論理ボリュームを定義するときは、次の表に示す項目を指定する必要があります。

項目	推奨される設定値
ドライブ	2 つ以上の物理ドライブが必要です。パフォーマンスが重視される順次 I/O を実行するときは、2 つ以上の物理ドライブのアクティビティを最小にする必要があります。場合によっては、複数のアダプタ間で論理ボリュームをストライプ化する必要があります。
ストライプ・ユニット・サイズ	ストライプ・ユニット・サイズには、2 の累乗（2 ~ 128KB の範囲）を指定できます。ただし、ほとんどの作業負荷には、32KB と 64KB のストライプ・サイズで十分です。Oracle Database ファイルでは、ストライプ・サイズをデータベース・ブロック・サイズの倍数にする必要があります。
サイズ	論理ボリュームに割り当てる物理パーティションの数は、使用するディスク・ドライブ数の倍数にする必要があります。
属性	ミラー化することはできません。copies 属性の値を 1 に設定します。

#### その他の考慮事項

LVM を使用したときのパフォーマンスの向上度は、使用する LVM や作業負荷の特性によって大きく異なります。DSS 作業負荷では、パフォーマンスが大幅に向上します。OLTP タイプまたは複合作業負荷の場合も、かなりのパフォーマンスの向上を期待できます。

## A.2.2 ジャーナル・ファイル・システムを使用した場合と RAW 論理ボリュームを使用した場合の相違

ジャーナル・ファイル・システムまたは RAW 論理ボリュームを使用するかどうかを決定する場合は、次のことを考慮してください。

- ファイル・システムは、実装の多様化に伴い、継続的に改善されています。場合によっては、RAW デバイスよりもファイル・システムを使用したほうが I/O パフォーマンスが向上することがあります。
- ファイル・システムには追加構成 (AIX の `minservers` および `maxservers` パラメータ) が必要であり、ファイル・システムの非同期 I/O がカーネルの外部で実行されるため、CPU オーバーヘッドが少し増加します。
- 様々なベンダーが、各ディスクの特長を生かすために、様々な方法でファイル・システム・レイヤーを実装しています。この結果、プラットフォーム間でのファイル・システムの比較が難しくなっています。
- 強力な LVM インタフェースを導入すると、RAW 論理ボリュームに基づく論理ディスクの設定やバックアップ作業が大幅に減少します。
- AIX 5L に組み込まれているダイレクト I/O およびコンカレント I/O 機能によって、ファイル・システムのパフォーマンスは、RAW 論理ボリュームに匹敵するレベルまで向上します。

ジャーナル・ファイル・システムを使用した場合は、RAW デバイスを使用した場合と比較して、データベース・ファイルの管理や保守が容易になります。以前のバージョンの AIX では、ファイル・システムはバッファに対する読取り / 書込みのみサポートしており、`inode` ロックが不完全なために余計な競合が発生していました。この 2 つの問題は、JFS2 コンカレント I/O 機能および GPFS ダイレクト I/O 機能によって解決されており、最高のパフォーマンスが必要な場合でも、RAW デバイスのかわりにファイル・システムを使用できます。

---

**注意:** RAC オプションを使用するには、ASM ディスク・グループ内の RAW デバイスまたは GPFS ファイル・システムにデータファイルを配置する必要があります。JFS または JFS2 は使用できません。GPFS を使用すると、ダイレクト I/O が暗黙的に有効になります。

---

### ファイル・システム・オプション

AIX 5L では、ダイレクト I/O およびコンカレント I/O がサポートされています。ダイレクト I/O およびコンカレント I/O のサポートによって、データベース・ファイルがファイル・システム上に存在できるようになります。これは、Oracle Database が提供する機能を使用して、オペレーティング・システムのバッファ・キャッシュを回避し、冗長な `inode` ロック操作を排除することによって実現されます。

Oracle のデータファイルが含まれるファイル・システムでは、コンカレント I/O またはダイレクト I/O をできるかぎり有効にすることをお勧めします。次の表に、AIX で使用可能なファイル・システムとその推奨設定を示します。

ファイル・システム	オプション	説明
JFS	<code>dio</code>	JFS ではコンカレント I/O は使用できません。ダイレクト I/O を使用できます。ただし、コンカレント I/O を使用した JFS2 に比べてパフォーマンスは劣ります。
JFS ラージ・ファイル	<code>none</code>	128KB の位置合せ制約によってダイレクト I/O の使用が回避されるため、サイズの大きい JFS ファイルを Oracle Database に使用することはお勧めしません。

ファイル・システム	オプション	説明
JFS2	cio	<p>コンカレント I/O は同一のファイルに対して複数のコンカレント・リーダー / ライターをサポートしているため、JFS2 では、ダイレクト I/O よりもコンカレント I/O を設定するほうが有効です。ただし、JFS2/CIO に対する AIX 制限のため、コンカレント I/O は、Oracle データファイル、制御ファイルおよびログ・ファイルでのみ使用されます。コンカレント I/O は、このような目的専用のファイル・システムにのみ適用してください。同じ理由から、cio オプションでマウントする JFS2 ファイル・システムでは、Oracle ホーム・ディレクトリはサポートされません。たとえば、インストール時に、Oracle ホーム・ディレクトリを cio オプションでマウントする JFS2 ファイル・システムに配置するように誤って指定した場合は、Oracle に再リンクしようとする、次のエラーが表示される場合があります。</p> <pre>ld: 0711-866 INTERNAL ERROR: Output symbol table size miscalculated</pre>
GPFS	該当なし	<p>Oracle Database は、最適なパフォーマンスとなるように GPFS に対してダイレクト I/O を暗黙的に有効にします。GPFS のダイレクト I/O は、すでに複数のノード上の複数のリーダー / ライターをサポートしています。したがって、GPFS の場合、ダイレクト I/O とコンカレント I/O は同じです。</p>

### JFS および JFS2 の考慮事項

JFS2 ファイル・システムに Oracle Database ログを置いている場合、agblksize=512 オプションを使用してファイル・システムを作成し、cio オプションでこれをマウントすると最適な構成となります。これによって、ロギング・パフォーマンスが RAW デバイスのパフォーマンスの数パーセント以内に抑えられます。

Oracle Database 10g より前のバージョンでは、JFS/JFS2 においてファイル・レベルでダイレクト I/O およびコンカレント I/O を有効にできませんでした。したがって、最適なパフォーマンスを得るために、Oracle ホーム・ディレクトリおよびデータファイルを独立したファイル・システムに配置する必要がありました。つまり、Oracle ホーム・ディレクトリをデフォルト・オプションでマウントしたファイル・システムに配置し、データファイルおよびログを dio または cio オプションを使用して、マウントしたファイル・システムに配置する必要がありました。

Oracle Database 10g では、JFS/JFS2 において、ダイレクト I/O およびコンカレント I/O をファイル・レベルで有効にできます。そのためには、サーバー・パラメータ・ファイルの FILESYSTEMIO\_OPTIONS パラメータを setall または directIO に設定します。これにより、すべてのデータファイル I/O について、JFS2 でのコンカレント I/O と JFS でのダイレクト I/O が有効になります。これは、directIO 設定により、通常は使用されない非同期 I/O が無効になるためです。この 10g の機能によって、Oracle ホーム・ディレクトリと同じ JFS/JFS2 ファイル・システムにデータファイルを配置し、ダイレクト I/O またはコンカレント I/O を使用してパフォーマンスを改善できます。前述のように、最適なパフォーマンスを得るには、Oracle Database ログを独立した JFS2 ファイル・システムに配置する必要があります。

### GPFS の考慮事項

GPFS を使用している場合は、すべての目的に同じファイル・システムを使用できます。Oracle ホーム・ディレクトリや、データファイルとログの格納などに使用できます。最適なパフォーマンスを得るためには、大規模な GPFS ブロック・サイズ (通常は 512KB 以上) を使用してください。GPFS はスケーラビリティを確保するように設計されており、データ量が単一の GPFS ファイル・システムに収まるかぎり、複数の GPFS ファイル・システムを作成する必要はありません。

**ジャーナル・ファイル・システムから RAW 論理ボリュームへの移動**

すべてのデータを手動で再ロードせずに、ジャーナル・ファイル・システムから RAW デバイスにデータを移動するには、次の手順を root ユーザーで実行します。

1. 新規 RAW 論理ボリューム・デバイス・タイプ (-T O) を使用して、RAW デバイス (BigVG を推奨) を作成します。これにより、最初の Oracle ブロックをオフセット 0 (ゼロ) にでき、最適なパフォーマンスを得ることができます。

```
# mklv -T O -y new_raw_device VolumeGroup NumberOfPartitions
```

---

**注意：** この新規 RAW デバイスは、既存のファイルより大きくする必要があります。また、新規 RAW デバイスのサイズが領域を無駄にしないように注意する必要があります。

---

2. RAW デバイスに対する権限を設定します。
3. 次のように、dd を使用して JFS ファイルの内容を変換し、新規 RAW デバイスにコピーします。

```
# dd if=old_JFS_file of=new_raw_device bs=1m
```

4. データファイルの名前を変更します。

**RAW 論理ボリュームからジャーナル・ファイル・システムへの移動**

RAW 論理ボリュームの最初の Oracle ブロックは、必ずしもオフセット 0 (ゼロ) ではありません。ただし、ファイル・システム上の最初の Oracle ブロックは常にオフセット 0 (ゼロ) です。オフセットを決定し、RAW 論理ボリュームの最初のブロックを検索するには、\$ORACLE\_HOME/bin/offset コマンドを使用します。オフセットは、AIX 論理ボリュームでは 4096 バイトまたは 128KB となり、mklv -T O オプションにより作成された AIX 論理ボリュームでは 0 (ゼロ) となります。

オフセットの決定後、dd コマンドを使用し、オフセットをスキップして、RAW 論理ボリュームからファイル・システムにデータをコピーできます。次の例では、オフセットを 4096 バイトと仮定しています。

```
# dd if=old_raw_device bs=4k skip=1|dd of=new_file bs=256
```

RAW 論理ボリュームの最大容量よりも小さいブロック数を使用するように、Oracle Database に指示できます。このように指示する場合、count 句を追加して、必ず Oracle ブロックが含まれるデータのみをコピーする必要があります。次の例では、オフセットが 4096 バイト、Oracle ブロック・サイズが 8KB、ブロック数が 150000 であると仮定しています。

```
# dd if=old_raw_device bs=4k skip=1|dd bs=8k count=150000|dd of=new_file bs=256k
```

## A.2.3 非同期 I/O の使用

Oracle Database では、AIX が提供する非同期 I/O (AIO) を最大限に利用して、データベース・アクセスの高速化を図っています。

AIX 5L は、ファイル・システム・パーティションおよび RAW デバイスで作成されたデータベース・ファイルに対して、非同期 I/O (AIO) をサポートしています。RAW デバイスに対する AIO は、AIX カーネル内に完全に実装されるため、サーバーのプロセスで AIO 要求を処理する必要はありません。ファイル・システムに対して AIO を使用するときは、要求がキューから取り出されてから処理が完了するまで、カーネル・サーバー・プロセス (aioserver) が各要求を制御します。aioserver サーバーの数によって、システムで同時に処理できる AIO 要求の数が決定します。このため、Oracle Database のデータファイルの格納にファイル・システムを使用する場合には、aioserver プロセスの数をチューニングすることが重要です。

サーバー数を設定するには、次のいずれかのコマンドを使用します。このコマンドは、RAW デバイスではなくファイル・システムで非同期 I/O を使用している場合에만適用されます。

- `smit aio`
- `chdev -l aio0 -a maxservers='m' -a minservers='n'`

### 関連項目：

- システム管理インタフェース・ツール (SMIT) の詳細は、SMIT のオンライン・ヘルプを参照してください。
- `smit aio` および `chdev` の各コマンドの詳細は、man ページを参照してください。

---

**注意：** AIX 5L リリース 5.2 以降は、2 つの AIO サブシステムを使用できません。Oracle Database 10g では、Oracle プリインストール・スクリプトにより LEGACY AIO (aio0) および POSIX AIO (posix\_aio0) が有効になる場合にも、LEGACY AIO (aio0) が使用されます。どちらの AIO サブシステムの場合もパフォーマンス特性は同じです。

---

最小値には、システムの起動時に起動するサーバーの数を指定します。最大値には、多数の同時要求に応答するために起動できるサーバーの数を指定します。これらのパラメータは、ファイル・システムにのみ適用されます。RAW デバイスには適用されません。

サーバーの最小数のデフォルト値は 1 です。最大数のデフォルト値は 10 です。カーネル化された AIO を使用していない場合、CPU を 4 個以上搭載する大規模なシステムで Oracle Database を実行するには、通常、これらの値では低すぎます。次の表に示されている値に設定することをお勧めします。

パラメータ	値
minservers	初期値としては、システム上の CPU の数または 10 のうち、いずれか小さい値を設定することをお勧めします。
maxservers	AIX 5L リリース 5.2 からは、このパラメータで CPU ごとの AIO サーバーの最大数がカウントされるようになりました。以前のバージョンの AIX では、システム全体の値がカウントされていました。GPFS を使用している場合は、maxservers を、CPU の数で割った worker1threads に設定します。これが最適な設定です。maxservers の値を大きくしても、I/O のパフォーマンスは向上しません。  JFS または JFS2 を使用している場合、初期値を、10 × 論理ディスク数 ÷ CPU 数に設定します。pstat または ps コマンドを使用して、典型的な作業負荷で起動された aioservers の実際の数を監視します。実際にアクティブな AIO サーバー数が maxservers と等しい場合は、maxservers の値を増加します。

パラメータ	値
maxreqs	初期値を、4 × 論理ディスク数 × キューの深さに設定します。キューの深さは次のコマンドを実行することによって判断できます。  \$ lsattr -E -l hdiskxx  通常、キューの深さは3です。

maxservers または maxreqs パラメータの値が低すぎると、次の警告メッセージが繰り返し表示される場合があります。

```
Warning: lio_listio returned EAGAIN
Performance degradation may be seen.
```

これらのエラーが表示されないようにするには、maxservers パラメータの値を大きくします。稼動している AIO サーバーの数を表示するには、次のコマンドを root ユーザーで入力します。

```
# pstat -a | grep -c aios
# ps -k | grep aioserver
```

アクティブな AIO サーバーの数を定期的にチェックし、必要に応じて minservers および maxservers パラメータの値を変更してください。パラメータの変更は、システムが再起動されるときに有効になります。

## A.2.4 I/O スレーブ

I/O スレーブは、I/O のみを実行する特別な Oracle プロセスです。非同期 I/O がデフォルトであり、AIX に対する I/O 操作は Oracle での実行が推奨されている方法であるため、I/O スレーブは AIX ではほとんど使用されません。I/O スレーブは、共有メモリー・バッファから割り当てられます。I/O スレーブには、次の表に示す初期化パラメータを使用します。

パラメータ	許容値	デフォルト値
DISK_ASYNC_IO	true/false	true
TAPE_ASYNC_IO	true/false	true
BACKUP_TAPE_IO_SLAVES	true/false	false
DBWR_IO_SLAVES	0-999	0
DB_WRITER_PROCESSES	1-20	1

通常、この表のパラメータは調整しません。ただし、作業負荷が大きい場合に、データベース・ライターがボトルネックになることがあります。その場合は、DB\_WRITER\_PROCESSES の値を大きくします。このデータベース・ライター・プロセスの数は、システムまたはパーティション内で CPU のペアにつき 1 つですが、原則として、この数は増やさないでください。

デバッグのためにオラクル社カスタマ・サポート・センターにより指示された場合などは、非同期 I/O の無効化が必要な場合があります。DISK\_ASYNC\_IO および TAPE\_ASYNC\_IO の各パラメータを使用して、ディスクまたはテープ・デバイスに対する非同期 I/O を無効にできます。各プロセス・タイプの I/O スレーブ数のデフォルトは 0 (ゼロ) であるため、デフォルトでは I/O スレーブは割り当てられません。

DBWR\_IO\_SLAVES パラメータは、DISK\_ASYNC\_IO または TAPE\_ASYNC\_IO パラメータが false の場合にのみ、0 (ゼロ) より大きい値に設定します。設定しないと、データベース・ライター・プロセス (DBWR) がボトルネックになります。この場合、AIX での DBWR\_IO\_SLAVES パラメータの最適値は 4 です。

## A.2.5 DB\_FILE\_MULTIBLOCK\_READ\_COUNT パラメータの使用

Oracle Database 10g でダイレクト I/O またはコンカレント I/O を使用している場合、AIX ファイル・システムでは順次スキャンでの先読みは実行されません。このため、ダイレクト I/O またはコンカレント I/O が Oracle データファイルで有効な場合、サーバー・パラメータ・ファイルの DB\_FILE\_MULTIBLOCK\_READ\_COUNT の値を増やす必要があります。DB\_FILE\_MULTIBLOCK\_READ\_COUNT 初期化パラメータの指定どおり、Oracle Database により先読みが実行されます。

DB\_FILE\_MULTIBLOCK\_READ\_COUNT 初期化パラメータの値を大きくすると、通常は順次スキャンでの I/O スループットが向上します。AIX では、このパラメータの範囲は 1 ~ 512 ですが、16 を超える値を使用しても、通常はそれ以上のパフォーマンス効果は得られません。

このパラメータの値は、DB\_BLOCK\_SIZE パラメータの値との積が LVM ストライプ・サイズよりも大きくなるように設定します。このように設定することによって、使用できるディスク数が増加します。

## A.2.6 後書きの使用

後書き機能を使用すると、オペレーティング・システムで書込み I/O をパーティションのサイズまでグループ化できるようになります。これによって、I/O 処理数が少なくなるため、パフォーマンスが向上します。ファイル・システムでは、各ファイルが複数の 16KB パーティションに分割されるため、書込みパフォーマンスが向上し、メモリー内の使用済ページ数が制限され、ディスクの断片化が最小限に抑えられます。特定のパーティションのページは、プログラムによって次の 16KB パーティションの 1 バイト目が書き込まれたときに、ディスクに書き込まれます。後書き用バッファのサイズを 8 個の 16KB パーティションに設定するには、次のコマンドを入力します。

```
à# /usr/sbin/vmo -o numclust=8
```

後書き機能を無効にするには、次のコマンドを入力します。

```
à# /usr/sbin/vmo -o numclust=0
```

## A.2.7 順次先読みのチューニング

---

**注意：** この項の内容は、ダイレクト I/O もコンカレント I/O も使用していないファイル・システムにのみ適用されます。

---

VMM は、順次ファイルのページの必要性を予測します。また、プロセスがファイルにアクセスするパターンを監視しています。プロセスがファイルの 2 つのページに連続してアクセスすると、VMM はプログラムがファイルへの順次アクセスを続行すると予測し、ファイルに対する後続の順次読み込みをスケジュールします。この結果、プログラム処理がオーバーラップし、プログラムはデータをより高速に利用できます。次の VMM しきい値がカーネル・パラメータとして実装されており、先読みするページ数の決定に使用されます。

- minpgahead

このパラメータは、VMM が最初に順次アクセス・パターンを検出したときに先読みするページ数を格納します。

- maxpgahead

このパラメータは、順次ファイルから VMM が先読みする最大ページ数を格納します。

minpgahead および maxpgahead パラメータは、アプリケーションに適した値に設定してください。デフォルト値は、それぞれ 2 および 8 です。これらの値を変更するには、/usr/sbin/vmo コマンドを使用します。ストライプ化論理ボリュームの順次パフォーマンスを重視する場合は、maxpgahead パラメータの値を高めに設定します。minpgahead パラメータを 32 ページに、maxpgahead パラメータを 64 ページに設定するには、次のコマンドを root ユーザーで実行します。

```
-o minpgahead=32 -o maxpgahead=64
```

minpgahead および maxpgahead パラメータは、2 の累乗に設定します。たとえば、2、4、8、... 512, 1042, ... などです。

## A.2.8 ディスク I/O の歩調合せのチューニング

ディスク I/O の歩調合せとは、システム管理者がファイルに対して保留される I/O 要求の数を制限するための AIX のメカニズムです。このメカニズムによって、ディスク I/O が頻繁に発生するプロセスのために CPU が飽和状態に陥ることを回避できます。この結果、対話型プロセスや CPU 使用量の多いプロセスの応答時間に遅延が発生しません。

ディスク I/O の歩調合せを行うには、最高水位標および最低水位標という 2 つのシステム・パラメータを調整します。保留中の I/O 要求が最高水位標に達しているファイルに対してプロセスが書き込みを実行すると、プロセスはスリープ状態になります。未処理の I/O 要求の数が最低水位標以下になると、プロセスはスリープ状態から解放されます。

最高水位標および最低水位標を変更するには、smit コマンドを使用します。試行錯誤を重ねて適切な水位標を決定します。水位標はパフォーマンスに影響を与えるため、設定するときには注意が必要です。ディスク I/O が 4KB を超える場合は、最高水位標および最低水位標をチューニングしてもほとんど効果はありません。

ディスク I/O の飽和状態を判断するには、iostat の結果、特に iowait および tm\_act の割合を分析します。特定のディスクの iowait の割合と tm\_act の割合が大きい場合、ディスクが飽和状態であることを表します。iowait の割合のみが大きい場合は、必ずしも I/O ボトルネックを表すものではないことに注意してください。

## A.2.9 Oracle Database でのミラー復元

RAW 論理ボリューム (LV) に割り当てられた Oracle データファイルに対してミラー書き込み整合性 (MWC) を無効にすると、Oracle Database のクラッシュ・リカバリ・プロセスによるシステム・エラー後のリカバリで、ミラーの復元が行われます。このミラー復元プロセスを実行することによって、データベースの不整合や破損を防止できます。

クラッシュ・リカバリ時に、論理ボリューム上のデータファイルに複数のコピーが割り当てられている場合、ミラー復元プロセスでは、それらのすべてのコピーのデータ・ブロックに対してチェックサムを実行します。その後、次のいずれかの処理を実行します。

- コピー内のデータ・ブロックのチェックサムが有効である場合、ミラー復元プロセスではそのコピーを使用して、チェックサムが無効なコピーを更新します。
- すべてのコピーについてブロックのチェックサムが無効である場合は、REDO ログ・ファイルの情報を使用してブロックを再構築します。次に、そのデータファイルを論理ボリュームに書き込み、すべてのコピーを更新します。

AIX の場合、ミラー復元プロセスは、RAW 論理ボリュームに割り当てられたデータファイルのうち、MWC が無効になっているデータファイルに対してのみ有効です。ミラー化論理ボリューム上のデータファイルのうち、MWC が有効になっているデータファイルには、MWC によってすべてのコピーの同期が保証されているため、ミラー復元は必要ありません。

以前のリリースの Oracle Database のアップグレード中にシステムがクラッシュし、論理ボリューム上のデータファイルの MWC が無効になっていた場合は、syncvg コマンドを実行してミラー化 LV を同期化してから、Oracle Database を起動してください。ミラー化 LV を同期せずにデータベースを起動すると、LV コピーからデータが正しく読み込めないことがあります。

---

**注意：** ディスク・ドライブに障害が発生した場合、ミラー復元は行われません。その場合は、syncvg コマンドを実行してから、LV を再度アクティブにする必要があります。

---



---

**注意：** ミラー復元は、データファイルに対してのみサポートされています。このため、REDO ログ・ファイルの MWC は無効にしないでください。

---

## A.2.10 RAW デバイスのバックアップ

RAW デバイスをバックアップするには RMAN の使用をお勧めします。dd コマンドを使用して RAW デバイスのバックアップを実行する場合は、この項で説明する指示に従ってください。

RAW デバイスの最初の Oracle ブロックのオフセットは、デバイス・タイプによって 0 (ゼロ)、4K または 128K になります。offset コマンドを使用して、適切なオフセットを判断できます。

論理ボリュームを作成する場合、オフセット 0 (ゼロ) を使用することをお勧めします。これは、-T O オプションを使用した場合に可能です。ただし、以前のバージョンの Oracle Database で作成された既存の RAW 論理ボリュームでは、通常は 0 (ゼロ) 以外のオフセットです。次の例では、最初の Oracle ブロックのオフセットが 4K である RAW デバイスのバックアップおよびリストア方法を示しています。

```
$ dd if=/dev/raw_device of=/dev/rmt0.1 bs=256k
```

RAW デバイスをテープからリストアするには、次のようなコマンドを入力します。

```
$ dd if=/dev/rmt0.1 of=/dev/raw_device count=63 seek=1 skip=1 bs=4k
$ mt -f /dev/rmt0.1 bsf 1
$ dd if=/dev/rmt0.1 of=/dev/raw_device seek=1 skip=1 bs=256k
```

## A.3 CPU のスケジューリングおよびプロセスの優先順位

CPU も、プロセスの競合が発生する可能性のあるシステム・コンポーネントです。AIX カーネルによってほとんどの CPU 時間は効果的に割り当てられますが、プロセスの多くは CPU サイクルをめぐって競合します。複数の CPU (SMP) が搭載されている場合は、各 CPU で様々なレベルの競合が発生する可能性があります。

