Oracle9*i* for UNIX Systems

管理者リファレンス

リリース 2（9.2.0.1.0）

2002 年 7 月

部品番号 : J06454-01

ORACLE®

Oracle9*i* for UNIX Systems 管理者リファレンス , リリース 2（9.2.0.1.0）

部品番号 : J06454-01

原本名：Oracle9*i* Administrator's Reference Release 2 (9.2.0.1.0) for UNIX Systems: AIX-Based Systems, Compaq Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel and Sun Solaris

原本部品番号：A97297-01

原本著者：Platform Technologies Division Documentation Team

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

Printed in Japan.

制限付権利の説明

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

当プログラムのリバース・エンジニアリング等は禁止されております。

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

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

危険な用途への使用について

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

Restricted Rights Notice

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

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

目次

はじめに xiii

- 対象読者 xiv
- Oracle9*i* Standard Edition および Oracle9*i* Enterprise Edition xiv
- 用語 xiv
- このマニュアルで使用する表記規則 xv
- コマンドの構文 xv
- インストールしたドキュメントの表示 xvi
- 関連ドキュメント xvi
- Oracle サービスおよびサポート xvii

1 Oracle9*i* の管理

- **概要** 1-2
- **環境変数** 1-2
  - Oracle9*i* の環境変数 1-2
  - UNIX 環境変数 1-6
  - 共通の環境設定 1-8
    - oraenv スクリプト・ファイル 1-8
    - ローカル bin ディレクトリ 1-9
    - データベース間の移動 1-9
  - 現行のセッションでの環境変数値の設定およびエクスポート 1-9
  - システム時刻の設定 1-10
- **共有ライブラリをロードするための環境変数 LD_PRELOAD（HP のみ）** 1-10
- **実行可能ファイルの再リンク** 1-12
- **システム・グローバル領域** 1-13
  - SGA サイズの確認 1-14

Intimate Shared Memory（Solaris のみ） 1-14
AIX での共有メモリー 1-15
Oracle9*i* のメモリー要件 1-17
データベースの制限 1-17
オペレーティング・システムのアカウントとグループ 1-18
Oracle ソフトウェア所有者アカウント 1-18
OSDBA、OSOPER および ORAINVENTORY グループ 1-18
グループとセキュリティ 1-20
データベース・ファイルのセキュリティ 1-20
外部認証 1-20
orapwd ユーティリティの実行 1-21
パスワード管理 1-22
初期化ファイルのカスタマイズ 1-22
Oracle HTTP Server 1-25
Oracle HTTP Server のログ・ファイル 1-26
デモ・ファイル 1-27
SQL*Loader のデモ 1-27
PL/SQL デモ 1-28
PL/SQL カーネル・デモ 1-28
PL/SQL プリコンパイラ・デモ 1-30
SQL*Loader の管理 1-31
固定長レコードの改行文字 1-31
改行文字の削除 1-32

2 Oracle9*i* のチューニング

チューニングの重要性 2-2
パフォーマンスのボトルネックの種類 2-2
オペレーティング・システムのツール 2-3
共通のツール 2-3
Linux のツール 2-6
Solaris のツール 2-6
AIX のツール 2-7
AIX System Management Interface Tool 2-7
Base Operating System ツール 2-7
AIX Performance Toolbox 2-8
HP のツール 2-9

パフォーマンス・チューニング・ツール 2-9
HP パフォーマンス分析ツール 2-9
メモリー管理のチューニング 2-10
十分なスワップ領域の割当て 2-10
ページングの制御 2-12
Oracle ブロック・サイズの調整 2-12
ディスク I/O のチューニング 2-13
適切なファイル・システム・タイプの選択 2-13
ディスク・パフォーマンスの監視 2-14
UNIX カーネル・パラメータのチューニング 2-15
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 2-15
RAW デバイス / ボリュームの使用 2-16
RAW デバイス / ボリュームを使用する場合のガイドライン 2-16
RAW デバイスの設定 2-18
トレース・ファイルおよびアラート・ファイルの使用 2-18
トレース・ファイル 2-18
アラート・ファイル 2-19

3 SQL*Plus および *i*SQL*Plus の管理

コマンドライン SQL*Plus の管理 3-2
設定ファイルの使用 3-2
サイト・プロファイル・ファイルの使用 3-2
ユーザー・プロファイル・ファイルの使用 3-2
PRODUCT_USER_PROFILE 表の使用 3-3
デモンストレーション表の使用 3-4
EMP および DEPT 表の使用 3-4
デモンストレーション表の手動作成 3-4
デモンストレーション表の削除 3-4
SQL*Plus のコマンドライン・ヘルプ 3-5
SQL*Plus のコマンドライン・ヘルプのインストール 3-5
SQL*Plus のコマンドライン・ヘルプの削除 3-6
***i*SQL*Plus の管理** 3-6
*i*SQL*Plus の無効化および再有効化 3-6
*i*SQL*Plus 構成ファイルの編集 3-7
セキュリティ 3-9
*i*SQL*Plus の Oracle HTTP Server 認証の設定 3-9

パスワード・ファイルへのユーザー名およびパスワードの追加 3-10
新しいパスワード・ファイルが使用されるように Oracle HTTP Server を設定する方法 3-11
*i*SQL*Plus からのデータベース・アクセスの制限 3-12
コマンドライン SQL*Plus の使用 3-13
SQL*Plus からのシステム・エディタの使用 3-13
SQL*Plus からのオペレーティング・システム・コマンドの実行 3-14
SQL*Plus への割込み 3-14
SPOOL コマンドの使用 3-14
SQL*Plus の制限事項 3-15
ウィンドウのサイズ変更 3-15
リターン・コード 3-15
パスワードの隠ぺい 3-15

4 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 4-2
プリコンパイラ構成ファイル 4-2
プリコンパイラ実行可能ファイルの再リンク 4-3
プリコンパイラの README ファイル 4-4
すべてのプリコンパイラに共通の問題 4-4
大文字から小文字への変換 4-4
ベンダー提供のデバッグ・プログラム 4-4
IRECLEN および ORECLEN の値 4-4
静的および動的リンク 4-5
クライアント共有ライブラリ 4-5
32 ビットおよび 64 ビットのクライアント・アプリケーションのサポート 4-7
Pro*C/C++ プリコンパイラ 4-8
Pro*C/C++ のデモ・プログラム 4-9
Pro*C/C++ のユーザー・プログラム 4-10
Pro*COBOL プリコンパイラ（AIX、HP、Solaris および Tru64 のみ） 4-11
Pro*COBOL の環境変数 4-11
Merant Server Express COBOL コンパイラ 4-11
Pro*COBOL の Oracle ランタイム・システム 4-13
Pro*COBOL のデモ・プログラム 4-13
Pro*COBOL のユーザー・プログラム 4-14
FORMAT プリコンパイラ・オプション 4-15
Pro*FORTRAN プリコンパイラ（AIX、HP、Solaris および Tru64 のみ） 4-15

Pro*FORTRAN のデモ・プログラム 4-16
Pro*FORTRAN のユーザー・プログラム 4-17
SQL*Module for Ada（Solaris 32 ビットおよび AIX のみ） 4-18
SQL*Module for Ada のデモ・プログラム。.......... 4-18
SQL*Module for Ada のユーザー・プログラム 4-19
Oracle Call Interface 4-19
OCI のデモ・プログラム 4-20
OCI のユーザー・プログラム 4-20
カスタム Make ファイル 4-22
未定義シンボルの修正（Solaris のみ） 4-23
マルチスレッド・アプリケーション 4-24
シグナル・ハンドラの使用方法 4-24
XA 機能 4-27

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 5-5
IPC プロトコル・サポート 5-5
TCP/IP プロトコル・サポート 5-6
SSL 付き TCP/IP プロトコル・サポート 5-7
TCP/IP または SSL 付き TCP/IP 用のリスナーの設定 5-8
Oracle Enterprise Manager 5-8
Oracle SNMP での Oracle Intelligent Agent の構成 5-9
マスター・エージェントの構成 5-9
カプセル化機能の構成 5-9
start_peer スクリプトの SNMP デーモンの保存場所の確認 5-10
SNMP コンポーネントの起動 5-10
データベース・サブエージェントの構成および起動 5-11
Oracle Advanced Security 5-11
PL/SQL からの 32 ビット外部プロシージャのコール（AIX、HP および Solaris 64 ビットのみ） 5-12

A Oracle9*i* for AIX-Based Systems のチューニング

メモリーおよびページング A-2
バッファ・キャッシュ間のページング・アクティビティの制御 A-2
MINFREE および MAXFREE パラメータのチューニング A-3

AIX ファイル・バッファ・キャッシュのチューニング A-3
MINPERM および MAXPERM パラメータのチューニング A-4
十分なページング領域の割当て A-5
ページングの制御 A-5
データベース・ブロック・サイズの設定 A-6
ログ・アーカイブ・バッファのチューニング A-6
I/O バッファおよび SQL*Loader A-7
インポート・ユーティリティ用の BUFFER パラメータ A-7
ディスク I/O の問題 A-8
AIX 論理ボリューム・マネージャ A-8
ストライプ化された論理ボリュームの設計 A-8
ストライプ化論理ボリュームのパラメータ（推奨） A-9
その他の考慮事項 A-9
ジャーナル・ファイル・システムを使用した場合と RAW パーティションを使用した場合の相違 A-9
ジャーナル・ファイル・システムから RAW デバイスへの移動 A-10
ジャーナル・ファイル・システムおよび RAW デバイスの同時利用 A-11
非同期 I/O の使用 A-11
I/O スレーブ A-13
DB_FILE_MULTIBLOCK_READ_COUNT パラメータの使用 A-14
RAID 機能の使用 A-14
後書きの使用 A-14
順次ファイルの先読みのチューニング A-15
ディスク I/O の歩調合せのチューニング A-16
ディスク形状の考慮事項 A-16
リモート I/O 操作の最小化 A-16
VSD キャッシュ・バッファ A-17
CPU のスケジューリングおよびプロセスの優先順位 A-18
プロセスのランタイム・スライスの変更 A-18
SMP システム上でのプロセッサ・バインディングの使用 A-18
ネットワークのクライアントおよびサーバー環境でのプロセッサ・バインディング A-19
ローカル環境でのプロセッサ・バインディング A-20
UDP のチューニング A-21
RAW デバイスのバックアップ A-21
Oracle9*i* でのミラー復元 A-22

B Oracle9*i* for HP 9000 Series HP-UX のチューニング

64 ビットの 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 SCHED_NOAGE スケジューリング・ポリシー B-3
Oracle9*i* の SCHED_NOAGE の有効化 B-3
軽量タイマーの実装 B-4
非同期 I/O B-4
MLOCK 権限 B-4
非同期 I/O の実装 B-5
非同期 I/O の検証 B-7
SGA の非同期フラグ B-7

C Oracle9*i* for Linux Intel のチューニング

拡張バッファ・キャッシュのサポート C-2
非同期 I/O サポート C-3

D Oracle9*i* for Compaq Tru64 UNIX のチューニング

Oracle9*i* 指定配置最適化の有効化 D-2
指定配置最適化を実行するための要件 D-3
Oracle 指定配置最適化の有効化 D-3
Oracle 指定配置最適化の無効化 D-4
Oracle 指定配置最適化の使用 D-4
Oracle 初期化パラメータ D-4
Tru64 UNIX システム・パラメータ D-4
RAD に対するプロセスの親和性 D-5
複合 CPU システムのサポート D-6
Tru64 でのデータベース統計の収集 D-6
Tru64 での Oracle9*i* Real Application Clusters D-7
Reliable Data Gram D-7
要件 D-7
UDP IPC の有効化 D-8
CLUSTER_INTERCONNECTS 初期化パラメータ（旧 TRU64_IPC_NET）........ D-9
非同期 I/O のチューニング D-11
aio_task_max_num パラメータ D-11
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 D-12
単一インスタンスの要件 D-12

クラスタ化されたシステム D-12
Tru64 UNIX V5.1 クラスタ化システム D-13
複数インスタンスの要件（Oracle9*i* Real Application Clusters）.......... D-13
ダイレクト I/O サポートの無効化 D-14
ファイルの断片化の防止 D-15
リアルタイム・クロックへのアクセスの有効化 D-16
RAW デバイスの設定 D-17
SPIKE 最適化ツール D-19
SPIKE の使用 D-20

E Oracle *inter*Media、Oracle Text、および Oracle Spatial のデモの実行

Oracle *inter*Media E-2
Oracle *inter*Media Audio，Image，Video E-2
Oracle *inter*Media Annotator E-3
Locator E-3
Oracle Text E-3
Oracle Spatial E-3

F Oracle Cluster Management Software for Linux

概要 F-2
Watchdog デーモン F-3
Cluster Manager F-3
OCMS の起動 F-5
Watchdog デーモンの起動 F-5
Cluster Manager の構成 F-6
Cluster Manager の起動 F-9
クラスタ再構成のタイミングの設定 F-10
Watchdog デーモンおよび Cluster Manager の起動オプション F-11

G Optimal Flexible Architecture

Optimal Flexible Architecture G-2
OFA に準拠したデータベースの特長 G-2
UNIX で適用される Optimal Flexible Architecture G-4
マウント・ポイント G-4
マウント・ポイントの作成 G-4

マウント・ポイントの構文 G-4
超大規模データベース（VLDB）のマウント・ポイントの名前 G-4
ディレクトリの名前 G-5
ホーム・ディレクトリの構文 G-5
パス名の参照 G-5
ソフトウェア・ディレクトリ G-6
サブディレクトリの名前 G-6
データベース・ファイルの名前 G-7
要件に応じた個々のセグメント G-8
表領域の名前 G-9
OFA 構造に基づいた Oracle ファイル G-9
OFA ファイルのマッピング G-10
複数インスタンスにおける OFA に準拠したデータベースのファイル・マッピング G-11
ディレクトリ構造 G-13
Oracle ベース・ディレクトリ G-13
Oracle ホーム・ディレクトリ G-13
製品のサブディレクトリの例 G-15
製品のサブディレクトリの内容 G-15
admin ディレクトリのファイルのネーミング規則 G-16
ファイル名の拡張子 G-17

索引

表目次

1-1 UNIX での Oracle9*i* 環境変数 1-3
1-2 Oracle9*i* で使用する UNIX 環境変数 1-6
1-3 再リンク・スクリプトの引数 1-12
1-4 CREATE CONTROLFILE および CREATE DATABASE のパラメータ 1-17
1-5 ファイル・サイズの制限 1-17
1-6 UNIX グループ 1-18
1-7 初期化パラメータ 1-23
2-1 sar -b 出力の列 2-14
4-1 Oracle プリコンパイラのシステム構成ファイル 4-2
4-2 製品とそれに対応する実行可能ファイル 4-3
4-3 プリコンパイラの README ファイルの保存場所 4-4
4-4 Pro*COBOL のネーミング規則 4-11
4-5 2 タスク通信に使用するシグナル 4-24
D-1 RDG サブシステムのオペレーティング・システム・パラメータの設定 D-7
D-2 SPIKE のシステム・リソース要件 D-20
E-1 Oracle *inter*Media のデモ・プログラム E-2
F-1 watchdogd デーモンの引数 F-6
F-2 cmcfg.ora ファイルの Cluster Manager パラメータ F-7
F-3 `oracm` 実行可能プログラムの引数 F-9
G-1 マウント・ポイントの名前の構文 G-5
G-2 ホーム・ディレクトリの名前の構文 G-5
G-3 Oracle9*i* ソフトウェアを保存するディレクトリ名の構文 G-6
G-4 データベース管理ファイルのサブディレクトリ G-6
G-5 特殊な表領域 G-8
G-6 ファイルのクラスを識別するためのディレクトリ構造の構文 G-9
G-7 OFA インストールの階層型ファイル・マッピング G-10
G-8 二重インスタンスを持つ Oracle9*i* Real Application Clusters の管理ディレクトリ構造 G-11
G-9 Oracle ベース・ディレクトリ構造および内容 G-13
G-10 Oracle ホーム・ディレクトリ構造および内容（一部） G-13
G-11 製品のサブディレクトリの例 G-15
G-12 製品のサブディレクトリの内容 G-15
G-13 admin ディレクトリのファイルのネーミング規則 G-16
G-14 ファイル名の拡張子 G-17

はじめに

このマニュアルおよび『Oracle9*i* for UNIX Systems インストレーション・ガイド』では、UNIX システムで Oracle9*i* リリース 2（9.2.0.1.0）を管理および構成する方法について説明します。

対象読者

このマニュアルは、UNIX システムで Oracle9*i* リリース 2（9.2.0.1.0）の管理および構成を担当する方を対象としています。

Oracle9*i* Standard Edition および Oracle9*i* Enterprise Edition

このマニュアルの情報は、特に明記されていないかぎり、Oracle9*i* Standard Edition および Oracle9*i* Enterprise Edition に共通です。

用語

このマニュアルおよび『Oracle9*i* for UNIX Systems インストレーション・ガイド』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 - Based Systems	AIX **注意 :** AIX 4.3.3 と AIX 5.1 の相違点については本文中に記載します。
HP 9000 Series HP-UX	HP
Linux Intel（32 ビット）	Linux
Sun Solaris（32 ビットおよび 64 ビット）	Solaris **注意** : 32 ビット・システムと 64 ビット・システムの相違点については本文中に記載します。
Compaq Tru64 UNIX	Tru64

このマニュアルで使用する表記規則

このマニュアルでは、次の表記規則を使用しています。

表記規則	説明
固定幅フォント	固定幅フォントは、UNIX コマンド、ディレクトリ名、ユーザー名、パス名およびファイル名を表します。
イタリック体	イタリック体は、変数（ファイル名の中で変化する部分を含む）を表します。
大文字	大文字は、SQL の予約語、初期化パラメータおよび環境変数を表します。
<cr>	この文字列は、改行文字を表します。

コマンドの構文

UNIX コマンドの構文は固定幅フォントで表し、Bourne シェルの使用を前提としています。UNIX コマンド例の先頭の `$` 文字は、デフォルトの UNIX コマンド・プロンプトです。コマンドの一部ではありませんので、入力しないでください。

表記規則	説明

バックスラッシュ \	バックスラッシュは、コマンドが 1 行に入りきらない場合に使用します。このマニュアルで記載しているとおりに入力する（バックスラッシュを付ける）か、バックスラッシュを付けずに 1 行で入力します。
	`dd if=/dev/rdsk/c0t1d0s6 of=/dev/rst0 bs=10b \`
	`count=10000`

中カッコ { }	中カッコは、必ず選択する項目を表します。 `.DEFINE {macro1}`
大カッコ [ ]	大カッコは、任意に選択する項目を表します。 `cvtcrt` *`termname`* `[`*`outfile`*`]`
省略記号 ...	省略記号は、同じ項目を任意の回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。変数の箇所を適切な値に置き換えてください。 *`library_name`*
縦棒線 |	縦棒線は、中カッコまたは大カッコで囲まれている選択項目を表します。 *`SIZE filesize`* `[K|M]`

インストールしたドキュメントの表示

Oracle9*i* for UNIX Systems リリース 2（9.2.0.1.0）のドキュメントには、このマニュアルおよび『Oracle9*i* for UNIX Systems インストレーション・ガイド』があります。PDF（Adobe Portable Document Format、表示するには Acrobat Reader が必要）形式のドキュメントをインストールできます。UNIX 固有のドキュメントのファイルは、Oracle9*i* CD-ROM にあります。Oracle9*i* 関連ドキュメントのファイルは、Oracle9*i* ドキュメント・セット CD にあります。ドキュメントのファイルの正確な位置は、次の規則に従って決定されます。

- 環境変数 ORACLE_DOC が定義されている場合、この変数に定義したディレクトリにファイルがインストールされます。
- 環境変数 ORACLE_DOC は定義されていないが、環境変数 ORACLE_BASE が定義されている場合、ファイルは `$ORACLE_BASE/doc` ディレクトリにインストールされます。
- ORACLE_DOC または ORACLE_BASE のどちらの環境変数も定義されていない場合、ファイルは `$ORACLE_HOME/doc` ディレクトリにインストールされます。

ドキュメントを参照するには、ドキュメント・ディレクトリに移動します。HTML ドキュメントを表示する場合は、ブラウザを使用して `index.htm` ファイルを開きます。紙マニュアルが必要な場合は、PDF ファイルから印刷してください。

Oracle 製品のドキュメント

Oracle9*i* 製品のドキュメントは、Oracle9*i* ドキュメント CD にあります。CD-ROM のドキュメントの表示およびインストール方法については、CD-ROM の最上位ディレクトリにある README ファイルを参照してください。

関連ドキュメント

リレーショナル・データベース管理システム関連の概念または用語に慣れていない場合は、インストールを開始する前に、『Oracle9*i* データベース概要』を参照してください。さらに、『Oracle9*i* for UNIX Systems インストレーション・ガイド』のインストレーション・チェックリストを使用して、必要な情報が準備できていること、インストール前に必要な作業が完了していることを確認してください。

本番データベース・システム用のシステム管理およびチューニングの詳細は、次のマニュアルを参照してください。

- 『Oracle9*i* for UNIX Systems インストレーション・ガイド』
- 『Oracle9*i* データベース管理者ガイド』
- 『Oracle9*i* Net Services 管理者ガイド』
- 『Oracle9*i* データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』

前回のリリースの Oracle サーバーからのアップグレードの詳細は、『Oracle9*i* データベース移行ガイド』を参照してください。

Oracle サービスおよびサポート

オラクル社の Web ページの URL は次のとおりです。

http://www.oracle.co.jp

オラクル社は、この Web サイトを通じて、Oracle Consulting サービスおよび Oracle Worldwide Customer Support などの幅広いサービスを提供することにより、企業システム・ソリューションをサポートしています。無償の試用版ソフトウェア、Oracle 製品およびサービスの最新情報、製品パンフレットおよびデータ・シートも提供しています。

Oracle サポート・サービス

テクニカル・サポートへの登録方法および連絡先は、次の URL に示されています。

http://www.oracle.co.jp/support

Oracle サポート・サイトでは、より迅速に問題を解決できるように、お電話をいただく前に問題に関する情報を準備していただくためのテンプレートを用意しています。該当する場合は、CSI 番号（カスタマ・サポート番号）か SAC 番号（サポート・アクセス・コード）、または詳しい連絡先情報（特別なプロジェクト情報がある場合は、それも含む）も記入します。

オラクル社カスタマ・サポート・センター

オラクル社カスタマ・サポート・センターの連絡先は、次の URL から参照できます。

http://www.oracle.co.jp/support/

ドキュメントの外部 Web サイトへのリンクのアクセス可能性

このドキュメントには、オラクル社が所有または管理しない他の企業や組織の Web サイトへのリンクが含まれている場合があります。オラクル社では、これらの Web サイトのアクセス可能性に関して評価することも陳述することもいっさいありません。

1

Oracle9*i* の管理

この章では、AIX、HP、Linux、Solaris および Tru64 で Oracle9*i* を管理する方法について説明します。この章では、次の項目について説明します。

- 概要
- 環境変数
- 共有ライブラリをロードするための環境変数 LD_PRELOAD（HP のみ）
- 実行可能ファイルの再リンク
- システム・グローバル領域
- Oracle9i のメモリー要件
- データベースの制限
- オペレーティング・システムのアカウントとグループ
- 初期化ファイルのカスタマイズ
- Oracle HTTP Server
- デモ・ファイル

概要

Oracle9*i* を使用するには、Oracle9*i* の環境変数、パラメータ、およびユーザー設定を準備する必要があります。この章では、AIX、HP、Linux、Solaris および Tru64 で動作する Oracle9*i* の各種設定について説明します。

Oracle9*i* 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、Oracle9*i* 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 2M
```

同様に、アットマーク（@）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指定する場合、次のように入力します。

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

環境変数

この項では、通常使用される Oracle9*i* および UNIX の環境変数について説明します。Oracle9*i* をインストールする前に、環境変数をいくつか定義する必要があります。これらの環境変数は、『Oracle9*i* for UNIX Systems インストレーション・ガイド』に記載されています。

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、次のように入力します。

```
$ env | grep ORACLE_SID
```

Oracle9*i* の環境変数

表 1-1 に、Oracle9*i* で使用する環境変数の構文および例を示します。

表 1-1 UNIX での Oracle9*i* 環境変数

環境変数	項目	定義
EPC_DISABLED	機能	Oracle Trace を使用禁止にします。
	構文	`TRUE ｜FALSE`
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 で設定するキャラクタ・セットは、端末または端末エミュレータのキャラクタ・セットと一致する必要があります。NLS_LANG で設定されたキャラクタ・セットがデータベース側のキャラクタ・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、『Oracle9*i* 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	*`language_territory.characterset`*
	例	`french_france.we8dec`
ORA_NLS33	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	*`directory_path`*
	例	`$ORACLE_HOME/ocommon/nls/admin/data`
ORA_TZFILE	機能	タイムゾーン・ファイルのフルパス名を指定します。この環境変数は、データベース内のデータに対して大規模タイムゾーン・ファイル（`$ORACLE_HOME/oracore/zoneinfo/timezlrg.dat`）のタイムゾーンを使用する場合に設定する必要があります。大規模タイムゾーン・ファイルには、タイムゾーンに関する情報がデフォルトのタイムゾーン・ファイル（`$ORACLE_HOME/oracore/zoneinfo/timezone.dat`）よりも多く含まれています。 情報を共有するデータベースはすべて、同一のタイムゾーン・ファイルを使用する必要があります。この環境変数を変更するときは、データベースをいったん停止してから、再起動する必要があります。
	構文	*`directory_path`*
	例	`$ORACLE_HOME/oracore/zoneinfo/timezlrg.dat`
ORACLE_BASE	機能	Optimal Flexible Architecture（OFA）に準拠した Oracle データベース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	*`directory_path`*
	例	`/u01/app/oracle`

表 1-1 UNIX での Oracle9*i* 環境変数（続き）

環境変数	項目	定義
ORACLE_HOME	機能	Oracle ソフトウェアがあるディレクトリを指定します。
	構文	*directory_path*
	例	`$ORACLE_BASE/product/9.2.0.1.0`
ORACLE_PATH	機能	SQL*Plus（`*.sql`）、Oracle Forms（`*.frm`）、Oracle Reports（`*.rpt`）など、Oracle アプリケーションが使用するファイルの検索パス名を指定します。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を見つけます。
	構文	ディレクトリをコロンで区切ったリスト *directory1:directory2:directory3*
	例	`/u01/app/oracle/product/9.2.0.1.0/bin:.` **注意 :** 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 システム識別子（Oracle System Identifier: SID）を指定します。
	構文	英字で始まる数字と文字からな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薦めします。この環境変数の詳細は、『Oracle9*i* for UNIX Systems 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` に設定すると、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。
	構文	`T` または `T` 以外
ORAENV_ASK	機能	`coraenv` または `oraenv` スクリプトが、環境変数 ORACLE_SID または ORACLE_HOME 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指定します。`NO` を設定すると、プロンプトは表示されません。`NO` 以外を設定するとプロンプトが表示されます。
	構文	文字列。
	例	`NO` または `NO` 以外
SQLPATH	機能	SQL*Plus が `login.sql` ファイルを見つけるために検索する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。

表 1-1 UNIX での Oracle9*i* 環境変数（続き）

環境変数	項目	定義
	構文	ディレクトリをコロンで区切ったリスト *directory1:directory2:directory3*
	例	/home:/home/oracle:/u01/oracle
TNS_ADMIN	機能	Oracle Net Services 構成ファイルが入っているディレクトリを指定します。
	構文	*directory_path*
	例	$ORACLE_HOME/network/admin
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。この接続識別子を接続文字列に指定する必要はありません。たとえば、環境変数 TWO_TASK が sales に設定されている場合は、CONNECT *username/password*@sales コマンドではなく CONNECT *username/password* 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・メソッドで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	PRODDB_TCP

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

UNIX 環境変数

表 1-2 に、Oracle9*i* で使用する UNIX 環境変数の構文および例を示します。

表 1-2 Oracle9*i* で使用する UNIX 環境変数

環境変数	項目	定義
ADA_PATH (Solaris 32 ビットおよび AIX のみ)	機能	Ada コンパイラが入っているディレクトリを指定します。
	構文	*directory_path*
	例	/usr/lpp/powerada
CLASSPATH	機能	Java アプリケーションで使用します。この環境変数の設定は、Java アプリケーションごとに異なります。詳細は、ご使用の 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 *directory1:directory2:file1:file2*
	例	デフォルトの設定はありません。CLASSPATH は、次のディレクトリ内にある必要があります。 $ORACLE_HOME/JRE/lib:$ORACLE_HOME/product/jlib
DISPLAY	機能	X ベースのツールで使用します。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、各ベンダーの X Window のドキュメントを参照してください。
	構文	*hostname*:*server*[.*screen*] *hostname* はコンピュータ名（IP アドレスまたは別名）、*server* はサーバーの順次コード番号、*screen* は画面の順次コード番号です。モニターが 1 つの場合、*server.screen* は 0.0 です。 **注意 :** モニターが 1 つの場合、*screen* はオプションです。
	例	135.287.222.12:0.0 bambi:0
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	*directory_path*
	例	/home/oracle
LANG または LANGUAGE	機能	メッセージなどを出力するためにオペレーティング・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および『Oracle9*i* for UNIX Systems インストレーション・ガイド』を参照してください。
LD_OPTIONS	機能	デフォルトのリンカー・オプションを指定します。詳細は、ld の man ページを参照してください。

表 1-2 Oracle9*i* で使用する UNIX 環境変数（続き）

環境変数	項目	定義
LPDEST（Solaris のみ）	機能	デフォルト・プリンタの名前を指定します。
	構文	*string*
	例	docprinter
LDPATH（Solaris のみ）	機能	共有オブジェクト・ライブラリを検索するためにリンカーが使用するデフォルトのディレクトリを指定します。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。
LD_LIBRARY_PATH	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・ライブラリを検索するディレクトリのリストを示します。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 *directory1:directory2:directory3*
	例	/usr/dt/lib:$ORACLE_HOME/lib
LIBPATH（AIX のみ）	機能	共有ライブラリ・ローダーの実行時に共有オブジェクト・ライブラリを検索するディレクトリのリストを示します。この環境変数の詳細は、ld の man ページを参照してください。
	構文	ディレクトリをコロンで区切ったリスト *directory1:directory2:directory3*
	例	/usr/dt/lib:$ORACLE_HOME/lib
PATH	機能	シェルが実行可能プログラムの場所を見つけるために使用します。$ORACLE_HOME/bin ディレクトリが含まれている必要があります。
	構文	ディレクトリをコロンで区切ったリスト *directory1:directory2:directory3*
	例	/bin:/usr/bin:/usr/local/bin:/usr/bin/X11:$ORACLE_HOME/bin:$HOME/bin:. **注意** : 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・プリンタの名前を定義します。
	構文	*string*
	例	docprinter
SHELL	機能	ホスト・コマンドで使用するコマンド・インタプリタを指定します。
	構文	*shell_path*
	許容値	/bin/sh、/bin/csh、/bin/ksh、または他の UNIX コマンド・インタプリタ
	例	/bin/sh

表 1-2 Oracle9*i* で使用する UNIX 環境変数（続き）

環境変数	項目	定義
SHLIB_PATH（HP 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・ライブラリを検索するディレクトリのリストを示します。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 *`directory1:directory2:directory3`*
	例	`/usr/dt/lib:$ORACLE_HOME/lib`
TEMP および TMPDIR	機能	一時ファイル用のデフォルト・ディレクトリを示します。設定すると、一時ファイルを作成するツールは、このディレクトリに一時ファイルを作成します。
	構文	*`directory_path`*
	例	`/u02/oracle/tmp`
XENVIRONMENT	機能	X-Window システムのリソース定義を含むファイルを指定します。詳細は、X-Window のドキュメントを参照してください。

共通の環境設定

この項では、`oraenv` コマンドを使用して共通の UNIX 環境を設定する方法について説明します。C シェルの場合は、`coraenv` コマンドを同じ方法で使用できます。

oraenv スクリプト・ファイル

`oraenv` スクリプトは、インストール時に作成されます。このスクリプトは、`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に必要なユーザー・アカウントを更新するための主な方法
- `oratab` ファイルに指定されている Oracle9*i* データベース間を移動するためのメカニズム

開発システムに対してデータベースの追加および削除を頻繁に行う場合や、同一のシステムにインストールされている複数の異なる Oracle データベース間をユーザーが移動する場合があります。`oraenv` 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、データベース間を移動することができます。

`oraenv` スクリプトは通常、ユーザーの `.profile` ファイルからコールされ、ORACLE_SID、ORACLE_HOME、PATH の各環境変数を設定して `$ORACLE_HOME/bin` ディレクトリを含めるために使用します。データベース間を移動するときに、`oraenv` 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。

> **参照：** 共通の環境設定については、『Oracle9*i* for UNIX Systems インストレーション・ガイド 』を参照してください。

ローカル bin ディレクトリ

oraenv 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込みアクセス権限が必要です。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 に追加してください。インストール後に root.sh スクリプトを実行した場合は、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、oraenv および dbhome スクリプトがそのディレクトリに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 $ORACLE_HOME/bin ディレクトリからローカル bin ディレクトリに oraenv および dbhome スクリプトをコピーできます。

データベース間の移動

あるデータベースまたはデータベース・インスタンスから別のデータベースまたはデータベース・インスタンスに移動する場合は、oraenv スクリプトをコールします。移動先のデータベースの環境変数 ORACLE_SID の値を使用してプロンプトに応答します。環境変数 PATH にローカル bin ディレクトリが含まれていない場合は、oraenv コマンド・ファイルのフルパス名を入力します。たとえば、次のように入力します。

```
$ . /usr/local/bin/oraenv
ORACLE_SID= [default]? sid
```

現行のセッションでの環境変数値の設定およびエクスポート

環境にエクスポートされた環境変数値を表示する場合は、env コマンドを使用します。Bourne シェルおよび Korn シェルを使用すると、エクスポートせずに値を設定できます。

Bourne または Korn シェルの場合、次のように入力します。

```
$ ORACLE_SID=test
$ export ORACLE_SID
```

C シェルの場合、次のように入力します。

```
% setenv ORACLE_SID test
```

この例の *test* は、環境変数 ORACLE_SID の値です。

システム時刻の設定

環境変数 TZ には、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、ファイルのタイムスタンプを決定したり、date コマンドの出力を生成したり、現在の SYSDATE の値を取得する際に使用します。

個人用の TZ 値は、変更しないことをお薦めします。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（Oracle Financials など）が影響を受けます。この問題を回避するために、表の順序を決定する場合は、日付列ではなく順序番号を使用してください。

共有ライブラリをロードするための環境変数 LD_PRELOAD（HP のみ）

共有ライブラリやファンクション・コールを使用する Oracle9*i* アプリケーションを HP で開発している場合、各アプリケーションのユーザーは環境変数 LD_PRELOAD を設定する必要があります。

必要なメモリーが少なくて済むため、多数のプログラムが共有ライブラリを使用します。ほとんどの場合、共有ライブラリを使用するアプリケーションが起動すると、64 ビット版 HP-UX の動的ローダー dld.sl が自動的に起動されます。動的ローダーは実行時に、プログラムにリンクされているすべての共有ライブラリをプロセスに暗黙的に連結します。この中には、HP-UX の Thread Local Storage（TLS）ライブラリも含まれます。またプログラムは、shl_load() HP-UX 関数コールを使用して、次の処理を行うこともできます。

- 64 ビット版 HP-UX の動的ローダーに明示的にアクセスします。
- 実行時に共有ライブラリをプロセスに連結します。
- 共有ライブラリ内に定義されているシンボルのアドレスを計算します。
- 終了時にライブラリを連結解除します。

テストでは、アプリケーションが shl_load() 関数コールを使用して、直接または間接的に HP-UX TLS ライブラリを含む共有ライブラリを連結すると、エラーが発生します。これらのエラーには、HP-UX TLS ライブラリ libpthread.sl および libcl.sl に現在リンクされている Oracle 共有ライブラリ（libclntsh.sl など）が含まれます。

次の例では、prog.c プログラムが shl_load() 関数をコールして、libclntsh.sl ライブラリをロードします。

```
shl_load("Oracle_home_directory/rdbms/lib/libclntsh.sl", BIND_IMMEDIATE |
BIND_VERBOSE | DYNAMIC_PATH | 0L);
```

このプログラムを実行すると、次のエラーが生成されます。

```
/usr/lib/pa20_64/dld.sl: Cannot dlopen load module '/usr/lib/pa20_64/libpthread.1'
```

```
because it contains thread specific data.

/usr/lib/pa20_64/dld.sl: Cannot dlopen load module '/usr/lib/pa20_64/libcl.2'
because it contains thread specific data.
```

新しい環境変数 LD_PRELOAD では、これらのエラーが解決されています。

アプリケーションが `shl_load()` 関数コールを使用して、直接または間接的に HP-UX TLS ライブラリを含む共有ライブラリを連結したときにエラーが発生しないようにするには、次の手順を実行します。

1. 環境変数 LD_PRELOAD の値に、プログラムが使用する TLS ライブラリをコロンまたは空白で区切ったリストが含まれるように設定します。たとえば、次のように入力します。

   ```
   $ export LD_PRELOAD=/usr/lib/pa20_64/libpthread.1:/usr/lib/pa20_64/libcl.2
   ```

   動的ローダーは、環境変数 LD_PRELOAD によって指定されたライブラリをリンク・ラインの最初のライブラリとみなし、アプリケーションの起動時にこれらのライブラリを暗黙的に事前ロードします。

2. 次のコマンドを入力して、プログラムを実行します。*prog* は実行するプログラムの名前です。

   $ *prog*

   前のページに示したエラーが今度は表示されません。

3. メモリーのオーバーヘッドを避けるため、環境変数 LD_PRELOAD の設定を解除します。

   ```
   $ unset LD_PRELOAD
   ```

参照： 環境変数 LD_PRELOAD の詳細は、ご使用の HP-UX システムのドキュメントを参照してください。

実行可能ファイルの再リンク

`$ORACLE_HOME/bin` ディレクトリ内の再リンク用のシェル・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。再リンクは、オペレーティング・システムのパッチの適用後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：** リスナーや Oracle Intelligent Agent など、ORACLE_HOME 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シャットダウンしてください。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すべてシャットダウンしてください。

再リンク・スクリプトを使用すれば、Oracle 製品の実行可能ファイルを手動で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た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場合は、次のコマンドを入力します。*argument* には、表 1-3 に示した値のいずれかを指定します。

```
$ relink argument
```

表 1-3 再リンク・スクリプトの引数

値	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 サーバーの実行可能ファイルのみ
network	Oracle Net クライアント、Oracle Net サーバー、Connection Manager、Oracle Names
client	Oracle Net クライアント、Oracle Trace、PL/SQL
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	Oracle Intelligent Agent **注意：** dbsnmp 実行可能ファイルに正しい権限を付与するには、Oracle Intelligent Agent 実行可能ファイル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	Oracle Internet Directory

システム・グローバル領域

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 カーネル・パラメータ（Tru64 では SHM_MAX）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	システムに搭載された物理メモリーのサイズ。 **参照 :** HP の SHMMAX パラメータの詳細は、B-2 ページの「64 ビットの 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。
Solaris	4 GB - 16 MB。64 ビットシステムでは 4 GB より大きくても構いません。
Tru64	4 GB - 16 MB。 **注意 :** Oracle インスタンスを起動する場合は、SHM_MAX パラメータの値を 16 MB 以上にする必要があります。Oracle8*i* および Oracle9*i* 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「2 GB - 16 MB」に設定する必要があります。
Linux	システムに搭載された物理メモリーの半分のサイズ。

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、Oracle9*i* 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を連結しようとします。SHMSEG カーネル・パラメータ（Tru64 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、次のファイル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

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。値が大きすぎると、コンピュータの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て、パフォーマンスが低下します。

SGA サイズの確認

次のどちらかの方法で、SGA サイズを確認できます。

- 次の SQL*Plus コマンドを入力して、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きに、SGA のサイズを確認します。SGA サイズは、ヘッダー「Total System Global Area」の横に表示されます。

Intimate Shared Memory（Solaris のみ）

Solaris システム上の Oracle9*i* 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 Intimate Shared Memory（ISM）が使用されます。Solaris 2.6 および Solaris 7 上の Oracle9*i* では、デフォルトで 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 では、ページング可能な Dynamic Intimate Shared Memory（DISM）を利用できます。DISM を使用すると、Oracle9*i* 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われる物理メモリーをロックする必要がありません。

Oracle9*i* では、次の基準に基づいて、ISM または DISM のどちらを使用するかが起動時に自動的に決められます。

- Oracle9*i* では、使用しているシステムで DISM が利用可能であり、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。つまり Oracle9*i* では、使用される物理メモリー量だけがロックされます。
- Oracle9*i* では、起動時に共有メモリー・セグメント全体が使用中であるか、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。

Oracle9*i* では、ISM または DISM のどちら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、およびラージ・プール間でメモリーを交換できます。Oracle9*i* 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 を使用するときは、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、Oracle9*i* 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、Oracle9*i* は使用中のメモリーに対してロック操作を明示的に行います。動的な SGA 操作によって共有メモリーが解放されると、Oracle9*i* は解放されたメモリーに対してロック解除操作を明示的に行います。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。

Oracle9*i* は、共有メモリーのロックとロック解除を行うときに、新しいユーティリティ `oradism` を使用します。LOCK_SGA パラメータが TRUE に設定されている場合、Oracle9*i* は起動時に SGA 全体をロックしようとします。この場合、SGA のサイズが変更されたときにロックまたはロック解除は行われません。

Oracle9*i* リリース 2（9.2.0.1.0）を使用している場合、`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。このため、動的 SGA を使用するための設定作業を行う必要はありません。

注意： `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 を使用しているときは、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かどうか、およびスーパーユーザー権限が付与されているかどうかを確認してください。

AIX での共有メモリー

AIX では、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。

AIX で Oracle9*i* を使用している場合は、共有メモリーを確保して、ページングを回避したり、入出力のオーバーヘッドを減らすことができます。これを行うには、LOCK_SGA パラメータを TRUE に設定します。また、AIX 5.1 で Oracle9*i* リリース 2（9.2.0.1.0）を起動している場合、基礎となるハードウェアがラージ・ページ機能をサポートしているときは、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 AIX システムの Oracle9*i* で利用できるようにするには、次のコマンドを入力します。

```
$ vmtune -s 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（パーセント）を設定するには、次のコマンドを入力します。*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。

```
$ vmtune -M percent_of_real_memory
```

-M フラグ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 + 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。

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、AIX の svmon コマンドを使用します。Oracle9*i* 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 ベース・システムでのラージ・ページ機能

AIX Power4 ベース・システムでは、次の方法によってラージ・ページ機能を有効にできます。

1. vmtune コマンドの -g および -L フラグを使用します。-g フラグには、ラージ・ページのサイズを指定します。Power4 プロセッサの場合は、このフラグを 256 MB（-g 268435456）に設定する必要があります。-L フラグには、-g フラグで指定したラージ・ページの数を指定します。このページ数が、Oracle9*i* に予約されます。
2. bosboot コマンドを実行してから、システムをリブートします。

SGA 全体を格納できるだけのラージ・ページを指定することをお薦めします。Oracle9*i* リリース 2（9.2.0.1.0）を起動している場合、その Oracle インスタンスでは LOCK_SGA パラメータが TRUE に設定されているときにラージ・ページを割り当てようとします。SGA のサイズが確保されたメモリーまたはラージ・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。

> **参照：** 確保されたメモリーやラージ・ページを有効にするコマンドおよびチューニングするコマンドについては、AIX のドキュメントを参照してください。

Oracle9*i* のメモリー要件

Oracle9*i* インスタンスで使用する合計メモリーは、SGA のサイズにプログラム・グローバル領域（PGA）のサイズを加えた大きさとほぼ同じです。PGA のサイズを指定するには、PGA_AGGREGATE_TARGET パラメータを設定します。

データベースの制限

表 1-4 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 1-4 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 1-5 に、UNIX 固有の Oracle9*i* ファイル・サイズ制限をバイト単位で示します。

表 1-5 ファイル・サイズの制限

ファイル・タイプ	オペレーティング・システム	最大サイズ
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	16TB 未満
	AIX、HP、Linux、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、Linux、Solaris: 32 ビット（64 ビットのファイルを扱う）	無制限
	AIX、HP、Linux、Solaris: 64 ビット	無制限
制御ファイル	Solaris、HP、Linux	20000 データベース・ブロック

表 1-5 ファイル・サイズの制限（続き）

ファイル・タイプ	オペレーティング・システム	最大サイズ
	AIX	10000 データベース・ブロック
	Tru64	19200 データベース・ブロック

オペレーティング・システムのアカウントとグループ

Oracle9*i* では、次のような特殊なオペレーティング・システム・アカウントとグループが必要です。

- Oracle ソフトウェア所有者アカウント
- OSDBA、OSOPER および ORAINVENTORY グループ

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、インストール後の保守作業では、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。

ORAINVENTORY グループはプライマリ・グループとして所有し、OSDBA グループはセカンダリ・グループとして所有することをお薦めします。

OSDBA、OSOPER および ORAINVENTORY グループ

表 1-6 に、Oracle9*i* に必要な特殊 UNIX グループを示します。

表 1-6 UNIX グループ

グループ	代表的な名前	説明
OSDBA	`dba`	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。 OSDBA グループおよび SYSDBA 権限の詳細は、『Oracle9*i* データベース管理者ガイド』および『Oracle9*i* for UNIX Systems インストレーション・ガイド』を参照してください。

表 1-6 UNIX グループ（続き）

グループ	代表的な名前	説明
OSOPER	`oper`	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。 OSOPER グループおよび SYSOPER 権限の詳細は、『Oracle9*i* データベース管理者ガイド』および『Oracle9*i* for UNIX Systems インストレーション・ガイド』を参照してください。
ORAINVENTORY	`oinstall`	UNIX システムで Oracle ソフトウェアをインストールする全ユーザーは、ORAINVENTORY グループと呼ばれる同一の UNIX グループに所属する必要があります。このグループは、インストール時に Oracle ソフトウェア所有者のプライマリ・グループにする必要があります。インストール後、このグループはシステムにインストールされたすべての Oracle ファイルを所有します。

Oracle9*i* 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、および実行時にそのユーザー ID を変更するプログラム機能が含まれます。