次の各項では、CPU のスケジューリングおよびプロセスの優先順位について説明します。

- プロセスのランタイム・スライスの変更
- SMP システムでのプロセッサ・バインディングの使用

### A.3.1 プロセスのランタイム・スライスの変更

AIX RR ディスパッチャのランタイム・スライスのデフォルト値は 10 ミリ秒 (msec) です。タイム・スライスを変更するには、schedo コマンドを使用します。タイム・スライスを長くすると、アプリケーションの自発的な切替率の平均が低い場合に、コンテキスト切替率も低くなります。その結果、プロセスのコンテキスト切替率に消費される CPU サイクルが少なくなるため、システムのスループットが向上します。

ただし、ランタイム・スライスが長いと、応答時間が遅くなる可能性があります (特に単一プロセッサ・システムの場合)。デフォルトのランタイム・スライスは通常、ほとんどのアプリケーションに使用できます。実行キューが大きく、多くのアプリケーションと Oracle シャドウ・プロセスがかなり長時間にわたって実行できる場合は、次のコマンドを入力して、タイム・スライスを長くできます。

```
# /usr/sbin/schedo -t n
```

前のコマンドで、n の値に 0 (ゼロ) を設定すると、タイム・スライスは 10msec になり、1 を設定すると 20msec、2 を設定すると 30msec になります。

### A.3.2 SMP システムでのプロセッサ・バインディングの使用

SMP システムでは、複数のプロセスを 1 台のプロセッサにバインドすると、パフォーマンスが大幅に向上する場合があります。プロセッサ・バインディングは、AIX 5L ですべての機能を利用できます。

ただし、AIX 5L リリース 5.2 以降は、AIX スケジューラの特異な改良により、プロセッサ・バインディングを使用せずに Oracle Database プロセスを最適にスケジューリング設定できます。したがって、AIX 5L リリース 5.2 以降で実行する場合は、プロセッサへのプロセスのバインドは、特にお勧めしません。

## A.4 環境変数 AIXTHREAD\_SCOPE の設定

AIX のスレッドは、プロセス全体の競合範囲 (M:N) またはシステム全体の競合範囲 (1:1) により動作できます。環境変数 AIXTHREAD\_SCOPE は、使用する競合範囲を制御します。

環境変数 AIXTHREAD\_SCOPE のデフォルト値は、P (プロセス全体の競合範囲) です。プロセス全体の競合範囲を使用する場合、Oracle スレッドはカーネル・スレッドのプールにマップされます。Oracle がイベントで待機中に Oracle スレッドがスワップ・アウトされると、スレッド ID が異なる別のカーネル・スレッドに戻る場合があります。Oracle では待機中のプロセスのポストにスレッド ID を使用するため、スレッド ID が同じままであることが重要です。システム全体の競合範囲を使用する場合、Oracle スレッドはカーネル・スレッドに 1 対 1 で静的にマップされます。このため、システム全体の競合を使用することをお勧めします。システム全体の競合の使用は、特に RAC インスタンスには重要です。

さらに、AIX 5L リリース 5.2 以上では、システム全体の競合範囲を設定した場合、各 Oracle プロセスに割り当てられるメモリー量がかなり少なくなります。

Oracle Database インスタンスまたは Oracle Net Listener プロセスにおいて、環境変数 ORACLE\_HOME または ORACLE\_SID の設定に使用する環境スクリプトで、次のように、環境変数 AIXTHREAD\_SCOPE の値を S に設定することをお勧めします。

- Bourne、Bash または Korn シェルの場合

~/.profile または /usr/local/bin/oraenv スクリプトに次の行を追加します。

```
AIXTHREAD_SCOPE=S; export AIXTHREAD_SCOPE
```

- C シェルの場合

~/.login または /usr/local/bin/coraenv スクリプトに次の行を追加します。

```
setenv AIXTHREAD_SCOPE S
```

これにより、すべての Oracle プロセスの実行に対してシステム全体のスレッド範囲が有効になります。

## A.5 ネットワーク情報サービス (NIS) の外部ネーミングのサポート

AIX システムでは、NIS 外部ネーミング・アダプタがサポートされています。NIS 外部ネーミングの構成方法と使用方法の詳細は、『Oracle Database Net Services 管理者ガイド』の外部ネーミング・メソッドの構成に関する項を参照してください。

## A.6 AIX 5.3 システムでのマルチスレッド同時処理 (SMT)

マルチスレッド同時処理 (SMT) が有効で、AIX 5.3 オペレーティング・システムを使用している場合、v\$osstat ビューは、オンライン論理 (NUM\_LCPUS) および仮想 CPU (NUM\_VCPUS) に対応した 2 つの行を追加して報告します。

AIX 5.2 システムまたは AIX 5.3 システムで SMT を使用せずに Oracle を実行している場合、これらの行は報告されません。



---

---

## HP-UX システムでの Oracle Database の管理

この付録では、HP-UX システムで Oracle Database を管理する方法について説明します。次の項目について説明します。

- Oracle インスタンス用の HP-UX 共有メモリー・セグメント
- HP-UX SCHED\_NOAGE スケジューリング・ポリシー
- 軽量タイマーの実装
- 非同期 I/O
- 大規模メモリーの割当てと Oracle Database のチューニング
- CPU\_COUNT 初期化パラメータおよび HP-UX 動的プロセッサ再構成
- ネットワーク情報サービス (NIS) の外部ネーミングのサポート

## B.1 Oracle インスタンス用の HP-UX 共有メモリー・セグメント

Oracle Database は、インスタンスの起動時に、Oracle 共有グローバル領域 (SGA) の作成用に割り当てられた共有メモリーを HP-UX `shmmx` カーネル・パラメータの値で除算して、メモリー・セグメントを作成します。たとえば、1つの Oracle インスタンスに割り当てられた共有メモリーが 64GB で、`shmmx` パラメータの値が 1GB の場合、Oracle Database はそのインスタンスに対して 64 個の共有メモリー・セグメントを作成します。

1つの Oracle インスタンスに対して複数の共有メモリー・セグメントが作成されると、パフォーマンスが低下する可能性があります。これは、Oracle Database でインスタンスが作成されるたびに、各共有メモリー・セグメントが一意的プロテクション・キーを受け取るためです。使用できるプロテクション・キーの数は、次の表に示すように、システム・アーキテクチャによって異なります。

アーキテクチャ	プロテクション・キーの数
PA-RISC	6
Itanium	14

Oracle インスタンスが作成する共有メモリー・セグメントがプロテクション・キーの数より多い場合、HP-UX オペレーティング・システムは、プロテクション・キー・フォルトを表示します。

`shmmx` パラメータの値は、システムで使用できる物理メモリーの量に設定することをお勧めします。この設定によって、1つの Oracle インスタンスの共有メモリー全体が 1つの共有メモリー・セグメントに割り当てられ、インスタンスに必要なプロテクション・キーが 1つで済みます。

システムのアクティブな共有メモリー・セグメントのリストを表示するには、次のコマンドを実行します。

```
$ ipcs -m
```

Oracle Database がそのインスタンスに対してプロテクション・キーの数よりも多いセグメントを作成する場合は、`shmmx` カーネル・パラメータの値を大きくします。

**関連項目：** 推奨されるカーネル・パラメータの最小値については、『Oracle Database インストレーション・ガイド』を参照してください。

## B.2 HP-UX SCHED\_NOAGE スケジューリング・ポリシー

HP-UX システムのほとんどのプロセスは、タイム・シェアリング・スケジューリング・ポリシーを使用します。タイム・シェアリングが適用されると、重要な処理 (ラッチの保持など) が実行されるときに Oracle プロセスがスケジューリングから除外されるため、パフォーマンスが低下することがあります。HP-UX の `SCHED_NOAGE` は、特にこの問題に対処した修正済のスケジューリング・ポリシーです。通常のタイム・シェアリング・ポリシーとは異なり、`SCHED_NOAGE` によってスケジューリングが設定されたプロセスは、優先順位が上下したり、優先使用されることがありません。

この機能は、オンライン・トランザクション処理 (OLTP) 環境に適しています。これは、OLTP 環境では、重要なリソースを対象に競合が発生することがあるためです。Oracle Database の OLTP 環境で `SCHED_NOAGE` ポリシーを使用した場合、パフォーマンスが 10 パーセント以上も向上する可能性があります。

`SCHED_NOAGE` ポリシーを意思決定支援環境で使用しても、同じレベルのパフォーマンス上の効果は得られません。これは、リソースの競合がほとんど発生しないためです。アプリケーションとサーバーの環境はそれぞれ異なるため、使用している環境に `SCHED_NOAGE` ポリシーが適切であることをテストおよび検証する必要があります。`SCHED_NOAGE` を使用する場合、Oracle プロセスに最も高い優先順位を割り当てることには注意が必要です。`SCHED_NOAGE` の最も高い優先順位を Oracle プロセスに割り当てると、システムの CPU リソースを使い果たし、他のユーザー・プロセスの応答が停止する可能性があります。

## B.2.1 Oracle Database での SCHED\_NOAGE の有効化

Oracle Database で SCHED\_NOAGE スケジューリング・ポリシーを使用するには、OSDBA グループ（通常は dba グループ）が RTSCHED および RTPRIO 権限を取得し、スケジューリング・ポリシーを変更したり、Oracle プロセスの優先順位レベルを設定する必要があります。dba グループにこれらの権限を付与するには、次の手順を実行します。

1. root ユーザーでログインします。
2. テキスト・エディタを使用して /etc/privgroup ファイルを開くか、必要な場合は作成します。
3. OSDBA グループの名前で始まる次の行を追加または編集し、システムが再起動するたびにこのグループに付与する RTPRIO 権限と RTSCHED 権限を指定します。

```
dba RTPRIO RTSCHED
```

4. ファイルを保存してテキスト・エディタを終了します。
5. 次のコマンドを入力し、OSDBA グループに権限を付与します。

```
# /usr/sbin/setprivgrp -f /etc/privgroup
```

6. 次のコマンドを入力し、権限が正しく設定されていることを確認します。

```
# /usr/sbin/getprivgrp dba
```

HP-UX SCHED\_NOAGE 初期化パラメータを各インスタンスのパラメータ・ファイルに追加し、このパラメータに対してプロセスの優先順位レベル（整数値）を設定します。サポートされる値の範囲は 178 ~ 255 です。値が小さくなるほど優先順位が高くなります。パラメータの設定が範囲外にある場合、Oracle Database は自動的にそのパラメータを許容値に設定し、新しい値が設定された SCHED\_NOAGE ポリシーを使用して処理を続行します。また、新しい設定に関するメッセージを alert\_sid.log ファイルに生成します。ユーザーまたは自動再調整によって、Oracle プロセスに対して最高レベルの優先順位が割り当てられた場合、Oracle Database は、システムの CPU リソースが使い果たされる可能性があることを警告するメッセージを alert\_sid.log ファイルに生成します。このパラメータには、Oracle プロセスに必要な優先順位レベルを割り当てておくことをお勧めします。

**関連項目：** 優先順位ポリシーと優先順位の範囲の詳細は、HP-UX のドキュメント、および rtsched(1) と rtsched(2) の各 man ページを参照してください。

## B.3 軽量タイマーの実装

Oracle Database 10g では、動的初期化パラメータ STATISTICS\_LEVEL が TYPICAL（デフォルト）または ALL に設定されている場合、いつでもランタイム統計を収集できます。このパラメータ設定は、TIMED\_STATISTICS 初期化パラメータを暗黙的に true に設定します。

HP-UX システム上の Oracle Database では、gethrtime() システム・ライブラリ・コールを使用して、統計の収集中に経過時間を計算します。この軽量システム・ライブラリ・コールを使用することによって、パフォーマンスに影響を与えることなく、Oracle インスタンスを実行しながら、いつでもランタイム統計を収集できます。

TIMED\_STATISTICS 初期化パラメータが明示的に true に設定されているときに、gethrtime() システム・ライブラリ・コールを使用すると、使用しない場合に比べて Oracle のパフォーマンスを最大 10 パーセント改善できます。また、ランタイム統計の収集に gethrtime() システム・ライブラリ・コールを使用しても、Oracle Database の OLTP 環境のパフォーマンスが低下することはありません。

## B.4 非同期 I/O

非同期 I/O 擬似ドライバを HP-UX システム上で使用すると、Oracle Database が RAW ディスク・パーティションへの I/O を非同期方式で実行できるようになり、I/O のオーバーヘッドが減少してスループットが向上します。非同期 I/O 擬似ドライバは、HP-UX のサーバーとワークステーションの両方で使用できます。

### B.4.1 MLOCK 権限

Oracle Database で非同期 I/O 操作を処理するには、OSDBA グループ (dba) に MLOCK 権限を付与する必要があります。dba グループに MLOCK 権限を付与するには、次の手順を実行します。

1. root ユーザーでログインします。
2. テキスト・エディタを使用して /etc/privgroup ファイルを開くか、必要な場合は作成します。
3. OSDBA グループの名前で始まる次の行を追加または編集し、MLOCK 権限を指定します。

---

**注意：** このファイルでは、特定のグループに対する権限指定に 1 行のみを使用する必要があります。このファイルに dba グループに関する行がすでに含まれている場合は、その行に MLOCK 権限を追加してください。

---

```
dba RTPRIO RTSCHED MLOCK
```

4. ファイルを保存してテキスト・エディタを終了します。
5. 次のコマンドを入力し、OSDBA グループに権限を付与します。
 

```
# /usr/sbin/setprivgrp -f /etc/privgroup
```
6. 次のコマンドを入力し、権限が正しく設定されていることを確認します。
 

```
# /usr/sbin/getprivgrp dba
```

### B.4.2 非同期 I/O の実装

HP-UX システムで非同期 I/O を使用する場合は、データベース・ファイルに対して次の記憶域オプションを使用する必要があります。

- RAW デバイス (パーティションまたは論理ボリューム)
- RAW パーティションを使用する自動ストレージ管理ディスク・グループ

**関連項目：** HP-UX システムで自動ストレージ管理および RAW 論理ボリュームを構成する方法は、『Oracle Database インストレーション・ガイド』を参照してください。

いずれかの記憶域オプションを使用して非同期 I/O を実装する前に、System Administrator Management (SAM) ユーティリティを使用して、非同期ディスク・ドライバを HP-UX カーネルに設定する必要があります。

SAM ユーティリティを使用して非同期ディスク・ドライバを追加し、カーネルを設定するには、次の手順を実行します。

1. root ユーザーで次のコマンドを実行します。
 

```
# sam
```
2. 「Kernel Configuration」領域を選択します。
3. 「Drivers」領域を選択します。

4. 非同期ディスク・ドライバ (asyncdsk) を選択します。
5. 「Actions」 → 「Add Driver to Kernel」を選択します。
6. 「List」 → 「Configurable Parameters」を選択します。
7. MAX\_ASYNC\_PORTS パラメータを選択します。
8. 「Action」 → 「Modify Configurable Parameter」を選択します。
9. 次のガイドラインを使用して、パラメータに新しい値を指定し、「OK」をクリックします。

MAX\_ASYNC\_PORTS パラメータは、設定可能な HP-UX カーネル・パラメータの 1 つで、`/dev/async` ファイルを同時にオープンできる最大プロセス数を制御します。

最大数のプロセスが `/dev/async` ファイルをオープンした後で別のプロセスがそのファイルを開こうとすると、エラー・メッセージが表示されます。多数のシャドウ・プロセスやパラレル問合せスレーブが非同期 I/O を実行しているシステムで、このエラーが発生すると、システムのパフォーマンスが低下することがあります。このエラーは記録されません。このエラーを回避するには、`/dev/async` ファイルにアクセスする最大プロセス数を予測し、MAX\_ASYNC\_PORTS パラメータにその値を設定します。

10. 「Actions」 → 「Process a New Kernel」を選択します。
11. 次のオプションのいずれかを選択し、「OK」をクリックします。
  - 「Move Kernel Into Place and Shutdown System/Reboot Now」
  - 「Do Not Move Kernel Into Place: Do Not Shutdown/Reboot Now」

2 番目のオプションを選択した場合は、新しいカーネル `vmunix_test` とその作成に使用される `system.SAM` 構成ファイルが、`/stand/build` ディレクトリに作成されます。

新しいカーネルを使用するには、次の手順を実行します。

1. 次のコマンドを入力し、新しいカーネルを所定の場所に移動します。

```
# /usr/sbin/kmupdate
```

2. 次のコマンドを入力し、システムを再起動します。

```
# /sbin/shutdown -r now
```

HP-UX 非同期デバイス・ドライバを使用して非同期 I/O 操作を可能にするには、次の手順を実行します。

1. root ユーザーでログインします。
2. 次のコマンドを入力し、新しいデバイス・ファイルを作成します。

```
# /sbin/mknod /dev/async c 101 0x0
```

3. 次のコマンドを入力し、メジャー番号が 101 の `/dev/async` デバイス・ファイルが存在することを確認します。

```
# ls -l /dev/async
```

このコマンドの出力は、次のようになります。

```
crw----- 1 oracle dba 101 0x000000 Oct 28 10:32 /dev/async
```

4. 必要に応じて、このデバイス・ファイルに対し、Oracle ソフトウェア所有者および OSDBA グループとの整合性があるオペレーティング・システム所有者およびアクセス権を設定します。

Oracle ソフトウェア所有者が `oracle` で、OSDBA グループが `dba` の場合は、次のコマンドを実行します。

```
# /usr/bin/chown oracle:dba /dev/async
```

```
# /usr/bin/chmod 660 /dev/async
```

## B.4.3 非同期 I/O の検証

非同期 I/O を検証するには、最初に、HP-UX 非同期ドライバが Oracle Database に対して設定されていることを検証します。次に、Oracle Database が HP-UX デバイス・ドライバを介して非同期 I/O を実行していることを検証します。

### B.4.3.1 HP-UX 非同期ドライバが Oracle Database に対して設定されているかどうかの検証

HP-UX 非同期ドライバが Oracle Database に対して適切に設定されていることを検証するには、次の手順を実行します。

1. 少ない数のパラレル問合せスレーブ・プロセスを持つ Oracle Database を起動します。
2. 次のコマンドを入力して、GlancePlus/UX ユーティリティを起動します。

```
$ gpm
```

3. メイン・ウィンドウで、「Reports」→「Process List」をクリックします。
4. 「Process List」ウィンドウで、パラレル問合せスレーブ・プロセスを 1 つ選択し、「Reports」→「Process Open Files」を選択します。

パラレル問合せスレーブ・プロセスによって現在オープンされているファイルのリストが表示されます。

5. オープン・ファイルのリストから、/dev/async ファイルまたはモード 101 0x000000 を検索します。

/dev/async ファイルまたはモード 101 0x000000 のいずれかがリストに含まれている場合は、パラレル問合せスレーブ・プロセスによって /dev/async ファイルがオープンされています。つまり、HP-UX 非同期デバイス・ドライバは、Oracle プロセスが非同期 I/O を実行できるように正しく設定されています。/dev/async ファイルのファイル記述子番号をメモしておいてください。

### B.4.3.2 Oracle Database が非同期 I/O を使用しているかどうかの検証

Oracle Database が HP-UX 非同期デバイス・ドライバを介して非同期 I/O を使用しているかどうかを検証するには、次の手順を実行します。

1. HP-UX システムの tusc ユーティリティを、前述の手順で GlancePlus ユーティリティを使用して選択した Oracle パラレル問合せスレーブに接続します。
2. 使用している環境で I/O バウンド問合せを実行します。
3. tusc 出力で read/write コールのパターンをチェックします。

このためには、たとえば、次のコマンドを入力します。pid は、非同期 I/O を処理する予定のパラレル問合せスレーブのプロセス ID です。

```
$ tusc -p pid > tusc.output
```

4. 問合せの実行後、[Ctrl] キーを押しながら [C] キーを押してプロセスから切断し、tusc.output ファイルを開きます。

次に、tusc.output ファイルのサンプルを示します。

```
( Attached to process 2052: "ora_p000_tpch" [ 64-bit ] )
.....
.....
[2052] read(9, "80\0\001\013 \b\0\0\0\0\0\0"..., 388) .. = 28
[2052] write(9, "\0\0\00e\0\0\0\080\0\001\013D \0"..., 48) .. = 48
[2052] read(9, "80\0\001\013¢ 18\0\0\0\0\0\0"..., 388) .. = 28
[2052] write(9, "\0\0\00e\0\0\0\080\0\001\01bd4\0"..., 48) .. = 48
```

DISK\_ASYNC\_IO 初期化パラメータが明示的に false に設定されていない場合（デフォルトの true になっている場合）、tusc.output ファイルには、同じファイル記述子（前述の例では 9）の非同期 read/write コールのパターンが連続して表示されます。

tusc.output ファイルのファイル記述子番号を GlancePlus の /dev/async ファイルで使用されている番号にマップします。この番号は、特定の平行問合せスレーブ・プロセスについて一致しています。これによって、HP-UX 非同期デバイス・ドライバを介した I/O が非同期であることが検証されます。同期 I/O の場合または DISK\_ASYNC\_IO 初期化パラメータが明示的に false に設定されている場合、前述の非同期 read/write パターンは表示されません。かわりに、lseek または pread/pwrite のコールが表示されます。また、1つのファイル記述子のみでなく、多数の異なるファイル記述子 (read/write の最初の引数) が表示されます。

## B.4.4 SGA の非同期フラグ

HP-UX システム上の Oracle Database では、HP-UX 非同期ドライバが提供する非ブロック・ポーリング機能を使用して、I/O 操作の状態がチェックされます。非同期ドライバは、送信された I/O 操作の状態に基づいてフラグを更新します。このポーリングではこのフラグがチェックされます。HP-UX では、このフラグが共有メモリーに読み込まれている必要があります。

Oracle Database では、非同期フラグを各 Oracle プロセスの SGA 内に設定します。HP-UX システム上の Oracle Database は、真の非同期 I/O メカニズムを備えています。つまり、以前に発行された I/O 操作の一部が完了していなくても、I/O 要求を発行できます。このメカニズムにより、パフォーマンスが向上し、平行 I/O プロセスのスケラビリティが保証されます。

リリース 8.1.7 より前の Oracle Database のリリースでは、I/O 操作は、HP-UX 非同期ドライバを使用して共有メモリーからのみ実行できました。新しい HP-UX 非同期ドライバでは、Oracle Database 10g は共有メモリーとプロセス専用領域の両方から I/O 操作を実行できます。ただし、非同期ドライバを介した I/O 操作は、本質的には非同期ではありません。Oracle Database では、非同期ドライバに送信された I/O 操作の状態をチェックするために、ブロック待機が必要です。このため、データベース・ライター・プロセスなど一部の Oracle プロセスでは、本質的には同期 I/O が処理されています。

## B.5 大規模メモリーの割当てと Oracle Database のチューニング

Oracle Database 10g 上で実行するアプリケーションは、以前のリリースで実行するアプリケーションと比べて非常に大きいメモリーを使用する場合があります。これには、次の 2つの理由があります。

- CURSOR\_SPACE\_FOR\_TIME 初期化パラメータのデフォルト値が、false から true に変更されたこと。
- HP-UX システム上の Oracle Database 10g で、仮想メモリー・データ・ページのデフォルト設定値が D (4KB) から L (4GB) に変更されたこと。

### B.5.1 永続的な専用 SQL 領域とメモリー

ユーザーが SQL 文を発行すると、Oracle Database は、次のメモリー割当てステップを自動的に実行します。

1. Oracle SGA の共有プールをチェックし、同一文に対する共有 SQL 領域がすでに存在しているかどうかを確認します。共有 SQL 領域が存在する場合は、その領域を使用して、文の以降の新しいインスタンスを実行します。共有 SQL 領域が存在しない場合は、共有プール内の新しい共有 SQL 領域を SQL 文に割り当てます。
2. ユーザー・セッションのために、専用 SQL 領域を割り当てます。

専用 SQL 領域には、処理対象 SQL 文のバインド情報やランタイム・メモリー構造などのデータが格納されます。また、解析対象文および情報を処理するその他の文も格納されます。

同じ SQL 文を実行するすべてのユーザーには、単一の共有 SQL 領域を使用するカーソルが割り当てられます。このようにして、多数の専用 SQL 領域を同じ共有 SQL 領域に関連付けることができます。ユーザー・セッションが専用サーバーを介して接続された場合、専用 SQL 領域はサーバー・プロセスの PGA 内に設定されます。ただし、ユーザー・セッションが共有サーバーを介して接続された場合は、専用 SQL 領域の一部が SGA 内に保持されます。

CURSOR\_SPACE\_FOR\_TIME 初期化パラメータは、SQL カーソルの割当てをライブラリ・キャッシュから解除して、新規 SQL 文のために領域を確保できるかどうかを指定します。このパラメータを true に設定すると、共有 SQL 領域の割当てが Oracle ライブラリ・キャッシュから解除されるのは、SQL 文に関連付けられたすべてのアプリケーション・カーソルがクローズしている場合のみになります。また、このパラメータが true の場合、オープンしているカーソルに関連付けられた専用 SQL 領域の割当て解除も防止されるため、ユーザーの専用 SQL 領域を永続的にすることができます。