Oracle9*i* 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。

> **参照：** セキュリティ問題の詳細は、『Oracle9*i* データベース管理者ガイド』を参照してください。

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、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薦めします。

- oracle アカウントのプライマリ・グループは、oinstall グループ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。
- DBA 権限を必要とするユーザー・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のは、oracle アカウントのみです。

データベース・ファイルのセキュリティ

データベース・ファイルの適切な権限については、『Oracle9*i* for UNIX Systems インストレーション・ガイド』を参照してください。

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=""
```

参照： 外部認証の詳細は、『Oracle9*i* データベース管理者ガイド』を参照してください。

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することができます。パスワード・ファイルを作成するには、次の手順で行います。

1. Oracle ソフトウェア所有者としてログインします。
2. `$ORACLE_HOME/bin/orapwd` ユーティリティを使用します。このユーティリティの構文は次のとおりです。

   ```
   $ 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。

変数	説明
filename	パスワード情報が書き込まれているファイル名です。ファイル名は orapw*sid* 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て、ユーザーには読めません。このパラメータの入力は必須です。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。
password	このパラメータは、SYS ユーザーのパスワードを設定します。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。
max_users	パスワード・ファイルに格納するエントリの最大数です。

参照： `orapwd` ユーティリティの使用方法については、『Oracle9*i* データベース管理者ガイド』を参照してください。

パスワード管理

Database Configuration Assistant のデータベース作成の完了を示すウィンドウが表示されたら、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 を使用することはできません。

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、SYSTEM、SCOTT 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にログインする場合は、事前にそれらのロックを解除し、パスワードを変更する必要があります。

パスワードを変更するには、Database Configuration Assistant のデータベース作成の完了を示すウィンドウの「パスワード管理」ボタンをクリックします。

また、SQL*Plus を使用して SYSDBA としてデータベースに接続し、次のコマンドを入力して変更することもできます。

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

初期化ファイルのカスタマイズ

デフォルトの初期化ファイル（init*sid*.ora）は、Oracle9*i* ソフトウェアとともに提供されます。Database Creation Assistant は、そのファイルを $ORACLE_BASE/admin/*sid*/pfile ディレクトリに作成します。初期化ファイルのサンプルは、$ORACLE_HOME/dbs ディレクトリ内にあります。

表 1-7 に、UNIX での初期化パラメータのデフォルト値を示します。init*sid*.ora ファイルのこれらの初期化パラメータに別の値を指定しなければ、すべての Oracle9*i* インスタンスでこれらの値が使用されます。init*sid*.ora ファイルには、デフォルトとは異なる値のパラメータのみを指定してください。

システムの現在のパラメータ値を表示するには、SQL*Plus の SHOW PARAMETERS コマンドを使用します。

> **参照：** Tru64 の場合、初期化パラメータの詳細については、D-9 ページの「CLUSTER_INTERCONNECTS 初期化パラメータ（旧 TRU64_IPC_NET）」、『Oracle9*i* データベース・リファレンス』、『Oracle9*i* データベース管理者ガイド』および『Oracle9*i* データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』を参照してください。

表 1-7 初期化パラメータ

パラメータ	デフォルト値	許容値
BACKGROUND_DUMP_DEST	`?/rdbms/log`	有効なディレクトリ名
BITMAP_MERGE_AREA_SIZE	1048576	65536 ～（無制限）
COMMIT_POINT_STRENGTH	1	0 ～ 255
CONTROL_FILES	`?/dbs/cntrl`*`sid`*`.dbf`	有効なファイル名
CREATE_BITMAP_AREA_SIZE	8388608	65536 ～（無制限）
DB_BLOCK_SIZE	2048	2048 ～ 16384 （Linux、Solaris 32 ビット） 2048 ～ 32768 （AIX、HP、Tru64、Solaris 64 ビット）
DB_CACHE_SIZE	8MB	8MB ～（無制限）
DB_FILES	200	1 ～ 2000000
DB_FILE_DIRECT_IO_COUNT	64	0 ～ 1048576/ ブロック・サイズ
DB_FILE_MULTIBLOCK_READ_COUNT	8	1 ～次の値のうちいずれか小さい値 ■ DB_CACHE_SIZE の値を 4 で割った値 ■ 1048576 を DB_BLOCK_SIZE の値で割った値
HASH_AREA_SIZE	SORT_AREA_SIZE の値に 2 を掛けた値	0 ～（無制限）
HASH_MULTIBLOCK_IO_COUNT	0（セルフ・チューニング）	1 ～次の値のうちの最小値 ■ 127 ■ DB_CACHE_SIZE の値を 4 で割った値 ■ 1048576 を DB_BLOCK_SIZE の値で割った値
JAVA_POOL_SIZE	24MB	1000000 ～ 1000000000
LOCK_SGA	FALSE	TRUE、FALSE
LOG_ARCHIVE_DEST	NULL	有効なディレクトリ名
LOG_ARCHIVE_FORMAT	`%t_%s.dbf`	有効なファイル名

表 1-7 初期化パラメータ（続き）

パラメータ	デフォルト値	許容値
LOG_BUFFER	512KB（または 128KB と CPU_COUNT の値の積のうち、いずれか大きな値）	66560 ～（無制限）
LOG_CHECKPOINT_INTERVAL	0	0 ～（無制限）
MAX_DISPATCHERS	5	1 ～オペレーティング・システムでオープンできるプロセスの最大数
MAX_SHARED_SERVERS	SHARED_SERVERS の値が 21 以上である場合は、2 と SHARED_SERVERS の値の積、それ以外の場合は 20	SHARED_SERVERS の値と PROCESSES の値の間
SHARED_SERVERS	DISPATCHERS が指定されている場合は 1、それ以外の場合は 0	1 と PROCESSES 値の間
NLS_LANGUAGE	AMERICAN	有効な言語名
NLS_TERRITORY	AMERICA	有効な地域名
OBJECT_CACHE_MAX_SIZE_PERCENT	10	0 ～（無制限）
OBJECT_CACHE_OPTIMAL_SIZE	100KB	10KB ～（無制限）
OPEN_CURSORS	50	1 ～（無制限）
OS_AUTHENT_PREFIX	`ops$`	任意の文字列
PGA_AGGREGATE_TARGET	0（自動メモリー管理が設定されていない）	10 MB ～ 4 TB（設定されている場合）
PROCESSES	PARALLEL_AUTOMATIC_TUNING でない場合は 30	6 ～（無制限）
SHARED_POOL_SIZE	64 ビット・システムでは 64MB、32 ビット・システムでは 8MB	4194304 ～（無制限）
SORT_AREA_SIZE	65536	0 ～（無制限）

Oracle HTTP Server

Oracle HTTP Server を管理するには、そのサーバーが動作しているローカル・システムへのアクセス権限を持っている必要があり、場合によっては root でのアクセス権限が必要です。

Oracle HTTP Server は、インストール後にデフォルト・ポート 7777 で自動的に起動します。サーバーが動作しているかどうかを確認するには、次のコマンドを入力します。

```
$ ps -elf | grep httpd
```

注意： この製品には、Apache Software Foundation によって開発されたソフトウェアが含まれています。

Oracle HTTP Server の起動および停止

構成を変更した場合は、サーバーを再起動する必要があります。SSL に対応しているサーバーを起動するには、root ユーザーとしてログインする必要があります。

サーバーを停止するには、次のコマンドを入力します。

```
$ cd $ORACLE_HOME/Apache/Apache/bin
$ su root
# ./apachectl stop
```

サーバーを再起動するには、次のコマンドを入力します。

```
$ cd $ORACLE_HOME/Apache/Apache/bin
$ su root
# ./apachectl {start|startssl}
```

SSL 非対応サーバーを起動するには、start フラグを使用します。SSL 対応サーバーを起動するには、startssl フラグを使用します。SSL 非対応サーバーのデフォルト・ポートは 80 です。SSL 対応サーバーのデフォルト・ポートは 443 です。

注意： SSL 非対応サーバーを起動するときに、ポート 80 が利用できない場合、Oracle HTTP Server は次に利用できるポート（7777 ～ 7877）をデフォルト・ポートとして使用します。

SSL 対応サーバーを起動するときに、ポート 443 が利用できない場合、Oracle HTTP Server は次に利用できるポート（4443 ～ 4543）をデフォルト・ポートとして使用します。

デフォルトの初期静的ページへのアクセス

デフォルトの初期静的ページには、オンライン・マニュアルおよび各コンポーネントのデモへのリンクが含まれます。初期静的ページにアクセスするには、インターネット・ブラウザを使用して次のいずれかの URL を表示します。

- SSL に対応していないサーバーの場合

  http://*ServerName*:7777/

- SSL に対応しているサーバーの場合

  http://*ServerName*:80/

- SSL に対応しているサーバー（安全性が高く、HTTPS を使用）の場合

  https://*ServerName*:4443/

この例の *ServerName* は、HTTP Server の `httpd.conf` 構成ファイルに設定されています。構成ファイル内で使用されている値を調べるには、次のコマンドを入力します。

```
$ grep ServerName $ORACLE_HOME/Apache/Apache/conf/httpd.conf
```

Oracle HTTP Server のログ・ファイル

多くのログ・ファイルがサーバーによって生成されます。定期的にログ・ファイルをチェックして、サーバーが正常に動作していることを確認してください。デフォルトでは、構成ファイルのエラー・ログ・レベルは、`warn` に設定されています。デフォルトのエラー・レベルを変更するには、該当する構成ファイルを編集し、サーバーを再起動します。

サーバーによって、次のログ・ファイルが生成されます。

```
$ORACLE_HOME/Apache/Apache/logs/access_log
$ORACLE_HOME/Apache/Apache/logs/error_log
$ORACLE_HOME/Apache/Apache/logs/ssl_engine_log
$ORACLE_HOME/Apache/Jserv/logs/jserv.log
$ORACLE_HOME/Apache/Jserv/logs/mod_jserv.log
```

デモ・ファイル

この項では、Oracle9*i* とともにインストールされる 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。

SQL*Loader のデモ

次の SQL*Loader デモ・ファイルは、Oracle9*i* とともに `$ORACLE_HOME/rdbms/demo` ディレクトリ内にあります。これらのデモ・プログラムを昇順に実行します。

`ulcase1`	`ulcase4`	`ulcase6`
`ulcase3`	`ulcase5`	`ulcase7`

デモの作成と実行

デモを実行するには、ユーザー SCOTT/TIGER としてログインします。SCOTT/TIGER スキーマがシステムに存在しない場合は、`$ORACLE_HOME/rdbms/admin/utlsampl.sql` スクリプトを実行してそれを作成します。

次の要件を確認してください。

- ユーザー SCOTT/TIGER に、CONNECT および RESOURCE 権限が付与されていること
- EMP および DEPT の各表が存在すること

次の手順での *n* は、前の項で示したデモ番号を表しています。デモを作成および実行するには、次の手順で行います。

1. 実行するデモに対応する `ulcase`*n*`.sql` スクリプトを実行します。

   ```
   $ sqlplus SCOTT/TIGER @ulcasen.sql
   ```

2. デモ・データをオブジェクトにロードします。

   ```
   $ sqlplus SCOTT/TIGER @ulcasen.sql
   ```

次に、`ulcase6`、`ulcase7` の各デモについて詳しく説明します。

- `ulcase6` デモの場合、`ulcase6.sql` スクリプトを実行し、次のコマンドを入力します。

  ```
  $ sqlldr SCOTT/TIGER ulcase6 DIRECT=true
  ```

- `ulcase7` デモの場合、`ulcase7s.sql` スクリプトを実行し、次のコマンドを入力します。

  ```
  $ sqlldr SCOTT/TIGER ulcase7
  ```

デモを実行した後、`ulcase7e.sql` を実行して、このデモで使用されたトリガーおよびパッケージを削除します。

PL/SQL デモ

PL/SQL には、ロードできるデモ・プログラムがいくつかあります。デモ・プログラムを操作するには、Oracle9*i* データベースをオープンし、マウントする必要があります。

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。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には、次の手順で行います。

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. 次のコマンドを入力して、データベース・オブジェクトを作成し、サンプル・データをロードします。

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

注意： デモの作成は、必要な権限を持つ Oracle ユーザーが行ってください。デモの実行は、同一の Oracle ユーザーが行ってください。

PL/SQL カーネル・デモ

次の PL/SQL カーネル・デモを利用できます。

`examp1.sql`	`examp5.sql`	`examp11.sql`	`sample1.sql`
`examp2.sql`	`examp6.sql`	`examp12.sql`	`sample2.sql`
`examp3.sql`	`examp7.sql`	`examp13.sql`	`sample3.sql`
`examp4.sql`	`examp8.sql`	`examp14.sql`	`sample4.sql`
`extproc.sql`			

PL/SQL カーネル・デモ `examp`*n*`.sql` または `sample`*n*`.sql` をコンパイルして実行するには、次の手順で行います。

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入力してデモを実行します。*demoname*.sql はデモの名前です。

```
SQL> @demoname
```

extproc.sql デモを実行するには、次の手順で行います。

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

2. 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。

```
SID_LIST_LISTENER =
  (SID_LIST =
     (SID_DESC=
        (SID_NAME=PLSExtProc)
        (ORACLE_HOME=/u01/app/oracle/product/9.2.0.1.0)

(ENVS=EXTPROC_DLLS=/u01/app/oracle/product/9.2.0.1.0/plsql/demo/extproc.so,
LD_LIBRARY_PATH=/u01/app/oracle/product/9.2.0.1.0/plsql/demo)
        (PROGRAM=extproc)
      )
   )
```

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。

---

3. 次のコマンドを入力して、extproc.so 共有オブジェクトを作成し、必要なデータベース・オブジェクトを構築し、サンプル・データをロードします。

```
$ 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・データがロード済である場合は、次のコマンドを入力します。

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

4. SQL*Plus から、次のコマンドを入力します。

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

5. デモを実行するには、次のコマンドを入力します。

```
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

これらの PL/SQL プリコンパイラ・デモをすべて作成するには、次のコマンドを入力します。

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

デモを 1 つだけ作成する場合は、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、次のように入力します。

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

examp9 デモを実行するには、次のコマンドを入力します。

```
$ ./examp9
```

SQL*Loader の管理

SQL*Loader は、データベース管理者と Oracle9*i* ユーザーによって使用されます。SQL*Loader は、標準のオペレーティング・システム ファイルのデータを Oracle データベース表にロードします。

> **参照：** SQL*Loader の使用方法については、『Oracle9*i* データベース・ユーティリティ』を参照してください。

SQL*Loader の制御ファイルには、次のファイル処理オプションが含まれています。デフォルトは `str` で、引数はありません。

```
[ "str" | "fix n" | "var n" ]
```

次の表は、これらの処理オプションの説明です。

文字列	説明
`"str"`	レコード・ストリームを示します。各レコードは、改行文字で終わり、一度に 1 レコードずつ読み取られます（デフォルト）。
`"fix n"`	ファイルが *n* バイトの固定長レコードで構成されていることを示します。*n* は整数です。
`"var n"`	ファイルが可変長レコードで構成されていることを示します。各レコードの長さは、最初の *n* 文字で指定されます。*n* の値を指定しない場合、SQL*Loader は値を 5 と見なします。

ファイル処理オプションを選択しない場合、デフォルトにより、情報はレコード・ストリーム（`str`）として処理されます。`"fix"` オプションでは、レコード終了記号がスキャンされないため、デフォルトの `"str"` オプションに比べて高いパフォーマンスが得られます。

固定長レコードの改行文字

`"fix"` オプションを使用して、各レコードが改行文字で終了する固定長レコードを含むファイルを読み取る場合は、改行文字の長さ（1 文字）を含めたレコード長を SQL*Loader に指定します。

たとえば、次のファイルを読み取るには、`"fix 3"` ではなく `"fix 4"` を指定して、追加の改行文字を含めます。

```
AAA<cr>
BBB<cr>
CCC<cr>
```

固定長レコードのファイルの最後のレコードを改行文字で終了しない場合は、その他のレコードも改行文字で終了しないでください。同様に、最後のレコードを改行文字で終了する場合は、すべてのレコードを改行文字で終了してください。

注意： vi などの特定のテキスト・エディタを使用すると、ファイルの最後のレコードは自動的に改行文字で終了します。この場合、ファイル内のその他のレコードが改行文字で終了していないと、不整合が発生します。

改行文字の削除

改行文字をロードするのではなく固定長レコードから削除するには、制御ファイルで position(*x*:*y*) 関数を使用します。

たとえば、次の行を制御ファイルに入力して、4 桁目の改行文字を削除します。

```
load data
infile xyz.dat "fix 4"
into table abc
( dept position(01:03) char )
```

これらの行を使用すると、SQL*Loader によって各固定長レコードの 4 桁目にある改行文字が削除されます。

2

Oracle9*i* のチューニング

Oracle9*i* アプリケーションの複雑さが増すほど、システムのチューニングを頻繁に行って、パフォーマンスの最適化を図り、データのボトルネックの発生を防ぐ必要があります。この章では、Oracle9*i* のインストール環境を設定して、パフォーマンスを最適化する方法について説明します。この章では、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
- UNIX カーネル・パラメータのチューニング
- 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 / ボリュームの使用
- トレース・ファイルおよびアラート・ファイルの使用

参照： システムのチューニングについては、次のマニュアルおよび付録を参照してください。

- 『Oracle9*i* データベース・パフォーマンス・プランニング』
- 『Oracle9*i* データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』
- 『Oracle9*i* Real Application Clusters 概要』
- 『Oracle9*i* Real Application Clusters 管理』
- 付録 A「Oracle9i for AIX-Based Systems の チューニング」
- 付録 B「Oracle9i for HP 9000 Series HP-UX の チューニング」
- 付録 C「Oracle9i for Linux Intel のチューニング」
- 付録 D「Oracle9i for Compaq Tru64 UNIX の チューニング」

チューニングの重要性

Oracle9*i* 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。この章の説明は単一ノード・コンピュータ・システムを対象に書かれていますが、パフォーマンスのチューニングに関するヒントの多くは、Oracle9*i* Real Application Clusters や、Oracle9*i* で利用できる機能を使用する場合にも適用できます。

システムのチューニングを始める前に、2-3 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、通常の動作を監視します。

パフォーマンスのボトルネックの種類

パフォーマンスのボトルネックは、次のことが原因でよく発生します。

- メモリーの競合

  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。メモリーの競合が発生すると、メモリーとディスクとの間でプロセスのページングやスワッピングが行われます。

- ディスクの I/O 競合

  ディスクの I/O 競合は、メモリー管理が不適切である場合、ディスク間の表領域とファイルの配分が不適切である場合、あるいはそれらが組み合された場合に発生します。

- CPU の競合

  通常、CPU リソースは UNIX カーネルによって効果的に割り当てられます。ただし、多くのプロセスが CPU サイクルをめぐって競り合った場合、競合が発生することがあ

ります。マルチプロセッサ環境に Oracle9*i* をインストールした場合は、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。

- Oracle リソースの競合

  ロックやラッチなどの競合も、Oracle リソースでよく発生します。

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めるのに役立つオペレーティング・システム・ツールがいくつか利用できます。これらのツールは、Oracle プロセスの統計だけでなく、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・スイッチング、I/O についての統計情報も提供します。

共通のツール

次の各項では、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、lsps

> **参照：** 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 UNIX の man ページを参照してください。

vmstat

プロセス、仮想メモリー、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、コマンドで切り換えます。CPU アクティビティのサマリーを 5 秒間隔で 8 回表示する場合は、次のどちらかのコマンドを入力します。

- HP および Solaris

```
$ vmstat -S 5 8
```

- AIX、Linux および Tru64

```
$ vmstat 5 8
```

次の例は、Solaris でこのコマンドを実行した場合のサンプル出力です。

```
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  0   87   23   21  0  0 100
 0 0 0  85356  8372   0    0  0  0  0  0  0  0  0  0  0   48   41   23  0  0 100
 0 0 0  85356  8372   0    0  0  0  0  0  0  0  0  0  0   44   20   18  0  0 100
 0 0 0  85356  8372   0    0  0  0  0  0  0  0  0  0  0   51   71   24  0  0 100
```

procs 列の下にある w 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 以外の場合は、スワッピングが発生し、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 秒あたりのスワップ・インおよびスワップ・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 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 秒あたりのページインおよび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。使用できるメモリーが十分備わっているシステムでも、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。ご使用のプラットフォームでの出力の解釈については、man ページを参照してください。

---

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 コマンドを使用します。表示内容はコマンドで切り換えます。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
```

次の例は、Solaris でこのコマンドを実行した場合のサンプル出力です。

```
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
```

注意： Tru64 システムでは、`sar` コマンドは、UNIX SVID2 互換サブセットの OSFSVID2400 で利用できます。

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、コマンドで切り換えます。`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際に役立ちます。

次のコマンドは、端末およびディスク・アクティビティのサマリーを 5 秒間隔で 5 回表示します。

```
$ iostat 5 5
```

次の例は、Solaris でこのコマンドを実行した場合のサンプル出力です。

```
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   31    0   0   18    3   0   42   0  0  0 99
   0   16   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  0 100
   0   16   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、つまり 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 または 0 に近い値であることが理想的です。

swap、swapinfo、swapon、lsps

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 lsps の各コマンドを使用します。スワップ領域が不足すると、プロセスが生成できなくなり、応答時間が長くなったり、応答が停止したりすることがあります。次の表は、プラットフォームごとのコマンドの一覧です。

プラットフォーム	コマンド
AIX	`lsps -a`
HP	`swapinfo -m`
Linux	`swapon -s`
Solaris	`swap -l`
Tru64	`swapon -s`

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
```

Linux のツール

Linux システムでは、free 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。スワップ領域が不足すると、プロセスが生成できなくなり、応答時間が長くなったり、応答が停止したりすることがあります。

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。2 行目以降の各行は、アクティビティの間隔を表示します。特に指定しない限り、すべての値は割合（1 秒あたりのイベント数）です。引数は、統計および反復回数の時間間隔を示します。次の例は、mpstat コマンド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
```

AIX のツール

次の項では、AIX システムで利用できるツールについて説明します。

> **参照：** これらのツールの詳細は、AIX 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。

AIX 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 は、実行するジョブを中心に、様々なツールにナビゲートするためのツールです。

Base Operating System ツール

AIX の Base Operating System（BOS）は、UNIX システムに従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・ツールで構成されます。次の表は、最も重要な BOS ツールの一覧です。

ツール	説明
`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）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、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。
`vmtune`	仮想メモリー・マネージャや他の AIX コンポーネントの操作パラメータを変更します。

AIX 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 / 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て表示します。PTX Agent は、xmserd ユーティリティ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、次の監視 / チューニング・ツールがあります。

ツール	説明
fdpr	特定の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。
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	トレース機能を使用して、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を終了します。

参照： PTX の詳細は、『Performance Toolbox for AIX; Guide and Reference 1.2 and 2 』を参照してください。一部のツールの構文は、『AIX Performance Tuning Guide Version 3.2 and 4』および『AIX5L Performance Management Guide』を参照してください。

HP のツール

次の項では、HP システムで利用できるツールについて説明します。

パフォーマンス・チューニング・ツール

次の表は、HP のパフォーマンス・チューニング・ツールの一覧です。

参照： PTX の詳細は、HP-UX 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。

ツール	説明
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、特定の関数にコールします。
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	プロセッサごとの統計を報告します。
nettl	ロギングとトレースによって、ネットワーク上のイベントまたはパケットを獲得します。
prof	いくつかの C プログラムからなる実行プロファイルを作成し、プログラムのパフォーマンス統計を表示して、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。