Oracle Database で CURSOR\_SPACE\_FOR\_TIME 初期化パラメータを true に設定した場合、以前のリリースの Oracle と比べて次のメリットがあります。

- アクティブな各カーソルの SQL 領域がメモリーに存在してエージ・アウトしないため、SQL の実行コールが高速になります。
- 共有 SQL 領域がライブラリ・キャッシュに存在することを Oracle Database が検証する必要がないため、アプリケーションのパフォーマンスが向上します。次の SQL 文の実行まで専用 SQL 領域が保持されるため、カーソルの割当てと初期化の時間が節約されます。

Oracle Database で CURSOR\_SPACE\_FOR\_TIME 初期化パラメータを true に設定した場合、以前のリリースの Oracle Database と比べて次のデメリットがあります。

- 永続的な専用 SQL 領域用のメモリー割当てが増加するため、ユーザー・プロセスのメモリー要件が増加します。
- カーソルのメモリーが大幅に増加するため、Oracle Database のシャドウ・プロセス用のメモリー割当てが大きくなります。

CURSOR\_SPACE\_FOR\_TIME パラメータを false に設定すると、SQL 全体の実行速度が遅くなり、パフォーマンスが低下することがあります。このパラメータを false に設定すると、共有 SQL 領域の割当てがライブラリ・キャッシュから早期に解除される可能性があります。

## B.5.2 デフォルトの大規模仮想メモリー・ページ・サイズ

デフォルトでは、Oracle Database は、プロセス専用メモリーの割当てに、HP-UX システムの使用可能な最大の仮想メモリー・ページ・サイズ設定を使用します。このサイズは、値 L (最大) で定義され、現在 HP-UX v2 (11i.23) です。この値は、Oracle 実行可能ファイルのリンク時に、LARGE\_PAGE\_FLAGS オプションの 1 つとして設定されます。

仮想メモリー・ページ・サイズが L に設定されていると、HP-UX は、使用可能なプロセス専用メモリーを、1GB 制限まで、または割り当てられたメモリー量の合計に達するまで、1MB、4MB、16MB などのサイズのページに割り当てます。Oracle PGA に十分なメモリーが割り当てられていて、大きなデータ・ページ・サイズ単位でのメモリー割当てが可能な場合は、オペレーティング・システムは、一度に最大のページ・サイズを割り当てます。たとえば、Oracle PGA に 48MB を割り当てている場合、システムでは、16MB を 3 ページ設定するか、またはより小さい倍数の単位サイズのページを組み合せることができます。たとえば、1MB を 4 ページ、4MB を 3 ページおよび 16MB を 2 ページとするなどです。PGA に 64MB を割り当てた場合、データ・ページ単位サイズと使用可能なメモリー量とが一致するため、1 ページ (64MB) がオペレーティング・システムによって割り当てられます。

一般的には、大規模メモリー・ページによってアプリケーションのパフォーマンスは向上します。これは、オペレーティング・システムが処理する必要のある仮想メモリー変換時のエラー数が減り、より多くの CPU リソースをアプリケーションに解放できるためです。また、プロセス専用メモリーの割当てに必要なデータ・ページ数の合計も減ります。これによって、プロセス・レベルでの Translation Lookaside Buffer (TLB) ミスの可能性が減ります。

ただし、アプリケーションにメモリーの制約があり、非常に多くのプロセスを実行する傾向がある場合は、この大幅なページ・サイズの増加によってプロセスに大規模なメモリーの割当てが指示されるため、メモリー不足エラー・メッセージが発生する可能性があります。メモリー不足エラーが発生する場合は、ページ・サイズの値を小さくして、D (デフォルト) サイズの 4KB と L (最大) サイズの 4GB の間に設定する必要があります。

最小ページ・サイズ設定値 (4KB) を使用すると、最大ページ・サイズ設定値を使用した場合よりも CPU 使用率が 20 パーセント以上高くなります。最大設定値 L を使用すると、4MB の設定値を使用した場合よりもメモリー使用率が 50 パーセント高くなります。システムにメモリー

制約がある場合は、使用できるメモリー・リソースの制約内で、ページ・サイズを特定のアプリケーションの要件と一致するように設定することをお勧めします。

たとえば、設定値 L では問題が発生するアプリケーションで、設定値 4MB の仮想メモリー・ページを使用すると、適切なパフォーマンスが得られる場合があります。

### B.5.3 チューニングに関する推奨事項

永続的な専用 SQL 領域および大規模仮想メモリー・ページ・サイズへのメモリー割当ての増加に対応してチューニングを行うには、次の推奨事項があります。

- CURSOR\_SPACE\_FOR\_TIME パラメータの値は、true にしておきます。ただし、この設定で、アプリケーションの実行時にライブラリ・キャッシュ・ミスが発生する場合があります。この場合、共有プールが、同時にオープンしているすべてのカーソル用の SQL 領域を保持できる最小限のサイズになっている可能性があります。
- 必要に応じて、Oracle Database の仮想メモリー・データ・ページ・サイズを減らします。次のコマンドを使用して、ページ・サイズ設定を変更します。

```
# /usr/bin/chatr +pd newsize $ORACLE_HOME/bin/oracle
```

この例の *newsize* は仮想メモリー・ページ・サイズの新しい値を表します。

chatr コマンドを次のように使用して、新しい設定を表示します。

```
# /usr/bin/chatr $ORACLE_HOME/bin/oracle
```

## B.6 CPU\_COUNT 初期化パラメータおよび HP-UX 動的プロセッサ再構成

HP-UX 11i では、プロセッサ・セット (Psets) の動的ランタイム再構成、および有効なユーザーによるプロセッサ・セット間の作業負荷の動的再割当てをサポートしています。

HP-UX Virtual Partitions (VPAR) を使用すると、ユーザーは、各自のシステムを複数の論理パーティションで構成し、各パーティションに独自のプロセッサ、メモリーおよび I/O リソースのセットを割り当てて、HP-UX オペレーティング・システムの個別のインスタンスを実行できます。vPars に組み込まれている HP-UX Processor Sets を使用すると、仮想パーティションを再起動せずに、仮想パーティション間で動的プロセッサを移行できます。これによって、アプリケーション間でリソースのパーティション化を効率的に行うことができるため、HP-UX サーバー上で実行する各アプリケーションへのインタフェースおよび保証を必要とするリソースの割当てが最小限で済みます。

Oracle Database の CPU\_COUNT 初期化パラメータは、Oracle Database で使用可能な CPU 数を指定します。HP-UX 11i システムの Oracle Database 10g では、オペレーティング・システムを定期的に問い合わせることによって、CPU ホスト構成の変更を動的に検出できます。システムの CPU 数が増えられた場合、Oracle は CPU\_COUNT パラメータを適切な値に調整して、その内部リソースを再割当てします。これによって、新しい作業負荷は新しく追加されたプロセッサを利用できるため、高い CPU 使用率が原因でボトルネックが発生している場合は、DBA による変更を必要とせずに、データベースのパフォーマンスを改善できます。

初期化パラメータの一部の値は、システムの起動時に CPU\_COUNT 値に基づいて計算されます。システムの起動後に CPU 数が増えられた場合、これらの初期化パラメータは動的に更新されず、新しい CPU 数は考慮されません。このため、新しい CPU 数が元の CPU 数と大きく異なる場合は、最適でないデータベース構成となる場合があります。システムで CPU 数が大幅に増加すると、データベースは追加された処理能力を利用できない可能性があります。

システムで増加した CPU 数が小さい場合 (たとえば、2 から 4 に変更した場合) に必要な処理はありません。

システムで増加した CPU 数が大きい場合 (たとえば、2 から 32 に変更した場合) は、次の手順に従ってください。

1. 次のどちらかの方法で、CPU\_COUNT 初期化パラメータを新しい値に設定します。
  - データベースでサーバー・パラメータ・ファイル (spfiledbname.ora) を使用している場合は、次の SQL\*Plus コマンドを SYS ユーザーで実行し、新しいパラメータ値を指定します。

```
SQL> ALTER SYSTEM SET CPU_COUNT=32 SCOPE=SPFILE
```
  - データベースで初期化パラメータ・ファイル (init.ora) を使用している場合は、このファイルを編集して新しいパラメータ値を指定します。
2. データベースを再起動します。

## B.7 ネットワーク情報サービス (NIS) の外部ネーミングのサポート

HP-UX システムでは、NIS 外部ネーミング・アダプタがサポートされています。NIS 外部ネーミングの構成方法と使用方法の詳細は、『Oracle Database Net Services 管理者ガイド』の外部ネーミング・メソッドの構成に関する項を参照してください。

---

## Linux システムでの Oracle Database の管理

この付録では、Linux システムで Oracle Database を管理する方法について説明します。次の項目について説明します。

- 拡張バッファ・キャッシュのサポート
- SUSE Linux Enterprise Server 9 または Red Hat Enterprise Linux 4 での hugetlbfs の使用
- Red Hat Enterprise Linux AS 3 での hugetlbfs の使用
- SGA アドレス空間の増加
- 非同期 I/O サポート
- ダイレクト I/O サポート
- `sentimedop` のサポート
- 高速ネットワークのサポート
- マルチスレッド同時処理 (SMT)

## C.1 拡張バッファ・キャッシュのサポート

---

---

**注意：** この項の内容は、Linux x86 にのみ適用されます。

---

---

Oracle Database では、データベース・バッファ・キャッシュとして、4GB を超えるメモリーを割り当てて使用できます。この項では、Linux x86 システムの拡張バッファ・キャッシュ機能の制限事項と要件について説明します。

**関連項目：** 拡張バッファ・キャッシュ機能の詳細は、『Oracle Database 概要』を参照してください。

### メモリー内ファイル・システム

拡張バッファ・キャッシュ機能を使用するには、データベース・バッファ・キャッシュに使用するメモリー量以上のサイズのメモリー内ファイル・システムを /dev/shm マウント・ポイントに作成します。たとえば、8GB のファイル・システムを /dev/shm マウント・ポイントに作成するには、次の手順を実行します。

1. root ユーザーで次のコマンドを実行します。

```
# mount -t tmpfs shmfs -o size=8g /dev/shm
```
2. システムの再起動時にメモリー内ファイル・システムが確実にマウントされるようにするには、/etc/fstab ファイルに次のようなエントリを追加します。

```
shmfs /dev/shm tmpfs size=8g 0 0
```

拡張バッファ・キャッシュ機能を有効にして Oracle Database を起動すると、Oracle バッファ・キャッシュに対応するファイルが /dev/shm ディレクトリに作成されます。

---

---

**注意：** メモリー内ファイル・システムが /dev/shm マウント・ポイントにすでにマウント済の場合は、その大きさがデータベース・バッファ・キャッシュに使用するメモリー量以上であることを確認します。

---

---

### USE\_INDIRECT\_DATA\_BUFFERS 初期化パラメータ

拡張バッファ・キャッシュ機能を有効にするには、パラメータ・ファイルで USE\_INDIRECT\_DATA\_BUFFERS 初期化パラメータを true に設定します。これによって、Oracle Database は大きいバッファ・キャッシュを指定できるようになります。

### 動的キャッシュ・パラメータ

拡張キャッシュ機能が有効になっている場合は、DB\_BLOCK\_BUFFERS パラメータを使用してデータベース・キャッシュ・サイズを指定する必要があります。

拡張バッファ・キャッシュ機能が有効になっている間は、次の動的キャッシュ・パラメータを使用しないでください。

- DB\_CACHE\_SIZE
- DB\_2K\_CACHE\_SIZE
- DB\_4K\_CACHE\_SIZE
- DB\_8K\_CACHE\_SIZE
- DB\_16K\_CACHE\_SIZE

**制限事項**

拡張バッファ・キャッシュ機能には、次の制限事項が適用されます。

- 作成または使用できる表領域はデフォルト・ブロック・サイズの表領域のみです。DB\_BLOCK\_SIZE パラメータで指定されたブロック・サイズのみを使用した表領域を作成できます。
- インスタンスの実行中はバッファ・キャッシュのサイズを変更できません。

**関連項目：** CREATE TABLESPACE コマンドで使用するデフォルト・ブロック・サイズの詳細は、『Oracle Database SQL リファレンス』を参照してください。

---

**注意：** デフォルトの VLM ウィンドウ・サイズは 512MB です。このメモリー・サイズは、プロセスのアドレス空間に割り当てられます。この値を増減させるには、環境変数 VLM\_WINDOW\_SIZE をバイト単位の新しいサイズに設定します。たとえば、VLM\_WINDOW\_SIZE を 256MB に設定するには、次のコマンドを実行します。

```
$ export VLM_WINDOW_SIZE=268435456
```

環境変数 VLM\_WINDOW\_SIZE に指定する値は、64KB の倍数である必要があります。

---

**Red Hat Enterprise Linux 3 の場合のみ：VLM ウィンドウ・サイズ**

VLM ウィンドウ・サイズに対応するには、プロセスごとのロック・メモリーのデフォルトの最大サイズを増やす必要があります。このサイズを増やすには、`/etc/security/limits.conf` ファイルに次の行を追加します。この場合、`oracle` はデータベースを管理するユーザーです。

```
oracle      soft    memlock    3145728
oracle      hard    memlock    3145728
```

ssh を使用してシステムにログインする場合、`/etc/ssh/sshd_config` ファイルに次の要素を追加して、ssh セッションの開始時にデフォルト値を使用できるようにします。

```
UsePrivilegeSeparation no
```

## C.2 SUSE Linux Enterprise Server 9 または Red Hat Enterprise Linux 4 での hugetlbfs の使用

SUSE Linux Enterprise Server 9 または Red Hat Enterprise Linux 4 において、Oracle Database でラージ・ページ (Huge ページとも呼ばれる) を使用可能にするには、`vm.nr_hugepages` カーネル・パラメータの値を設定し、予約するラージ・ページ数を指定します。データベース・インスタンスの SGA 全体を保持するだけの十分なラージ・ページ数を指定する必要があります。必要なパラメータ値を判断するには、インスタンスの SGA サイズをラージ・ページのサイズで除算してから、結果の端数を切り上げて最も近い整数にします。

デフォルトのラージ・ページ・サイズを判断するには、次のコマンドを実行します。

```
# grep Hugepagesize /proc/meminfo
```

たとえば、`/proc/meminfo` にラージ・ページのサイズが 2MB とリストされ、インスタンスの総 SGA サイズが 1.6GB の場合は、`vm.nr_hugepages` カーネル・パラメータの値を 820 ( $1.6\text{GB} / 2\text{MB} = 819.2$ ) に設定します。

## C.3 Red Hat Enterprise Linux AS 3 での hugetlbfs の使用

---

---

**注意:** Red Hat Linux AS 3 では、Linux on POWER はサポートされていません。

---

---

Red Hat Enterprise Linux AS 3 でラージ・ページを使用するには、次の手順を実行します。

1. ラージ・ページ・プールに必要なメモリーを決定します。

この値を決定するには、インスタンスの SGA のサイズを MB に変換し、4MB 単位で端数を切り上げます。たとえば、SGA が 2.7GB の場合、適切な値は 2768MB です。

2. 使用するブート・ローダーのタイプに応じて、次のいずれかの手順を実行します。

LILO の場合

- a. /etc/lilo.conf ファイルの適切なイメージ・セクションに、hugepages オプションを追加して、ページ数を指定します。

```
append = "hugepages=1024"
```

- b. /sbin/lilo を実行します。

- c. システムを再起動します。

GRUB の場合

- a. /etc/grub.conf ファイルの kernel コマンドに、hugepages オプションを追加して、次のようにページ数を指定します。

```
kernel /vmlinuz-2.4.9 root=/dev/hda5 hugepages=1024
```

- b. システムを再起動します。

3. /etc/sysctl.conf ファイルに次のエントリを追加または編集して、ラージ・ページ・プールのサイズを MB で指定します。

```
vm.hugetlb_pool = 2768
```

4. 次のコマンドを実行してカーネル・パラメータの値を設定します。

```
# sysctl -p /etc/sysctl.conf
```

5. このメモリー量がラージ・ページ・プールに正常に移動したことを確認するには、次のコマンドを実行します。

```
# cat /proc/meminfo
```

最後に表示される行は、ラージ・ページ・プールに移動したメモリー・ページ数を示します。

6. Oracle Database を起動します。

## C.4 SGA アドレス空間の増加

---

**注意：** この項の内容は、Linux x86 にのみ適用されます。

---

Linux のディストリビューションに応じて、次のいずれかの項の説明を適用して SGA アドレス空間を増やします。

- [SUSE Linux Enterprise Server 9](#)
- [Red Hat Enterprise Linux AS 3](#)

### SUSE Linux Enterprise Server 9

SUSE Linux Enterprise Server 9 で SGA アドレス空間を増やすには、次の手順を実行します。

1. oracle ユーザーでログインします。
2. \$ORACLE\_HOME/rdbms/lib ディレクトリで、次のコマンドを実行します。

```
$ genksms -s 0x15000000 > ksms.s
$ make -f ins_rdbms.mk ksms.o
$ make -f ins_rdbms.mk ioracle
```

---

**注意：** 前述の手順を完了しても Oracle Database が起動しない場合、またはランタイム・メモリー・エラーが存在する場合は、最初のコマンドで指定した 16 進の数値を大きくしてください。たとえば、0x15000000 の値で Oracle Database が起動しない場合は、0x20000000 の値を指定します。この値を小さくすると SGA アドレス空間が大きくなりますが、PGA アドレス空間は小さくなる場合があります。

---

3. 次のコマンドを実行し、oracle ユーザーのシェル・プロセスのプロセス ID を確認します。
 

```
$ echo $$
```

 戻される数値はプロセス ID です。
4. 次のコマンドを実行し、ユーザーを root に切り替えます。
 

```
$ su - root
```
5. 次のコマンドを実行し、oracle ユーザーのシェル・プロセスの mapped\_base 設定を変更します。pid は、手順 3 で識別したプロセス ID です。
 

```
# echo 268435456 > /proc/pid/mapped_base
```
6. exit コマンドを実行して oracle ユーザーのシェル・プロセスに戻り、Oracle リスナーと Oracle Database を起動します。

---

**注意：** Oracle プロセスはすべて、この変更した mapped\_base 値を取得する必要があります。mapped\_base が変更されているシェルからリスナーを起動すると、クライアント接続を正しく接続できます。

---

**Red Hat Enterprise Linux AS 3**

Red Hat Enterprise Linux AS 3 または 4 で SGA アドレス空間を増やすには、次の手順を実行します。

1. oracle ユーザーでログインします。
2. \$ORACLE\_HOME/rdbms/lib ディレクトリで、次のコマンドを実行します。

```
$ genksms -s 0x15000000 > ksms.s
$ make -f ins_rdbms.mk ksms.o
$ make -f ins_rdbms.mk ioracle
```

3. Oracle Database を起動します。

## C.5 非同期 I/O サポート

---

**注意：** Linux では、デフォルトにより、自動ストレージ管理で非同期 I/O が使用されます。NFS ファイル・システムに格納されたデータベース・ファイルでは、非同期 I/O はサポートされません。

---

Oracle Database では、カーネルの非同期 I/O がサポートされます。この機能は、デフォルトでは無効になっています。

デフォルトでは、パラメータ・ファイルの DISK\_ASYNC\_IO 初期化パラメータは true に設定され、RAW デバイスに対する非同期 I/O が有効になっています。ファイル・システムのファイルに対して非同期 I/O を有効にするには、次の手順を実行します。

1. すべての Oracle Database ファイルが、非同期 I/O をサポートしているファイル・システム上にあることを確認します。
2. パラメータ・ファイル内の FILESYSTEMIO\_OPTIONS 初期化パラメータを次のいずれかの値に設定します。

Linux ディストリビューション	推奨値
SUSE Linux Enterprise Server 9	SETALL
その他のディストリビューション	ASYNCH

## C.6 ダイレクト I/O サポート

ダイレクト I/O サポートは、Red Hat Enterprise Linux 3 および SUSE Linux Enterprise Server 9 では使用可能であり、サポートされています。

---

**注意：** Red Hat Enterprise Linux 3 でダイレクト I/O を使用するには、使用するドライバが vary I/O をサポートしている必要があります。

Linux on POWER の場合、Red Hat Linux 4 でダイレクト I/O を使用できません。

---

ダイレクト I/O サポートを使用可能にするには、次の手順を実行します。

- FILESYSTEMIO\_OPTIONS 初期化パラメータを DIRECTIO に設定します。
- 非同期 I/O オプションを使用している場合、FILESYSTEMIO\_OPTIONS 初期化パラメータを SETALL に設定します。

## C.7 semtimedop のサポート

---

**注意:** Linux on POWER では、semtimedop はサポートされていません。

---

Red Hat Enterprise Linux 3 および SUSE Linux Enterprise Server 9 では、Oracle Database で `semtimedop()` システム・コール (時間制限付きセマフォ) がサポートされます。この機能のサポートを有効にするには、`$ORACLE_HOME/rdbms/lib` ディレクトリで `oracle` ユーザーとして次のコマンドを実行します。

```
$ make -f ins_rdbms.mk smt_on
```

`semtimedop()` のサポートを無効にするには、`$ORACLE_HOME/rdbms/lib` ディレクトリで `oracle` ユーザーとして次のコマンドを実行します。

```
$ make -f ins_rdbms.mk smt_off
```

## C.8 高速ネットワークのサポート

---

**注意:** この項の内容は、Linux x86 にのみ適用されます。

---

Oracle Database 10g リリース 1 (10.1) の Oracle Net では、Red Hat Enterprise Linux AS 3 で InfiniBand ネットワーク・アーキテクチャを介した Sockets Direct プロトコル (SDP) がサポートされます。このリリースでは、SDP のサポートは同期 I/O のみの限定です。SDP を介した非同期 I/O 使用のサポートの詳細は、次のドキュメントを参照してください。

[http://www.oracle.com/technology/products/oraclenet/files/Oracle\\_Net\\_High-Speed\\_Interconnect\\_Support.doc](http://www.oracle.com/technology/products/oraclenet/files/Oracle_Net_High-Speed_Interconnect_Support.doc)

---

**注意:** このドキュメントで特に示されていないかぎり、Oracle Net の `NET_ASYNC_IO` および `SDP_ASYNC_IO` 構成パラメータを設定しないでください。

---

## C.9 マルチスレッド同時処理 (SMT)

マルチスレッド同時処理 (SMT) が有効になっている場合、`v$osstat` ビューは、オンライン論理 (NUM\_LCPUS) および仮想 CPU (NUM\_VCPUS) に対応した 2 つの行を追加して報告します。



# D

---

---

## Mac OS X システムでの Oracle Database の管理

この付録では、Mac OS X で Oracle Database を管理する方法について説明します。

## D.1 使用可能および使用済のスワップ領域の決定

Mac OS X では、`/private/var/vm` ディレクトリに必要な応じてスワップ・ファイルが動的に作成されます。Oracle Database を実行する場合、新規作成されたスワップ・ファイルに対応するために、ルート (`/`) ファイル・システムに使用可能なディスク領域が 1GB 以上あることを確認してください。

`/private/var/vm` ディレクトリで使用可能なディスク領域のサイズを判別するには、次のコマンドを実行します。

```
$ df -k /private/var/vm
```

現在使用中のスワップ領域のサイズを判別するには、次のコマンドを実行します。

```
$ du -sk /private/var/vm/swapfile*
```

---

---

## Solaris システムでの Oracle Database の管理

この付録では、Solaris システムで Oracle Database を管理する方法について説明します。

## E.1 緊密共有メモリー

Solaris システム上の Oracle Database では、Oracle プロセス間で仮想メモリー・リソースを共有するため、共有メモリー・セグメントとして緊密共有メモリー (ISM) が使用されます。ISM を使用すると、共有メモリー・セグメント全体の物理メモリーが自動的にロックされます。

Solaris 8 および Solaris 9 システムでは、ページング可能な Dynamic ISM (DISM) を使用できます。DISM を使用すると、Oracle Database はセグメントを共有するプロセス間で仮想メモリー・リソースを共有でき、同時にメモリーのページングも可能になります。このため、オペレーティング・システムでは、共有メモリー・セグメント全体で使用される物理メモリーをロックする必要がありません。

Oracle Database では、次の基準に基づいて、ISM または DISM が自動的に選択されます。

- Oracle Database では、使用しているシステムで DISM を使用でき、SGA\_MAX\_SIZE 初期化パラメータの値が、結合されたすべての SGA コンポーネントに必要なサイズよりも大きい場合に、DISM が使用されます。したがって、Oracle Database では、使用される物理メモリー量のみがロックされます。
- Oracle Database では、起動時に共有メモリー・セグメント全体が使用中の場合、または SGA\_MAX\_SIZE パラメータの値が、結合されたすべての SGA コンポーネントに必要なサイズ以下の場合に、ISM が使用されます。

Oracle Database では、ISM または DISM のいずれを使用するかに関係なく、インスタンスの起動後に、常に動的にサイズ変更できるコンポーネント (バッファ・キャッシュなど)、共有プールおよびラジ・プール間でメモリーを交換できます。Oracle Database では、動的 SGA コンポーネントからメモリーを解放し、それを別のコンポーネントに割り当てることができます。

DISM の使用時は、共有メモリー・セグメントがメモリー内で暗黙的にロックされないため、Oracle Database は起動時に使用している共有メモリーを明示的にロックします。動的な SGA 操作によって共有メモリーが追加されると、Oracle Database は使用中のメモリーを明示的にロックします。動的な SGA 操作によって共有メモリーが解放されると、Oracle Database は解放されたメモリーに対するロックを明示的に解除します。その結果、解放されたメモリーが他のアプリケーションで使用できるようになります。

Oracle Database は、oradism ユーティリティを使用して共有メモリーのロックとロック解除を実行します。oradism ユーティリティは、インストール時に自動的に設定されます。したがって、動的 SGA を使用するための設定作業は不要です。

---

---

### 注意:

- サーバー・パラメータ・ファイルの LOCK\_SGA パラメータは、TRUE に設定しないでください。TRUE に設定した場合、Oracle Database 10g は起動できません。
- oradism ユーティリティのプロセス名は、ora\_dism\_sid です。sid はシステム識別子です。DISM を使用している場合、このプロセスはインスタンスの起動時に開始され、インスタンスがシャットダウンされると自動的に終了します。

oradism ユーティリティが正しく設定されていないことを示すメッセージがアラート・ログに表示された場合は、oradism ユーティリティが \$ORACLE\_HOME/bin ディレクトリに格納されていること、およびスーパーユーザー権限が付与されていることを確認してください。

---

---

---

---

# Tru64 UNIX システムでの Oracle Database の 管理

この付録では、Tru64 UNIX システムで Oracle Database を管理する方法について説明します。  
次の項目について説明します。

- Oracle Database 指定配置最適化の有効化
- 複合 CPU システムのサポート
- Tru64 UNIX でのデータベース統計の収集
- 非同期 I/O のチューニング
- ダイレクト I/O サポートおよびコンカレント・ダイレクト I/O サポート
- リアルタイム・クロックへのアクセスの有効化
- RAW デバイスの設定
- Spike 最適化ツール

## F.1 Oracle Database 指定配置最適化の有効化

HP AlphaServer GS Series、ES47 および ES80 システムは、リソース・アフィニティ・ドメイン (RAD) と呼ばれる小さな構築ブロックで構成されています。RAD は、高速インターコネクで接続された密結合 CPU、メモリー・モジュールおよび I/O コントローラの集合です。第 2 レベルのインターコネクでは、各 RAD が接続されて、より大きな構成が形成されます。

CPU、メモリーおよび I/O コントローラ間の共有インターコネクが 1 つのみであった以前のサーバーとは異なり、GS Series、ES47 および ES80 サーバーでは、特定の CPU によるその RAD 内のメモリーへのアクセス時またはローカル I/O コントローラ使用時のパフォーマンスとメモリー・アクセス時間が向上しています。インターコネクが切り替わるため、特定の RAD 内の I/O アクティビティやメモリー・アクセスによって、別の RAD 内の I/O アクティビティやメモリー・アクセスが妨げられることはありません。ただし、RAD 境界にまたがって配置されている CPU とメモリー・モジュール間にメモリー・アクセスを行った場合は、2 つのレベルのインターコネク階層を横断する必要があります。この場合、1 つの RAD 内でメモリーを参照する場合に比べ、多くの参照時間を必要とします。

メモリーとプロセスの指定配置サポートを使用することによって、高度なアプリケーション固有のプロセスおよびメモリーのレイアウトに関する要件をオペレーティング・システムに伝達できます。この機能によって、特定の RAD 内のメモリー参照を詳細に設定できるため、パフォーマンスが向上します。

Oracle Database では、GS Series、ES47、ES80 などの高性能サーバーの特別な機能に対するサポートが強化されています。指定配置最適化では、これらのサーバーで使用できる階層型インターコネクを利用して、以前のサーバーでは、共有インターコネクが 1 つのみであったため、指定配置最適化の利点を直接受けることもなく、これらのサーバーのパフォーマンスが低下することもあります。したがって、Oracle Database では指定配置最適化がデフォルトで無効になっています。

次の各項では、指定配置最適化について説明します。

- [指定配置最適化を実行するための要件](#)
- [Oracle 指定配置最適化の有効化](#)
- [Oracle 指定配置最適化の無効化](#)
- [Oracle 指定配置最適化の使用](#)
- [Oracle 初期化パラメータ](#)
- [Tru64 UNIX サブシステム属性](#)
- [RAD に対するプロセスの親和性](#)
- [システムに存在する RAD 数のサブセットへの Oracle Database の実行制限](#)

### F.1.1 指定配置最適化を実行するための要件

Oracle Database の指定配置最適化を使用するには、次のシステム要件を満たす必要があります。

- HP GS Series、ES47 または ES80 AlphaServer など、ローカル・パフォーマンスを重視したシステムを使用する必要があります。Oracle Database の最適化は、ローカル・パフォーマンスを重視したシステムにのみ効果があります。
- オペレーティング・システムは Tru64 UNIX V5.1B 以上を使用する必要があります。

## F.1.2 Oracle 指定配置最適化の有効化

Oracle 指定配置最適化を有効にするには、次の手順を実行します。

1. Oracle インスタンスをシャットダウンします。
2. 次のコマンドを入力して、Oracle Database を再リンクします。

```
$ cd $ORACLE_HOME/rdbms/lib
$ make -f ins_rdbms.mk numa_on
$ make -f ins_rdbms.mk ioracle
```

互換性のあるバージョンの Tru64 UNIX を使用していない場合は、次のメッセージが表示されます。

```
Operating System Version Does not Support NUMA.
Disabling NUMA!
```

Oracle 指定配置最適化を有効にしてから非互換バージョンの Tru64 UNIX に変更する場合は、次の項で説明する手順に従って、Oracle 指定配置最適化を無効にしてください。

## F.1.3 Oracle 指定配置最適化の無効化

Oracle 指定配置最適化を無効にするには、次の手順を実行します。

1. Oracle インスタンスをシャットダウンします。
2. numa\_off オプションを使用して、Oracle Database を再リンクします。

```
$ cd $ORACLE_HOME/rdbms/lib
$ make -f ins_rdbms.mk numa_off
$ make -f ins_rdbms.mk ioracle
```

## F.1.4 Oracle 指定配置最適化の使用

Oracle 指定配置最適化は、等しくパーティション化された構成を想定しています。つまり、すべての RAD は同じ数の CPU と同じ量のメモリーで構成されていることを想定します。Oracle Database は、システム上のすべての RAD にわたって稼動することを想定します。

## F.1.5 Oracle 初期化パラメータ

Oracle Database では、ローカル環境を最も効率的に使用するために、オペレーティング・システムから報告されたサーバー構成に応じて、いくつかの初期化パラメータが自動的に調整されます。この調整によって、これらの初期化パラメータ間のいくつかの依存関係を計算する際に発生する一般的なエラーが除去されます。

## F.1.6 Tru64 UNIX サブシステム属性

NUMA システムの利点をすべて実現するには、次の表に示すサブシステム属性を設定する必要があります。

サブシステム	属性	設定
ipc	ssm_threshold	0
	shm_allocate_striped	1 (デフォルト)
vm	rad_gh_regions[0]、 rad_gh_regions[1]、 以下同様	<p>rad_gh_regions[n] 属性を次のように設定します。</p> <ol style="list-style-type: none"> <li>次のコマンドを実行します。 \$ ipcs -b</li> <li>出力から SEGSZ 列の値を追加し、合計を 1048576 で除算して MB に変換します。</li> <li>LOG_BUFFER 初期化パラメータの値が 8388608 (8MB) 未満の場合は、合計から 8388608 を差し引きます。LOG_BUFFER の値が 8388608 ~ 16777216 (16MB) の間の場合、SEGSZ 出力の合計から 16777216 を差し引きます。</li> <li>手順 3 の結果をシステムの RAD 数で除算します。</li> <li>rad_gh_regions [0] パラメータを、手順 4 の結果に 8MB または 16MB を加えた値に設定します。どちらを加えるかは、LOG_BUFFER 初期化パラメータおよびその他の有効な rad_gh_regions [n] 属性の値によって異なります。</li> </ol> <p>これらの手順では、システム上のすべての RAD が Oracle に割り当てられていることを前提としています。</p> <p>システムに存在する RAD 数のサブセットを使用するように Oracle が制限されている場合は、F-5 ページの「システムに存在する RAD 数のサブセットへの Oracle Database の実行制限」の項を参照してください。 numa_config_sid.ora ファイルにリストされた最初の RAD については、前述の手順のかわりに、rad_gh_regions [0] 設定を rad_gh_regions [n] パラメータに適用する必要があります。</p>

Tru64 UNIX V5.1B の vm サブシステムには、rad\_gh\_regions 属性が 63 個あります。システム上の RAD の合計数に応じて、これらの属性を設定してください。たとえば、システムに 4 つの RAD があり、これらが Oracle で使用され、SEGSZ 列の合計が 10248MB で、LOG\_BUFFER 初期化パラメータが 2097152 (2MB) に設定されている場合は rad\_gh\_regions [0] を 2568 に設定し、rad\_gh\_regions [1]、rad\_gh\_regions [2] および rad\_gh\_regions [3] を 2560 に設定します。インスタンスを正常に起動させるには、この値をわずかに (1 または 2) 高くする必要があります。

CPU とメモリーがオフラインの場合、Oracle Database は引き続き動作しますが、パフォーマンスは低下します。

## F.1.7 RAD に対するプロセスの親和性

特定の RAD 上でプロセスを実行するようにオペレーティング・システムに設定することによって、パフォーマンスを向上させることができます。Oracle リスナー・プロセスを介してデータベースに接続し、対応するネットワーク・インターコネクト・アダプタが RAD 上にある場合は、各 RAD 上でリスナーを実行できます。特定の RAD 上でリスナーを実行するには、次のコマンドを実行します。rad\_number は RAD の数です。

```
$ runon -r rad_number lsnrctl start [listener_name]
```

すべての Oracle シャドウ・プロセスが、Oracle リスナーと同じ RAD 上に自動的に作成されます。

## F.1.8 システムに存在する RAD 数のサブセットへの Oracle Database の実行制限

Oracle Database の実行を、システムに存在する RAD の数のサブセットに制限できます。インスタンスの起動時に、Oracle Database は \$ORACLE\_HOME/dbs ディレクトリの numa\_config\_sid.ora ファイルを検索します。ファイルがない場合、Oracle Database はシステム上の RAD を使用します。ファイルが見つかった場合、Oracle Database はインスタンスに対して、ファイルに指定されている RAD のみを使用して、指定の RAD のみでプロセスを実行するように指示します。

numa\_config\_sid.ora ファイルは、次のような形式になっています。

```
number_RADs
group1
...
groupn
```

この例では、number\_RADs は Oracle が使用するプロセッサ・グループまたは RAD の数、group1 ~ groupn は、Oracle が使用する RAD の番号です。

ファイルを作成するときは、次のガイドラインに従ってください。

- 空白の行は挿入しない
- 個々の行に番号を指定する
- 番号以外は行に一切指定しない

各 RAD には、特定のシステムに対する番号が 0 (ゼロ) から最大数まで付けられます。システムが完全に統合化されている AlphaServer GS320 の場合、RAD には、0、1、2、3、4、5、6、7 と番号が付きます。たとえば、完全に構成された GS320 の RAD のうち、Oracle Database が 1 と 3 を使用するように制限するには、numa\_config\_sid.ora ファイルを次のように作成します。

```
2
1
3
```

グループ番号は、連続したり昇順の番号にする必要はありません。

Bequeath 接続のためのパラレル問合せスレーブとシャドウ・プロセスは、ファイルに指定されている RAD 上に作成されます。リモート TCP 接続の場合は、RAD にリスナーを明示的にバインドする必要があります。詳細は、F-5 ページの「[RAD に対するプロセスの親和性](#)」の項を参照してください。

## F.2 複合 CPU システムのサポート

Tru64 UNIX V5.1B 以上を使用する Tru64 UNIX システムでは、異なる CPU 速度およびタイプを組み合わせて動作できます。1つの RAD 内の CPU はすべて、速度およびキャッシュ・サイズが同じであることが必要です。別の RAD には、速度およびキャッシュ・サイズの異なる CPU を指定できます。

複合 CPU システムのパフォーマンスは、低速 CPU と高速 CPU の比率によって変わります。また、システム上の Oracle プロセスの配置にも左右されます。高速トランザクションのオンライン・トランザクション処理 (OLTP) 環境では、データベース・ライター・プロセスおよびログ・ライター・プロセスを低速 CPU に配置すると、パフォーマンスが低下する可能性があります。一方、データ・ウェアハウスや意思決定支援環境では、データベース・ライター・プロセスおよびログ・ライター・プロセスを低速 CPU に配置しても、パフォーマンスはまったく変わらないことがあります。

CPU システムを組み合わせることによって、ハードウェア資産を保護できます。古い CPU を交換せずに、より高速で強力な CPU をシステムに追加できます。HP 社とオラクル社では、複合 CPU システムのテストを完了し、現在この機能をサポートしています。

---

**注意：** この複合 CPU システムのパフォーマンスは、最高速 CPU のみで構成された複合 CPU システムと同等のパフォーマンスにはなりません。ただし、最低速 CPU のみで構成された複合 CPU システムよりは高くなります。複合 CPU システムの詳しい規則と制限事項は、HP 社にお問い合わせください。

---

## F.3 Tru64 UNIX でのデータベース統計の収集

Oracle Database 10g は、Tru64 UNIX V5.1B 以上でのみ動作します。これは、Tru64 UNIX の long double データ型のサイズが、V4.0x では 64 ビットでしたが、V5.x では 128 ビットに変更されたためです。この変更によって、いくつかの操作の精度が高くなりました。これらの操作を実行すると、表または索引が分析され、統計情報がデータ・ディクショナリに格納されます。

Oracle Database 内の問合せ最適マイザでは、データ・ディクショナリに格納されている統計情報を使用して、最適な問合せ方法を判断します。最適な問合せ方法を判断しているときに、データ・ディクショナリに格納されている統計と問合せ最適マイザによって算出された統計とが一致しない場合、問合せ最適マイザは誤った方法で問合せを実行することがあります。この場合、問合せが正しく実行されなかったり、失敗する可能性があります。

このため、Oracle8i リリース 8.1.7 以前から Oracle Database 10g にアップグレードした後は、各スキーマのオブジェクト統計をすべて分析する必要があります。Oracle9i リリース 1 (9.0.1) またはリリース 2 (9.0.2) から Oracle Database 10g にアップグレードした後は、スキーマを分析しなおす必要はありません。DBMS\_STATS.GATHER\_SCHEMA\_STATS プロシージャを使用して、分析を実行し、スキーマごとに統計情報を収集できます。DBMS\_STATS パッケージでは、新しい統計に問題が発生した場合に備えて、現在の表または索引の統計を表に保存しています。

**関連項目：** データベース統計の収集の詳細は、『Oracle Database PL/SQL パッケージ・プロシージャおよびタイプ・リファレンス』を参照してください。

## F.4 非同期 I/O のチューニング

Oracle Database for Tru64 UNIX システムでは、同期 I/O または非同期 I/O を実行できます。パフォーマンスを向上させるには、非同期 I/O を使用することをお勧めします。非同期 I/O を有効にするには、DISK\_ASYNC\_IO 初期化パラメータを true に設定します。

Oracle Database では、自動ストレージ管理ディスク・グループ内の AdvFS ファイル・システムまたはクラスタ・ファイル・システム (CFS)、あるいは RAW デバイスに格納されているデータファイルに対して、非同期 I/O を使用できます。非同期 I/O のパフォーマンスを最適化するには、いくつかのカーネル・サブシステム属性をチューニングする必要があります。

### F.4.1 aio\_task\_max\_num 属性

単一インスタンスの aio\_task\_max\_num カーネル・サブシステム属性は、8193 に設定してください。

aio\_task\_max\_num 属性の設定は、単一ノードで複数の Oracle Database インスタンスを使用する場合など、非同期 I/O を使用する他のすべてのアプリケーションを考慮して調整する必要があります。このパラメータの値は、アプリケーションが発行できる I/O 要求の最大数に設定してください。たとえば、3つのアプリケーションが動作しているときに、アプリケーション 1 が最大 10 個の非同期 I/O 要求を同時に発行でき、アプリケーション 2 が 100 個の非同期 I/O 要求を同時に発行でき、アプリケーション 3 が 1000 個の非同期 I/O 要求を同時に発行できる場合は、aio\_task\_max\_num パラメータを 1000 以上に設定します。

aio\_task\_max\_num 属性を設定しないと、Oracle Database のパフォーマンスが低下し、I/O エラーが発生する可能性があります。これらのエラーは、アラート・ログ・ファイルおよびトレース・ファイルに記録されます。

## F.5 ダイレクト I/O サポートおよびコンカレント・ダイレクト I/O サポート

この項では、ダイレクト I/O およびコンカレント I/O のサポートについて説明します。次の項目について説明します。

- [単一インスタンスの要件](#)
- [クラスタ・ファイル・システム](#)
- [Tru64 UNIX V5.1B クラスタ・ファイル・システム](#)
- [ダイレクト I/O サポートの無効化](#)

### F.5.1 単一インスタンスの要件

単一インスタンスをインストールする場合の Oracle Database の要件は、次のとおりです。

- Tru64 UNIX V5.1B 以上で、適切なパッチ・キットを適用している必要があります。

**関連項目：** Tru64 UNIX のパッチ・キットの詳細は、『Oracle Database インストール・ガイド』を参照してください。

- Oracle データファイルは、AdvFS ファイル・システムまたは自動ストレージ管理ディスク・グループに格納されている必要があります。
- AdvFS ファイル・システムを使用するディスクは、Oracle Database インスタンスを実行しているコンピュータに物理的に接続する必要があります。これには、ファイバー・チャネルによって接続されるディスクも含まれます。物理的な接続がなく、I/O 処理を別のノードで行う必要がある場合は除きます。

Tru64 UNIX V5.1B 以上のシステムが非クラスタ化システム環境で稼動している場合は、ファイル・システム・キャッシュが使用されないため、RAW デバイスのほとんどは AdvFS ファイル・システムとダイレクト I/O によって処理されます。さらに、このファイル・システムを使用すると、データベース・ファイルの管理が簡素化されます。

## F.5.2 クラスタ・ファイル・システム

Tru64 UNIX V5.1B 以上のシステムは、クラスタ・ファイル・システム (CFS) に対応しています。CFS では、クラスタ内のすべてのノードで単一名前空間ファイル・システムが使用されています。クラスタにマウントされたすべてのファイル・システムは、同じクラスタ内のすべてのノードに自動的に公開されます。CFS ファイル・システムは、AdvFS ファイル・システムの最上位に位置するため、非クラスタ化システムの特性をほぼ継承します。

## F.5.3 Tru64 UNIX V5.1B クラスタ・ファイル・システム

CFS ファイル・システムは、現在コンカレント・ダイレクト I/O モデルに対応しているため、Tru64 UNIX V5.1B 以上のシステムでのみサポートされています。ドライブに物理的に接続されているノードは、所有元のノードに問い合わせなくてもそのファイル・システムにデータ I/O を発行できます。

アクセスまたは変更日の延長、クローズ、変更など、ファイルのメタデータを変更すると、その変更はすべて所有ノードによって処理されるため、クラスタのインターコネクトが飽和状態になる可能性があります。インターコネクトが飽和状態になると、CFS ファイル・システムでの CREATE TABLESPACE、ALTER TABLESPACE、ADD DATAFILE、ALTER DATABASE DATAFILE または RESIZE コマンドのパフォーマンスが、RAW デバイスよりも低下することがあります。

## F.5.4 ダイレクト I/O サポートの無効化

ダイレクト I/O サポートが有効になっている AdvFS ファイル・システム上で動作している Oracle Database は、RAW デバイスで動作している Oracle Database と同等のパフォーマンスを実現します。ほとんどの場合、ダイレクト I/O サポートが有効になっている AdvFS ボリュームに格納されている Oracle Database は、ダイレクト I/O サポートが無効になっている同等のデータベース以上のパフォーマンスを実現します。ただし、ダイレクト I/O サポートが有効になっている場合は、次の作業負荷条件によってパフォーマンスが低下する可能性があります。

- 読取り / 書込みの比率が高い
- 問合せでパラレル問合せスレーブが使用されるため、Oracle データ・ブロックが SGA にキャッシュされない
- UNIX パツファ・キャッシュ (UBC) のサイズが数 MB 以上ある
- 全表スキャンによって、同じ表セットが繰り返シスキャンされる
- スキャンされる表が UBC にキャッシュされる

ダイレクト I/O サポートが無効になっている場合、前述のほとんどの条件に当てはまる作業負荷は UBC に大きく依存します。スキャンされる表のすべてが UBC にキャッシュされるわけではありませんが、ほとんどの場合、パラレル問合せによって発行された I/O 要求は UBC から返されます。したがって、データをディスクから読み込む場合、つまりダイレクト I/O が有効になっている場合に比べ、問合せがより速く返されます。

ダイレクト I/O サポートが有効になっている場合、Oracle データ・ブロックは UBC にキャッシュされません。それらのデータ・ブロックは、プロセス専用メモリーに読み込まれます。つまり、以前にスキャンした表を読み込む問合せの場合も、ディスクへの I/O 要求を実行して、データを取得する必要があります。ディスク I/O 待機時間は、メモリー待機時間よりも大幅に長くなります。このため、問合せの実行速度が遅くなり、パフォーマンスが低下します。

作業負荷が前述のほとんどの条件に当てはまる場合は、ダイレクト I/O サポートを無効にすることによってパフォーマンスが改善する可能性があります。ただし、多くの場合、様々な種類の問合せが同時に実行されています。データを単に読み込む問合せもある一方で、データを挿入、変更または削除する問合せもあります。これらの問合せの比率は環境ごとに異なります。一般に、OLTP 作業負荷の比率が高い環境では、ダイレクト I/O サポートを無効にしてもパフォーマンスは改善しません。

Oracle Database 10g では、ダイレクト I/O サポートがデフォルトで有効になっています。Oracle9i リリース 1 (9.0.1) でダイレクト I/O サポートを無効にするための `_TRU64_DIRECTIO_DISABLED` 初期化パラメータ (マニュアルには記載されていません) は、Oracle

Database 10g では削除されました。かわりに、汎用の FILESYSTEMIO\_OPTIONS 初期化パラメータが使用されます。次の表に、Tru64 UNIX で解釈される FILESYSTEMIO\_OPTIONS 初期化パラメータの有効な値を示します。

値	説明
directIO	AdvFS ファイル・システムのファイルの I/O に対して、ダイレクト I/O サポートを有効にし、非同期 I/O サポートを無効にします。
asynch	none と等価です。AdvFS ファイルに対して、ダイレクト I/O サポートが有効である場合にのみ非同期 I/O サポートを有効にします。
setall	AdvFS ファイルに対して、ダイレクト I/O サポートおよび非同期 I/O サポートを有効にします。これはデフォルトのオプションです。
none	AdvFS ファイルに対して、ダイレクト I/O サポートおよび非同期 I/O サポートを無効にします。

**関連項目：** FILESYSTEMIO\_OPTIONS 初期化パラメータの詳細は、『Oracle Database リファレンス』を参照してください。

DISK\_ASYNC\_IO 初期化パラメータは、ファイル・システムまたは RAW デバイスのいずれのファイルであっても、すべてのデータベース・ファイルの非同期 I/O 状態を制御します。したがって、DISK\_ASYNC\_IO 初期化パラメータが false に設定されている場合、ファイル・システムのファイルへの I/O 要求はすべて、FILESYSTEMIO\_OPTIONS 初期化パラメータの値に関係なく同期化されます。DISK\_ASYNC\_IO 初期化パラメータは、デフォルトで true に設定されています。

## F.6 リアルタイム・クロックへのアクセスの有効化

Oracle プロセスの多くは、特に TIMED\_STATISTICS 初期化パラメータが true に設定されている場合は、各種処理時間が測定されます。これらのタイミング機能は、Tru64 UNIX カーネルを呼び出すため、Oracle Database のパフォーマンスに影響する可能性があります。Tru64 UNIX では、プロセスからリアルタイム・クロックに直接アクセスできるため、負荷の大きいシステムのパフォーマンスを改善できます。

リアルタイム・クロックへのアクセスを有効にするには、次の手順を実行します。

1. root ユーザーでログインします。
2. 次のコマンドを実行します。

```
# mknod /dev/timedev c 15 0
# chmod a+r /dev/timedev
```