HP パフォーマンス分析ツール

次の HP-UX パフォーマンス分析ツールを利用できます。

- GlancePlus/UX
- HP PAK

GlancePlus/UX

この HP-UX ユーティリティは、システム・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することもできます。

HP PAK

HP Programmer's Analysis Kit（HP PAK）は、Puma および Thread Trace Visualizer（TTV）という 2 つのツールから構成されます。

- Puma 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。収集した統計は、グラフィカルに表示および分析します。
- TTV は、インストルメント・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。

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、まず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調べます。システムのメモリー使用量を調べ終わったら、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てバッファ・キャッシュをチューニングすると、Oracle9*i* のパフォーマンスが大幅に向上する可能性があります。各システムの Oracle9*i* 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 の優先順位によって異なります。

十分なスワップ領域の割当て

スワッピングは、UNIX 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` または `vmstat` コマンドを入力します。`sar` または `vmstat` 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、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、使用可能なメモリーを増やします。
- UNIX 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）。

スワップ領域の使用量を調べるには、次のいずれかのコマンドを入力します。

プラットフォーム	コマンド
AIX	`lsps -a`
HP	`swapinfo -m`

プラットフォーム	コマンド
Linux	`swapon -s`
Solaris	`swap -l`
Tru64	`swapon -s`

スワップ領域をシステムに追加するには、次のいずれかのコマンドを入力します。

プラットフォーム	コマンド
AIX	`chps` または `mkps`
HP	`swapon`
Linux	`swapon -a`
Solaris	`swap -a`
Tru64	`swapon -a`

スワップ領域は、システムの物理メモリーの 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。

> **参照：** これらのコマンドの詳細は、ご使用のオペレーティング・システムのドキュメントを参照してください。

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。少しくらい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` または `sar` コマンドを使用します。プラットフォームの監視結果の解釈については、該当する 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。Solaris では、これらのコマンドの出力から得られる次の列が重要です。

列	説明
`vflt/s`	アドレス変換ページ・フォルトの数を示します。アドレス変換ページ・フォルトは、プロセスが、メモリー中にない有効ページを参照したときに発生します。
`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 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。
- SGA で使用するメモリーを少なく設定します。

Oracle ブロック・サイズの調整

UNIX システムでは、オペレーティング・システムのブロック全体をディスクから読み取ります。データベースのブロック・サイズが UNIX ファイル・システムのバッファ・サイズよりも小さい場合は、I/O 帯域幅の効率が悪くなります。Oracle データベースのブロック・サイズがオペレーティング・システムのブロック・サイズの倍数になるように調整すると、パフォーマンスを最高 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。このブロック・サイズを変更するには、データベースを再作成します。

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 PARAMETERS コマンドを入力します。

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 を均等に分散させて、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させてください。

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・システムから選択することができます。ファイル・システムの特性はそれぞれ異なり、それらがデータにアクセスする方法は、データベース・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・システムとそれらが利用できるプラットフォームを示します。

ファイル・システム	プラットフォーム	説明
S5	AIX、HP、Solaris	UNIX System V ファイル・システム
UFS	AIX、HP、Solaris、Tru64	Unified ファイル・システム（BSD UNIX から派生）
VXFS	AIX、Solaris	Veritas ファイル・システム
Raw	AIX、HP、Linux、Solaris、Tru64	RAW デバイス / ボリューム（ファイル・システムなし）
EXT2	Linux	Linux 用拡張ファイル・システム
AdvFS	Tru64	Advanced ファイル・システム
CFS	Tru64	クラスタ・ファイル・システム
JFS	AIX	ジャーナル・ファイル・システム
JFS2	AIX 5.1	ジャーナル・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限りません。たとえば、UFS 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 0 〜 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配分します。
- ログ・ファイルとデータベース・ファイルを切り離します。

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。

表 2-1 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。

表 2-1 sar -b 出力の列

列	説明
bread/s、bwrit/s	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）
pread/s、pwrit/s	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 を処理する 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。

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。そうしないと、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 以上である場合、システムは I/O バウンドになります。

UNIX カーネル・パラメータのチューニング

Oracle ユーザーと Oracle プロセスの優先順位をすべて同じにすることで、パフォーマンスを向上させることができます。UNIX カーネルでは通常、事前に物理メモリーを割り当てるため、Oracle プロセスなど他のプロセスで使用できるメモリーが少なくなります。

従来は、NBUF、NFILE、NOFILES などのカーネル・パラメータを使用してカーネル・サイズを調整していました。ただし、ほとんどの UNIX は、それらのパラメータを実行時に動的に調整します。それらのパラメータが UNIX 構成ファイルに含まれている場合も例外ではありません。

注意： 必ず UNIX カーネルのバックアップを取っておいてください。バックアップ・コピーの作成方法については、ご使用のオペレーティング・システムのドキュメントを参照してください。

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、Oracle9*i* バッファ・キャッシュのサイズを調整し、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・キャッシュのサイズも調整します。

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に転送される、メモリー内のデータ・ブロックが保持されます。

Oracle9*i* バッファ・キャッシュは、Oracle データベース・バッファを格納するためのメモリー内の領域です。Oracle9*i* 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・キャッシュを使用する必要はありません。

RAW デバイスを使用する場合は、Oracle9*i* 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。`sar` コマンドの詳細は、UNIX の man ページを参照してください。

RAW デバイス / ボリュームの使用

次の項では、RAW デバイス / ボリュームの使用方法について説明します。

注意： RAW デバイス / ボリュームのチューニングの詳細は、次の付録を参照してください。

- 付録 A「Oracle9i for AIX-Based Systems の チューニング」
- 付録 C「Oracle9i for Linux Intel のチューニング」
- 付録 D「Oracle9i for Compaq Tru64 UNIX の チューニング」

RAW デバイス / ボリュームを使用する場合のガイドライン

UNIX で RAW デバイス / ボリュームを使用する場合、次のようなデメリットがあります。

- RAW デバイス / ボリュームでは、ファイル・サイズの書込み制限に関する問題を解決できない場合があります。

注意： 現在のファイル・サイズの制限を表示するには、次のコマンドを入力します。

```
$ ulimit -a
```

- 小規模クライアント・システムでは、十分な大きさの RAW デバイス / ボリュームのパーティションを使用できない場合があります。
- 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すと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け入れられるサイズのセクションがないことがよくあります。このため、RAW デバイス / ボリューム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 / ボリュームは、ファイル・システムに格納されているデータ・ファイルに比べて、管理が難しい場合があります。

この項で説明した要因以外に、RAW デバイス / ボリュームを使用するかどうかを決定する場合、次の問題を考慮する必要があります。

- Oracle9*i* Real Application Clusters のインストール

  Oracle9*i* Real Application Clusters の各インスタンスでは、個別のログ・ファイルが使用されます。そのため、表領域および制御ファイルに必要なパーティションの他に、各インスタンスのログ・ファイル用に、3 つ以上のパーティションが必要です。これらのファイルはすべて、クラスタの全ノードで共有できるディスク上にある必要があります。

- RAW ディスク・パーティションの可用性

  使用しているサイトに、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 デバイス / ボリュームを使用してください。ディスク領域が問題になっており、RAW ディスク・パーティションがフォーマット済である場合は、データ・ファイルのサイズをパーティションのサイズにできるだけあわせて、無駄な領域をなくします。

  また、多数のディスク上の領域を一部だけ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する
  – データを複数のディスクにまたがって不連続に格納したり、レプリケートしたり、動的に拡張したりできるようにする

- 動的パフォーマンスのチューニング

  データベースがオンラインで動作しているときは、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すれば、ディスク・パフォーマンスを最適化できます。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 Graphical User Interface 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もあります。

  Oracle9*i* Real Application Clusters では、単一の UNIX システムに対応付けられているドライブだけでなく、UNIX クラスタに接続された複数のコンピュータ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、Oracle9*i* Real Application Clusters に関連付けられている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。

RAW デバイスの設定

RAW デバイスを作成する場合、次のことに注意してください。

- ボリュームの作成時に、所有者が `oracle` であり、グループが `oinstall` であることを確認します。
- RAW パーティションに作成した 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より（Oracle ブロック・サイズ× 2 以上）小さくなければなりません。

> **参照：** RAW デバイスの作成方法については、ご使用のオペレーティング・システムのドキュメントを参照してください。また、Tru64 システムで RAW デバイスを作成する方法については、D-17 ページの「RAW デバイスの設定」を参照してください。

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、Oracle9*i* 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。

トレース・ファイル

各サーバー・プロセスとバックグラウンド・プロセスでは、関連するトレース・ファイルに情報を書き込むことができます。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・ファイル名の形式は、*`processname_unixpid_sid`*`.trc` です。各項目の説明は次のとおりです。

- *`processname`* は、トレース・ファイルを作成した Oracle9*i* 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です（`pmon`、`dbwr`、`ora`、`reco` など）。
- *`unixpid`* は、UNIX プロセス ID 番号です。
- *`sid`* は、インスタンスのシステム識別子です。

トレース・ファイル名のサンプルは、`$ORACLE_BASE/admin/TEST/bdump/lgwr_1237_TEST.trc` です。

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・ディレクトリ `$ORACLE_HOME/rdbms/log` が使用されます。

サーバー・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。MAX_DUMP_FILE 初期化パラメータを 5000 以上に設定して、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。

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グローバル・データベースに影響を及ぼす情報はすべて、このファイルに記録されます。このファイルは、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ているデータベースに関連付けられます。この初期化パラメータを設定しない場合は、デフォルト・ディレクトリ $ORACLE_HOME/rdbms/log が使用されます。

3

SQL*Plus および *i*SQL*Plus の管理

この章では、Oracle9*i* で SQL*Plus および *i*SQL*Plus を使用および管理する方法について説明します。この章では、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
- iSQL*Plus の管理
- コマンドライン SQL*Plus の使用
- SQL*Plus の制限事項

> **参照：** *i*SQL*Plus および 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。

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。この項の例で、疑問符（?）が指定されている箇所では、環境変数 ORACLE_HOME の値が使用されます。

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 をインストールしたときにそのファイルが上書きされます。SQL*Plus を削除すると、既存のサイト・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 を使って指定したディレクトリを検索して、このファイルを見つけます。この環境変数には、ディレクトリ・リストをコロンで区切って設定します。SQL*Plus は、これらのディレクトリを指定された順番に検索して、login.sql ファイルを見つけます。

login.sql ファイルに設定されているオプションによって、glogin.sql ファイルに設定されているオプションが上書きされます。

参照： ユーザー・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。

PRODUCT_USER_PROFILE 表の使用

Oracle9*i* 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、初期データベースがインストールされるインストール・タイプを選択すると自動的に作成されます。次の環境では、初期データベースのインストールも、デモンストレーション表の作成も行われません。

- 「ソフトウェアのみ」データベース構成オプションを選択している場合
- カスタム・インストール時に「Example Schemas」オプションを選択していない場合

> **参照：** インストール・オプションについては、『Oracle9*i* for UNIX Systems インストレーション・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 スキーマに入っている `$ORACLE_HOME/sqlplus/admin/pupbld.sql` スクリプトを実行します。

```
$ORACLE_HOME/sqlplus/admin/pupbld.sql
```

たとえば、次のコマンドを入力します。*`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、シェル・スクリプト `$ORACLE_HOME/bin/pupbld` 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。プロンプトを表示しないで `pupbld` スクリプトを実行する必要がある場合は、環境変数 SYSTEM_PASS に SYSTEM ユーザー名およびパスワードを設定してください。たとえば、次のコマンドを入力します。*`SYSTEM_PASSWORD`* は SYSTEM ユーザーのパスワードです。

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

デモンストレーション表の使用

Oracle9*i* には、テスト用のデモンストレーション表が用意されています。テスト用のデモンストレーション表は、初期データベースがインストールされるインストール・タイプを選択すると自動的に作成されます。次の環境では、初期データベースのインストールも、デモンストレーション表の作成も行われません。

- 「ソフトウェアのみ」データベース構成オプションを選択している場合
- カスタム・インストール時に「Example Schemas」オプションを選択していない場合

> **参照：** デモンストレーション表については、『Oracle9*i* サンプル・スキーマ』を参照してください。また、インストール・オプションについては、『Oracle9*i* for UNIX Systems インストレーション・ガイド』を参照してください。

EMP および DEPT 表の使用

この項では、デモンストレーション表 EMP および DEPT を手動で作成および削除する方法について説明します。

デモンストレーション表の手動作成

デモンストレーション表 EMP および DEPT を作成するには、SQL スクリプト `$ORACLE_HOME/sqlplus/demo/demobld.sql` を使用します。SQL*Plus では、任意のユーザー名を使用して `demobld.sql` スクリプトを実行し、スキーマにデモンストレーション表を作成できます。たとえば、次のように入力します。

```
$ sqlplus SCOTT/TIGER
SQL> @?/sqlplus/demo/demobld.sql
```

また、次のように、シェル・スクリプト `$ORACLE_HOME/bin/demobld` を使用して、`demobld.sql` スクリプトを実行することもできます。

```
$ demobld SCOTT TIGER
```

デモンストレーション表の削除

デモンストレーション表 EMP および DEPT を削除するには、`$ORACLE_HOME/sqlplus/demo/demodrop.sql` スクリプトを使用します。SQL*Plus では、任意のユーザー名を使用して、ユーザーのスキーマからデモンストレーション表を削除できます。たとえば、次のように入力します。

```
$ sqlplus SCOTT/TIGER
SQL> @?/sqlplus/demo/demodrop.sql
```

また、次のように、シェル・スクリプト `$ORACLE_HOME/bin/demodrop` を使用して、`demodrop.sql` スクリプトを実行することもできます。

```
$ demodrop SCOTT TIGER
```

SQL*Plus のコマンドライン・ヘルプ

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および削除方法について説明します。

> **参照：** 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。

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。

- 初期データベースがインストールされるインストールを実行します。

  インストールの実行時に、事前に作成したデータ・ファイルを含む初期データベースをコピーすると、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。

- シェル・スクリプト `$ORACLE_HOME/bin/helpins` を使用して、手動でコマンドライン・ヘルプを SYSTEM スキーマにインストールします。

  `helpins` スクリプトは、SYSTEM ユーザーのパスワードを要求するプロンプトを表示します。プロンプトを表示しないでこのスクリプトを実行する必要がある場合は、環境変数 SYSTEM_PASS に、SYSTEM ユーザー名およびパスワードを設定してください。たとえば、次のコマンドを入力します。*`SYSTEM_PASSWORD`* は SYSTEM ユーザーのパスワードです。

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

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

  たとえば、次のコマンドを入力します。*`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・ヘルプ表を削除します。

SQL*Plus のコマンドライン・ヘルプの削除

手動で SQL*Plus のコマンドライン・ヘルプ表を SYSTEM スキーマから削除するには、`$ORACLE_HOME/sqlplus/admin/help/helpdrop.sql` スクリプトを実行します。たとえば、次のコマンドを入力します。*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
```

*i*SQL*Plus の管理

*i*SQL*Plus は、次の 3 層モデルで SQL*Plus 処理機構を使用するブラウザ・ベースのインタフェースです。

層	説明
クライアント	*i*SQL*Plus ユーザー・インタフェース（通常は Web ブラウザ）
中間	*i*SQL*Plus サーバー、Oracle Net、および Oracle HTTP Server
データベース	Oracle9*i* データベース

*i*SQL*Plus の無効化および再有効化

Oracle9*i* をインストールすると、デフォルトで *i*SQL*Plus が有効になります。この項では、*i*SQL*Plus を無効化および再有効化する方法について説明します。

*i*SQL*Plus を無効にするには、次の手順で行います。

1. Oracle HTTP Server を実行しているシステムに Oracle ソフトウェア所有者（`oracle`）としてログインします。
2. ディレクトリを Oracle HTTP Server の構成ディレクトリに変更します。

   ```
   $ cd $ORACLE_HOME/Apache/Apache/conf
   ```

3. 任意のテキスト・エディタで `oracle_apache.conf` 構成ファイルを開きます。
4. 次のように、`isqlplus.conf` 構成ファイルを含むディレクティブの前にコメント文字（#）を挿入します。

   ```
   #include "ORACLE_HOME/sqlplus/admin/isqlplus.conf"
   ```

   この例の *ORACLE_HOME* は、Oracle ホーム・ディレクトリのパスです。
5. ファイルを保存して、テキスト・エディタを終了します。
6. Oracle HTTP Server の構成ファイルを解析することによって、変更内容を確認します。

```
$ $ORACLE_HOME/Apache/Apache/bin/apachectl configtest
```

必要に応じて、このコマンドによって表示されたエラーを修正します。

7. Oracle HTTP Server を再起動します。

```
$ $ORACLE_HOME/Apache/Apache/bin/apachectl restart
```

*i*SQL*Plus を再度有効にするには、前述の手順を繰り返しますが、手順 4 でコメント文字（#）を削除します。

*i*SQL*Plus 構成ファイルの編集

`$ORACLE_HOME/sqlplus/admin/isqlplus.conf` 構成ファイルの次の *i*SQL*Plus サーバーのパラメータ値を変更できます。

パラメータ	説明
`iSQLPlusNumberOfThreads`	*i*SQL*Plus サーバーが同時に処理できる HTTP 要求の最大数を設定します。
`iSQLPlusLogLevel`	*i*SQL*Plus でログ・ファイルをメンテナンスできるようにし、ログ・ファイルのメンテナンス・レベルを決めます。ログ・ファイルのデフォルトの保存場所は、`$ORACLE_HOME/sqlplus/log/isqlplus/log.xml` です。*i*SQL*Plus には、`$ORACLE_HOME/sqlplus/log` ディレクトリに対する書込み権限が必要です。このディレクトリには、所有者の書込み権限があります。*i*SQL*Plus が `nobody` ユーザーとして動作しているために、ログ・ファイルに書き込めない場合は、ログ・ファイルに対する書込み権限を手動で *i*SQL*Plus に付与する必要があります。一時的なテストまたはトレース以外の目的で、このファイルに対する `world`（全ユーザーを対象とした）書込み権限を付与しないようにしてください。
`iSQLPlusTimeOutInterval`	セッションが期限切れになるまでのアイドル時間を設定します。*i*SQL*Plus セッションがタイムアウトになると、システムの負荷が減り、リソースを最大限に活用できます。
`iSQLPlusHashTableSize`	*i*SQL*Plus サーバーが同時に処理できる *i*SQL*Plus セッションの最大数を設定します。デフォルト値は、`iSQLPlusNumberOfThreads` パラメータの値から取得されます。
`iSQLPlusConnectIdList`	ユーザーがログイン画面の「接続識別子」テキスト・フィールドを使用しないで、*i*SQL*Plus からアクセスできるデータベースのドロップダウン・リストを設定します。これにより、ホスト環境での *i*SQL*Plus サーバーのセキュリティが向上します。

パラメータ	説明
iSQLPlusAllowUserEntMap	*i*SQL*Plus のユーザーが SET MARKUP HTML ENTMAP および COLUMN ENTMAP パラメータの設定を変更できるかどうかを制御します。 HTML では、エンティティ・マッピングによって特別な意味を持つ文字が印刷可能な表現に置き換えられます。デフォルトでは、エンティティ・マッピングが有効になるため、iSQLPlusAllowUserEntMap パラメータの値は none に設定されます。この値が設定されると、ユーザーはエンティティ・マッピングの設定を変更できないため、*i*SQL*Plus の出力でユーザー定義の HTML を使用できません。 iSQLPlusAllowUserEntMap パラメータの値を「all」に設定すると、ユーザーはエンティティ・マッピングの設定を変更できます。
-idle-timeout	Oracle HTTP Server が *i*SQL*Plus からの結果を待機する時間を設定します。

次の例は、isqlplus.conf ファイルに入っている FastCgiServer ディレクティブのサンプルを示しています。

```
FastCgiServer ORACLE_HOME/isqlplus -initial-env
iSQLPlusNumberOfThreads=20 -initial-env iSQLPlusTimeOutInterval=30 -initial-env
iSQLPlusLogLevel=warn -idle-timeout 3600
```

この例の *ORACLE_HOME* は、Oracle ホーム・ディレクトリのパスです。

セキュリティ

この項では、*i*SQL*Plus 固有のセキュリティ問題について説明します。

> **参照：** *i*SQL*Plus のセキュリティの詳細は、『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。

*i*SQL*Plus の Oracle HTTP Server 認証の設定

ユーザーは、次の 2 つの方法で *i*SQL*Plus にアクセスできます。

- SYSDBA または SYSOPER 権限を持つユーザーとして、アクセスする。
- SYSDBA または SYSOPER 権限を持たない通常ユーザーとして、アクセスする。

SYSDBA または SYSOPER 権限を持つユーザーとして *i*SQL*Plus にアクセスするには、通常ユーザーとして接続するときに使用する URL とは異なる URL を使用します。デフォルトでは、権限を持つユーザーが使用する URL は Oracle HTTP Server 認証を使用して保護されます。必要に応じて、Oracle HTTP Server 認証を使用して、通常ユーザーが使用する URL を保護することもできます。

保護された *i*SQL*Plus の URL にアクセスするには、ログイン画面にアクセスする前に、ユーザー名とパスワードを指定する必要があります。これらは、Oracle のユーザー名やパスワードとは異なります。これらのユーザー名とパスワードは、パスワード・ファイルに格納されています。

Oracle9*i* をインストールすると、SYSDBA または SYSOPER 権限を持つユーザー用の空のパスワード・ファイルがインストールされます。最初に権限を持つユーザーとして *i*SQL*Plus にアクセスする前に、このパスワード・ファイルに 1 つまたは複数のユーザー名およびパスワードを追加する必要があります。

同様に、通常ユーザーに Oracle HTTP Server 認証を使用する場合は、別のパスワード・ファイルを作成して、ユーザー名およびパスワードをそれに追加する必要があります。さらに、このファイルを使って通常ユーザーを認証するように、*i*SQL*Plus 構成ファイルを変更する必要があります。

以降の項では、ユーザーをパスワード・ファイルに追加し、必要に応じて *i*SQL*Plus 構成ファイルを変更する方法について説明します。

パスワード・ファイルへのユーザー名およびパスワードの追加

ユーザー名およびパスワードをパスワード・ファイルに追加するには、次の手順で行います。

1. Oracle HTTP Server を実行しているシステムに Oracle ソフトウェア所有者（`oracle`）としてログインします。
2. ディレクトリを `$ORACLE_HOME/Apache/Apache/bin` ディレクトリに変更します。

```
$ cd $ORACLE_HOME/Apache/Apache/bin
```

3. 次のどちらかのコマンドを入力して、ユーザー名およびパスワードを該当するパスワード・ファイルに追加します。

> **注意：** 権限を持つユーザー用のパスワード・ファイルを作成する必要はありません。このファイル（`iplusdba.pw`）は、デフォルトでインストールされます。

- 通常ユーザー用のパスワード・ファイルを作成して、ユーザー名およびパスワードをそれに追加するには、次のように入力します。

```
$ htpasswd -c $ORACLE_HOME/sqlplus/admin/filename.pw username
```

- ユーザー名およびパスワードを既存のパスワード・ファイルに追加するには、次のように入力します。