---

**注意：** 特殊ファイル /dev/timedev は、再起動後もシステムに残ります。

---

3. Oracle Database インスタンスを再起動します。

/dev/timedev ファイルが存在しているかどうかは、インスタンスの起動時にのみチェックされます。

## F.7 RAW デバイスの設定

---



---

**注意：** RAW デバイスの設定には、経験の豊富なシステム管理者の協力と、使用しているシステムに関する専門知識が必要です。

---



---



---



---

**注意：** データベース・ファイル管理作業を簡素化するために、ダイレクト I/O には RAW デバイスよりも自動ストレージ管理または AdvFS を使用することをお勧めします。

---



---

Tru64 UNIX システム上で RAW デバイスおよび RAW ボリュームを設定するには、次の手順を実行します。

1. RAC を使用している場合、追加するパーティションは共有ディスク上に配置します。
2. 空きディスク・パーティションの名前を確認します。

空きディスク・パーティションは、Tru64 UNIX ファイル・システムに使用されていないパーティションで、次の制限に準拠している必要があります。

- /usr/sbin/mount コマンドの実行で、リストに表示されないこと。
- スワップ・デバイスとして使用されていないこと。
- スワップ・パーティションをオーバーラップしていないこと。
- 他の Tru64 UNIX アプリケーション（Oracle Database の他のインスタンスなど）で使用されていないこと。
- Tru64 UNIX ファイル・システムをオーバーラップしていないこと。
- ファイル・システムによってすでに使用されている領域を使用しないこと。

パーティションが空いているかどうかを判断するには、デバイス上のパーティションの開始位置とサイズを示す詳細なマップを出力し、空き領域を確認します。パーティションの中には、現在マウントされずに /usr/sbin/mount の出力に表示されないファイル・システムもあります。

---



---

**注意：** パーティションがシリンダ 0（ゼロ）で開始していないことを確認してください。

---



---

3. Oracle Database で使用する RAW デバイスを設定します。

ディスクがパーティション化されているかどうかを確認します。パーティション化されていない場合は、disklabel コマンドを使用してディスクをパーティション化します。

4. ls コマンドを実行し、デバイス・ファイルの所有者と権限を表示します。たとえば、次のように入力します。

```
$ ls -la
```

5. パーティションの所有者が Oracle ソフトウェア所有者であることを確認します。必要に応じて、chown コマンドを使用して、デバイスのブロック・ファイルおよびキャラクタ・ファイルの所有者を変更します。たとえば、次のように入力します。

```
# chown oracle:dba /dev/rdisk/dsk10c
```

6. パーティションに正しい権限が付与されていることを確認します。必要に応じて、chmod コマンドを使用して、Oracle ソフトウェア所有者のみがそのパーティションにアクセスできるようにします。たとえば、次のように入力します。

```
# chmod 600 /dev/rdisk/dsk10c
```

7. 目的の RAW デバイスへのシンボリック・リンクを作成します。たとえば、次のように入力します。

```
$ ln -s /dev/rdisk/dsk10c /oracle_data/datafile.dbf
```

シンボリック・リンクが作成されたことを確認するには、文字型特殊デバイス（ブロック型特殊デバイスではなく）を使用し、次のコマンドを実行します。

```
$ ls -l datafile
```

次の出力が表示されます。

```
crwxrwxrwx oracle dba datafile
```

---

**注意：** このシンボリック・リンクは、クラスタの各ノードに設定する必要があります。複数のシンボリック・リンクが同一の RAW デバイスを指定していないことを確認してください。

---

8. 新しいパーティションを新しいデータベースに作成または追加します。  
新しいパーティションを作成するには、SQL\*Plus から次のコマンドを実行します。

---

**注意：** RAW パーティションに作成する Oracle データファイルのサイズは、RAW パーティションのサイズと比較して、64KB に Oracle ブロック・サイズ 1 つ分を加えた値以上小さくする必要があります。

---

```
SQL> CREATE DATABASE sid
  2 LOGFILE '/oracle_data/log1.dbf' SIZE 100K
  3 '/oracle_data/log2.dbf' SIZE 100K
  3 DATAFILE '/oracle_data/datafile.dbf' SIZE 10000K REUSE;
```

パーティションを既存の Oracle Database の表領域に追加するには、次のコマンドを実行します。

```
SQL> ALTER TABLESPACE tablespace_name
  2 ADD DATAFILE '/dev/rdisk/dsk10c' SIZE 10000K REUSE;
```

同じ手順を使用して、REDO ログ・ファイルの RAW デバイスを設定できます。

## F.8 Spike 最適化ツール

Spike 最適化ツール (Spike) は、Tru64 UNIX バイナリのパフォーマンスを改善するパフォーマンス最適化ツールです。OLTP 作業負荷を伴ったテスト環境で Spike (フィードバック付き) を使用したところ、Oracle Database のパフォーマンスが最大 23 パーセント向上しました。

Spike の詳細は、Tru64 UNIX のドキュメントを参照するか、または次のいずれかのコマンドを実行してください。

- `man spike`
- `spike`

Oracle Database には、Spike V5.2 (510 USG、GEM: 48C5K、LIBMLD: 2.4、DATE: 2003 年 9 月 28 日) 以上が必要です。

---

**注意:** Spike のバージョンが V5.2 (510 USG、GEM: 48C5K、LIBMLD: 2.4、DATE: 2003 年 9 月 28 日) より前の場合は、HP 社に連絡してパッチ・キットを入手してください。

---

Spike のバージョンを確認するには、次のコマンドを実行します。

```
$ spike -V
```

HP 社の Web サイトから最新バージョンの Spike をダウンロードできます。

---

**注意:** オラクル社では、`spike` コマンドを使用して最適化された Oracle 実行可能ファイルをサポートしていません。Spike を使用して最適化された Oracle Database バイナリに問題がある場合は、元の最適化されていないバイナリを使用してその問題を再現してください。問題が解決しない場合は、Oracle サポート・サービスにご連絡ください。

---

### F.8.1 Spike の使用

この項では、Spike に必要なシステム・リソース、Spike 最適化フラグを使用する方法と理由、および Spike を実行するいくつかの方法について説明します。

#### システム・リソースの設定

表 F-1 に、Spike の実行に必要なシステム・リソースを示します。

**表 F-1 Spike のシステム・リソース要件**

リソース	最小値
物理メモリー	1024MB
max-per-proc-address-space サブシステム属性の値	1024MB
max-per-proc-data-space サブシステム属性の値	1024MB
vm-maxvas サブシステム属性の値	1024MB

/etc/sysconfigtab ファイルのこれらのサブシステム属性値を設定するには、次の行を追加します。

```
proc:
    max-per-proc-address-space = 0x40000000
    max-per-proc-data-size = 0x40000000
vm:
    vm-maxvas = 0x40000000
```

シェル環境の制限を最大値に設定します。C シェルの場合は、次のコマンドを実行します。

```
% limit datasize unlimited
% limit memoryuse unlimited
% limit vmemoryuse unlimited
```

スタックサイズの制限を高く設定しすぎると、Spike によって仮想メモリー不足が発生する可能性があります。この問題を回避するには、次の C シェル・コマンドを実行します。

```
% limit stacksize 8192
```

## 最適化フラグのチェック

Spike には、多数の最適化フラグが用意されています。ただし、spike コマンドによる最適化すべてを Oracle Database に使用できるわけではありません。Oracle Database では、次の Spike 最適化フラグの実行が認定されています。

```
-arch, -controlOpt, -fb, -feedback, -map, -nosplit, -nochain, -noporder,
-noaggressiveAlign, -o, optThresh, -splitThresh, -symbols_live, -tune, -v, -V
```

Spike を実行すると、最適化フラグのコピーが、最適化しているバイナリのイメージ・ヘッダー・コメント・セクションに格納されます。Oracle Database では、インスタンスの起動時に、使用される Spike 最適化をチェックします。Oracle Database で認定されていない最適化が検出された場合、または以前に OM (HP 社製の Spike の先行モデル) を使用してバイナリを最適化していた場合は、インスタンスの起動に失敗し、ORA-04940 エラー・メッセージが表示されます。インスタンスの起動に失敗した場合は、アラート・ログ・ファイルの詳細情報を確認してください。

---

**注意：** Oracle Database では、Spike 最適化フラグ `-symbols_live` を使用する必要があります。

---

## Spike の実行

Spike を使用して実行可能ファイルを最適化するには、次のいずれかの方法を使用します。

- 静的 Spike
- フィードバック付き Spike の実行

静的 Spike は、いくつかの手順の設定のみで、フィードバック付き Spike の約半分のパフォーマンスを実現できます。

フィードバック付き Spike は、静的 Spike によるすべての最適化以外に、作業負荷関連の最適化をいくつか実行できます。フィードバック付き Spike を実行すると、パフォーマンスは最も向上しますが、静的 Spike に比べて多くの操作を必要とします。

フィードバック付き Spike および静的 Spike を両方実行する場合は、Spike を実行した Oracle バイナリをテスト環境で実行してから、本番環境に移行することをお勧めします。

### 静的 Spike

静的 Spike では、グローバル・ポインタ (gp) レジスタの操作や CPU アーキテクチャの有効利用など、作業負荷以外の最適化が実行されます。テスト環境では、Spike によるパフォーマンス最適化の約半分が、静的 Spike で実現されました。さらに、静的 Spike は比較的わかりやすく、簡単に実行できます。静的 Spike は、操作が簡単でパフォーマンスの向上度が大きく、実用的です。

静的 Spike を使用するには、次の手順を実行します。

1. データベースをシャットダウンします。
2. 次のコマンドを入力して、oracle イメージを Spike します。

```
$ spike oracle -o oracle.spike -symbols_live
```

3. 次のコマンドを入力して、元のイメージを保存し、Spike されたイメージへのシンボリック・リンクを作成します。

```
$ mv oracle oracle.orig
$ ln -s oracle.spike oracle
```

4. データベースを起動します。

---

**注意：** オラクル社カスタマ・サポート・センターに連絡するときは、元のイメージを使用して問題を再現してください。

---

### フィードバック付き Spike の実行

フィードバック付き Spike では、静的 Spike で実行できるすべての最適化以外に、ホット / コールド基本ブロック移動などの作業負荷に関連した最適化も実行されます。テスト環境では、Spike によるパフォーマンス最適化の約半分が、フィードバック情報に依存する最適化で実現されました。フィードバック付き Spike では、静的 Spike に比べて多数の手順と多くの操作が必要です。しかし、パフォーマンスを重視する場合は、工数は余分に必要ですが、静的 Spike のみでは得られない効果を期待できます。

フィードバック付き Spike を使用するには、次の手順を実行します。

1. 次のコマンドを入力して、Oracle バイナリをインストールメント処理します。

```
$ pixie -output oracle.pixie -dirname dir -pids oracle_image
```

この例の `oracle_image` は元のイメージです。 `dir` は、インストールメント処理された実行可能プログラムがプロファイルしているデータファイルを書き込むディレクトリの名前です。

---

**注意：** `-dirname` オプションを指定すると、`oracle.Counts.pid` ファイルが `dir` ディレクトリに保存されます。これらのファイルは容量が大きく、作業負荷に応じてファイル数も多くなるため、ディレクトリに十分なディスク領域が確保されていることを確認してください。

---

この手順では、後で必要になる `oracle.Addr`s ファイルも作成されます。

`pixie` コマンドの出力には、エラーが含まれている場合があります。これらのエラーは無視してもかまいません。

2. データベースをシャットダウンします。
3. 次のコマンドを入力して、元のイメージを保存し、`pixie` イメージへのシンボリック・リンクを作成します。

```
$ mv oracle oracle.orig
$ ln -s oracle.pixie oracle
```

4. データベースを起動し、対応する作業負荷を実行します。

`pixie` 実行可能プログラムは容量が大きく処理速度が遅いため、標準の実行可能プログラムを使用した場合と同じ数のユーザーは実行できません。Oracle Database を使用しているときは、複数の `oracle.Counts.pid` ファイルが作成されます。 `pid` は、対応する Oracle プロセスのプロセス ID です。最適化する各 Oracle プロセスのプロセス ID を追跡します。これらは、クライアントの Oracle シャドウ・プロセスになります。

5. データベースをシャットダウンします。
6. 次のコマンドを入力し、シンボリック・リンクを作成して、元の実行可能ファイルと置換します。

```
$ ln -s oracle.orig oracle
```

7. 特定の `oracle.Counts.pid` ファイルを代表的な作業負荷として選択できる場合は、手順 a を実行します。複数のカウント・ファイルをマージして作業負荷を表現する場合は、手順 b を実行します。
- a. `pixie` コマンドによって作成された `oracle.Addrs` ファイル、`oracle.Counts.pid` ファイル、および元の Oracle 実行可能ファイルが使用可能であることを確認します。
- プロセス ID (`pid`) を使用して代表的な `oracle.Counts.pid` ファイルを選択し、次のコマンドを入力してコピーします。
- ```
$ cp oracle.Counts.pid oracle.Counts
```
- b. `prof` ユーティリティを使用して、複数の `oracle.Counts.pid` ファイルをマージします。このユーティリティの詳細は、`prof` の `man` ページを参照してください。
- パラレル問合せオプションを使用している場合は、問合せスレーブと問合せコーディネータによって生成された `oracle.Counts.pid` ファイルをマージします。マージされたファイルは、問合せ開始クライアントの Oracle シャドウ・プロセスとなります。
- パラレル問合せオプションを使用していない場合は、各 Oracle フォアグラウンド・プロセスからメモリー使用量の最も多い `oracle.Counts.pid` ファイルを選択してマージします。
- `oracle.Counts.pid` ファイルをマージするには、次のコマンドを実行します。
- ```
$ prof -pixie -merge oracle.Counts $ORACLE_HOME/bin/oracle \
oracle.Addrs oracle.Counts.pid1 oracle.Counts.pid2
```
8. `oracle.Addrs` および `oracle.Counts` ファイルがカレント・ディレクトリで使用可能になっていることを確認し、フィードバック情報を使用して `Spike` を実行します。次のコマンドを入力します。
- ```
$ spike oracle -fb oracle -o oracle.spike_fb -symbols_live
```
- `spike` コマンドの出力には、エラーが含まれている場合があります。これらのエラーは無視してもかまいません。
9. 次のコマンドを入力し、新しい `oracle` イメージへのシンボリック・リンクを作成します。
- ```
$ ln -s oracle.spike_fb oracle
```
10. データベースを起動します。



---

---

# Oracle ODBC Driver の使用

---

---

**注意：** Oracle ODBC ドライバは、Linux x86、Linux Itanium および Solaris SPARC 64 の各プラットフォームに対してのみサポートされます。

---

---

この付録では、Oracle ODBC Driver の使用方法について説明します。次の項目について説明します。

- サポートされていない機能
- データ型の実装
- データ型に関する制限事項
- `SQLDriverConnect` 関数の接続文字列の書式
- プログラムでのロック・タイムアウトの削減
- ODBC アプリケーションのリンク
- ROWID に関する情報の取得
- WHERE 句の ROWID
- 結果セットの有効化
- EXEC 構文の有効化
- サポートされている機能
- Unicode のサポート
- パフォーマンスとチューニング
- エラー・メッセージ

## G.1 サポートされていない機能

Oracle ODBC Driver は、次の ODBC 3.0 機能をサポートしていません。

- 間隔データ型
- SQL\_C\_UBIGINT および SQL\_C\_SBIGINT C データ型識別子
- 共有接続
- 共有環境
- SQLSetConnectAttr の SQL\_LOGIN\_TIMEOUT 属性
- 期限切れパスワード・オプション

Oracle ODBC Driver は、次の表に示す SQL 関数をサポートしていません。

文字列関数	数値関数	時間関数、日付関数および間隔関数
BIT_LENGTH	ACOS	CURRENT_DATE
CHAR_LENGTH	ASIN	CURRENT_TIME
CHARACTER_LENGTH	ATAN	CURRENT_TIMESTAMP
DIFFERENCE	ATAN2	EXTRACT
OCTET_LENGTH	COT	TIMESTAMPDIFF
POSITION	DEGREES	
	RADIANS	
	RAND	
	ROUND	

## G.2 データ型の実装

この項では、DATE、TIMESTAMP および浮動小数点の各データ型について説明します。

### DATE および TIMESTAMP

Oracle の DATE および TIMESTAMP データ型のセマンティクスは、同名の ODBC データ型と必ずしも正確に対応していません。Oracle の DATE データ型には、日付と時間の両方の情報が格納されています。これに対して、SQL\_DATE データ型に格納されているのは、日付情報のみです。Oracle の TIMESTAMP データ型にも日付と時間の情報が格納されていますが、その小数秒の精度は他方に比較して高くなります。Oracle ODBC Driver は、情報が失われるのを防ぐために、Oracle の DATE 列と TIMESTAMP 列の両方のデータ型を SQL\_TIMESTAMP として報告します。同様に、Oracle ODBC Driver は SQL\_TIMESTAMP パラメータを Oracle の TIMESTAMP 値としてバインドします。

**関連項目：** パフォーマンスとチューニングに関する DATE および TIMESTAMP データ型の詳細は、G-22 ページの「[DATE および TIMESTAMP データ型](#)」を参照してください。

### 浮動小数点データ型

リリース 10.1 以上の Oracle Database に接続すると、Oracle ODBC Driver は、Oracle の浮動小数点データ型 BINARY\_FLOAT と BINARY\_DOUBLE を ODBC データ型 SQL\_REAL と SQL\_DOUBLE にそれぞれマップします。これより前のリリースでは、SQL\_REAL と SQL\_DOUBLE が Oracle の汎用数値データ型にマップされていました。

## G.3 データ型に関する制限事項

Oracle ODBC Driver および Oracle Database には、データ型に関する制限事項があります。次の表に制限事項を示します。

制限があるデータ型	説明
リテラル	Oracle Database では、SQL 文のリテラルは 4000 バイトに制限されています。
SQL_LONGVARCHAR および SQL_WLONGVARCHAR	Oracle では、列型が LONG の SQL_LONGVARCHAR データは 2,147,483,647 バイトに制限されます。また、Oracle では、列型が CLOB の SQL_LONGVARCHAR データは 4GB に制限されます。これは、クライアント・ワークステーションのメモリーによる制限です。
SQL_LONGVARCHAR および SQL_LONGVARBINARY	Oracle Database で使用できるのは、表ごとに 1 列の LONG データ列のみです。LONG データ型とは、SQL_LONGVARCHAR (LONG) および SQL_LONGVARBINARY (LONG RAW) です。かわりに、CLOB および BLOB 列の使用をお勧めします。1 つの表で使用できる CLOB および BLOB の列数に制限はありません。

## G.4 SQLDriverConnect 関数の接続文字列の書式

SQLDriverConnect 関数は、Oracle ODBC Driver によって実装される関数の 1 つです。次の表に、SQLDriverConnect 関数コールの接続文字列の引数に含めることができるキーワードを示します。

キーワード	意味	値
DSN	ODBC データソース名	ユーザー指定の名前。 これは必須のキーワードです。
DBQ	TNS サービス名	ユーザー指定の名前。 これは必須のキーワードです。
UID	ユーザー ID または ユーザー名	ユーザー指定の名前。 これは必須のキーワードです。
PWD	パスワード	ユーザー指定の名前。 パスワードがない場合は PWD=; と指定します。 これは必須のキーワードです。
DBA	データベース属性	W は書込みアクセスを意味します。 R は読取り専用アクセスを意味します。
APA	アプリケーション属性	T はスレッド・セーフティを有効にすることを意味します。 F はスレッド・セーフティを無効にすることを意味します。
RST	結果セット	T は結果セットを有効にすることを意味します。 F は結果セットを無効にすることを意味します。
QTO	問合せタイムアウト・ オプション	T は問合せタイムアウトを有効にすることを意味します。 F は問合せタイムアウトを無効にすることを意味します。
CSR	カーソルのクローズ	T はカーソルのクローズを有効にすることを意味します。 F はカーソルのクローズを無効にすることを意味します。

キーワード	意味	値
BAM	バッチ自動コミット・モード	IfAllSuccessful は、すべての文が成功した場合のみコミットすることを意味します (古い動作)。 UpToFirstFailure は、最初に失敗した文までコミットすることを意味します。これは ODBC バージョン 7 の動作です。 AllSuccessful は、成功したすべての文をコミットすることを意味します。
FBS	フェッチ・バッファ・サイズ	ユーザー指定の数値 (0 バイト以上の値を指定します)。 デフォルトは 60,000 バイトです。
FEN	フェイルオーバー	T はフェイルオーバーを有効にすることを意味します。 F はフェイルオーバーを無効にすることを意味します。
FRC	フェイルオーバー再試行数	ユーザー指定の数値。 デフォルトは 10 です。
FDL	フェイルオーバーの遅延	ユーザー指定の数値。 デフォルトは 10 です。
LOB	LOB 書込み	T は LOB を有効にすることを意味します。 F は LOB を無効にすることを意味します。
FWC	SQL_WCHAR 強制サポート	T は Force SQL_WCHAR を有効にすることを意味します。 F は Force SQL_WCHAR を無効にすることを意味します。
EXC	EXEC 構文	T は EXEC 構文を有効にすることを意味します。 F は EXEC 構文を無効にすることを意味します。
XSM	スキーマ・フィールド	Default はデフォルト値が使用されることを意味します。 Database はデータベース名が使用されることを意味します。 Owner は所有者名が使用されることを意味します。
MDI	METADATA ID デフォルトの設定	T は SQL_ATTR_METADATA_ID のデフォルト値が SQL_TRUE であることを意味します。 F は SQL_ATTR_METADATA_ID のデフォルト値が SQL_FALSE であることを意味します。
DPM	SQLDescribeParam の無効化	T は SQLDescribeParam を無効にすることを意味します。 F は SQLDescribeParam を有効にすることを意味します。
BTD	TIMESTAMP を DATE としてバインド	T は SQL_TIMESTAMP が Oracle の DATE としてバインドされることを意味します。 F は SQL_TIMESTAMP が Oracle の TIMESTAMP としてバインドされることを意味します。
NUM	数値の設定	NLS は、グローバリゼーション・サポートの数値の設定が使用されることを意味します (小数点およびグループのセパレータを判断するため)。

## G.5 プログラムでのロック・タイムアウトの削減

Oracle Database は、トランザクション間のロックの競合が解決されるまで無期限に待機します。ただし、Oracle Database がロックの解決を待機する時間は制限できます。制限するには、ODBC の `SQLSetStmtAttr` 関数のコール時に `SQL_ATTR_QUERY_TIMEOUT` 属性を設定してから、データソースに接続します。

## G.6 ODBC アプリケーションのリンク

プログラムをリンクする場合は、そのプログラムを Driver Manager ライブラリ `libodbc.so` にリンクする必要があります。

## G.7 ROWID に関する情報の取得

ODBC の `SQLSpecialColumns` 関数は、表内の列に関する情報を返します。Oracle ODBC Driver で使用すると、この関数は Oracle 表に関連付けられた Oracle ROWID に関する情報を返します。

## G.8 WHERE 句の ROWID

ROWID は、SQL 文の WHERE 句で使用できます。ただし、ROWID 値はパラメータ・マーク内存在している必要があります。

## G.9 結果セットの有効化

Oracle 参照カーソル（結果セットとも呼ばれます）によって、アプリケーションでは、ストアード・プロシージャとストアード・ファンクションを使用してデータを取得できます。ここでは、参照カーソルを使用して ODBC を介して結果セットを有効にする方法を説明します。

- ストアド・プロシージャをコールするには、ODBC 構文を使用する必要があります。システム固有の PL/SQL は、ODBC を介してサポートされていません。次のコード例は、プロシージャまたはファンクションを、パッケージなしでコールする方法、およびパッケージ内でコールする方法を示します。このコード例でのパッケージ名は `RSET` です。

```
Procedure call:
{CALL Example1(?)}
{CALL RSET.Example1(?)}
Function Call:
{? = CALL Example1(?)}
{? = CALL RSET.Example1(?)}
```

- PL/SQL の参照カーソル・パラメータは、プロシージャのコール時には省略されます。たとえば、プロシージャ `Example2` には 4 つのパラメータが定義されているとします。パラメータ 1 と 3 は参照カーソル・パラメータで、パラメータ 2 と 4 は文字列です。コールは、次のように指定されます。

```
{CALL RSET.Example2("Literal 1", "Literal 2")}
```

次のサンプル・アプリケーションは、Oracle ODBC Driver を使用して結果セットを返す方法を示します。

```
/*
 * Sample Application using Oracle reference cursors through ODBC
 *
 * Assumptions:
 *
 * 1) Oracle Sample database is present with data loaded for the EMP table.
 *
 * 2) Two fields are referenced from the EMP table, ename and mgr.
 */
```

```
* 3) A data source has been setup to access the sample database.
*
*
* Program Description:
*
* Abstract:
*
* This program demonstrates how to return result sets using
* Oracle stored procedures
*
* Details:
*
* This program:
* Creates an ODBC connection to the database.
* Creates a Packaged Procedure containing two result sets.
* Executes the procedure and retrieves the data from both result sets.
* Displays the data to the user.
* Deletes the package then logs the user out of the database.
*
*
* The following is the actual PL/SQL this code generates to
* create the stored procedures.
*
DROP PACKAGE  ODBCRefCur;
CREATE PACKAGE ODBCRefCur AS
    TYPE ename_cur IS REF CURSOR;
    TYPE mgr_cur   IS REF CURSOR;
PROCEDURE EmpCurs(ENAME IN OUT ename_cur, Mgr IN OUT mgr_cur, pjob IN VARCHAR2);

END;

/
CREATE PACKAGE BODY ODBCRefCur AS
PROCEDURE EmpCurs(ENAME IN OUT ename_cur, Mgr IN OUT mgr_cur, pjob IN VARCHAR2)
AS
    BEGIN
        IF NOT ENAME%ISOPEN
        THEN
            OPEN ENAME FOR SELECT ename FROM emp;
        END IF;

        IF NOT MGR%ISOPEN
        THEN
            OPEN MGR FOR SELECT mgr FROM emp WHERE job = pjob;
        END IF;
    END;
END;

/

*
* End PL/SQL for Reference Cursor.
*/

/*
* Include Files
*/
#include <stdio.h>
#include <sql.h>
#include <sqlext.h>
/*
* Defines
*/
```

```
#define JOB_LEN 9
#define DATA_LEN 100
#define SQL_STMT_LEN 500
/*
 * Procedures
 */
void DisplayError( SWORD HandleType, SQLHANDLE hHandle, char *Module );
/*
 * Main Program
 */
int main()
{
SQLHENV hEnv;
SQLHDBC hDbc;
SQLHSTMT hStmt;
SQLRETURN rc;
char *DefUserName ="scott";
char *DefPassWord ="tiger";
SQLCHAR ServerName[DATA_LEN];
SQLCHAR *pServerName=ServerName;
SQLCHAR UserName[DATA_LEN];
SQLCHAR *pUserName=UserName;
SQLCHAR PassWord[DATA_LEN];
SQLCHAR *pPassWord=PassWord;
char Data[DATA_LEN];
SQLINTEGER DataLen;
char error[DATA_LEN];
char *charptr;
SQLCHAR SqlStmt[SQL_STMT_LEN];
SQLCHAR *pSqlStmt=SqlStmt;
char *pSalesMan = "SALESMAN";
SQLINTEGER sqlnts=SQL_NTS;
/*
 * Allocate the Environment Handle
 */
rc = SQLAllocHandle( SQL_HANDLE_ENV, SQL_NULL_HANDLE, &hEnv );
if (rc != SQL_SUCCESS)
{
printf( "Cannot Allocate Environment Handle\n");
printf( "\nHit Return to Exit\n");
charptr = gets ((char *)error);
exit(1);
}
/*
 * Set the ODBC Version
 */
rc = SQLSetEnvAttr( hEnv,SQL_ATTR_ODBC_VERSION,(void *)SQL_OV_ODBC3,0);
if (rc != SQL_SUCCESS)
{
printf( "Cannot Set ODBC Version\n");
printf( "\nHit Return to Exit\n");
charptr = gets ((char *)error);
exit(1);
}
/*
 * Allocate the Connection handle
 */
rc = SQLAllocHandle( SQL_HANDLE_DBC, hEnv, &hDbc );
if (rc != SQL_SUCCESS)
{
printf( "Cannot Allocate Connection Handle\n");
printf( "\nHit Return to Exit\n");
charptr = gets ((char *)error);
}
```

```
    exit(1);
}
/*
 * Get User Information
 */
strcpy ((char *) pUserName, DefUserName );
strcpy ((char *) pPassWord, DefPassWord );
/*
 * Data Source name
 */
printf( "\nEnter the ODBC Data Source Name\n" );
charptr = gets ((char *) ServerName);
/*
 * User Name
 */
printf ( "\nEnter User Name Default [%s]\n", pUserName);
charptr = gets ((char *) UserName);
if (*charptr == '\0')
{
    strcpy ((char *) pUserName, (char *) DefUserName );
}
/*
 * Password
 */
printf ( "\nEnter Password Default [%s]\n", pPassWord);
charptr = gets ((char *) PassWord);
if (*charptr == '\0')
{
    strcpy ((char *) pPassWord, (char *) DefPassWord );
}
/*
 * Connection to the database
 */
rc = SQLConnect( hDbc,pServerName,(SQLSMALLINT) strlen((char
*)pServerName),pUserName,(SQLSMALLINT) strlen((char
*)pUserName),pPassWord,(SQLSMALLINT) strlen((char *)pPassWord));
if (rc != SQL_SUCCESS)
{
    DisplayError(SQL_HANDLE_DBC, hDbc, "SQLConnect");
}
/*
 * Allocate a Statement
 */
rc = SQLAllocHandle( SQL_HANDLE_STMT, hDbc, &hStmt );
if (rc != SQL_SUCCESS)
{
    printf( "Cannot Allocate Statement Handle\n");
    printf( "\nHit Return to Exit\n");
    charptr = gets ((char *)error);
    exit(1);
}
/*
 * Drop the Package
 */
strcpy( (char *) pSqlStmt, "DROP PACKAGE ODBCRefCur");
rc = SQLExecDirect(hStmt, pSqlStmt, strlen((char *)pSqlStmt));
/*
 * Create the Package Header
 */
strcpy( (char *) pSqlStmt, "CREATE PACKAGE ODBCRefCur AS\n");
strcat( (char *) pSqlStmt, " TYPE ename_cur IS REF CURSOR;\n");
strcat( (char *) pSqlStmt, " TYPE mgr_cur IS REF CURSOR;\n\n");
strcat( (char *) pSqlStmt, " PROCEDURE EmpCurs (Ename IN OUT ename_cur,");
```

```

strcat( (char *) pSqlStmt, "Mgr IN OUT mgr_cur,pjob IN VARCHAR2);\n\n");
strcat( (char *) pSqlStmt, "END;\n");
rc = SQLExecDirect(hStmt, pSqlStmt, strlen((char *)pSqlStmt));
if (rc != SQL_SUCCESS)
{
    DisplayError(SQL_HANDLE_STMT, hStmt, "SQLExecDirect");
}
/*
 * Create the Package Body
 */
strcpy( (char *) pSqlStmt, "CREATE PACKAGE BODY ODBCRefCur AS\n");
strcat( (char *) pSqlStmt, " PROCEDURE EmpCurs (Ename IN OUT ename_cur,");
strcat( (char *) pSqlStmt, "Mgr IN OUT mgr_cur, pjob IN VARCHAR2)\n AS\n BEGIN\n");
strcat( (char *) pSqlStmt, " IF NOT Ename%ISOPEN\n THEN\n");
strcat( (char *) pSqlStmt, " OPEN Ename for SELECT ename from emp;\n");
strcat( (char *) pSqlStmt, " END IF;\n\n");
strcat( (char *) pSqlStmt, " IF NOT Mgr%ISOPEN\n THEN\n");
strcat( (char *) pSqlStmt, " OPEN Mgr for SELECT mgr from emp where job = pjob;\n");
strcat( (char *) pSqlStmt, " END IF;\n");
strcat( (char *) pSqlStmt, " END;\n");
strcat( (char *) pSqlStmt, "END;\n");
rc = SQLExecDirect(hStmt, pSqlStmt, strlen((char *)pSqlStmt));
if (rc != SQL_SUCCESS)
{
    DisplayError(SQL_HANDLE_STMT, hStmt, "SQLExecDirect");
}
/*
 * Bind the Parameter
 */
rc = SQLBindParameter(hStmt,1,SQL_PARAM_INPUT,SQL_C_CHAR,SQL_CHAR,JOB_
LEN,0,pSalesMan,0,&sqlInts);
/*
 * Call the Store Procedure which executes the Result Sets
 */
strcpy( (char *) pSqlStmt, "{CALL ODBCRefCur.EmpCurs(?)}");
rc = SQLExecDirect(hStmt, pSqlStmt, strlen((char *)pSqlStmt));
if (rc != SQL_SUCCESS)
{
    DisplayError(SQL_HANDLE_STMT, hStmt, "SQLExecDirect");
}
/*
 * Bind the Data
 */
rc = SQLBindCol( hStmt,1,SQL_C_CHAR,Data,sizeof(Data),&DataLen);
if (rc != SQL_SUCCESS)
{
    DisplayError(SQL_HANDLE_STMT, hStmt, "SQLBindCol");
}
/*
 * Get the data for Result Set 1
 */
printf( "\nEmployee Names\n\n");
while ( rc == SQL_SUCCESS )
{
    rc = SQLFetch( hStmt );
    if ( rc == SQL_SUCCESS )
    {
        printf("%s\n", Data);
    }
    else
    {
        if (rc != SQL_NO_DATA)
        {

```

```
        DisplayError(SQL_HANDLE_STMT, hStmt, "SQLFetch");
    }
}
printf( "\nFirst Result Set - Hit Return to Continue\n");
charptr = gets ((char *)error);
/*
 * Get the Next Result Set
 */
rc = SQLMoreResults( hStmt );
if (rc != SQL_SUCCESS)
{
    DisplayError(SQL_HANDLE_STMT, hStmt, "SQLMoreResults");
}
/*
 * Get the data for Result Set 2
 */
printf( "\nManagers\n\n");
while ( rc == SQL_SUCCESS )
{
    rc = SQLFetch( hStmt );
    if ( rc == SQL_SUCCESS )
    {
        printf("%s\n", Data);
    }
    else
    {
        if (rc != SQL_NO_DATA)
        {
            DisplayError(SQL_HANDLE_STMT, hStmt, "SQLFetch");
        }
    }
}
printf( "\nSecond Result Set - Hit Return to Continue\n");
charptr = gets ((char *)error);
/*
 * Should Be No More Results Sets
 */
rc = SQLMoreResults( hStmt );
if (rc != SQL_NO_DATA)
{
    DisplayError(SQL_HANDLE_STMT, hStmt, "SQLMoreResults");
}
/*
 * Drop the Package
 */
strcpy( (char *) pSqlStmt, "DROP PACKAGE ODBCRefCur");
rc = SQLExecDirect(hStmt, pSqlStmt, strlen((char *)pSqlStmt));
/*
 * Free handles close connections to the database
 */
SQLFreeHandle( SQL_HANDLE_STMT, hStmt );
SQLDisconnect( hDbc );
SQLFreeHandle( SQL_HANDLE_DBC, hDbc );
SQLFreeHandle( SQL_HANDLE_ENV, hEnv );
printf( "\nAll Done - Hit Return to Exit\n");
charptr = gets ((char *)error);
return(0);
}
/*
 * Display Error Messages
 */
void DisplayError( SWORD HandleType, SQLHANDLE hHandle, char *Module )
```

```

{
SQLCHAR MessageText[255];
SQLCHAR SQLState[80];
SQLRETURN rc=SQL_SUCCESS;
long NativeError;
SWORD RetLen;
SQLCHAR error[25];
char *charptr;
rc =
SQLGetDiagRec(HandleType,hHandle,1,SQLState,&NativeError,MessageText,255,&RetLen);
printf( "Failure Calling %s\n", Module );
if (rc == SQL_SUCCESS || rc == SQL_SUCCESS_WITH_INFO)
{
printf( "\t\t\t State: %s\n", SQLState);
printf( "\t\t\t Native Error: %d\n", NativeError );
printf( "\t\t\t Error Message: %s\n", MessageText );
}
printf( "\nHit Return to Exit\n");
charptr = gets ((char *)error);
exit(1);
}

```

## G.10 EXEC 構文の有効化

使用している SQL Server の EXEC 文の構文を変更せずに同等の Oracle プロシージャ・コールに容易に変換できる場合、その構文は、このオプションを有効にすると、Oracle ODBC Driver で変換できます。

SQL Server プロシージャの完全な名前は、次に示す最大 4 つの識別子で構成されます。

- サーバー名
- データベース名
- 所有者名
- プロシージャ名

この名前の書式は次のとおりです。

```
[[[server.] [database.] [owner_name] .]procedure_name
```

Microsoft SQL Server データベースから Oracle Database に移行する際、各 SQL Server プロシージャまたはファンクションの定義は同等の Oracle Database 構文に変換され、Oracle Database のスキーマで定義されます。移行したプロシージャは、多くの場合、次のいずれかの方法で再編成（およびスキーマで作成）されます。

- すべてのプロシージャが 1 つのスキーマに移行されます（デフォルト・オプション）。
- 1 つの SQL Server データベースに定義されているすべてのプロシージャが、そのデータベース名の付いたスキーマに移行されます。
- 特定のユーザーが所有するすべてのプロシージャが、そのユーザー名の付いたスキーマに移行されます。

移行したプロシージャを編成するこれら 3 通りの方法をサポートするためには、いずれかのスキーマ名オプションを指定してプロシージャ名を変換できます。変換した Oracle プロシージャ・コールのオブジェクト名では、大 / 小文字が区別されません。

## G.11 サポートされている機能

この項では、Oracle ODBC Driver でサポートされている機能について説明します。次の項目について説明します。

- [API への準拠](#)
- [ODBC API 関数の実装](#)
- [ODBC SQL 構文の実装](#)
- [データ型の実装](#)

### G.11.1 API への準拠

Oracle ODBC Driver リリース 10.2.0.1.0 以上では、コア、レベル 2 およびレベル 1 のすべての関数をサポートしています。

### G.11.2 ODBC API 関数の実装

次の表に、Oracle ODBC Driver が特定の関数を実装する方法を示します。

機能	説明
SQLConnect	SQLConnect で必要なのは、DBQ、ユーザー ID およびパスワードのみです。
SQLDriverConnect	SQLDriverConnect では、DSN、DBQ、UID および PWD の各キーワードが使用されます。
SQLSpecialColumns	SQL_BEST_ROWID 属性を指定して SQLSpecialColumns をコールすると、常に ROWID 列を返します。
SQLProcedures および SQLProcedureColumns	次の項の説明を参照してください。
すべてのカタログ関数	SQL_ATTR_METADATA_ID 文の属性が SQL_TRUE に設定されている場合、文字列の引数は、識別子の引数として処理されますが、このケースはあまり重要ではありません。この場合、アンダースコア ( ) およびパーセント記号 (%) は検索用パターン文字ではなく、実際の文字として処理されます。これに対して、この属性を SQL_FALSE に設定すると、文字列の引数は、通常の引数またはパターン値の引数となり、リテラルに処理されるため、このケースは重要です。

#### SQLProcedures および SQLProcedureColumns

SQLProcedures コールおよび SQLProcedureColumns コールは、パッケージ内に含まれている場合でも、すべてのプロシージャおよびファンクションに関する情報を検索して返すように変更されました。これより前のリリースでは、これらのコールで検索されるのは、パッケージ外のプロシージャとファンクションのみでした。次の例では、SQL\_ATTR\_METADATA\_ID 属性を SQL\_FALSE に設定した場合に返されるプロシージャまたはファンクションを示します。

この例では、次のストアド・プロシージャがあるとします。

```
"BAR"
"BARX"
"XBAR"
"XBARX"
"SQLPROCTEST.BAR"
"SQLPROCTEST.BARX"
"SQLPROCTEST.XBAR"
"SQLPROCTEST.XBARX"
```

% または %%%% と指定して検索すると、これら 8 つのプロシージャがすべて返されます。

`%_` または `_%` と指定して検索すると、次のプロシージャが返されます。

```
BAR
BARX
XBAR
XBARX
```

`..`、`.%`、`%.%`、`SQLPROC%` または `SQLPROC%.%` と指定して検索すると、次のプロシージャが返されます。

```
SQLPROCTEST.BAR
SQLPROCTEST.BARX
SQLPROCTEST.XBAR
SQLPROCTEST.XBARX
```

`%BAR` と指定して検索すると、次のプロシージャが返されます。

```
BAR
XBAR
```

`%.BAR` または `%.%BAR` と指定して検索すると、次のプロシージャが返されます。

```
SQLPROCTEST.BAR
SQLPROCTEST.XBAR
```

`SQLPROC%` または `.SQLPROC%` と指定して検索すると、次のプロシージャが返されます。

```
nothing (0 rows)
```

### G.11.3 ODBC SQL 構文の実装

比較述語で比較の 2 番目の式としてパラメータ・マーカーが使用され、そのパラメータの値が `SQLBindParameter` を使用して `SQL_NULL_DATA` に設定されている場合、比較は失敗します。これは、ODBC SQL の NULL 述語構文と一致しています。

### G.11.4 データ型の実装

プログラマがデータ型を実装する際に最も考慮する必要があるのは、`CHAR`、`VARCHAR` および `VARCHAR2` の各データ型です。

`SQL_VARCHAR` の値が `fSqlType` の場合、`SQLGetTypeInfo` は Oracle Database データ型 `VARCHAR2` を返します。`SQL_CHAR` の値が `fSqlType` の場合、`SQLGetTypeInfo` は Oracle Database データ型 `CHAR` を返します。

## G.12 Unicode のサポート

この項では、Unicode のサポートについて説明します。次の項目について説明します。

- [ODBC 環境内での Unicode のサポート](#)
- [ODBC API での Unicode のサポート](#)
- [SQLGetData のパフォーマンス](#)
- [Unicode のサンプル](#)

### G.12.1 ODBC 環境内での Unicode のサポート

ODBC Driver Manager を使用すると、すべての ODBC ドライバは、Unicode をサポートしているかどうかに関係なく Unicode 準拠と同様に動作します。これによって、ODBC アプリケーションは、基礎となる ODBC ドライバの Unicode 機能に関係なく記述できます。

Driver Manager が ANSI ODBC ドライバに対する Unicode サポートをエミュレートできる範囲は、Unicode データとローカル・コード・ページ間で可能な変換内容によって制限されます。Driver Manager がデータを Unicode からローカル・コード・ページに変換する際、データが失

われる場合があります。基礎となる ODBC ドライバが Unicode をサポートしていないかぎり、Unicode の完全なサポートは不可能です。Oracle ODBC Driver は、Unicode を完全にサポートしています。

## G.12.2 ODBC API での Unicode のサポート

ODBC API は、w および A 接尾辞の変換を使用して、Unicode および ANSI エントリ・ポイントの両方をサポートします。ODBC アプリケーション開発者は、接尾辞を指定してエントリ・ポイントを明示的にコールする必要はありません。UNICODE および \_UNICODE プリプロセッサ定義を使用してコンパイルされた ODBC アプリケーションによって、適切なコールが生成されます。たとえば、SQLPrepare へのコールは、SQLPrepareW としてコンパイルされます。

C データ型の SQL\_C\_WCHAR が ODBC インタフェースに追加されたため、アプリケーションでは、入力パラメータを Unicode としてエンコードするように指定するか、または Unicode として返された列データを要求できます。マクロの SQL\_C\_TCHAR は、Unicode および ANSI の両方で作成する必要があるアプリケーションで有効です。SQL\_C\_TCHAR マクロは、Unicode アプリケーションの場合は SQL\_C\_WCHAR として、ANSI アプリケーションの場合は SQL\_C\_CHAR としてコンパイルされます。

SQL データ型の SQL\_WCHAR、SQL\_WVARCHAR および SQL\_WLONGVARCHAR が ODBC インタフェースに追加され、表内で定義された列が Unicode で表現されるようになりました。これらの値は、SQLDescribeCol、SQLColAttribute、SQLColumns および SQLProcedureColumns へのコールから返すことも可能です。

Unicode エンコーディングは、SQL 列型の NCHAR、NVARCHAR2 および NCLOB に対してサポートされています。さらに、文字セマンティクスが列定義で指定されている場合、Unicode エンコーディングは SQL 列型の CHAR および VARCHAR2 に対してもサポートされています。

Oracle ODBC Driver は、これらの SQL 列型をサポートして ODBC SQL データ型にマップします。次の表に、サポートされている SQL データ型および対応する ODBC SQL データ型を示します。

SQL データ型	ODBC SQL データ型
CHAR	SQL_CHAR または SQL_WCHAR
VARCHAR2	SQL_VARCHAR または SQL_WVARCHAR
NCHAR	SQL_WCHAR
NVARCHAR2	SQL_WVARCHAR
NCLOB	SQL_WLONGVARCHAR

## G.12.3 SQLGetData のパフォーマンス

SQLGetData 関数を使用すると、ODBC アプリケーションはデータ型を指定して、データのフェッチ後に列を取得できます。OCI では、Oracle ODBC Driver はデータ型を指定してからフェッチする必要があります。この場合、Oracle ODBC Driver は（データベースで定義された）列のデータ型に関する情報を使用し、OCI を介して列をフェッチする最適なデフォルト方法を判断します。

文字データを含む列が SQLBindCol によってバインドされていない場合、Oracle ODBC Driver は、その列を Unicode としてフェッチするか、またはローカル・コード・ページとしてフェッチするかを判断する必要があります。ドライバは、列を Unicode として受け入れるように常にデフォルト設定できます。ただし、この設定では、2 回の不要な変換が実行されます。たとえば、データベース内のデータが ANSI としてエンコードされた場合、データを Oracle ODBC Driver にフェッチするために ANSI から Unicode への変換が実行されます。次に、ODBC アプリケーションがそのデータを SQL\_C\_CHAR として要求すると、データを元のエンコーディングに戻すための変換が実行されます。

Oracle Database Client のデフォルト・エンコーディングは、データをフェッチする際に使用されます。ただし、ODBC アプリケーションでは、このデフォルトを上書きし、列またはパラメータを WCHAR データ型としてバインドすることによって、データを Unicode としてフェッチできます。

## G.12.4 Unicode のサンプル

Oracle ODBC Driver 自身が TCHAR マクロを使用して実装されているため、ODBC アプリケーション・プログラムでは、TCHAR を使用して、このドライバを利用することをお勧めします。

次の例では、UNICODE および \_UNICODE を指定してコンパイルすると WCHAR データ型になる TCHAR の使用方法を示します。

### 例 G-1 データベースへの接続

このコードを使用するには、SQLConnect に Unicode リテラルのみ指定する必要があります。

```
HENV          envHnd;
HDBC          conHnd;
HSTMT        stmtHnd;
RETCODE       rc;

rc = SQL_SUCCESS;

// ENV is allocated
rc = SQLAllocEnv(&envHnd);
// Connection Handle is allocated
rc = SQLAllocConnect(envHnd, &conHnd);
rc = SQLConnect(conHnd, _T("stpc19"), SQL_NTS, _T("scott"), SQL_NTS, _T("tiger"), SQL_NTS);
.
.
.
if (conHnd)
    SQLFreeConnect(conHnd);
if (envHnd)
    SQLFreeEnv(envHnd);
```

### 例 G-2 単純な取得

次の例では、EMP 表から従業員名と役職名を取得します。すべての ODBC 関数に TCHAR 準拠のデータを指定する必要があることを除いて、ANSI の場合と違いはありません。Unicode アプリケーションの場合は、SQLBindCol のコール時にバッファ長を BYTE 長に指定する必要があります。たとえば、sizeof(ename) の場合は次のとおりです。

```
/*
** Execute SQL, bind columns, and Fetch.
** Procedure:
**
**   SQLExecDirect
**   SQLBindCol
**   SQLFetch
**
*/
static SQLTCHAR *sqlStmt = _T("SELECT ename, job FROM emp");
SQLTCHAR  ename[50];
SQLTCHAR  job[50];
SQLINTEGER enamelen, joblen;

_tprintf(_T("Retrieve ENAME and JOB using SQLBindCol 1.../n[%s]/n"), sqlStmt);

// Step 1: Prepare and Execute
rc = SQLExecDirect(stmtHnd, sqlStmt, SQL_NTS); // select
checkSQLErr(envHnd, conHnd, stmtHnd, rc);
```

```

// Step 2: Bind Columns
rc = SQLBindCol(stmtHnd,
                1,
                SQL_C_TCHAR,
                ename,
                sizeof(ename),
                &enamelen);
checkSQLErr(envHnd, conHnd, stmtHnd, rc);

rc = SQLBindCol(stmtHnd,
                2,
                SQL_C_TCHAR,
                job,
                sizeof(job),
                &joblen);
checkSQLErr(envHnd, conHnd, stmtHnd, rc);

do
{
    // Step 3: Fetch Data
    rc = SQLFetch(stmtHnd);
    if (rc == SQL_NO_DATA)
        break;
    checkSQLErr(envHnd, conHnd, stmtHnd, rc);
    _tprintf(_T("ENAME = %s, JOB = %s/n"), ename, job);
} while (1);
_tprintf(_T("Finished Retrieval/n/n"));