```
$ htpasswd $ORACLE_HOME/sqlplus/admin/filename.pw username
```

この例では、*`filename`*`.pw` は変更または作成するパスワード・ファイルの名前であり、*`username`* は追加するユーザー名です。権限を持つユーザーのパスワード・ファイルの名前は、`iplusdba.pw` です。権限を持たないユーザーの場合、パスワード・ファイルに `iplus.pw` という名前を使用することをお薦めします。

4. プロンプトが表示されたら、指定したユーザー名のパスワードを入力します。
5. パスワード・ファイルを新しく作成した場合は、次の項の説明に従って、そのパスワード・ファイルが使用されるように Oracle HTTP Server を設定します。

新しいパスワード・ファイルが使用されるように Oracle HTTP Server を設定する方法

通常ユーザー用のパスワード・ファイルを新しく作成したら、*i*SQL*Plus の URL にアクセスするユーザーを認証するときにこのパスワード・ファイルが使用されるように、Oracle HTTP Server を設定する必要があります。

新しいパスワード・ファイルが使用されるように Oracle HTTP Server を設定するには、次の手順で行います。

1. Oracle HTTP Server を実行しているシステムに Oracle ソフトウェア所有者（`oracle`）としてログインします。
2. ディレクトリを `$ORACLE_HOME/sqlplus/admin` ディレクトリに変更します。

   ```
   $ cd $ORACLE_HOME/sqlplus/admin
   ```

3. 任意のテキスト・エディタで `isqlplus.conf` 構成ファイルを開きます。
4. このファイルから次のセクションを見つけます。

   ```
   #
   # Enable handling of all virtual paths beginning with "/isqlplus"
   #
   <Location /isqlplus>
     SetHandler fastcgi-script
     Order deny,allow

     # Comment "Allow ..." and uncomment the four lines "AuthType ..."
     # to "Require ..." if Oracle HTTP authentication access is required
     # for the http://.../isqlplus URL
     Allow from all
     #AuthType Basic
     #AuthName 'iSQL*Plus'
     #AuthUserFile ORACLE_HOME/sqlplus/admin/iplus.pw
     #Require valid-user
   </Location>
   ```

   この例の *`ORACLE_HOME`* は、Oracle ホーム・ディレクトリのパスです。

5. 次の例に示すように、このセクションを変更します。

   ```
   #
   # Enable handling of all virtual paths beginning with "/isqlplus"
   #
   <Location /isqlplus>
    SetHandler fastcgi-script
    Order deny,allow
    AuthType Basic
    AuthName 'iSQL*Plus'
   ```

```
 AuthUserFile ORACLE_HOME/sqlplus/admin/filename.pw
 Require valid-user
</Location>
```

この例では、*ORACLE_HOME* は Oracle ホーム・ディレクトリのパスであり、*filename*.pw は通常ユーザー用に作成したパスワード・ファイルの名前（通常は iplus.pw）です。

6. Oracle HTTP Server の構成ファイルを解析することによって、変更内容を確認します。

```
$ $ORACLE_HOME/Apache/Apache/bin/apachectl configtest
```

必要に応じて、このコマンドによって表示されたエラーを修正します。

7. Oracle HTTP Server を再起動します。

```
$ $ORACLE_HOME/Apache/Apache/bin/apachectl restart
```

Oracle HTTP Server が再起動すると、ブラウザにログイン画面が表示される前に、*i*SQL*Plus の URL にアクセスするユーザーに対して、Oracle HTTP Server 認証のユーザー名とパスワードを要求するプロンプトが表示されます。

*i*SQL*Plus からのデータベース・アクセスの制限

*i*SQL*Plus からのデータベースへのアクセスを制限することができます。データベース・アクセスの制限を有効にすると、ログイン画面の「接続識別子」テキスト・フィールドを使用しないで、利用できるデータベースのドロップダウン・リストが表示されます。これにより、ホスト環境での *i*SQL*Plus サーバーのセキュリティが向上します。接続識別子は、iSQLPlusConnectIdList パラメータに定義した順に表示されます。

isqlplus.conf ファイルを編集して、データベース・アクセスの制限を有効にすることができます。次の行を変更してください。*SID1, SID2, ...* は、使用できるデータベースを示す Oracle Net 接続識別子をカンマで区切ったリストです。

```
FastCgiServer ... -initial-env "iSQLPlusConnectIdList=SID1, SID2,..."
```

たとえば、次のように入力します。

```
FastCgiServer ... -initial-env "iSQLPlusConnectIdList=ABC1, PROD2, DEV3"
```

参照： isqlplus.conf ファイルの編集方法については、3-7 ページの「iSQL*Plus 構成ファイルの編集」を参照してください。

接続識別子の中に引用符を指定したり、空白を埋め込むことはできませんが、前述の例に示すように、iSQLPlusConnectIdList= 引数全体を引用符で囲む必要があります。接続識別子には大文字と小文字の区別がなく、引数内に示された各接続識別子は tnsnames.ora ファイルに定義されている別名と同じである必要があります。

接続識別子を一度設定すると、ログイン画面を使用して行われたすべての接続、すべての動的レポート、および CONNECT コマンドで試みたすべての接続のうち、制限リストに指定されたデータベースへの接続以外は拒否されます。

同様に、システム変数 SET INSTANCE を使用した場合は、定義された接続識別子が `iSQLPlusConnectIdList` パラメータのエントリと一致している必要があります。一致していない場合は、接続が拒否されます。

接続識別子が指定されていない場合、または `iSQLPlusConnectIdList` パラメータのエントリと一致しない場合、データベースの接続は拒否され、次のエラーが発生します。

```
SP2-0884: データベース database_name への接続は許可されません。
```

この例の *database_name* は、接続しようとしているデータベースの名前です。

コマンドライン SQL*Plus の使用

この項では、UNIX システムでコマンドライン SQL*Plus を使用する方法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 プロンプトで ED または EDIT コマンドを入力すると、`ed`、`emacs` または `ned`、`vi` などのオペレーティング・システム・エディタが起動されます。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 _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・プロファイルや `login.sql` ユーザー・プロファイルに定義することも、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、次のように入力します。

```
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 のどちら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 の値が優先されます。_EDITOR、EDITOR、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。

エディタを起動した場合、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することができます。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。

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際に使用するシェルは、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（`/bin/sh`）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` コマンドを入力するか、[Ctrl] キーを押しながら [D] キーを押します。

たとえば、特定のコマンドを実行するには、次のように入力します。

```
SQL>! command
```

`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 ! コマンドを入力してから、[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。

SQL*Plus への割込み

SQL*Plus の実行時に、[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。

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
```

SQL*Plus の制限事項

この項では、SQL*Plus の制限事項を説明します。

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・サイズは自動的に調整されません。

リターン・コード

UNIX 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。

パスワードの隠ぺい

環境変数 SYSTEM_PASS に SYSTEM ユーザーのユーザー名およびパスワードを設定した場合、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ときにだけ SYSTEM パスワードを入力してください。

自動的にスクリプトを実行させる場合は、Oracle9*i* へのログインが外部で認証されるなど、パスワードを格納する必要のない認証方法の使用を考慮してください。セキュリティの低い環境では、スクリプト・ファイルに UNIX パイプを使用して、パスワードを SQL*Plus に渡すことを考えてみてください。たとえば、次のように入力します。

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

また、コマンド・プロンプトで次の行を入力することもできます。

```
$ 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。

4

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 について説明します。この章では、次の項目について説明します。

- Oracle プリコンパイラの概要
- 32 ビットおよび 64 ビットのクライアント・アプリケーションのサポート
- Pro*C/C++ プリコンパイラ
- Pro*COBOL プリコンパイラ（AIX、HP、Solaris および Tru64 のみ）
- Pro*FORTRAN プリコンパイラ（AIX、HP、Solaris および Tru64 のみ）
- SQL*Module for Ada（Solaris 32 ビットおよび AIX のみ）
- Oracle Call Interface
- カスタム Make ファイル
- 未定義シンボルの修正（Solaris のみ）
- マルチスレッド・アプリケーション
- シグナル・ハンドラの使用方法
- XA 機能

参照： SQL*Plus を使用してデモンストレーション表を作成する方法については、3-4 ページの「デモンストレーション表の使用」を参照してください。

Oracle プリコンパイラの概要

Oracle プリコンパイラは、Oracle データベースの SQL 文と高水準言語で書かれたプログラムとを組み合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、Oracle9*i* 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。

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 4-1 は、各プリコンパイラの構成ファイルの一覧です。

表 4-1 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++ リリース 9.2.0.1.0	`pcscfg.cfg`
Pro*COBOL リリース 9.2.0.1.0（AIX、HP、Solaris および Tru64 のみ）	`pcbcfg.cfg`
Pro*COBOL リリース 1.8.77.0.0（AIX、HP、Solaris および Tru64 のみ）	`pcccob.cfg`
Pro*FORTRAN リリース 1.8.77.0.0（AIX、HP、Solaris および Tru64 のみ）	`pccfor.cfg`
Object Type Translator リリース 9.2.0.1.0	`ottcfg.cfg`
Oracle SQL*Module for Ada リリース 9.2.0.1.0（Solaris 32 ビットおよび AIX のみ）	`pmscfg.cfg`

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
```

このコマンドを実行すると、まず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 ディレクトリに移されます。新しい実行可能ファイルを作成したときに、そのファイルを $ORACLE_HOME/bin ディレクトリに移動しない場合は、次のコマンドを入力します。

```
$ make -f ins_precomp.mk executable
```

executable は、表 4-2 に示している製品の実行可能ファイルです。

表 4-2 製品とそれに対応する実行可能ファイル

製品	実行可能ファイル
Pro*C/C++ リリース 9.2.0.1.0	proc
Pro*COBOL リリース 9.2.0.1.0（AIX、HP、Solaris および Tru64 のみ）	procob または rtsora
Pro*COBOL リリース 1.8.77.0.0（AIX、HP、Solaris および Tru64 のみ）	procob18 または rtsora
Pro*FORTRAN リリース 1.8.77.0.0（AIX、HP、Solaris および Tru64 のみ）	profor
Object Type Translator リリース 9.2.0.1.0	ott
Oracle SQL*Module for Ada リリース 9.2.0.1.0（Solaris 32 ビットおよび AIX のみ）	modada

プリコンパイラの README ファイル

表 4-3 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 4-3 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++ リリース 9.2.0.1.0	`$ORACLE_HOME/precomp/doc/proc/readme.doc`
Pro*COBOL リリース 9.2.0.1.0	`$ORACLE_HOME/precomp/doc/procob2/readme.doc`
Pro*COBOL リリース 1.8.77.0.0 および Pro*FORTRAN リリース 1.8.77.0.0	`$ORACLE_HOME/precomp/doc/pro1x/readme.txt`

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るものです。

> **注意：** Oracle プリコンパイラのデモを実行するには、Oracle9*i* をインストールしておく必要があります。

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。これが原因で、ユーザーが存在しないことを示すエラー・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文字 / 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グ・プログラム

ベンダー提供のデバッガとプリコンパイラ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に同様に動作するとは限りません。

IRECLEN および ORECLEN の値

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

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 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ことがあります。

動的リンクの場合、実行コードの一部分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いいます。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。

- ディスク要件が少なくなる : 複数のアプリケーションまたは同一アプリケーションの呼出しで、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る : 同一の動的ライブラリ・イメージをメイン・メモリーに 1 度だけロードしておけば、複数のアプリケーションでそれを共有できます。

クライアント共有ライブラリ

クライアント共有ライブラリは、`$ORACLE_HOME/lib` ディレクトリ内にあります。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。

実行可能ファイルの起動時に、次のいずれかのエラー・メッセージが表示される場合があります。

- AIX システムの場合

```
$ sample1
exec(): 0509-036 Cannot load program ./sample1 because of the following
errors:
0509-022 Cannot load library libclntsh.a [shr.o]
0509-026 System error: A file or directory in the pathname does not exist.
```

- HP システムの場合

```
$ sample1
/usr/lib/dld.sl: Can't open shared library:
/u01/app/oracle/product/9.2.0.1.0/lib/libclntsh.sl.9.0
/usr/lib/dld.sl: No such file or directory
Abort (core dumped)
```

- Solaris および Linux システムの場合

```
$ sample1
ld.so.1: sample1: fatal: libclntsh.so.1.0: can't open file: errno=2
Killed
```

- Tru64 システムの場合

```
$ sample1
/sbin/loader: Fatal Error: Cannot map libclntsh.so
Killed
```

これらのエラー・メッセージのいずれかが表示された場合は、使用しているプラットフォームに応じて、次のいずれかの環境変数を設定します。

- AIX システムの場合

```
$ LIBPATH=$ORACLE_HOME/lib:${LIBPATH}
$ export LIBPATH
```

- HP システム（64 ビットのアプリケーション）、Linux、Solaris および Tru64 システムの場合

```
$ LD_LIBRARY_PATH=$ORACLE_HOME/lib:${LD_LIBRARY_PATH}
$ export LD_LIBRARY_PATH
```

- HP システム（32 ビットのアプリケーション）の場合

```
$ SHLIB_PATH=$ORACLE_HOME/lib32:${SHLIB_PATH}
$ export SHLIB_PATH
```

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作成しなおす必要がある場合は、次の手順で行います。

1. クライアント共有ライブラリを使用するクライアント・アプリケーションをすべて終了します。この中には、SQL*Plus や Recovery Manager などの Oracle クライアント・アプリケーションもすべて含まれます。
2. `oracle` ユーザーとしてログインし、次のように入力します。

```
$ genclntsh
```

32 ビットおよび 64 ビットのクライアント・アプリケーションのサポート（AIX、HP および Solaris 64 ビットのみ）

AIX、HP および Solaris 64 ビットの各システムの Oracle9*i* リリース 2（9.2.0.1.0）では、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サポートしています。デフォルトでは、リリース 2（9.2.0.1.0）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。ただし、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することができます。

AIX、HP および Solaris 64 ビットの各システムの Oracle9*i* リリース 2（9.2.0.1.0）では、32 ビットおよび 64 ビットのアプリケーションを実行できます。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/lib/libclntsh.a`
HP	`$ORACLE_HOME/lib32/libclntsh.sl`	`$ORACLE_HOME/lib/libclntsh.sl`
Solaris 64 ビット	`$ORACLE_HOME/lib32/libclntsh.so`	`$ORACLE_HOME/lib/libclntsh.so`

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、次の手順で行います。

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。

```
$  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、Solaris 64 ビット	LIBPATH
HP（32 ビットのクライアント・アプリケーション）	SHLIB_PATH
HP（64 ビットのクライアント・アプリケーション）	LD_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

Oracle9*i* リリース 2（9.2.0.1.0）では、32 ビットおよび 64 ビットの Pro*C および OCI（Oracle Call Interface）カスタマ・アプリケーションをサポートしています。

> **参照：** 32 ビットの Pro*C アプリケーションの作成方法については、`$ORACLE_HOME/precomp/demo/demo_proc.mk` および `$ORACLE_HOME/precomp/demo/demo_proc32.mk` の各ファイルを参照してください。
>
> 32 ビットの OCI アプリケーションの作成方法については、`$ORACLE_HOME/rdbms/demo/demo_rdbms.mk` および `$ORACLE_HOME/rdbms/demo/demo_rdbms32.mk` の各ファイルを参照してください。

32 ビットの実行可能ファイルおよびディレクトリ（AIX および HP のみ）

HP および AIX システムの Oracle9*i* リリース 2（9.2.0.1.0）には、次の 32 ビットの実行可能ファイルおよびライブラリが含まれています。

- `$ORACLE_HOME/lib32`
- `$ORACLE_HOME/rdbms/lib32`
- `$ORACLE_HOME/hs/lib32`
- `$ORACLE_HOME/network/lib32`
- `$ORACLE_HOME/precomp/lib32`
- `$ORACLE_HOME/bin`

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。

> **参照：** 各プラットフォームに必要なコンパイラのバージョンについては、『Oracle9*i* for UNIX Systems 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については、『Pro*C/C++ Precompiler プログラマーズ・ガイド』を参照してください。

Pro*C/C++ のデモ・プログラム

デモ・プログラムは、Pro*C/C++ プリコンパイラのいくつか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++、Object プログラムという 3 つの種類があります。Object プログラムでは、Oracle9*i* Object の新機能がいくつか紹介されています。デモ・プログラムはすべて、`$ORACLE_HOME/precomp/demo/proc` ディレクトリ内にあります。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。

これらのプログラムでは、`$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、そのパスワードは TIGER であると仮定しています。

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、デモ・プログラム `sample1` をプリコンパイル、コンパイルおよびリンクするには、次のコマンドを入力します。

```
$ 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、次のコマンドを入力します。

```
$ 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、次のコマンドを入力します。

```
$ 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、次のコマンドを入力します。

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

デモ・プログラムによっては、`$ORACLE_HOME/precomp/demo/sql` ディレクトリ内にある SQL スクリプトを実行する必要があります。このスクリプトを実行しないと、実行を要求するメッセージが表示されます。デモ・プログラムを作成し、それに対応する SQL スクリプトを実行するには、「`make` *`macro_argument`* `RUNSQL=run`」をコマンドラインに追加します。たとえば、`calldemo` デモ・プログラムを作成し、必要な `$ORACLE_HOME/precomp/demo/sql/calldemo.sql` スクリプトを実行する場合は、次のように入力します。

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

Object デモ・プログラムをすべて作成し、必要な SQL スクリプトをすべて実行するには、次のように入力します。

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

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。Make ファイル demo_proc.mk を使用してプログラムを作成するコマンドの構文は、次のとおりです。

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

前述の例の各項目について説明します。

- *target* は、使用する Make ファイル・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、作成する実行可能プログラムのソースとタイプに応じて、次のいずれかのコマンドを入力します。

- 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、次のコマンドを入力します。

  ```
  $ make -f demo_proc.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、次のコマンドを入力します。

  ```
  $ 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、次のコマンドを入力します。

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

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

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

注意： AIX、HP および Solaris 64 ビットの各システムでは、Make ファイル demo_proc.mk はデフォルトで 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 ビットのユーザー・プログラムを作成することもできます。32 ビットのユーザー・プログラムの作成方法については、その Make ファイルを参照してください。

Pro*COBOL プリコンパイラ（AIX、HP、Solaris および Tru64 のみ）

このリリースの Pro*COBOL には、2 つのバージョンがあります。表 4-4 に、各バージョンのネーミング規則を示します。

表 4-4 Pro*COBOL のネーミング規則

項目	Pro*COBOL リリース 9.2.0.1.0	Pro*COBOL リリース 1.8.77.0.0
実行可能ファイル	procob	procob18
デモ・ディレクトリ	procob2	procob
Make ファイル（Merant Server Express COBOL）	demo_procob.mk	demo_procob18.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` 実行可能ファイルを使用します。

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。

Merant Server Express COBOL コンパイラ

Merant Server Express COBOL コンパイラの場合は、環境変数 COBDIR を設定し、使用しているオペレーティング・システムに応じて環境変数 LD_LIBRARY_PATH、LIBPATH または SHLIB_PATH を設定する必要があります。

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/cobol` ディレクトリにインストールされている場合は、次のように入力します。

```
$ COBDIR=${COBDIR}:/opt/cobol
$ export COBDIR
```

LD_LIBRARY_PATH（64 ビットのアプリケーションを扱う Solaris および HP のみ）　環境変数 LD_LIBRARY_PATH 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 `$COBDIR/coblib` ディレクトリにインストールされている場合は、次のように入力します。

```
$ LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:$COBDIR/coblib
$ export LD_LIBRARY_PATH
```

LIBPATH（AIX のみ） 環境変数 LIBPATH 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 `$COBDIR/coblib` ディレクトリにインストールされている場合は、次のように入力します。

```
$ LIBPATH=${LIBPATH}:$COBDIR/coblib
$ export LIBPATH
```

SHLIB_PATH（32 ビットのアプリケーションを扱う HP） 環境変数 SHLIB_PATH 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 `$COBDIR/coblib` ディレクトリにインストールされている場合は、次のように入力します。

```
$ SHLIB_PATH=${SHLIB_PATH}:$COBDIR/coblib
$ export SHLIB_PATH
```

環境変数 LD_LIBRARY_PATH、LIBPATH または SHLIB_PATH 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時に次のどちらかのエラー・メッセージが表示されます。

- Tru64 の場合

  ```
  14783 rtsora: /sbin/loader: Fatal Error: cannot map libwtc9.so
  ```

- AIX、HP、Solaris の場合

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

PATH 環境変数 PATH には、次のように、`/opt/SUNWnsun/bin` ディレクトリを設定します。

```
$ PATH ${PATH}:/opt/SUNWnsun/bin
$ export PATH
```

LD_LIBRARY_PATH 環境変数 LD_LIBRARY_PATH には、次のように、`/opt/SUNWnsun/bin` ディレクトリを設定します。

```
$ LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/opt/SUNWnsun/bin
$ export LD_LIBRARY_PATH
```

環境変数 LD_LIBRARY_PATH に `/opt/SUNWnsun/bin` 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時に次のエラー・メッセージが表示されます。

```
ld.so.1: cobol: fatal: liblicense.so: can't open file: errno=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 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、Merant 社が提供する cobrun ランタイム・システムのかわりに、rtsora 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
```

Pro*COBOL のデモ・プログラム

デモ・プログラムは、Pro*COBOL プリコンパイラのいくつか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、Pro*COBOL のリリースに応じて、$ORACLE_HOME/precomp/demo/procob ディレクトリまたは $ORACLE_HOME/precomp/demo/procob2 ディレクトリのどちらかにあります。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。

これらのプログラムでは、$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、そのパスワードは TIGER であると仮定しています。

該当する Make ファイルを使用して、デモ・プログラムを作成してください。

- Pro*COBOL リリース 9.2.0.1.0 の場合は、次の Make ファイルを使用します。

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

- Pro*COBOL リリース 1.8.77.0.0 の場合は、次の Make ファイルを使用します。

  ```
  $ORACLE_HOME/precomp/demo/procob/demo_procob18.mk
  ```

注意： 次の例では、Pro*COBOL リリース 9.2.0.1.0 を使用していることを前提とします。

---

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場合は、次のように入力します。

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

Pro*COBOL のデモ・プログラムをすべて作成するには、次のように入力します。

```
$ 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 を作成および実行するには、次のように入力します。

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

デモ・プログラムによっては、$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 *macro_argument* RUNSQL=run」をコマンドラインに追加します。たとえば、デモ・プログラム sample9 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、次のように入力します。

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

Pro*COBOL デモ・プログラムをすべて作成し、必要な SQL スクリプトをすべて実行するには、次のように入力します。

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

Pro*COBOL のユーザー・プログラム

使用している Pro*COBOL のリリースに応じて、適切なデモ用の Make ファイルを使用して、ユーザー・プログラムを作成できます。デモ用の Make ファイルを使用してプログラムを作成するコマンドの構文は、次のとおりです。

- Pro*COBOL リリース 9.2.0.1.0 の場合は、次のように入力します。

```
$ make -f demo_procob.mk target COBS="cobfile1 cobile2 ..." EXE=exename
```

- Pro*COBOL リリース 1.8.77.0.0 の場合は、次のように入力します。

```
$ make -f demo_procob18.mk target COBS="cobfile1 cobfile2 ..." EXE=exename
```

前述の例の各項目について説明します。

- *target* は、使用する Make ファイル・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、作成する実行可能プログラムのソースとタイプに応じて、次のいずれかのコマンドを入力します。

- 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、次のコマンドを入力します。

```
$ 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、次のコマンドを入力します。

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

- COBOL ソースの場合、rtsora で使用するための動的読取りプログラムを作成する場合、次のコマンドを入力します。

  ```
  $ make -f demo_procob.mk myprog.gnt
  ```

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。デフォルトの FORMAT=ANSI を指定した場合、カラム 1 〜 6 はオプションの順序番号、カラム 7 はコメントまたは継続行を示す標識です。段落名はカラム 8 〜 11 で開始され、カラム 12 〜 72 が文となります。

FORMAT=TERMINAL を指定した場合、カラム 1 〜 6 は削除され、カラム 7 が左端の列になります。

Pro*FORTRAN プリコンパイラ（AIX、HP、Solaris および Tru64 のみ）

Pro*FORTRAN プリコンパイラを使用する場合は、事前に適切なリリース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。

> **参照：** 各プラットフォームに必要なコンパイラのリリースについては、『Oracle9*i* for UNIX Systems インストレーション・ガイド』を参照してください。

Pro*FORTRAN のデモ・プログラム

デモ・プログラムは、Pro*FORTRAN プリコンパイラのいくつか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 ディレクトリ内にあります。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。

これらのプログラムでは、$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、そのパスワードは TIGER であると仮定しています。

デモ・プログラムを作成するには、$ORACLE_HOME/precomp/demo/profor ディレクトリ内にある Make ファイル demo_profor.mk を使用します。たとえば、sample1 というデモ・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場合は、次のように入力します。

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

Pro*FORTRAN のデモ・プログラムをすべて作成するには、次のように入力します。

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

デモ・プログラムによっては、$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 *macro_argument* RUNSQL=run」をコマンドラインに追加します。たとえば、デモ・プログラム sample11 を作成し、必要な $ORACLE_HOME/precomp/demo/sql/sample11.sql スクリプトを実行する場合は、次のように入力します。

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

Pro*FORTRAN デモ・プログラムをすべて作成し、必要な SQL スクリプトをすべて実行するには、次のように入力します。

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

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。Make ファイル `demo_proc.mk` を使用してプログラムを作成するコマンドの構文は、次のとおりです。

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

前述の例の各項目について説明します。

- *target* は、使用する Make ファイル・ターゲットです。
- *forfilen* は、プログラムの FORTRAN ソースです。
- *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、作成する実行可能プログラムのタイプに応じて、次のどちらかのコマンドを入力します。

- 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、次のコマンドを入力します。

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

- 静的にリンクさせるには、次のように入力します。

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

SQL*Module for Ada（Solaris 32 ビットおよび AIX のみ）

SQL*Module for Ada を使用する場合は、事前に適切なリリース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。

> **参照：** 各プラットフォームに必要なコンパイラのリリースについては、『Oracle9*i* for UNIX Systems インストレーション・ガイド』を参照してください。SQL*Module for Ada の詳細については、『SQL*Module for Ada Programmer's Guide 』を参照してください。

SQL*Module for Ada のデモ・プログラム

デモ・プログラムは、SQL*Module for Ada のいくつか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` ディレクトリ内にあります。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 `ch1_drv` では、`$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、そのパスワードは TIGER であると仮定しています。

`demcalsp` と `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 スキーマに入っ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・プログラムをすべて作成し、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、次のように入力します。

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

デモ・プログラムを 1 つだけ（`demohost`）作成し、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、次のように入力します。

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

SQL*Module for Ada デモ・プログラムをすべて作成し、大学のサンプル・データベースを作成しなおさない場合は、次のように入力します。

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

デモ・プログラムを 1 つだけ（`demohost`）作成し、大学のサンプル・データベースを作成しなおさない場合は、次のように入力します。

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

すべてのプログラムでは、Oracle Net の接続文字列または別名 INST1_ALIAS が定義されていて、適切な表が入っているデータベースに接続できるものとします。

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コマンドの構文は、次のとおりです。

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

前述の例の各項目について説明します。

- *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。

> **参照：** SQL*Module for Ada のユーザー・プログラムの詳細は、『SQL*Module for Ada Programmers Guide』を参照してください。

Oracle Call Interface

Oracle Call Interface（OCI）を使用する場合は、事前に適切なリリースの Pro*C/C++ がインストールされていることを確認してください。

> **参照：** ご使用のオペレーティング・システムに必要な Pro*C/C++ のリリースについては、『Oracle9*i* for UNIX Systems インストレーション・ガイド』を参照してください。Oracle Call Interface の詳細については、『Oracle Call Interface プログラマーズ・ガイド』を参照してください。

OCI のデモ・プログラム

デモ・プログラムは、OCI の機能をいくつか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 と C++ の 2 つの種類があります。デモ・プログラムはすべて、`$ORACLE_HOME/rdbms/demo` ディレクトリに入っています。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。

ほとんどのデモ・プログラムでは、`$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、そのパスワードは TIGER であると仮定しています。

デモ・プログラムを作成するには、`$ORACLE_HOME/rdbms/demo` ディレクトリに入っている Make ファイル `demo_rdbms.mk` を使用します。たとえば、`cdemo1` というデモ・プログラムをコンパイルおよびリンクするには、次のコマンドを入力します。

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

OCI の C デモ・プログラムをすべて作成するには、次のコマンドを入力します。

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

OCI の C++ デモ・プログラムをすべて作成するには、次のコマンドを入力します。

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

OCI 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。Make ファイル `demo_rdbms.mk` を使用してユーザー・プログラムを作成するコマンドの構文は、次のとおりです。

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

前述の例の各項目について説明します。

- *`target`* は、使用する Make ファイル・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/C++ ソース `myprog.c` からプログラム `myprog` を作成する場合、作成する実行可能プログラムのタイプに応じて、次のいずれかのコマンドを入力します。

- 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、次のコマンドを入力します。

  ```
  $ 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、次のコマンドを入力します。

```
$ make -f demo_rdbms.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、次のコマンドを入力します。

```
$ 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、次のコマンドを入力します。

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

注意： 前述の例では、ファイル `myprog.o` は、コンパイラによって生成されたオブジェクト・ファイルです。

AIX、HP および Solaris 64 ビットの各システムでは、Make ファイル `demo_rdbms.mk` はデフォルトで 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。32 ビットのユーザー・プログラムの作成方法については、Make ファイルを参照してください。

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 ファイルを修正する場合、またはカスタム Make ファイルを使用する場合は、次の制限事項があります。

- 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。

  Oracle ライブラリの順番が重要である理由は、次のとおりです。

  – Oracle ライブラリは、相互に参照し合います。ライブラリ A の関数はライブラリ B の関数を呼び出し、ライブラリ B の関数はライブラリ A の関数を呼び出します。

  – HP と Tru64 の各リンカーは、1 パス・リンカーです。AIX、Linux、Solaris の各リンカーは、2 パス・リンカーです。

- リンク・ラインに独自の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、マクロおよび接尾辞の処理について、プラットフォー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、プラットフォームの 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。

未定義シンボルの修正（Solaris のみ）

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・ファイルの場所を調べる場合に役立ちます。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 つと見なされ、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、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は、sqlcex シンボルを見つけるときに使用します。

```
$ symfind sqlcex

SymFind - Find Symbol <sqlcex> in <**>.a, .o, .so
------------------------------------------------------
Command:         /u01/app/oracle/product/9.2.0.1.0/bin/symfind sqlcex
Local Directory: /u01/app/oracle/product/9.2.0.1.0
Output File:     (none)
Note:            I do not traverse symbolic links
                 Use '-v' option to show any symbolic links

Locating Archive and Object files ...
[11645] |    467572|      44|FUNC |GLOB |0    |8      |sqlcex
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ ./lib/libclntsh.sl
[35]    |         0|      44|FUNC |GLOB |0    |5      |sqlcex
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ ./lib/libsql.a
```

マルチスレッド・アプリケーション

今回のリリースで提供されている Oracle ライブラリはスレッド・セーフで、マルチスレッド・アプリケーションをサポートできます。

シグナル・ハンドラの使用方法

Oracle9*i* では、2 タスク通信にいくつかのシグナルが使用されます。シグナルは、データベースに接続するとユーザー・プロセスに作成され、切断すると削除されます。

表 4-5 に、Oracle9*i* が 2 タスク通信に使用するシグナルを示します。

表 4-5 2 タスク通信に使用するシグナル

シグナル	説明
SIGCLD	Oracle プロセスが異常終了すると、パイプ・ドライバは SIGCLD（SIGCHLD ともいいます）を使用します。UNIX カーネルはユーザー・プロセスに SIGCLD シグナルを送ります。サーバー・プロセスが異常終了しているかどうかを調べるには、シグナル・ハンドラで wait() ルーチンを使用します。Oracle プロセスではなく、ユーザー・プロセスが SIGCLD を受け取ります。
SIGCONT	パイプ 2 タスク・ドライバが、バンド外ブレークをユーザー・プロセスから Oracle プロセスに送信する場合に、SIGCONT を使用します。
SIGINT	2 タスク・ドライバが、ユーザーの割込み要求を検出する場合に、SIGINT を使用します。Oracle プロセスではなく、ユーザー・プロセスが SIGINT を受け取ります。
SIGIO	Oracle Net プロトコル・メソッドが、ネットワーク・イベントの着信を示す場合に、SIGIO を使用します。
SIGPIPE	パイプ・ドライバが、通信チャネルのファイルの終わりを検出する場合に、SIGPIPE を使用します。パイプへの書込み時に、読取りプロセスが存在していなければ、SIGPIPE シグナルが書込みプロセスに送信されます。Oracle プロセスとユーザー・プロセスの両方が、SIGPIPE を受け取ります。SIGCLD は、SIGPIPE に似ていますが、ユーザー・プロセスだけに適用され、Oracle プロセスには適用されません。
SIGTERM	パイプ・ドライバが、ユーザー側から Oracle プロセスに割込みシグナルを送る場合に、SIGTERM を使用します。ユーザーが割込みキー（[Ctrl]+[C]）を押すと、このシグナルが送られます。ユーザー・プロセスではなく、Oracle プロセスが SIGTERM を受け取ります。
SIGURG	Oracle Net TCP/IP ドライバが、バンド外ブレークをユーザー・プロセスから Oracle プロセスに送信する場合に、SIGURG を使用します。

表に載っているシグナルは、すべてのプリコンパイラ・アプリケーションに影響します。Oracle プロセスへの接続時に、SIGCLD（または SIGCHLD）および SIGPIPE にシグナル・

ハンドラを 1 つだけインストールできます。osnsui() ルーチンを呼び出して設定すると、複数のシグナル・ハンドラを SIGINT 用にインストールできます。SIGINT の場合、osnsui() および osncui() を使用して、シグナル受取りルーチンを登録および削除します。

また、必要に応じて、他のシグナルにもシグナル・ハンドラをインストールできます。Oracle プロセスに接続していない場合は、複数のシグナル・ハンドラをインストールできます。

例 4-1 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 4-1 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 *handlp, 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 *handlp 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 4-2 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 4-2 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 my user interrupt handler */

        if (err = osnsui(&handle, sig_handler, (char *) 0))
        {
                /* if the return value is non-zero, an error has occurred
                   Take appropriate action for the error. */
                ...
        }

        ...

        /* clear my interrupt handler */

        if (err = osncui(handle))
        {
                /* if the return value is non-zero, an error has occurred
                   Take appropriate action for the error. */
                ...
        }
        ...
}
```

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（Oracle など）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 クライアント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`）を使用する場合にのみ必要です。

Oracle9*i* は、Oracle7 7.1.6 XA コールには対応していません（Oracle7 7.3 XA コールには対応しています）。このため、Oracle7 7.1.6 XA コールを使用する TP モニター XA アプリケーションは、Oracle9*i* XA ライブラリに再リンクする必要があります。Oracle9*i* の XA コールは、共有ライブラリ `$ORACLE_HOME/lib/libclntsh.sl` および静的ライブラリ `$ORACLE_HOME/lib/libclient9.a` の両方で定義されています。

5

Oracle Net Services の構成

この章では、AIX、HP、Linux、Solaris および Tru64 の各システムで Oracle Net Services を構成する方法について説明します。この章では、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 Enterprise Manager
- Oracle SNMP での Oracle Intelligent Agent の構成
- Oracle Advanced Security
- PL/SQL からの 32 ビット外部プロシージャのコール（AIX、HP および Solaris 64 ビットのみ）

> **参照：** Oracle ネットワーキングの詳細は、『Oracle9*i* Net Services 管理者ガイド』を参照してください。

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 ディレクトリに入っています。このファイルに応じて、Oracle Net は異なる検索順序でファイルを見つけます。

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（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（設定されている場合）
2. 次のディレクトリのどちらか
   - Solaris システムの場合、/var/opt/oracle ディレクトリ
   - AIX、HP、Linux および Tru64 の各システム場合、/etc ディレクトリ
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・ディレクトリに保存されている）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
sqlnet.ora	$HOME/.sqlnet.ora
tnsnames.ora	$HOME/.tnsnames.ora

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、names.ora、sqlnet.ora、および tnsnames.ora 構成ファイルのサンプルが入っています。

adapters ユーティリティ

ご使用のシステムで Oracle9*i* 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 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、次のコマンドを入力します。

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

`adapters` ユーティリティを実行すると、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
         Oracle Names Server 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
         CyberSAFE authentication
         RADIUS authentication
         ENTRUST authentication
```

クライアントのシステムで、構成済の Oracle トランスポート・プロトコル、ネーミング・メソッドおよびセキュリティ・オプションを表示する場合は、`adapters` ユーティリティを

実行します。クライアントで adapters ユーティリティを実行するには、次のコマンドを入力します。

```
$ cd $ORACLE_HOME/bin
$ adapters
```

adapters ユーティリティを実行すると、次のような出力が表示されます。

```
Installed Oracle Net transport protocols are:

         IPC
         BEQ
         TCP/IP
         SSL
         RAW

Installed Oracle Net naming methods are:

         Local Naming (tnsnames.ora)
         Oracle Directory Naming
         Oracle Host Naming
         Oracle Names Server Naming
         NIS Naming

Installed Oracle Advanced Security options are:

         RC4 40-bit encryption
         RC4 56-bit encryption
         RC4 128-bit encryption
         RC4 256-bit encryption
         DES40 40-bit encryption
         DES 56-bit encryption
         3DES 112-bit encryption
         3DES 168-bit encryption
         AES 128-bit encryption
         AES 192-bit encryption
         AES 256-bit encryption
         MD5 crypto-checksumming
         SHA-1 crypto-checksumming
         Kerberos v5 authentication
         CyberSAFE authentication
         RADIUS authentication
```

参照： adapters ユーティリティの詳細は、『Oracle9*i* Net Services 管理者ガイド』を参照してください。

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、SSL 付き TCP/IP の各プロトコル・サポートでは、Oracle Net Services の構成ファイルと、`init`*`sid`*`.ora` ファイルの DISPATCHER 初期化パラメータで使用されるアドレスを指定します。以降の項では、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。

> **参照：** Oracle protocol support の詳細は、『Oracle9*i* Net Services 管理者ガイド』を参照してください。

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 Oracle9*i* サーバー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・サポートの動作には、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 Oracle9*i* 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。大文字と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=PROD)
)
```

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 Oracle9*i* サーバーが同じシステムまたは別のシステムのどちら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。ご使用のシステムに TCP/IP プロトコルをインストールした場合は、TCP/IP プロトコル・サポートがすべてのクライアント・ツールと Oracle9*i* 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を表します。この値は TCP です。大文字と小文字は区別されません。
HOST	ホスト名またはホスト IP アドレスを表します。
PORT	TCP/IP のポートを表します。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)
)
```

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 を介してリモートの Oracle データベースと通信を行うことができます。SSL 付き TCP/IP を使用するには、Oracle Advanced Security をインストールしておく必要があります。

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。大文字と小文字は区別されません。
HOST	ホスト名またはホスト IP アドレスを表します。
PORT	SSL 付き TCP/IP のポートを表します。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)
)
```

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ようにしてください。デフォルトのポートは 1521 です。次のように、リスナー名とポート番号という形式で入力します。

```
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。

```
oraclelistenerssl    2484/tcps
```

この例の *oraclelistenerssl* 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。

Oracle Enterprise Manager

Tcl スクリプトをデバッグするときは、`oratclsh` 実行可能ファイルを使用します。`oratclsh` を実行する前に、環境変数 TCL_LIBRARY で、`$ORACLE_HOME/network/agent/tcl` ディレクトリを設定してください。

> **参照：** Tcl スクリプトのデバッグ方法については、『Oracle Intelligent Agent ユーザーズ・ガイド』を参照してください。

Oracle SNMP での Oracle Intelligent Agent の構成

Oracle Intelligent Agent では、Simple Network Management Protocol（SNMP）が機能している必要はありませんが、Intelligent Agent の起動前に Oracle SNMP サポートを構成することができます。後述する構成ファイルはすべて、`$ORACLE_HOME/network/snmp/peer` ディレクトリに保存されています。

マスター・エージェントの構成

`CONFIG.master` ファイルで、次の変更を行います。

1. MANAGER で始まる行を検索します。
2. MANAGER パラメータの値を、SNMP トラップ・メッセージの送信先システムの IP アドレスまたはホスト名に変更します。

   ファイル内の記述に従って、`CONFIG.master` ファイルに他の変更を加えることもできます。
3. AIX システムの場合だけは、次の行を `/etc/snmpd.conf` ファイルに追加します。*`ip_address`* は、Oracle サブエージェントの IP アドレスです。

   ```
   smux 0.0 " " ip_address
   ```

カプセル化機能の構成

カプセル化機能を構成するには、次の手順を実行します。

1. 次の行を `snmpd.conf` ファイルに追加します。*`hostname_or_IP_address`* は、ローカル・システムの IP アドレスまたはホスト名を表します。

   ```
   trap hostname_or_IP_address
   ```
2. AIX システムの場合だけは、必要に応じて、`start_peer` スクリプトの変数 NEW_SNMPD_PORT および NEW_TRAPD_PORT に指定されているポート番号を変更します。

   NEW_SNMPD_PORT 変数と NEW_TRAPD_PORT 変数には、異なるポート番号を指定する必要があります。
3. AIX 以外のすべてのプラットフォームでは、次の手順を実行します。
   - **a.** 必要に応じて、`CONFIG.encap` ファイルに指定されているポート番号を変更します。デフォルトのポート番号は 1161 です。
   - **b.** 手順 a でポート番号を変更した場合は、`start_peer` スクリプトの NEW_SNMPD_PORT 変数に指定されている値をこのポート番号に変更します。
   - **c.** 必要に応じて、NEW_TRAPD_PORT 変数に指定されている値を変更します。

この変数は、snmpd エージェントがトラップを送信する PEER カプセル化機能のポートを示します。このポート番号と NEW_SNMPD_PORT 変数に指定されているポート番号は異なっている必要があります。

start_peer スクリプトの SNMP デーモンの保存場所の確認

start_peer スクリプトには、次のような行が含まれています。*snmpd_executable_path* は、snmpd 実行可能ファイルへのパスです。

```
SNMPD=snmpd_executable_path
```

snmpd_executable_path は、システム上の snmpd 実行可能ファイルの保存場所でなければなりません。

SNMP コンポーネントの起動

SNMP コンポーネントを起動するには、次の手順を実行します。

1. 次のコマンドを入力して、SNMP コンポーネント（master_peer、encap_peer および snmpd）が動作していないことを確認します。

   ```
   $ ps -aef | grep peer
   $ ps -aef | grep snmp
   ```

   実行されているコンポーネントがあった場合、次に進む前に root ユーザーでログインして、kill コマンドを使用してプロセスを終了させます。

2. AIX システムの場合だけは、root ユーザーとして次のコマンドを入力し、システム固有の AIX SNMP エージェントを起動します。

   ```
   # startsrc -s snmpd "-f /tmp/snmpd.log"
   ```

   このコマンドを実行すると、SNMP デーモンが起動し、情報が /tmp/snmpd.log ファイルに記録されます。

3. AIX 以外のすべてのプラットフォームでは、次の手順を実行します。

   a. root ユーザーとして、start_peer スクリプトを実行し、PEER マスター・エージェント、PEER カプセル化機能およびシステム固有の UNIX SNMP エージェントを起動します。

      ```
      # cd $ORACLE_HOME/network/snmp/peer
      # ./start_peer -a
      ```

注意： ご使用のシステムに固有の UNIX SNMP エージェントがない場合は、PEER カプセル化機能を使用しないでください。マスター・エージェントのみを起動するには、`start_peer -m` を入力します。

---

b. 次のコマンドを入力して、SNMP コンポーネントが動作していることを確認します。

```
# ps -aef | grep peer
# ps -aef | grep snmp
```

データベース・サブエージェントの構成および起動

データベース・サブエージェント（Oracle Intelligent Agent）の構成および起動については、『Oracle Enterprise Manager 構成ガイド』を参照してください。

Oracle Advanced Security

Oracle Advanced Security のインストール中に、3 つの `.bak` ファイル（`naeet.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。

PL/SQL からの 32 ビット外部プロシージャのコール（AIX、HP および Solaris 64 ビットのみ）

64 ビットの Extproc 実行可能プログラム（extproc）および 32 ビットの Extproc 実行可能プログラム（extproc32）は、$ORACLE_HOME/bin ディレクトリにインストールされます。デフォルトでは、extproc 実行可能プログラムは 64 ビット外部プロシージャを実行できます。32 ビット外部プロシージャを使用可能にするには、次の手順で行います。

1. listener.ora ファイルの PROGRAM パラメータの値を次のように設定します。

   ```
   (PROGRAM=extproc32)
   ```

2. ご使用の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	SHLIB_PATH
Solaris 64 ビット	LD_LIBRARY_PATH

3. リスナーをいったんシャットダウンしてから、再起動します。

注意： 32 ビットまたは 64 ビットのどちらかの外部プロシージャを実行するようにリスナーを設定できますが、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。

A

Oracle9*i* for AIX-Based Systems の チューニング

この付録では、Oracle9*i* for AIX-Based Systems のチューニングについて説明します。

- メモリーおよびページング
- 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
- Oracle9i でのミラー復元

メモリーおよびページング

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、メモリーおよびディスク間でプログラムやデータのページングが行われます。

バッファ・キャッシュ間のページング・アクティビティの制御

ページング・アクティビティが必要以上に行われると、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）上に作成したデータベース・ファイルでは、ページングが頻繁に行われる可能性があります。この場合、複数の SGA データ・バッファ内の類似したジャーナル・ファイル・システムのバッファにも、参照頻度の高いデータが含まれている可能性があります。AIX ファイル・バッファ・キャッシュ・マネージャの動作によって、パフォーマンスが大きく低下することがあります。これによって、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ことがあります。

AIX では、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。次の AIX システム・パラメータをチューニングするには、`vmstat` コマンドを使用します。

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 用の最小永続バッファ・ページ数。
`MAXPERM`	ファイル I/O 用の最大永続バッファ・ページ数。

> **参照：** AIX システム・パラメータの詳細は、『AIX Performance Tuning Guide』または『AIX5L Performance Management Guide』を参照してください。