```

### 例 G-3 SQLGetData を使用した取得（フェッチ後のバインド）

この例では、SQLGetData の使用方法を示します。Unicode 固有の事項に関しては、ANSI アプリケーションとの違いはありません。

```

/*
** Execute SQL, bind columns, and Fetch.
** Procedure:
**
**   SQLExecDirect
**   SQLFetch
**   SQLGetData
*/
static SQLTCHAR *sqlStmt = _T("SELECT ename,job FROM emp"); // same as Case 1.
SQLTCHAR          ename[50];
SQLTCHAR          job[50];

_tprintf(_T("Retrieve ENAME and JOB using SQLGetData.../n[%s]/n"), sqlStmt);
if (rc != SQL_SUCCESS)
{
    _tprintf(_T("Failed to allocate STMT/n"));
    goto exit2;
}

// Step 1: Prepare and Execute
rc = SQLExecDirect(stmtHnd, sqlStmt, SQL_NTS); // select
checkSQLErr(envHnd, conHnd, stmtHnd, rc);

do
{
    // Step 2: Fetch
    rc = SQLFetch(stmtHnd);
    if (rc == SQL_NO_DATA)
        break;

```

```

checkSQLErr(envHnd, conHnd, stmtHnd, rc);

// Step 3: GetData
rc = SQLGetData(stmtHnd,
    1,
    SQL_C_TCHAR,
    (SQLPOINTER)ename,
    sizeof(ename),
    NULL);
checkSQLErr(envHnd, conHnd, stmtHnd, rc);
rc = SQLGetData(stmtHnd,
    2,
    SQL_C_TCHAR,
    (SQLPOINTER)job,
    sizeof(job),
    NULL);
checkSQLErr(envHnd, conHnd, stmtHnd, rc);
_tprintf(_T("ENAME = %s, JOB = %s/n"), ename, job);
} while (1);
_tprintf(_T("Finished Retrieval/n/n"));

```

#### 例 G-4 単純な更新

この例では、データの更新方法を示します。SQLBindParameter のデータ長は、Unicode アプリケーションの場合でも BYTE 長で指定する必要があります。

```

/*
** Execute SQL, bind columns, and Fetch.
** Procedure:
**
**   SQLPrepare
**   SQLBindParameter
**   SQLExecute
*/
static SQLTCHAR *sqlStmt = _T("INSERT INTO emp(empno,ename,job) VALUES(?,?,?)");
static SQLTCHAR *empno   = _T("9876");      // Emp No
static SQLTCHAR *ename   = _T("ORACLE");   // Name
static SQLTCHAR *job     = _T("PRESIDENT"); // Job

_tprintf(_T("Insert User ORACLE using SQLBindParameter.../n[%s]/n"), sqlStmt);

// Step 1: Prepare
rc = SQLPrepare(stmtHnd, sqlStmt, SQL_NTS); // select
checkSQLErr(envHnd, conHnd, stmtHnd, rc);

// Step 2: Bind Parameter
rc = SQLBindParameter(stmtHnd,
    1,
    SQL_PARAM_INPUT,
    SQL_C_TCHAR,
    SQL_DECIMAL,
    4,           // 4 digit
    0,
    (SQLPOINTER)empno,
    0,
    NULL);
checkSQLErr(envHnd, conHnd, stmtHnd, rc);

rc = SQLBindParameter(stmtHnd,
    2,
    SQL_PARAM_INPUT,
    SQL_C_TCHAR,
    SQL_CHAR,
    lstrlen(ename)*sizeof(TCHAR),

```

```

        0,
        (SQLPOINTER)ename,
        strlen(ename)*sizeof(TCHAR),
        NULL);
checkSQLErr(envHnd, conHnd, stmtHnd, rc);

rc = SQLBindParameter(stmtHnd,
        3,
        SQL_PARAM_INPUT,
        SQL_C_TCHAR,
        SQL_CHAR,
        strlen(job)*sizeof(TCHAR),
        0,
        (SQLPOINTER)job,
        strlen(job)*sizeof(TCHAR),
        NULL);
checkSQLErr(envHnd, conHnd, stmtHnd, rc);

// Step 3: Execute
rc = SQLExecute(stmtHnd);
checkSQLErr(envHnd, conHnd, stmtHnd, rc);

```

#### 例 G-5 長いデータ (CLOB) の更新と取得

ここでは、CLOB などの長いデータを Oracle Database で更新および取得する最も複雑な例を示します。データ長は常に BYTE 長であるため、BYTE 長を導出するには式 `strlen(TCHAR data)*sizeof(TCHAR)` が必要です。

```

/*
** Execute SQL, bind columns, and Fetch.
** Procedure:
**
**   SQLPrepare
**   SQLBindParameter
**   SQLExecute
**   SQLParamData
**   SQLPutData
**
**   SQLExecDirect
**   SQLFetch
**   SQLGetData
*/
static SQLTCHAR *sqlStmt1 = _T("INSERT INTO clobtbl(clob1) VALUES(?)");
static SQLTCHAR *sqlStmt2 = _T("SELECT clob1 FROM clobtbl");
SQLTCHAR      clobdata[1001];
SQLTCHAR      resultdata[1001];
SQLINTEGER    ind = SQL_DATA_AT_EXEC;
SQLTCHAR      *bufp;
int           clobdatalen, chunksize, dtsize, retchklen;

_tprintf(_T("Insert CLOB1 using SQLPutData.../n[%s]/n"), sqlStmt1);

// Set CLOB Data
{
    int i;
    SQLTCHAR ch;
    for (i=0, ch=_T('A'); i< sizeof(clobdata)/sizeof(SQLTCHAR); ++i, ++ch)
    {
        if (ch > _T('Z'))
            ch = _T('A');
        clobdata[i] = ch;
    }
    clobdata[sizeof(clobdata)/sizeof(SQLTCHAR)-1] = _T('/0');
}

```

```

clobdatalen = strlen(clobdata); // length of characters
chunksize   = clobdatalen / 7; // 7 times to put

// Step 1: Prepare
rc = SQLPrepare(stmtHnd, sqlStmt1, SQL_NTS);
checkSQLErr(envHnd, conHnd, stmtHnd, rc);

// Step 2: Bind Parameter with SQL_DATA_AT_EXEC
rc = SQLBindParameter(stmtHnd,
                      1,
                      SQL_PARAM_INPUT,
                      SQL_C_TCHAR,
                      SQL_LONGVARCHAR,
                      clobdatalen*sizeof(TCHAR),
                      0,
                      (SQLPOINTER)clobdata,
                      clobdatalen*sizeof(TCHAR),
                      &ind);
checkSQLErr(envHnd, conHnd, stmtHnd, rc);
// Step 3: Execute
rc = SQLExecute(stmtHnd);
checkSQLErr(envHnd, conHnd, stmtHnd, rc);

// Step 4: ParamData (initiation)
rc = SQLParamData(stmtHnd, (SQLPOINTER*)&bufp); // set value
checkSQLErr(envHnd, conHnd, stmtHnd, rc);

for (dtsize=0, bufp = clobdata;
     dtsize < clobdatalen;
     dtsize += chunksize, bufp += chunksize)
{
    int len;
    if (dtsize+chunksize<clobdatalen)
        len = chunksize;
    else
        len = clobdatalen-dtsize;

    // Step 5: PutData
    rc = SQLPutData(stmtHnd, (SQLPOINTER)bufp, len*sizeof(TCHAR));
    checkSQLErr(envHnd, conHnd, stmtHnd, rc);
}

// Step 6: ParamData (termination)
rc = SQLParamData(stmtHnd, (SQLPOINTER*)&bufp);
checkSQLErr(envHnd, conHnd, stmtHnd, rc);

rc = SQLFreeStmt(stmtHnd, SQL_CLOSE);
_tprintf(_T("Finished Update/n/n"));
rc = SQLAllocStmt(conHnd, &stmtHnd);
if (rc != SQL_SUCCESS)
{
    _tprintf(_T("Failed to allocate STMT/n"));
    goto exit2;
}

// Clear Result Data
memset(resultdata, 0, sizeof(resultdata));
chunksize   = clobdatalen / 15; // 15 times to put

// Step 1: Prepare
rc = SQLExecDirect(stmtHnd, sqlStmt2, SQL_NTS); // select
checkSQLErr(envHnd, conHnd, stmtHnd, rc);

```

```

// Step 2: Fetch
rc = SQLFetch(stmtHnd);
checkSQLErr(envHnd, conHnd, stmtHnd, rc);

for(dtsize=0, bufp = resultdata;
   dtsize < sizeof(resultdata)/sizeof(TCHAR) && rc != SQL_NO_DATA;
   dtsize += chunksize-1, bufp += chunksize-1)
{
    int len; // len should contain the space for NULL termination
    if (dtsize+chunksize<sizeof(resultdata)/sizeof(TCHAR))
        len = chunksize;
    else
        len = sizeof(resultdata)/sizeof(TCHAR)-dtsize;

    // Step 3: GetData
    rc = SQLGetData(stmtHnd,
        1,
        SQL_C_TCHAR,
        (SQLPOINTER)bufp,
        len*sizeof(TCHAR),
        &retchklen);
}
if (!_tcscmp(resultdata, clobdata))
{
    _tprintf(_T("Succeeded!!/n/n"));
}
else
{
    _tprintf(_T("Failed!!/n/n"));
}

```

## G.13 パフォーマンスとチューニング

この項では、次の項目について説明します。

- [ODBC プログラミングの一般的なガイドライン](#)
- [データソース構成オプション](#)
- [DATE および TIMESTAMP データ型](#)

### G.13.1 ODBC プログラミングの一般的なガイドライン

ODBC アプリケーションのパフォーマンスを向上させるには、次のプログラミング・ガイドラインを適用してください。

- アプリケーションがデータソースに頻繁に接続したり切断する場合は、接続プーリングを使用可能にします。プールされた接続の再利用は、接続の再確立に比べて非常に効率的です。
- 文を準備する回数を最小限にします。可能な場合は、バインド・パラメータを使用し、別のパラメータ値に対して文を再利用できるようにします。SQLExecute ごとに文を準備するよりも、1つの文を1回準備して複数回実行する方が効率的です。
- アプリケーションが取得しないことが判明している列、特に LONG 列を SELECT 文に含めないでください。LONG 列が SELECT 文に含まれていると、データベース・サーバー・プロトコルの特性上、アプリケーションが列をバインドするか SQLGetData 操作を実行するかに関係なく、Oracle ODBC Driver では列全体をフェッチする必要があります。
- データソースを更新しないトランザクションを実行する場合は、ODBC の SQLSetConnectAttr 関数の SQL\_ATTR\_ACCESS\_MODE 属性を SQL\_MODE\_READ\_ONLY に設定します。

- ODBC のエスケープ句を使用しない場合は、その ODBC の `SQLSetConnectAttr` 関数または `SQLSetStmtAttr` 関数の `SQL_ATTR_NOSCAN` 属性を `true` に設定します。
- 行数の多い表からデータを取得する場合は、ODBC の `SQLFetch` 関数のかわりに ODBC の `SQLFetchScroll` 関数を使用します。

## G.13.2 データソース構成オプション

この項では、次の ODBC データソース構成オプションを使用した場合のパフォーマンスへの影響を説明します。

- 結果セットを有効化

このオプションによって、プロシージャ・コールから結果セット (RefCursor など) を返すサポートが有効になります。デフォルトでは、結果セットを返すサポートが有効です。

Oracle ODBC Driver では、RefCursor パラメータが存在するかどうかを判断するために、データベース・サーバーを問い合わせるプロシージャのパラメータ・セットとそのデータ型を判別する必要があります。この問合せによって、最初にプロシージャが準備完了になり実行されると、追加のネットワーク・ラウンドトリップが発生します。

- LOB を有効化

このオプションによって、LOB を挿入および更新するサポートが有効になります。デフォルトではこのサポートが有効です。

Oracle ODBC Driver では、LOB パラメータがあるかどうかを判断するために、データベース・サーバーを問い合わせ、INSERT 文または UPDATE 文の各パラメータのデータ型を判別する必要があります。この問合せによって、最初に INSERT または UPDATE が準備完了になり実行されると、追加のネットワーク・ラウンドトリップが発生します。

- TIMESTAMP を DATE としてバインド

SQL\_TIMESTAMP パラメータを適切な Oracle Database データ型としてバインドします。このオプションを `TRUE` に設定すると、SQL\_TIMESTAMP は Oracle の DATE データ型としてバインドされます。このオプションを `FALSE` に設定すると、SQL\_TIMESTAMP は Oracle の TIMESTAMP データ型としてバインドされます。これがデフォルトです。

- カーソル・クローズを有効化

ODBC 関数 `SQLFreeStmt` の `SQL_CLOSE` オプションは、関連するカーソルを文とともにクローズし、保留中のすべての結果を廃棄します。アプリケーションでは文を再度実行することによってカーソルを再オープンでき、`SQLPrepare` を再度実行する必要はありません。このオプションを使用する典型的な例は、しばらくの間アイドル状態になり、後で同じ SQL 文を再利用するアプリケーションの場合です。この場合、アプリケーションがアイドル状態になっている間、関連するサーバー・リソースを解放できます。

Oracle ODBC Driver が位置する OCI では、カーソルをクローズする機能をサポートしていません。したがって、デフォルトでは、`SQL_CLOSE` オプションは Oracle ODBC Driver に影響を与えません。カーソルおよび関連するリソースはデータベース上でオープンしたままです。

このオプションを有効にすると、データベース・サーバー上の関連するカーソルがクローズします。ただし、その結果、SQL 文の解析コンテキストが失われます。ODBC アプリケーションは文を再度実行でき、`SQLPrepare` をコールする必要はありません。ただし、内部的には、Oracle ODBC Driver は文全体を準備して実行する必要があります。このオプションを有効にすると、文を 1 回準備して繰り返し実行するアプリケーションのパフォーマンスに大きな影響を与えます。

このオプションは、サーバー上のリソースを解放する必要がある場合のみ有効にしてください。

- フェッチ・バッファ・サイズ

`odbc.ini` ファイルのフェッチ・バッファ・サイズ (`FetchBufferSize`) を、バイト単位で指定した値に設定します。この値は、Oracle ODBC Driver が Oracle Database からクライアントのキャッシュに 1 回にプリフェッチするデータの行数を決定するのに必要なメモ

リー量で、アプリケーション・プログラムが1回の問合せで要求する行数とは関係ありません。これによって、パフォーマンスが向上します。

また、1回にフェッチするデータが20行未満であることが多いアプリケーションで、特に、速度の遅いネットワーク接続である場合や負荷の大きいサーバーからフェッチする場合は、アプリケーションの応答時間が改善されます。この設定値が高すぎると、応答時間に悪影響が生じたり、大量のメモリーが消費される可能性があります。デフォルトは64,000バイトです。アプリケーションに対して最適な値を選択してください。

LONG および LOB データ型がある場合、Oracle ODBC Driver がプリフェッチする行数はこのフェッチ・バッファ・サイズで決まりません。LONG および LOB データ型が含まれると、パフォーマンスの向上は最小限になり、過剰にメモリーが使用される可能性があります。Oracle ODBC Driver はこのフェッチ・バッファ・サイズを無視し、LONG および LOB データ型が存在する場合は一定の行数のみプリフェッチします。

**関連項目：** G-3 ページ「[SQLDriverConnect 関数の接続文字列の書式](#)」

### G.13.3 DATE および TIMESTAMP データ型

WHERE 句でデータベースの DATE 列が使用され、その列に索引がある場合は、パフォーマンスに影響を与える可能性があります。たとえば、次のように入力します。

```
SELECT * FROM EMP WHERE HIREDATE = ?
```

この例では、HIREDATE 列の索引を使用して、問合せを迅速に実行できます。ただし、HIREDATE が DATE 値で、Oracle ODBC Driver はこのパラメータ値を TIMESTAMP として提供しているため、Oracle Database の問合せオブティマイザは変換関数を適用する必要があります。誤った結果（パラメータ値に 0 以外的小数秒がある場合に発生する可能性があります）を防ぐために、オブティマイザは、HIREDATE 列に変換を適用するため、次のような文になります。

```
SELECT * FROM EMP WHERE TO_TIMESTAMP(HIREDATE) = ?
```

ただし、この場合は HIREDATE 列で索引を使用できなくなります。かわりに、サーバーでは表の順次スキャンを実行します。したがって、表に多数の行がある場合は時間がかかる可能性があります。このような場合の次善策として、Oracle ODBC Driver には TIMESTAMP を DATE としてバインドする接続オプションが用意されています。このオプションを有効にすると、Oracle ODBC Driver は、SQL\_TIMESTAMP パラメータを Oracle の TIMESTAMP データ型ではなく Oracle の DATE データ型としてバインドします。これによって、問合せオブティマイザは DATE 列で索引を使用できます。

---

---

**注意：** このオプションは、DATE 列を TIMESTAMP 列としてバインドする Microsoft Access や類似のプログラムで使用することを目的としています。実際の TIMESTAMP 列がある場合、またはデータが失われる可能性がある場合は使用しないでください。Microsoft Access では、主キーとして選択されたすべての列を使用してこの問合せを実行します。

---

---

## G.14 エラー・メッセージ

エラーが発生すると、Oracle ODBC Driver は、システム固有のエラー番号、SQLSTATE (ODBC エラー・コード) およびエラー・メッセージを返します。ドライバは、ドライバが検出したエラーおよび Oracle Database から返されたエラーの両方からこれらの情報を導出します。

### システム固有のエラー

データソースでエラーが発生した場合、Oracle ODBC Driver は、Oracle Database から返されたシステム固有のエラーを返します。Oracle ODBC Driver または Driver Manager がエラーを検出した場合、Oracle ODBC Driver はシステム固有のエラー番号として 0 (ゼロ) を返します。

### SQLSTATE

データソースでエラーが発生した場合、Oracle ODBC Driver は、返されたシステム固有のエラーを適切な SQLSTATE にマップします。Oracle ODBC Driver または Driver Manager がエラーを検出した場合は、適切な SQLSTATE を生成します。

### エラー・メッセージ

データソースでエラーが発生した場合、Oracle ODBC Driver は、Oracle Database から返されたメッセージに基づいてエラー・メッセージを返します。Oracle ODBC Driver または Driver Manager でエラーが発生した場合、Oracle ODBC Driver は、SQLSTATE に関連付けられたテキストに基づいてエラー・メッセージを返します。

エラー・メッセージの書式は次のとおりです。

```
[vendor] [ODBC-component] [data-source] error-message
```

大カッコ ([ ]) 内の接頭辞によって、エラーの発生場所を識別します。次の表に、Oracle ODBC Driver から返される接頭辞の値を示します。データソースでエラーが発生した場合は、vendor および ODBC-component 接頭辞によって、データソースからエラーを受け取った ODBC コンポーネントのベンダーと名前を識別します。

エラーの発生場所	接頭辞	値
Driver Manager	[vendor]	[unixODBC]
	[ODBC-component]	[Driver Manager]
	[data-source]	該当なし
Oracle ODBC Driver	[vendor]	[ORACLE]
	[ODBC-component]	[Oracle ODBC Driver]
	[data-source]	該当なし
Oracle Database	[vendor]	[ORACLE]
	[ODBC-component]	[Oracle ODBC Driver]
	[data-source]	[Oracle OCI]

たとえば、次の書式に示すように、エラー・メッセージに Ora 接頭辞が付いていない場合、そのエラーは Oracle ODBC Driver のエラーであることがわかります。

```
[Oracle] [ODBC] Error message text here
```

また、次の書式に示すように、エラー・メッセージに Ora 接頭辞が付いている場合、このエラーは Oracle ODBC Driver のエラーではありません。

```
[Oracle] [ODBC] [Ora] Error message text here
```

---

**注意：** エラー・メッセージに Ora 接頭辞が付いている場合でも、実際のエラーは複数のソースのいずれかで発生している場合があります。

---

エラー・メッセージのテキストが ORA- 接頭辞で始まっている場合、そのエラーの詳細は Oracle Database のドキュメントを参照してください。

---

---

## データベースの制限

この付録では、データベースの制限について説明します。

## H.1 データベースの制限

表 H-1 に、CREATE DATABASE 文または CREATE CONTROLFILE 文のパラメータのデフォルト値および最大値を示します。

**注意：** これらのパラメータ間の相互依存によって、許容値に影響を与える場合があります。

**表 H-1 CREATE CONTROLFILE および CREATE DATABASE のパラメータ**

パラメータ	デフォルト	最大値
MAXLOGFILES	16	255
MAXLOGMEMBERS	2	5
MAXLOGHISTORY	100	65534
MAXDATAFILES	30	65534
MAXINSTANCES	1	63

表 H-2 に、Oracle Database ファイル・サイズ制限をバイト単位で示します。

**表 H-2 ファイル・サイズの制限**

ファイル・タイプ	プラットフォーム	
データファイル	すべて	4,194,303 と DB_BLOCK_SIZE パラメータの値の積
インポート / エクスポート・ファイルおよび SQL*Loader ファイル	Tru64 UNIX	16TB
	AIX、HP-UX、Linux および Solaris: 32 ビット (32 ビットのファイルを扱う)	2,147,483,647 バイト
	AIX、HP-UX、Linux、Mac OS X および Solaris: 64 ビットのファイル	無制限
制御ファイル	HP-UX、Linux、Mac OS X および Solaris	20000 データベース・ブロック
	AIX	10000 データベース・ブロック
	Tru64 UNIX	19200 データベース・ブロック

# 索引

## 記号

@ 略称, 1-2

## 数字

32 ビット外部プロシージャ  
PL/SQL からのコール, 7-5

## A

A\_TERMCAP 環境変数, 6-10  
A\_TERM 環境変数, 6-10  
ADA\_PATH 環境変数, 1-4  
adapters ユーティリティ, 5-3  
aio\_task\_max\_num 属性, F-7  
AIXTHREAD\_SCOPE 環境変数, A-15  
AIX での LVM, A-6  
AIX のツール  
    Base Operation System ツール, 8-5  
    Performance Toolbox, 8-5  
    PTX Agent, 8-5  
    PTX Manager, 8-5  
    SMIT, 8-6  
    System Management Interface Tool, 8-6  
ASM\_DISKSTRING 初期化パラメータ, 1-8

## B

BACKGROUND\_DUMP\_DEST 初期化パラメータ, 1-14  
BUFFER パラメータ, A-5

## C

CLASSPATH 環境変数, 1-4  
COBDIR 環境変数, 6-11  
coraenv ファイル, 1-6  
CPU\_COUNT 初期化パラメータ, B-9  
CPU のスケジューリング, A-14  
CREATE CONTROLFILE のパラメータ, H-2  
CREATE DATABASE のパラメータ, H-2  
CURSOR\_SPACE\_FOR\_TIME 初期化パラメータ  
    Oracle Database のチューニング, B-7

## D

Database Control,  
    「Oracle Enterprise Manager Database Control」を  
    参照  
DB\_BLOCK\_SIZE 初期化パラメータ, 1-7, 8-12  
DB\_CACHE\_SIZE 初期化パラメータ, 8-12  
DB\_FILE\_MULTIBLOCK\_READ\_COUNT パラメータ,  
    A-12  
DBA,  
    「管理者」を参照  
dbhome ファイル, 1-7  
demo\_proc32.mk Make ファイル, 6-8  
demo\_proc32.mk ファイル, 6-8  
demo\_procb.mk ファイル, 6-13  
DISM, E-2  
DISPLAY 環境変数, 1-5  
DYLD\_LIBRARY\_PATH 環境変数, 1-5

## F

FORMAT プリコンパイラ, 6-13  
Pro\*COBOL, 6-14

## G

getprivgrp コマンド, B-3, B-4  
glogin.sql ファイル, 4-2  
GPFS  
    使用する際の考慮事項, A-8

## H

HP-UX 動的プロセッサ再構成, B-9  
HP-UX のツール, 8-6  
    Glance/UX, 8-6  
hugetlfs  
    Linux 上, C-4  
    SUSE 上, C-3

## I

ins\_precomp.mk ファイル, 6-2  
I/O  
    チューニング, 8-10  
    非同期, A-10, B-4, F-7  
iostat コマンド, 8-4

## I/O サポート

- コンカレント, F-7
- ダイレクト, C-6, F-7
- ダイレクトの無効化, F-8
- 非同期, C-6
- I/O スレーブ, A-11
- I/O バッファおよび SQL\*Loader, A-5
- IPC プロトコル, 5-4
- IRECLEN, 6-3
- ISM, E-2
- iSQL\*Plus, 2-5
  - 停止, 2-5

## J

- JAVA\_POOL\_SIZE 初期化パラメータ, 8-12
- JFS
  - 考慮事項, A-8
- JFS2
  - 考慮事項, A-8

## L

- LANGUAGE 環境変数, 1-5
- LANG 環境変数, 1-5
- LARGE\_POOL\_SIZE 初期化パラメータ, 8-12
- LD\_LIBRARY\_PATH 環境変数, 1-5, 6-10, 6-12
- LD\_OPTIONS 環境変数, 1-5
- LIBPATH 環境変数, 1-5, 6-12
- Linux のツール, 8-7
- LOG\_ARCHIVE\_DEST\_n 初期化パラメータ, 1-8
- LOG\_BUFFERS 初期化パラメータ, 8-12
- login.sql ファイル, 4-2
- LPDEST 環境変数, 1-5
- lspc コマンド, 8-4