これらの制限を変更するには、`root` ユーザーとしてログインし、`vmtune` コマンドを使用します。AIX `vmtune` コマンドは、オペレーティング・システムのバージョンに固有です。あるバージョンの `vmtune` コマンドを別のバージョンの AIX 上で実行すると、オペレーティング・システムに障害が発生することがあります。

MINFREE および MAXFREE パラメータのチューニング

MINFREE パラメータのサイズは、即時応答を必要とするプログラムの作業セットのサイズに基づいて設定できます。空きリストには十分な量のページを指定して、ロードされたプログラムが空きリストを補充しなくて済むようにする必要があります。プログラムの作業セットのサイズを決定するには、svmon -p コマンドを使用します。MAXFREE パラメータの値は、MINFREE パラメータの値と、MAXPGAHEAD パラメータの値または 8 ページのどちらか大きい値の合計値より大きくなければなりません。たとえば、MINFREE パラメータ値を 128 ページで、MAXPGAHEAD パラメータ値が 16 ページの場合、次のコマンドを入力して、MINFREE パラメータの値を 128 ページ、MAXFREE パラメータの値を 144 ページに設定します。

```
$ /usr/samples/kernel/vmtune -f 128 -F 144
```

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・システムの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えて、1 つまたは複数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。

> **参照：** AIX ファイル・バッファ・キャッシュの増加が及ぼす影響については、A-2 ページの「バッファ・キャッシュ間のページング・アクティビティの制御」を参照してください。

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）場合は、MINPERM パラメータ値を増やします。また、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て、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、簡単に数量化できません。マルチプログラミングの量や作業負荷の I/O 特性によって変わるためです。

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 と 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、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奪われます。
- 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、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奪われます。
- 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 および MAXPERM の間にある場合は、Virtual Memory Manager（VMM）によって通常ファイル・ページだけが奪われます。しかし、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奪われます。

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。

```
# vmtune -p 5 -P 20
```

デフォルト値は、それぞれ 20 パーセントおよび 80 パーセントです。

データベース・ファイルが RAW デバイス上にある場合は、MINPERM および MAXPERM パラメータを低い値に設定できます（それぞれ 5 パーセントおよび 20 パーセントなど）。RAW デバイスでは、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。このメモリーは、Oracle システム・グローバル領域など、他の目的で使用できます。

十分なページング領域の割当て

ページング領域を十分に割り当てないと、通常はシステムが停止したり、応答時間が非常に遅くなったりします。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、`lsps` 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、`smit pgsp` コマンドを使用します。

AIX 4.3.1 以前では、ページング領域の大きさは一般に実メモリーの 2 〜 3 倍にする必要がありました。ページング領域が使用されるかどうかに関係なく、プログラムが起動すると、ページング領域が事前に割り当てられていたためです。AIX 4.3.2 以上では、遅延ページングを使用するため、ページング領域が必要になるまでは割り当てられません。システムの実メモリーが不足した場合にだけ、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いため、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ければ、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。これに対し、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だけのページング領域が必要になります。

ページングの制御

必要以上のページングが常に発生する場合は、実メモリーが不足していることを示します。通常は、次のように対処してください。

- システムに高速の拡張記憶域が装備され、メモリーおよび拡張記憶域間のページングが、SGA およびディスク間のデータ読込みおよび書込み速度より十分に速くないかぎり、ページングが常に発生する状態は回避してください。
- 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。この処理は、場合によっては、メモリー・リソース要件およびその影響のバランスを取るために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挙げます。

  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 動的パフォーマンス表およびビューを問い合せ、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ことが判明したと想定しま

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た方が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*]

データベース・ブロック・サイズの設定

I/O スループットを向上させるために、Oracle データベース・ブロック・サイズを設定できます。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（デフォルトは 4KB）。Oracle データベースがジャーナル・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（AIX では 4KB）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 データベース・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 データベース・ブロック・サイズは、オンライン・トランザクション処理（OLTP）または複合的作業の環境では小さめ（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薦めします。

ログ・アーカイブ・バッファのチューニング

バッファの割当て量を大きくして、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。ただし、設定を大きくしすぎると、システム・パフォーマンスが全体的に低下することがあります。`LOG_ARCHIVE_BUFFER_SIZE` 初期化パラメータは 128KB まで増やすことができます。`LOG_ARCHIVE_BUFFER_SIZE` パラメータのデフォルト値は、4KB です。

このパラメータをチューニングするときは、通常のデータベース・アクティビティの全体的なパフォーマンスが急激に低下しないように注意してください。アーカイブ・プロセスのパフォーマンスは最大 20 パーセント向上させることができます。環境によっては、それ以上の向上が見られることもあります。

I/O バッファおよび SQL*Loader

SQL*Loader ダイレクト・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、データを高速でロードするときは、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。通常は、バッファの数を増やすことにより、CPU 使用率を上げることができ、スループット全体も向上します。

選択したバッファの数（SQL*Loader `BUFFERS` パラメータで設定）によって、利用できるメモリーの容量や CPU 使用率の上昇率が変わってきます。`BUFFERS` パラメータのファイル処理オプション文字列の調整については、『Oracle9*i* データベース・ユーティリティ』を参照してください。

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わります。

> **参照：** SQL*Loader ユーティリティの概要については、『Oracle9*i* データベース・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、大きな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、このパラメータの値を 1MB 以上に設定する必要があります。

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（ページングと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。すべてのディスクに I/O 負荷を均等に分散させます。

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）を利用して、複数のディスクにデータをストライプ化すれば、ディスクの競合を軽減させることができます。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込みおよび書込みのパフォーマンスを向上させることです。LVM で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、パフォーマンスが全体的に向上します。

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。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が最小要件です。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 〜 128KB の範囲で 2 の累乗を指定できます。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 データベース・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、使用されるディスク・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` 属性を値 1 に設定します。

ストライプ化論理ボリュームのパラメータ（推奨）

MINPGAHEAD、MAXPGAHEAD、MAXFREE の各パラメータを調整して、順次 I/O のスループットを最大限に上げるには、vmtune コマンドを使用します。推奨される設定値については、『AIX Performance Tuning Guide』または『AIX5L Performance Management Guide』を参照してください。

filemon、sar、iostat などのパフォーマンス・ツールを使用して、高い I/O アクティビティを持つディスクを識別し、複数のディスク・ドライブ間で I/O アクティビティが均等に分散されるようにしてください。

その他の考慮事項

LVM を利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応じて大きく変わります。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でも、かなりのパフォーマンス向上を期待できます。

ジャーナル・ファイル・システムを使用した場合と RAW パーティションを使用した場合の相違

ジャーナル・ファイル・システムと RAW パーティションのどちらを使用するかを決定するときは、次のことに考慮してください。

- ファイル・システムは、その実装とともに、継続的に更新されています。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方が、I/O パフォーマンスが向上します。
- 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 パーティションをベースにした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。

パフォーマンスの向上度は、作業負荷の I/O 特性によって大きく変わります。

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に比べて、データベース・ファイルの管理や保守が容易になります。しかし、現時点での JFS および JFS2 では、バッファに対する読込みおよび書込みだけがサポートされています。つまり、Oracle バッファ・キャッシュや SGA ではなく、I/O サブシステムに対してデータが送受信され、そのたびに AIX ファイル・バッファ・キャッシュが追加作成されます。この現象は、ジャーナル・ファイル・システムを使用した場合の大きな短所で、I/O サブシステムが頻繁に実行されるほど、その短所も大きくなります。

また、頻繁にロードされるシステムでは、AIX ファイル・バッファ・キャッシュの競合も大きなボトルネックになります。次の問題が発生します。

- pagedaemon コマンドの作業量が増加します。このコマンドは、AIX ファイル・バッファ・キャッシュのデータをディスクにフラッシュします。

- AIX ファイル・バッファ・キャッシュに割り当てられるメモリーが増加します。

使用しているサイトに、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 デバイス / ボリュームを使用してください。RAW ディスク・パーティションがフォーマット済の場合は、データ・ファイルのサイズをパーティションのサイズにできるだけあわせて無駄な領域をなくします。頻繁に使用されるデータベース構成では、RAW デバイスを使用します。

> **注意：** Oracle9*i* Real Application Clusters では、データベース・ファイルに対して、ジャーナル・ファイル・システムではなく RAW デバイスを使用する必要があります。

ジャーナル・ファイル・システムから RAW デバイスへの移動

すべてのデータを手動で再ロードしないで、ジャーナル・ファイル・システムから RAW デバイスに移動するには、`root` ユーザーとして次のコマンドを入力します。

```
# dd if=old_JFS_file of=new_raw_device seek=1 bs=4k
```

次のガイドラインに従ってください。

- RAW デバイスの最初の 4KB のブロックはスキップします（AIX 論理ボリューム・マネージャで使用するため）。

> **注意：** RAW デバイスがハッシュ共有ディスク（HSD）である場合は、最初の 4KB のブロックをスキップする必要はありません。

- 領域を無駄にしないように RAW デバイスのサイズを決定します。RAW デバイスは、既存の JFS または JFS2 ファイルよりも数ブロック大きくする必要があります。
- RAW パーティションに対する権限を設定します。
- データ・ファイルの名前を変更します。

順次読込みのパフォーマンスがわずかに低下することがあります。アプリケーションが全表スキャンを頻繁に行っている場合は、全表スキャンに対するサーバーのパフォーマンスが低下することがあります。全表スキャンされる表が、AIX ファイル・バッファ・キャッシュに格納されないためです。

メモリーが不足している構成では、AIX ファイル・バッファ・キャッシュのサイズを変更する必要があります。

AIX 環境では、RAW デバイスの名前が変更されることがあるため、できるだけシンボリック・リンクを使用してください。たとえば、ディスクの周辺装置を再構成したときや、データベースを新しいハードウェア構成に移動したときに、RAW デバイスの名前が変更されることがあります。詳細は、IBM のドキュメントを参照してください。

参照： RAW デバイスの使用に関する操作上の問題については、『Oracle9*i* for UNIX Systems インストレーション・ガイド』を参照してください。

ジャーナル・ファイル・システムおよび RAW デバイスの同時利用

AIX では、一部のデータ・ファイルにジャーナル・ファイル・システムを選択し、残りのデータ・ファイルに RAW パーティションを選択することができます。非同期 I/O では、RAW パーティションおよびジャーナル・ファイル・システムが同時に使用されます。データベース・オブジェクトに対する I/O アクティビティの特性をあらかじめ認識している場合は、特定のオブジェクトに対応するデータ・ファイルを、LVM（利用できる場合）を適用したジャーナル・ファイル・システムまたは RAW パーティションに配置することができます。

パフォーマンスが向上するかどうかは、作業負荷の特性やディスクまたはファイル・システム構成に大きく左右されます。

非同期 I/O の使用

Oracle9*i* 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。AIO は、複数の I/O を組み合せて I/O サブシステムのスループットを向上させます。AIO を効果的に使用するには、複数のディスク間にデータを有効に分散する必要があります。

LVM とストライプ機能を使用すると、AIO の効果がさらに高くなります。LVM は、複数のディスク・スピンドルにデータをストライプ化し、ディスクの競合を軽減しています。AIO を LVM と併用すると、RDBMS のパフォーマンスが大幅に向上します。

AIX バージョン 4 以上では、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（`kproc`）が各要求を管理します。カーネル・サーバー・プロセスは、FastPath が無効になっている仮想共有ディスク（VSD）や HSD による I/O でも使用されます。デフォルトでは、FastPath は有効になっています。`kproc` サーバーの数によって、システムで同時に実行できる AIO 要求の数が決まるため、ファイルシステムを使用して Oracle9*i* のデータファイルを格納するときに、`kproc` プロセスの数をチューニングすることが重要です。

注意： AIO FastPath が有効になっている（デフォルト）VSD や HSD で AIO を使用する場合は、最大バディ・バッファ・サイズを 128KB 以上に設定する必要があります。

サーバー数を設定するには、次のどちらかのコマンドを使用します。

- `smit aio`
- `chdev -l aio0 -a maxservers='`*m*`' -a minservers='`*n*`'`

> **参照：** システム管理インタフェース・ツール（SMIT）については、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` の各コマンドについては、man ページを参照してください。

最小値には、システム・ブート時に起動されるサーバーの数を指定します。最大値には、多数の要求が同時に送信されたときに、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、ファイルだけに適用され、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、これらの値は通常、Oracle サーバーの実行には低すぎます。次の値を設定することをお薦めします。

パラメータ	値
MINSERVERS	MAXSERVERS パラメータの値を 2 で除算して求めた商、またはアクセスされるディスク数を 10 で除算して求めた商のうち、どちらか大きい値まで増やすことができます。
MAXSERVERS	システム上の AIO サーバーに対する非同期要求の数によって変わります。推奨される初期値は 10 です。
MAXREQS	8192

MAXSERVERS パラメータの値が低すぎると、次のエラー・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、稼動している AIO サーバーの数よりも大きくします。稼動している AIO サーバーの数を表示するには、`root` ユーザーとして次のコマンドを入力します。

```
# pstat -a | grep -c aios
```

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ます。パラメータの変更は、システムが再起動されるときに行われます。

I/O スレーブ

I/O スレーブは、I/O だけを実行する特別なプロセスで、非同期 I/O が利用可能であるかどうかに関係なく動作できます。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 スレーブの動作を制御するには、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_ASYNCH_IO	TRUE/FALSE	TRUE
TAPE_ASYNCH_IO	TRUE/FALSE	TRUE
BACKUP_TAPE_IO_SLAVES	TRUE/FALSE	FALSE
DBWR_IO_SLAVES	0 〜 999	0
DB_WRITER_PROCESSES	1 〜 20	1

非同期 I/O が必要でなかったり、使用できない場合があります。この表に示されている最初の 2 つのパラメータ（DISK_ASYNCH_IO および TAPE_ASYNCH_IO）によって、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 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_ASYNCH_IO または TAPE_ASYNCH_IO パラメータが FALSE の場合に、0 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（DBWR）・プロセスがボトルネックになります。AIX 上での DBWR_IO_SLAVES パラメータの最適値は、4 です。

DB_WRITER_PROCESSES パラメータには、インスタンスに対するデータベース・ライター・プロセス数の初期値を指定します。DBWR_IO_SLAVES パラメータを使用した場合は、DB_WRITER_PROCESSES パラメータの設定に関係なく、使用されるデータベース・ライター・プロセスの数は 1 つだけです。

DB_FILE_MULTIBLOCK_READ_COUNT パラメータの使用

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 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、17 以上の値を使用しても、通常はそれ以上パフォーマンスは向上しません。

このパラメータに設定する値は、DB_BLOCK_SIZE パラメータの値との積が LVM ストライプ・サイズより大きくなるようにしてください。このような値を設定すると、使用できるディスク数が増えます。

RAID 機能の使用

RAID 5 を使用すると、順次読込みのパフォーマンスは向上しますが、書込みのパフォーマンスは全体的に低下します。RAID 5 は、書込みが頻繁に発生しない作業負荷に対して使用することをお薦めします。RAID 5 に対して書込みを頻繁に行うと、RAID を使用しない環境に比べて、パフォーマンスが低下することがあります。

RAID 0 および 1 を使用すると、一般にパフォーマンスが向上します。ハードウェア・レベルでストライプ化およびミラー化が導入され、AIX または Oracle レベルよりも効率的なためです。RAID 7 を使用すると、RAID 0 ～ 6 よりも読込みおよび書込みのパフォーマンスが多少向上します。

後書きの使用

後書き機能を使用すると、書込み I/O がパーティションのサイズまで蓄積されます。書込み I/O が蓄積されることによって、I/O 処理数が減るため、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を制限し、ディスクの断片化が最小限に抑えられます。指定された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 つの 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。

```
# vmtune -c 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。

```
# vmtune -c 0
```

順次ファイルの先読みのチューニング

仮想メモリー・マネージャ（VMM）は、順次ファイルのページに対する要求を予測します。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、プログラムはすぐにデータを利用できます。次の 2 つの 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`

  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。

- `MAXPGAHEAD`

  順次ファイルから VMM が先読みする最大ページ数。

`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、`vmtune` 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、次のコマンドを入力します。

```
# vmtune -r 32 -R 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... などに設定します。

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 メカニズムのことです。このメカニズムを利用すれば、ディスク I/O が頻繁に発生するプロセスによって CPU が飽和状態に達するのを回避できます。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込みを実行すると、スリープします。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。適切な水位標を決定するには、テストと検証が必要です。水位標はパフォーマンスに影響するので、設定するときは注意してください。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あまり効果はありません。

ディスク形状の考慮事項

AIX では、ディスク上の論理ボリュームの配置を、ある程度制御することができます。ディスク・アクティビティの大きい論理ボリュームを近くに配置すると、ディスクのシーク時間を短縮できるため、パフォーマンスが全体的に向上します。

リモート I/O 操作の最小化

SP アーキテクチャ上で動作する Oracle9*i* Real Application Clusters は、様々なノードのすべてのインスタンスからアクセスできる共通記憶域として、VSD または HSD を使用します。I/O 要求が VSD に対して発行されたときに、VSD がノードのローカル論理ボリューム上にある場合は、ローカル I/O が実行されます。ローカルでない VSD に対して発行された I/O 通信量は、ネットワーク通信レイヤーを介して処理されます。

パフォーマンスを向上させるには、リモート I/O をできるだけ最小限に抑えることが重要です。各インスタンスの REDO ログは、ローカル論理ボリューム上にある VSD に配置する必要があります。更新および挿入が頻繁に発生する場合は、ローカル論理ボリュームに配置された VSD 上に、各インスタンスのプライベート・ロールバック・セグメントを割り当てる必要があります。

各セッションでは、一時表領域を 1 ユーザーにつき 1 つだけ割り当てることができます。各一時表領域には、各ノードのローカル・データ・ファイルを 1 つ以上配置する必要があります。

リモート I/O が最小限に抑えられるように、アプリケーションおよびデータベースの設計（アプリケーションとデータベースのパーティション化など）には注意が必要です。

VSD キャッシュ・バッファ

次の理由から、通常の状況では VSD キャッシュ・バッファ（NOCACHE）を使用しないでください。

- VSD LRU キャッシュ・バッファは、より効果的に使用するために、確保されたカーネル・メモリーを使用します。
- このキャッシュ・バッファが使用可能になっていると、バッファが物理的にデータを読み込むたびに、キャッシュ・ブロックを検索して、ページをオーバーラップし、キャッシュ・バッファにデータをコピーするというオーバーヘッドが発生します。

VSD のパフォーマンスを調べるには、`statvsd` コマンドを使用します。`statvsd` コマンドによって、バディ・バッファを待機している要求がキューに常に表示される場合は、バディ・バッファをそれ以上追加しないでください。かわりに、切替え送信プールのサイズを増やしてください。

```
# /usr/lpp/ssp/css/chgcss -l css0 -a spoolsize=new_size_in_bytes
```

送信プールのサイズを増やした場合は、mbuf パラメータの上限値も大きくする必要があります。

```
# /etc/no -o thewall=new_size_in_kbytes
```

注意： 指定できる最大値は、64MB です。

---

mbuf パラメータの上限値は、ネットワーク・バッファに使用できるメモリーの最大量です。送信プールおよび受信プールの現在のサイズを調べるには、次のコマンドを入力します。

```
$ /usr/sbin/lsattr -El css0
```

参照： IBM の Web アドレスについては、Oracle9*i* for AIX-Based 4.3.3 Systems（64-bit）のリリース・ノートを参照してください。

CPU のスケジューリングおよびプロセスの優先順位

プロセスの競合は、CPU でも発生します。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ことがあります。

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 ミリ秒です。このタイム・スライスを変更するには、`schedtune` コマンドを使用します。ただし、このコマンドを使用するときは注意が必要です。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え率の平均が低い場合に、コンテキスト切替え率も低くなります。その結果、プロセスのコンテキスト切替え率に消費される 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ことがあります（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ときは、次のコマンドを入力して、タイム・スライスを長くすることができます。

```
# /usr/samples/kernel/schedtune -t n
```

たとえば、*n* の値に 0 を設定すると、タイム・スライスは 10 ミリ秒（ms）になり、1 を設定すると 20ms になり、2 を設定すると 30ms になります。

SMP システム上でのプロセッサ・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 バージョン 4 以上ですべての機能を利用できます。

プロセッサ・バインディングには、次の特長があります。

- 優先順位の高いアプリケーションに、比較的大きな割合の CPU 時間を割り当てられます。
- 長期間にわたってプロセス・コンテキストを保持できます。

AIX 上では、プロセッサ・バインディングは自動化されていません。マルチプロセッサ・システムでは、`bindprocessor` コマンドを使用して、プロセスをプロセッサに明示的にバインドする必要があります。`root` ユーザーまたは Oracle ソフトウェア所有者だけが、Oracle プロセスをプロセッサにバインドできます。子プロセスは、プロセッサ・バインディングを継承します。

Oracle バックグラウンド・プロセス（データベース・ライター・プロセスは除く）は複数のプロセッサにバインドし、データベース・ライター・プロセスには特定のプロセッサを割り

当てることをお薦めします。データベース・ライターを実行するプロセッサを確保できる上、CPU が飽和した場合にデータベース・ライター・プロセスを任意のプロセッサに移行することができます。

注意： プロセッサ・バインディングは、複雑な操作なので、注意して行う必要があります。プロセッサにバインドされているプロセスは、別のプロセッサが解放されている場合でも、そのプロセッサに移行することはできません。別のプロセッサに移行すると、アプリケーションのパフォーマンスが低下する場合があります。プロセッサ・バインディングは、負荷が分散された同種のアプリケーションで構成される環境に適しています。

プロセッサに対するプロセスのバインディングは、排他的ではありません。このプロセッサは、他のプロセスの実行に解放されています。

ネットワークのクライアントおよびサーバー環境でのプロセッサ・バインディング

Oracle クライアント・プロセスが Oracle Net Services リスナーを介して Oracle サーバー・プロセスに接続されているときは、リスナー・プロセスをバインドすることにより、サーバー・プロセスをプロセッサに簡単にバインドできます。その後リスナーによって生成された Oracle サーバー・プロセスは、すべて同一のプロセッサにバインドされます。

複数のリスナーを起動して、専用のポートでプロセスを待機させることもできます。`$ORACLE_HOME/network/admin/listener.ora` ファイルをカスタマイズして、リスナーごとにいくつかの行を追加する必要があります。まず、サーバー側で複数のリスナーを起動します。次に、クライアント側の `tnsnames.ora` ファイルをカスタマイズします。クライアントつまりアプリケーションごとに異なるポートに接続し、それぞれ独自のリスナーでプロセスを待機させます。たとえば、次の手順に従って、`listener.ora` ファイルを変更します。2 つのリスナー（L1 および L2）を指定し、それぞれポート 1521 および 1522 でプロセスを待機します。

1. `listener.ora` ファイルを次のように変更します。

```
L1 =
  (ADDRESS_LIST =
     (ADDRESS= (PROTOCOL= TCP)(Host= nowhere)(Port= 1521))
  )
SID_LIST_L1 =
  (SID_LIST =
    (SID_DESC =
      (ORACLE_HOME= /oracle)
      (SID_NAME = ordb)
    )
  )
L2 =
```

```
  (ADDRESS_LIST =
      (ADDRESS= (PROTOCOL= TCP)(Host= nowhere)(Port= 1522))
  )
SID_LIST_L2 =
  (SID_LIST =
    (SID_DESC =
      (ORACLE_HOME= /oracle)
      (SID_NAME = ordb)
    )
  )
```