## M

- Mac OS X のツール, 8-7
- Make ファイル
  - demo\_proc32.mk, 6-8
  - demo\_procob.mk, 6-13
  - ins\_precomp.mk, 6-2
- Make ファイル、カスタム, 6-20
- MAX\_DUMP\_FILE 初期化パラメータ, 1-14
- MAXDATAFILES パラメータ, H-2
- maxfree パラメータ, A-2
- MAXINSTANCES パラメータ, H-2
- MAXLOGFILES パラメータ, H-2
- MAXLOGHISTORY パラメータ, H-2
- MAXLOGMEMBERS パラメータ, H-2
- maxperm パラメータ, A-2, A-3
- MicroFocus COBOL コンパイラ, 6-11
- minfree パラメータ, A-2
- minperm パラメータ, A-2, A-3
- MLOCK 権限, B-4
- mpstat コマンド, 8-7

## N

- NUMA
  - 「指定配置最適化」を参照

## O

- OCCI, 6-17
  - ユーザー・プログラム, 6-18
- OCI, 6-17
  - ユーザー・プログラム, 6-18
- oinstall グループ, 1-9
- ORA\_NLS10 環境変数, 1-2
- ORA\_TZFILE 環境変数, 1-3
- Oracle Advanced Security, 5-6
- Oracle C++ Call Interface, 6-17
  - 「OCCI」を参照
- Oracle Call Interface
  - 「OCI」を参照
- Oracle Call Interface, 6-17
- Oracle Call Interface と Oracle C++ Call Interface
  - デモ・プログラム, 6-17
- Oracle CSS デーモン
  - 起動, 2-4
  - 停止, 2-4
- Oracle Database, 3-3
  - 再起動, 2-3
  - 停止, 2-2
- Oracle Database Configuration Assistant
  - 構成, 3-3
- Oracle Database アップグレード・アシスタント, 3-3
- 『Oracle Database サンプル・スキーマ』, 4-3
- Oracle Database の環境変数
  - Oracle Database の変数, 1-2
- Oracle Database のチューニング
  - 大規模メモリの割当て, B-7
  - チューニングに関する推奨事項, B-9
- Oracle Database のチューニングと大規模メモリの割当て, B-7
- Oracle Database プロセス
  - 停止, 2-2
- Oracle Enterprise Manager Database Control
  - 起動, 2-7
  - 停止, 2-7
- Oracle JDBC/OCI
  - デモ・プログラム, 6-19
- Oracle Management Agent
  - 起動, 2-8
  - 停止, 2-8
- Oracle Net Listener
  - 再起動, 2-5
  - 停止, 2-4
- Oracle Net Services
  - IPC プロトコル, 5-4
  - Oracle Advanced Security, 5-6
  - protocol support, 5-4
  - TCP/IP プロトコル, 5-5
  - 構成ファイル, 5-2
  - プロトコル, 5-4
- Oracle Net Services の SSL 付き TCP/IP プロトコル, 5-5
- Oracle Net コンフィギュレーション・アシスタント
  - 使用, 3-2
- Oracle ODBC Driver, G-1
- Oracle Protocol Support
  - IPC プロトコル, 5-4
  - SSL 付き TCP/IP プロトコル, 5-5
  - TCP/IP プロトコル, 5-5

Oracle Ultra Search  
 起動, 2-6  
 停止, 2-6  
ORACLE\_BASE 環境変数, 1-3  
ORACLE\_HOME 環境変数, 1-3  
ORACLE\_PATH 環境変数, 1-3  
ORACLE\_SID 環境変数, 1-2, 1-3  
ORACLE\_TRACE 環境変数, 1-3  
oracle アカウント, 1-10  
Oracle インベントリ・グループ, 1-9  
Oracle 環境変数  
  ORA\_NLS10, 1-2  
Oracle 初期化パラメータ, F-3  
Oracle 製品  
  追加に対するデータベースの構成, 3-2  
Oracle ソフトウェア所有者アカウント, 1-9  
Oracle バッファ・マネージャ, 8-8  
Oracle ブロック・サイズ、調整, 8-10  
Oracle ユーザー・アカウント  
  構成, 1-11  
oradism コマンド, E-2  
ORAENV\_ASK 環境変数, 1-3  
oraenv ファイル, 1-6  
orapwd コマンド, 1-10  
orapwd ユーティリティ, 1-10  
ORECLEN, 6-3  
OS\_AUTHENT\_PREFIX パラメータ, 1-10  
OSDBA グループ, 1-11  
osdba グループ, 1-9  
OSOPER グループ, 1-11  
osoper グループ, 1-9  
ottcfg.cfg ファイル, 6-2

## P

PATH 環境変数, 1-6, 4-4, 6-10, 6-11  
pcbcfg.cfg ファイル, 6-2  
pccfor.cfg ファイル, 6-2  
pcscfg.cfg ファイル, 6-2  
PL/SQL カーネル・デモ, 7-2  
PL/SQL のデモ, 7-2  
pmscfg.cfg ファイル, 6-2  
Polycenter Advanced ファイル・システム, 8-10  
PRINTER 環境変数, 1-6  
privgroup ファイル, B-3, B-4  
Pro\*C/C++  
  Make ファイル, 6-7  
  シグナル, 6-21  
  デモ・プログラム, 6-7  
  ユーザー・プログラム, 6-8  
Pro\*C/C++ プリコンパイラ, 6-7  
Pro\*COBOL  
  FORMAT プリコンパイラ, 6-13, 6-14  
  Oracle ランタイム・システム, 6-12  
  環境変数, 6-10  
  デモ・プログラム, 6-12  
  ネーミングの相違点, 6-9  
  ユーザー・プログラム, 6-13  
Pro\*COBOL プリコンパイラ, 6-9  
Pro\*FORTRAN のデモ・プログラム, 6-14  
PRODUCT\_USER\_PROFILE 表, 4-2  
Programmer's Analysis Kit (HP PAK), 8-6  
PTX Agent, 8-5

## R

rad\_gh\_regions パラメータ, F-4  
RAD に対するプロセスの親和性, F-5  
RAW ディスク, 1-12  
RAW デバイス, 1-12  
  RAW ディスク・パーティションの可用性, 1-12  
  使用方法のガイドライン, 1-12  
  設定, 1-12, 1-13, F-10  
  バックアップの作成, A-14  
  バッファ・キャッシュ・サイズ, 8-14  
RAW デバイスの設定, 1-13  
RAW デバイスのデータファイル, 1-13  
RAW 論理ボリューム, A-7, A-9  
root.sh スクリプト, 1-7  
root.sh ファイル, 1-7

## S

sar コマンド, 8-3, 8-9  
SCHED\_NOAGE パラメータ、スケジューリング・ポリシー, B-2  
SCHED\_NOAGE パラメータ、有効化, B-3  
semtimedop のサポート, C-7  
services ファイル, 5-6  
setprivgrp コマンド, B-3, B-4  
SGA, 8-12  
  サイズの確認, 8-13  
SGA\_MAX\_SIZE パラメータ, E-2  
SGA アドレス空間、増加, C-5  
SGA の非同期フラグ, B-7  
SHARED\_POOL\_SIZE 初期化パラメータ, 8-12  
SHLIB\_PATH 環境変数, 1-6, 6-12  
shm\_allocate\_striped パラメータ, F-4  
shm\_max パラメータ, 8-12  
shmmax パラメータ, 8-12  
shm\_seg パラメータ, 8-12  
shmseg パラメータ, 8-12  
SIGCLD シグナル, 6-21  
SIGCONT シグナル, 6-21  
SIGINT シグナル, 6-21  
SIGIO シグナル, 6-21  
SIGPIPE シグナル, 6-21  
SIGTERM シグナル, 6-21  
SIGURG シグナル, 6-21  
SMP システムでのプロセッサ・バインディング  
  使用, A-14  
Solaris のツール, 8-7  
Spike 最適化ツール, F-12  
SPOOL コマンド  
  SQL\*Plus, 4-5  
SQL\*Loader, A-5  
SQL\*Loader のデモ, 7-2  
SQL\*Module for Ada, 6-16  
  デモ・プログラム, 6-16  
  ユーザー・プログラム, 6-17  
SQL\*Plus  
  PRODUCT\_USER\_PROFILE 表, 4-2  
  SPOOL コマンド, 4-5  
  エディタ, 4-4  
  オペレーティング・システム・コマンドの実行, 4-4  
  コマンドライン SQL\*Plus の使用, 4-4  
  コマンドライン・ヘルプ, 4-3

サイト・プロファイル, 4-2  
システム・エディタ, 4-4  
制限事項, 4-5  
デフォルト・エディタ, 4-4  
ユーザー・プロファイル, 4-2  
割込み, 4-5  
SQL\*Plus のコマンドライン・ヘルプ  
インストール, 4-3  
削除, 4-3  
SQL\*Plus、割込み, 4-5  
SQLPATH 環境変数, 1-4  
SSL 付き TCP/IP プロトコル, 5-5  
ssm\_threshold パラメータ, F-4  
swapinfo コマンド, 8-4  
swapon コマンド, 8-4  
swap コマンド, 8-4  
symfind ユーティリティ, 6-20  
SYSDATE, 1-7  
SYSTEM アカウント, 1-11  
SYS アカウント, 1-11

## T

---

TCP/IP プロトコル, 5-5  
TIMED\_STATISTICS パラメータ, F-9  
TMPDIR 環境変数, 1-6  
TWO\_TASK 環境変数, 1-4

## U

---

Ultra Search, 「Oracle Ultra Search」を参照  
unified ファイル・システム, 8-10  
UNIX System V ファイル・システム, 8-10  
USE\_INDIRECT\_DATA\_BUFFERS パラメータ, C-2  
USER\_DUMP\_DEST 初期化パラメータ, 1-14  
UTLRP.SQL  
無効な SQL モジュールの再コンパイル, 3-3

## V

---

Veritas ファイル・システム, 8-10  
Virtual Memory Manager  
「VMM」を参照  
VLM ウィンドウ・サイズ, C-3  
VMM, A-12  
vmo コマンド, A-2  
vmstat コマンド, 8-2

## X

---

XA 機能, 6-23  
XENVIRONMENT 環境変数, 1-6  
X/Open Distributed Transaction Processing (DTP) XA  
インタフェース, 6-23

## あ

---

アーカイブ・バッファ  
チューニング, A-5  
アカウント  
SYS, 1-11  
SYSTEM, 1-11

アシスタント  
Oracle Database Configuration Assistant, 3-3  
Oracle Database アップグレード・アシスタント, 3-3  
Oracle Net コンフィギュレーション・アシスタント,  
3-2  
アップグレード, 3-3  
アップグレード済データベース  
構成, 3-3  
アラート・ファイル, 1-14

## い

---

移行, 3-3  
インストール  
SQL\*Plus のコマンドライン・ヘルプ, 4-3  
インストール後のタスク  
コンフィギュレーション・アシスタント, 3-2  
インポート・ユーティリティ, A-5

## う

---

受取りルーチン, 6-22  
例, 6-22

## お

---

オペレーティング・システム・グループ  
oinstall, 1-9  
Oracle インベントリ・グループ, 1-9  
OSDBA, 1-11  
OSOPER, 1-11  
osoper, 1-9  
オペレーティング・システム・コマンド  
getprivgrp, B-3, B-4  
setprivgrp, B-3, B-4  
オペレーティング・システム・コマンド、実行, 4-4  
オペレーティング・システムのツール  
AIX, 8-5  
iostat, 8-4  
lsps, 8-4  
sar, 8-3  
swap, 8-4  
swapinfo, 8-4  
swapon, 8-4  
vmstat, 8-2  
オペレーティング・システムのバッファ・キャッシュ、  
チューニング, 8-14

## か

---

拡張ファイル・システム, 8-10  
仮想メモリー・データ・ページ  
Oracle Database のチューニング, B-7  
仮想メモリー・ページ・サイズ、デフォルト, B-8  
環境変数, 6-11  
A\_TERM, 6-10  
A\_TERMCAP, 6-10  
ADA\_PATH, 1-4  
AIXTHREAD\_SCOPE, A-15  
all, 1-2  
CLASSPATH, 1-4  
COBDIR, 6-11  
DISPLAY, 1-5

DYLD\_LIBRARY\_PATH, 1-5  
HOME, 1-5  
LANG, 1-5  
LANGUAGE, 1-5  
LD\_LIBRARY\_PATH, 1-5, 6-10, 6-12  
LD\_OPTIONS, 1-5  
LIBPATH, 1-5, 6-12  
LPDEST, 1-5  
MicroFocus COBOL コンパイラ, 6-11  
ORA\_TZFILE, 1-3  
ORACLE\_BASE, 1-3  
ORACLE\_HOME, 1-3  
ORACLE\_PATH, 1-3  
ORACLE\_SID, 1-2, 1-3  
ORACLE\_TRACE, 1-3  
ORAENV\_ASK, 1-3  
PATH, 1-6, 4-4, 6-10, 6-11  
PRINTER, 1-6  
Pro\*COBOL, 6-10  
SHLIB\_PATH, 1-6, 6-12  
SQLPATH, 1-4  
TMPDIR, 1-6  
TNS\_ADMIN, 5-2  
TWO\_TASK, 1-4  
XENVIRONMENT, 1-6  
環境変数 HOME, 1-5  
管理者  
オペレーティング・システム・アカウント, 1-11

## き

---

起動, 2-5  
iSQL\*Plus, 2-5  
Mac OS X での Oracle プロセス, 2-2  
Oracle CSS デーモン, 2-4  
Oracle Enterprise Manager Database Control, 2-7  
Oracle Management Agent, 2-8  
Oracle Ultra Search, 2-6  
自動化, 2-9  
キャッシュ・サイズ, 8-14  
共通の環境  
設定, 1-6  
共有メモリー、AIX, 8-13  
共有メモリー・セグメント, B-2  
緊密共有メモリー, E-2

## く

---

クライアント共有ライブラリ, 6-4  
クライアント静的ライブラリ, 6-4  
クラスター・ファイル・システム, 8-10  
グループ  
osdba, 1-9  
グループ,  
「オペレーティング・システム・グループ」を参照

## け

---

軽量タイマー, B-3  
軽量タイマーの実装, B-3

## こ

---

構成  
Oracle Database, 3-3  
Oracle ユーザーのアカウント, 1-11  
構成ファイル  
ottcfg.cfg, 6-2  
pbcfg.cfg, 6-2  
pccfor.cfg, 6-2  
pcscfg.cfg, 6-2  
pmscfg.cfg, 6-2  
プリコンパイラ, 6-2  
コマンド  
iostat, 8-4  
lsps, 8-4  
orapwd, 1-10  
sar, 8-3  
SPOOL, 4-5  
swap, 8-4  
swapinfo, 8-4  
swapon, 8-4  
vmo, A-2  
vmstat, 8-2  
コマンドライン SQL\*Plus の使用, 4-4  
コマンドライン SQL の管理, 4-2  
コンカレント I/O, A-7

## さ

---

再起動  
Oracle Database, 2-3  
Oracle Net Listener, 2-5  
自動ストレージ管理, 2-3  
削除  
SQL\*Plus のコマンドライン・ヘルプ, 4-3  
サブシステム属性  
説明, F-4  
サンプル・スキーマ,  
「Oracle Database サンプル・スキーマ」を参照

## し

---

シグナル  
SIGCLD, 6-21  
SIGCONT, 6-21  
SIGINT, 6-21  
SIGIO, 6-21  
SIGPIPE, 6-21  
SIGTERM, 6-21  
SIGURG, 6-21  
シグナル・ハンドラ, 6-21  
シグナル・ルーチン, 6-22  
例, 6-22  
システム・エディタ  
SQL\*Plus, 4-4  
システム時刻, 1-7  
システムに存在する RAD 数のサブセットへの Oracle  
Database の実行制限, F-5  
実行可能ファイル  
再リンク, 3-4  
プリコンパイラ, 6-2  
実行可能ファイルの再リンク, 3-4

指定配置最適化, F-2  
  使用, F-3  
  無効化, F-3  
  有効化, F-3  
  要件, F-2  
自動化  
  起動, 2-9  
  停止, 2-9  
自動ストレージ管理  
  再起動, 2-3  
  停止, 2-2  
自動ストレージ管理、使用, 8-10  
自動ストレージ管理プロセス  
  再起動, 2-3  
  停止, 2-2  
ジャーナル・ファイル・システム, 8-10, A-7, A-9  
シャドウ・プロセス, 1-10  
順次先読み  
  チューニング, A-12  
初期化パラメータ, 1-7  
  BACKGROUND\_DUMP\_DEST, 1-14  
  CPU\_COUNT, B-9  
  DB\_BLOCK\_SIZE, 8-12  
  DB\_CACHE\_SIZE, 8-12  
  JAVA\_POOL\_SIZE, 8-12  
  LARGE\_POOL\_SIZE, 8-12  
  LOG\_BUFFERS, 8-12  
  MAX\_DUMP\_FILE, 1-14  
  SHARED\_POOL\_SIZE, 8-12  
  USER\_DUMP\_DEST, 1-14  
初期化パラメータ ASM\_DISKSTRING, 1-8  
初期化パラメータ DB\_BLOCK\_SIZE, 1-7  
初期化パラメータ LOG\_ARCHIVE\_DEST\_n, 1-8

## す

---

スクリプト  
  root.sh, 1-7  
ストライプ化された論理ボリューム  
  設計, A-6  
スレッドのサポート, 6-21  
スワップ領域, 8-8  
  使用可能および使用済の決定, D-2  
  チューニング, 8-8  
スワップ領域の割当て, 8-8

## せ

---

制限事項、SQL\*Plus, 4-5  
  ウィンドウのサイズ変更, 4-5  
  パスワード, 4-5  
  リターン・コード, 4-5  
静的リンク  
  Oracle ライブラリとプリコンパイラ, 6-3  
セキュリティ, 1-10  
  2タスク構造, 1-10  
  オペレーティング・システムの機能, 1-10  
  グループ・アカウント, 1-10  
  ファイル所有権, 1-10  
専用 SQL 領域, B-7  
専用メモリー, B-7

## そ

---

属性  
  aio\_task\_max\_num, F-7

## た

---

ダイレクト I/O, A-7

## ち

---

チューニング, 8-8  
  I/O ボトルネック, 8-10  
  順次先読み, A-12  
  ディスク I/O, 8-10  
  メモリー管理, 8-8  
チューニング・ツール  
  Glance/UX ユーティリティ, 8-6  
  iostat コマンド, 8-4  
  lspv コマンド, 8-4  
  mpstat, 8-7  
  Programmer's Analysis Kit (HP PAK), 8-6  
  PTX Agent, 8-5  
  PTX Manager, 8-5  
  sar コマンド, 8-3  
  swapinfo コマンド, 8-4  
  swapon コマンド, 8-4  
  swap コマンド, 8-4  
  vmstat コマンド, 8-2

## て

---

停止  
  iSQL\*Plus, 2-5  
  Oracle CSS デーモン, 2-4  
  Oracle Database, 2-2  
  Oracle Enterprise Manager Database Control, 2-7  
  Oracle Management Agent, 2-8  
  Oracle Net Listener, 2-4  
  Oracle Ultra Search, 2-6  
  自動化, 2-9  
  自動ストレージ管理, 2-2  
ディスク  
  パフォーマンスの監視, 8-11  
ディスク I/O  
  I/O スレーブ, A-11  
  チューニング, 8-10  
  ファイル・システム・タイプ, 8-10  
ディスク I/O の歩調合せ  
  チューニング, A-13  
データベース  
  ブロック・サイズ, A-5  
データベース統計、収集, F-6  
データベースの制限, H-2  
データベース・ブロック・サイズ  
  設定, A-5  
デバッガ・プログラム, 6-3  
デモ  
  PL/SQL, 7-2  
  SQL\*Loader, 7-2  
  プリコンパイラ, 7-4

## デモ・プログラム

Oracle Call Interface, 6-17  
Oracle JDBC/OCI, 6-19  
Pro\*C/C++, 6-7  
Pro\*COBOL, 6-12  
Pro\*FORTRAN, 6-14  
SQL\*Module for Ada, 6-16

## と

動的キャッシュ・パラメータ, C-2

動的リンク

Oracle ライブラリとプリコンパイラ, 6-3

特殊アカウント

Oracle ソフトウェア所有者アカウント, 1-9

トレース・アラート, 1-14

トレース・ファイル, 1-14

## ね

ネットワークのサポート, C-7

## は

バッファ・キャッシュ

拡張、制限事項, C-3

チューニング, A-2

バッファ・キャッシュ・サイズ

チューニング, 8-14

バッファ・キャッシュのサポート、拡張, C-2

バッファ・キャッシュのページング・アクティビティ  
制御, A-2

バッファ・マネージャ, 8-8

パフォーマンス・チューニング・ツール, 8-7

パラメータ

BUFFER, A-5

CREATE CONTROLFILE, H-2

CREATE DATABASE, H-2

DB\_FILE\_MULTIBLOCK\_READ\_COUNT, A-12

MAXDATAFILES, H-2

maxfree, A-2

MAXLOGFILES, H-2

MAXLOGHISTORY, H-2

MAXLOGMEMBERS, H-2

maxperm, A-2, A-3

minfree, A-2

minperm, A-2, A-3

OS\_AUTHENT\_PREFIX, 1-10

rad\_gh\_regions, F-4

SCHED\_NOAGE, B-2

SGA\_MAX\_SIZE, E-2

shm\_allocate\_stripped, F-4

shm\_max, 8-12

shmmx, 8-12

shm\_seg, 8-12

shmseg, 8-12

TIMED\_STATISTICS, F-9

USE\_INDIRECT\_DATA\_BUFFERS, C-2

動的キャッシュ, C-2

## ひ

ビット長サポート, 6-5

非同期 I/O, B-4

検証, B-6

非同期 I/O の実装, B-4

表

PRODUCT\_USER\_PROFILE, 4-2

## ふ

ファイル

coraenv, 1-6

dbhome, 1-7

demo\_proc32.mk, 6-8

demo\_proccob.mk, 6-13

glogin.sql, 4-2

ins\_precomp.mk, 6-2

login.sql, 4-2

Oracle Net Services の構成, 5-2

oraenv, 1-6

ottcfg.cfg, 6-2

pcbcfg.cfg, 6-2

pccfor.cfg, 6-2

pcscfg.cfg, 6-2

pmscfg.cfg, 6-2

privgroup, B-3, B-4

README, 6-3

root.sh, 1-7

services, 5-6

アラート, 1-14

トレース, 1-14

ファイル・システム, 8-10

AdvFS, 8-10

ext2/ext3, 8-10

GPFS, 8-11

JFS, 8-10

OCFS, 8-10

S5, 8-10

UFS, 8-10

VxFS, 8-10

ファイル・バッファ・キャッシュ

AIX でのチューニング, A-3

複合 CPU システム, F-6

複数のシグナル・ハンドラ, 6-21

プリコンパイラ

IRECLEN および ORECLN の値, 6-3

Pro\*C/C++, 6-7

Pro\*COBOL, 6-9

大文字から小文字への変換, 6-3

概要, 6-2

シグナル, 6-21

実行可能ファイル, 6-2

デモの実行, 7-4

ベンダー提供のデバッガ・プログラム, 6-3

プリコンパイラ構成ファイル

ファイル

プリコンパイラ構成, 6-2

プリコンパイラ実行可能ファイル

再リンク, 6-2

プリコンパイラの README ファイル, 6-3

プロセスごとの最大ロック・メモリー  
VLM ウィンドウ・サイズおよび Red Hat Enterprise  
Linux 3, C-3  
ブロック・サイズ, A-5  
調整, 8-10  
プロトコル, 5-4

## へ

---

ページアウト・アクティビティ, 8-9  
ページング  
制御, A-4  
ページング、制御, 8-9  
ページング領域, 8-8  
十分な割当て, A-4  
チューニング, 8-8, 8-9

## ま

---

マルチスレッド・アプリケーション, 6-21

## み

---

未定義シンボル, 6-20  
ミラー復元、Oracle Database, A-13

## め

---

メモリー  
競合, A-2  
チューニング, 8-8  
メモリー管理, 8-8  
スワップ領域, 8-8  
ページングの制御, 8-9  
メモリーとページング、AIX 上, A-2  
メモリー内ファイル・システム, C-2

## ゆ

---

ユーザー・プログラム  
OCCI, 6-18  
OCI, 6-18  
Pro\*C/C++, 6-8  
Pro\*COBOL, 6-13  
SQL\*Module for Ada, 6-17  
ユーザー・プロファイル  
SQL\*Plus, 4-2  
ユーザー割込みハンドラ, 6-22  
ユーティリティ  
adapters, 5-3  
orapwd, 1-10  
symfind, 6-20  
インポート, A-5

## ら

---

ライブラリ  
クライアント共有とクライアント静的, 6-4

## り

---

リアルタイム・クロック, F-9  
リスナー  
TCP/IP または SSL 付き TCP/IP 用の設定, 5-6

## ろ

---

論理ボリューム・マネージャ  
「LVM」を参照