2. 2 つのリスナーを起動します。

```
$ lsnrctl start L1
$ lsnrctl start L2
```

3. 2 つのリスナーのプロセス ID を決定します。

```
$ ps -ef | grep tnslsnr
```

4. リスナー・プロセスを特定のプロセッサにバインドします。

```
$ bindprocessor process_id_for_L1 id_for_process1
$ bindprocessor process_id_for_L2 id_for_process2
```

たとえば、*id_for_process1* は 0、*id_for_process2* は 1 です。

この方法を使用すると、特定のクライアントとの通信を処理する Oracle サーバー・プロセスはすべて、同一のプロセッサで実行されます。

ローカル環境でのプロセッサ・バインディング

プロセッサ・バインディングの操作は、クライアントおよび Oracle サーバーがパイプ・ドライバを 2 タスクで使用して同一コンピュータで稼動しているときは、より複雑になります。サーバー・プロセスごとにプロセス ID を決定し、プロセッサに手動でバインドする必要があります。管理の負荷が大きくなるため、サーバーのプロセス存続期間が長くない限り、お薦めしません。

Oracle プロセスのプロセッサ・バインディングは、同一システムで実行されている他のアプリケーションのパフォーマンスを低下させることがあります。このため、チューニングや監視には注意が必要です。

UDP のチューニング

Oracle9*i* Real Application Clusters では、プロセス間通信にユーザ・データグラム・プロトコル（UDP）を使用します。UDP カーネルの設定をチューニングすれば、Oracle のパフォーマンスを向上させることができます。AIX 上でカーネルの UDP バッファを変更するには、`udp_sendspace` および `udp_recvspace` パラメータを変更します。

- `udp_sendspace` パラメータの値を、最大値の 65536 に設定します。
- `udp_recvspace` パラメータの値は、`sb_max` パラメータの値より小さくします。

`udp_recvspace` パラメータの値は、`udp_sendspace` パラメータの値の 10 倍以上に設定する必要があります。UDP がパケットをアプリケーションに送信できない場合、別のパケットが着信することがあるためです。

`udp_recvspace` パラメータが適切かどうかを調べるには、次のコマンドを入力します。

```
$ netstat -s | grep "socket buffer overflows"
```

オーバーフローの数がゼロでない場合は、`udp_recvspace` パラメータの値を大きくします。

> **参照：** AIX チューニング・パラメータの詳細は、『AIX Performance Tuning Guide』または『AIX5L Performance Management Guide』を参照してください。

RAW デバイスのバックアップ

RAW デバイスのバックアップを作成するには、`dd` コマンドを使用します。RAW デバイスの最初の 4KB は、AIX 論理ボリューム・マネージャによって使用されるため、上書きしないでください。ブロックサイズを 4KB より大きくすると、デバイスのバックアップが速くなります。テープへのバックアップに最適なブロックサイズは、256KB です。

RAW デバイスのバックアップをテープに作成するには、次のようなコマンドを入力します。

```
$ 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
```

注意： HSD の場合は、最初の 4KB をスキップしないでください。前述の 3 つのコマンドではなく、次のコマンドを使用して RAW デバイスをリストアしてください。

```
$ dd if=/dev/remt0.1 of=/dev/raw_device bs=256K
```

Oracle9*i* 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込み一貫性（MWC）を無効にした場合、Oracle9*i* のクラッシュ・リカバリ・プロセスではシステム・クラッシュ後のリカバリにミラー復元を使用します。このミラー復元プロセスを実行すると、データベースの不整合や破損を防ぐことができます。

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。次に、以下のどちら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。
- すべてのコピーの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 では、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ミラー復元は必要ありません。MWC によってすべてのコピーの同期が保証されているからです。

以前のリリースの Oracle9*i* をアップグレードしているときにシステムがクラッシュしたときに、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` コマンドを入力して、ミラー化 LV を同期化してから、Oracle サーバーを起動してください。ミラー化 LV を同期化しないでサーバーを起動すると、LV コピーからデータが正しく読み取れ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` コマンドを入力してから、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。

B

Oracle9*i* for HP 9000 Series HP-UX の チューニング

この付録では、Oracle9*i* for HP 9000 Series HP-UX のチューニングについて説明します。

- 64 ビットの Oracle インスタンス用の HP-UX 共有メモリー・セグメント
- HP 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

64 ビットの Oracle インスタンス用の HP-UX 共有メモリー・セグメント

64 ビットの Oracle サーバーは、データベース・インスタンスを作成するときに、使用可能な共有メモリーを HP-UX `SHMMAX` カーネル・パラメータの値で割って、共有メモリー・セグメントを作成します。たとえば、1 つの Oracle インスタンスに使用できる共有メモリーが 64GB で、`SHMMAX` パラメータの値が 1GB である場合、Oracle サーバー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。

テストでは、1 つの Oracle インスタンスに対して共有メモリー・セグメントが 7 つ以上作成されると、パフォーマンスが低下しました。これは、Oracle サーバーがインスタンスを作成するときに、各共有メモリー・セグメントに一意のプロテクション・キーを割り当てるからです。PA-RISC プロセッサには、共有メモリー・セグメントに使用できるプロテクション・キーが 6 つあります。システムの共有メモリー・セグメントが 7 つ以上ある場合、HP-UX オペレーティング・システムはプロテクション・キー障害を表示します。

`SHMMAX` パラメータ値をシステムで使用できる物理メモリーの量に設定するようにしてください。こうしておけば、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。

システム上のアクティブな共有メモリー・セグメントを表示するには、次のコマンドを入力します。

```
$ ipcs -m
```

Oracle サーバーがそのインスタンスに対してセグメントを 7 つ以上作成する場合は、`SHMMAX` カーネル・パラメータの値を大きくします。

参照： 推奨されるカーネル・パラメータの設定については、『Oracle9*i* for UNIX Systems インストレーション・ガイド』を参照してください。

HP SCHED_NOAGE スケジューリング・ポリシー

HP 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ルから除外されるため、Oracle のパフォーマンスが低下することがあります。HP 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・ポリシーと異なり、SCHED_NOAGE によってスケジュール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。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。テストでは、Oracle9*i* の OLTP 環境で SCHED_NOAGE ポリシーを使用すると、パフォーマンスが最大 10 パーセント向上しました。

SCHED_NOAGE ポリシーを意思決定支援（DSS）環境で使用しても、ほとんど効果はありません。DSS 環境ではリソースの競合がほとんど起きないためです。アプリケーション / サーバーの構成は環境ごとに異なるため、SCHED_NOAGE ポリシーがお使いの環境に適切かどうかをテストおよび検証する必要があります。

Oracle9*i* の SCHED_NOAGE の有効化

Oracle9*i* で SCHED_NOAGE スケジューリング・ポリシーを使用するには、Oracle ソフトウェア所有者が属しているグループ（`dba`）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、`root` ユーザーとして次のコマンドを入力します。

```
# setprivgrp dba RTSCHED RTPRIO
```

リブート後もこれらの権限を保持するには、`/etc/privgroup` ファイルを作成し（まだ作成していない場合）、次の行をそのファイルに追加します。

```
dba RTSCHED RTPRIO
```

`HPUX_SCHED_NOAGE` パラメータを各インスタンスの初期化ファイルに追加し、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。このパラメータの範囲は、HP-UX 11.0 では 153 ～ 255、HP-UX 11i では 178 ～ 255 です。パラメータの設定が範囲外にある場合、Oracle9*i* はそのパラメータを許容値に設定し、SCHED_NOAGE ポリシーにもその新しい値を設定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。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設定するようにしてください。

> **参照：** 優先順位ポリシーおよび優先順位の範囲については、HP 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。

軽量タイマーの実装

Oracle9*i* リリース 1（9.0.1）とは異なり、HP-UX システムの Oracle9*i* リリース 2（9.2.0.1.0）では、初期化パラメータ `TIMED_STATISTICS` が `TRUE` または `FALSE` のどちらに設定されていても、`gethrtime()` システム・ライブラリ・コールを使用します。これにより、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。

テストでは、初期化パラメータ `TIMED_STATISTICS` が `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 システムのパフォーマンスが最大 10 パーセント向上しました。また、初期化パラメータ `TIMED_STATISTICS` が `FALSE` に設定されているときに、`gethrtime()` システム・ライブラリ・コールを使用しても、Oracle9*i* リリース 2（9.2.0.1.0）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。

参照： Oracle9*i* に必要な HP パッチについては、『Oracle9*i* for UNIX Systems インストレーション・ガイド』を参照してください。

非同期 I/O

非同期 I/O 疑似ドライバを HP 上で使用すると、Oracle サーバーは 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するため、I/O オーバーヘッドが減少し、スループットが向上します。非同期 I/O 疑似ドライバは、HP 9000 Series 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。

MLOCK 権限

Oracle9*i* で非同期 I/O 操作を実行するには、Oracle ソフトウェア所有者が属しているグループ（`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、次の手順で行います。

1. `root` としてログインし、次のコマンドを入力します。

```
# setprivgrp dba MLOCK
```

2. リブート後も MLOCK 権限を保持するには、`/etc/privgroup` ファイルを作成し（まだ作成していない場合）、次の行をそのファイルに追加します。

```
dba MLOCK
```

注意： Oracle ソフトウェア所有者が MLOCK 権限を持っていない場合は、HP-UX の Oracle9*i* によって次のエラーを含むトレース・ファイルが生成されます。

```
Ioctl ASYNC_CONFIG error, errno = 1
```

非同期 I/O の実装

HP 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 RAW パーティションを使用する必要があります。非同期ディスク・ドライバを HP カーネルに追加および設定するには、System Administrator Management（SAM）ユーティリティを使用します。

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、次の手順で行います。

1. `root` としてログインし、次のコマンドを入力します。

   ```
   # 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. 「Actions」>「Modify Configurable Parameter」を選択します。
9. パラメータに新しい値を指定し、「OK」を選択します。

   `MAX_ASYNC_PORTS` パラメータは、設定可能な HP 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オープンしようとすると、エラーが表示されます。このエラーが発生すると、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 を実行している場合、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でき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. 次のオプションのどちらかを選択し、「OK」を選択します。

    - 「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、次の手順で行います。

1. 次のコマンドを入力して、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て、システムをリブートします。

```
# shutdown -r
```

HP 非同期デバイス・ドライバを使用して非同期 I/O 操作を使用するには、次の手順で行います。

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て、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て、`/dev/async` デバイス・ファイルが存在すること、およびその中にメジャー番号 101 が含まれてい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 orcle   oracle     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 UNIX 所有者と権限を付与します。Oracle ソフトウェア所有者と同じ名前と権限を割り当ててください。

Oracle ソフトウェア所有者が `oracle` の場合は、次のコマンドを入力します。

```
# /usr/bin/chown oracle:dba /dev/async
# /usr/bin/chmod 660 /dev/async
```

非同期 I/O の検証

非同期 I/O が機能していることを検証するには、次の手順で行います。

1. `DISK_ASYNCH_IO` 初期化パラメータを TRUE に設定します。
2. Oracle データベースを起動します。
3. 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
   ```

4. メイン・ウィンドウで、「Reports」>「Process List」を選択します。
5. 「Process List」ウィンドウで、データベース・ライター・プロセスを選択し、「Reports」>「Process Open Files」をクリックします。

   データベース・ライター・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。

6. 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 のどちらかがリストに含まれている場合は、データベース・ライター・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、データベース・ライター・プロセスは HP 非同期デバイス・ドライバを介して I/O を実行し、正しく機能しています。

SGA の非同期フラグ

Oracle9*i* for HP 9000 Series HP-UX では、HP 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 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。

Oracle9*i* 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 の Oracle9*i* には、真の非同期 I/O メカニズムが備わっています。つまり、以前に送信された 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、パラレル I/O プロセスの拡張性が保証されます。

Oracle8*i* リリース 8.1.6 以下では、Oracle サーバーは HP 非同期ドライバを使用して、共有メモリーから I/O 操作を実行していました。Oracle8*i* リリース 8.1.7 では、新しい HP 非同期ドライバを使用して、共有メモリーとプロセス専用領域の両方から I/O 操作を実行します。ただし、非同期ドライバを介して実行される I/O 操作は、非同期では実行できません。Oracle8*i* 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になるためです。ブロック待機が行われると、データベース・ライター・プロセスなど、一部の Oracle プロセスでは同期 I/O が実行されます。

C

Oracle9*i* for Linux Intel のチューニング

この付録では、Oracle9*i* for Linux Intel のチューニング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
- 非同期 I/O サポート

拡張バッファ・キャッシュのサポート

Oracle9*i* では、データベース・バッファ・キャッシュとして、4GB よりも大きいメモリーを割り当てて使用することができます。この項では、Linux の拡張バッファ・キャッシュ・サポートの制限事項と要件について説明します。

> **参照：** 拡張キャッシュ機能については、『Oracle9*i* データベース概要』を参照してください。

メモリー内ファイル・システム

Linux で拡張バッファ・キャッシュ・サポートを使用するには、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と同じかそれ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、Linux で 8GB の `shmfs` 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、`root` ユーザーとして次のコマンドを入力します。

```
$ mount -t shm shmfs -o size=8g /dev/shm
```

拡張バッファ・キャッシュ機能を有効にして Oracle9*i* 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である場合は、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と同じかそれ以上であることを確認します。

USE_INDIRECT_DATA_BUFFERS 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、`init`*`sid`*`.ora` ファイルで `USE_INDIRECT_DATA_BUFFERS` パラメータを `true` に設定します。これにより、Oracle9*i* は大きなバッファ・キャッシュを指定できます。

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

- DB_32K_CACHE_SIZE

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・サイズを指定してください。

制限事項

Linux の拡張バッファ・キャッシュ機能には、次の制限事項が適用されます。

- インスタンスの実行中にバッファ・キャッシュのサイズを変更することはできません。
- ブロック・サイズが標準でない表領域を作成または使用することはできません。

> **参照：** CREATE TABLESPACE コマンドで使用する標準のブロック・サイズについては、『Oracle9*i* SQL リファレンス』を参照してください。

非同期 I/O サポート

Oracle9*i* では、カーネルの非同期 I/O をサポートしています。この機能は、デフォルトで無効になっています。カーネルの非同期 I/O をサポートしていて、非同期 I/O の使用をオラクル社が認定しているシステムで Oracle9*i* を実行している場合は、次の手順に従って非同期 I/O サポートを有効にします。

1. `root` ユーザーとして、ディレクトリを `$ORACLE_HOME/rdbms/lib` ディレクトリに変更します。
2. 次のコマンドを入力します。

```
$ make -f ins_rdbms.mk async_on
```

> **注意：**「`/usr/bin/ld: cannot find -laio`」というエラーが表示された場合、そのシステムでカーネルの非同期 I/O はサポートされていません。次のコマンドを入力して、Oracle インスタンスを使用可能な状態にリストアする必要があります。
>
> ```
> $ make -f ins_rdbms.mk async_off
> ```

デフォルトでは、`init`*`sid`*`.ora` ファイルの `DISK_ASYNCH_IO` パラメータが `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する非同期 I/O を有効にするには、次の手順で行います。

1. すべての Oracle データ・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. `init`*`sid`*`.ora` ファイルの `FILESYTEMIO_OPTIONS` パラメータを asynch に設定します。

D

Oracle9*i* for Compaq Tru64 UNIX の チューニング

この付録では、Oracle9*i* for Compaq Tru64 UNIX のチューニングについて説明します。

- 複合 CPU システムのサポート
- Tru64 でのデータベース統計の収集
- Tru64 での Oracle9i Real Application Clusters
- 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

Oracle9*i* 指定配置最適化の有効化

Compaq 社の GS80、GS160、および GS320 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から構成されます。RAD は、高速インターコネクトで接続された疎結合 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トでは、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トが 1 つだけだった以前のサーバーと異なり、Compaq GS80、GS160 および GS320 サーバーでは、特定の CPU がその RAD 内のメモリーにアクセスするときやローカル I/O コントローラを使用するときの、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。インターコネクトが切り替わるため、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がありません。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ト階層を横断する必要があります。この場合、1 つの RAD 内でメモリー参照する場合と比べて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（Tru64 UNIX V5.1 以上で利用可能）を使用すれば、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。

Oracle9*i* では、GS80、GS160、GS320 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、GS80、GS160 および GS320 クラスのサーバーで使用できる階層型インターコネクトを活用しています。以前のサーバーでは、共有インターコネクトが 1 つだけだったため、指定配置最適化の利点を直接受けることも、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せん。したがって、Oracle9*i* 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。

指定配置最適化を実行するための要件

Oracle9*i* 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。

- Compaq GS80、GS160、GS320 AlphaServer など、ローカル・パフォーマンスを重視した Compaq システムを使用する必要があります。Oracle9*i* 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だけに効果があります。
- オペレーティング・システムは、Compaq Tru64 UNIX V5.1 以上でなければなりません。それより前のオペレーティング・システムでは、Oracle9*i* でプロセスとメモリーの指定配置に必要な機能がサポートされていません。

Oracle 指定配置最適化の有効化

Oracle 指定配置最適化を使用可能にするには、次の手順で行います。

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 サーバーを再リンクします。

```
$ 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 指定配置最適化を使用可能にしてから、非互換バージョンの Tru64 に変更する場合は、次の項の手順で Oracle 指定配置最適化を無効にしてください。

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、次の手順で行います。

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 サーバーを再リンクします。

```
$ cd $ORACLE_HOME/rdbms/lib
$ make -f ins_rdbms.mk numa_off
$ make -f ins_rdbms.mk ioracle
```

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 が同じ数の 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 サーバーは、システム上のすべての RAD にまたがって稼動することを想定します。

Oracle 初期化パラメータ

Oracle9*i* 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を自動的に調整します。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ときに発生する、共通のエラーが除去されます。

Tru64 UNIX システム・パラメータ

NUMA システムの利点をすべて実現するには、次の表に示すシステム・パラメータを設定する必要があります。

サブシステム	パラメータ	設定
ipc	ssm_threshold	0
ipc	shm_allocate_striped	1（デフォルト）
vm	rad_gh_regions[0] rad_gh_regions[1]... 以下同様	共有グローバル領域のサイズ（MB）をシステム上の RAD 数で割った値

Tru64 V5.1 の vm サブシステムには、rad_gh_regions パラメータが 63 個あります。システム上の RAD の数に応じて、これらのパラメータを設定してください。たとえば、システム（GS160）上に 4 つの RAD があり、SGA サイズが 10GB である場合は、rad_gh_regions[0]、rad_gh_regions[1]、rad_gh_regions[2]、rad_gh_regions[3] をそれぞれ 2500 に設定します。インスタンスを正常に起動させるには、この値をわずかに高く（2501 または 2502）設定する必要があります。

CPU とメモリーがオフラインの場合、Oracle9*i* は引き続き動作しますが、パフォーマンスは低下します。RAD が頻繁にオフラインになったり、等しくパーティション化できないと見なされる場合は、RAD ごとに 1 つのインスタンスを使用して、Oracle9*i* Real Application Clusters を実行することをお薦めします。Oracle9*i* Real Application Clusters を使用すれば、実際の RAD 構成に応じて、初期化パラメータを様々に組み合せて固有のインスタンスを構成できます。また、アプリケーションを全体的に利用可能にしたままで、特定のインスタンスを起動またはシャットダウンすることができます。

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・システムに設定すれば、パフォーマンスを向上させることができます。データベースに対して Oracle リスナー・プロセスを介して接続し、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、次のコマンドを入力します。

```
$ runon -r 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。

複合 CPU システムのサポート

Tru64 V5.1A 以上を使用する Compaq システムでは、異なる CPU 速度およびタイプを組み合せることができます。1 つの RAD 内の CPU はすべて、同じ速度およびキャッシュ・サイズでなければなりません。別の RAD には、異なる速度およびキャッシュ・サイズの 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 と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・ライターおよびログ・ライターのプロセスを低速 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・ライターおよびログ・ライターのプロセスを低速 CPU に配置しても、パフォーマンスはまったく変わらない場合があります。

CPU システムを組み合せることができると、ハードウェア資産を保護することができます。古い CPU を交換しなくても、より高速かつ強力な CPU をシステムに追加できるからです。Compaq 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： 複合 CPU システムのパフォーマンスは、最高速 CPU だけで構成された複合 CPU システムよりは低くなります。ただし、複合 CPU システムのパフォーマンスは、最低速 CPU だけで構成された複合 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については、Compaq 社にお問合せください。

Tru64 でのデータベース統計の収集

Oracle9*i* リリース 2（9.2.0.1.0）は、Tru64 UNIX V5.1 以上でのみ動作します。Compaq 社の Tru64 UNIX における 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 Oracle 操作の精度が高くなりました。これらの操作を行うと、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 サーバー内の問合せオプティマイザでは、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決定します。最適な問合せ方法を決定しているときに、データ・ディクショナリに格納されている統計と問合せオプティマイザによって算出された統計とが一致しない場合は、問合せオプティマイザは不正な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、Oracle8*i* リリース 8.1.7 以前から Oracle9*i* リリース 2（9.2.0.1.0）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。Oracle9*i* リリース 1（9.0.1）から Oracle9*i* リリース 2 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することができます。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。

> **参照：** データベース統計の収集の詳細は、『Oracle9*i* 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。

Tru64 での Oracle9*i* Real Application Clusters

ここでは、Tru64 の Oracle9*i* Real Application Clusters について説明します。

Reliable Data Gram

Reliable Data Gram（RDG）は、Tru64 TruCluster プラットフォームの IPC インフラストラクチャの 1 つです。RDG は、Oracle9*i* for Compaq Tru64 UNIX のデフォルトの IPC 方式で、Oracle9*i* Real Application Clusters 環境に最適化されています。

要件

RDG を使用するには、ノードがクラスタに所属し、メモリー・チャネルを介して接続する必要があります。また、表 D-1 に示すオペレーティング・システム・パラメータをノード全体に設定することをお薦めします。

表 D-1 RDG サブシステムのオペレーティング・システム・パラメータの設定

パラメータ	設定
`max_objs`	ノードあたりの Oracle プロセス数の 5 倍以上で、10240 または Oracle プロセス数の 70 倍以下。
`msg_size`	データベースの `DB_BLOCK_SIZE` パラメータの最大値以上。 Oracle9*i* は各表領域の様々なブロック・サイズに対応しているため、推奨値は 32768 です。
`max_async_req`	100 またはオペレーティング・システムのデフォルト値の、どちらか大きい値 **注意：**多くの場合、1000 以上の値を指定すると、パフォーマンスが向上します。
`max_sessions`	Oracle プロセス数と 2 の和以上
`rdg_max_auto_msg_wires`	0 に設定してください。

UDP IPC の有効化

Oracle9*i* for Compaq Tru64 UNIX では、RDG がデフォルトの IPC 方式です。Oracle9*i* Real Application Clusters オプションが使用可能になっている場合は、グローバル・キャッシュ・サービス（GCS）、グローバル・エンキュー・サービス（GES）、インタープロセッサ・パラレル問合せ（IPQ）、およびキャッシュ・フュージョンで RDG が使用されます。ユーザ・データグラム・プロトコル（UDP）IPC の実装も利用できますが、明示的に使用可能にする必要があります。

UDP IPC を使用可能にする場合は、事前に Oracle9*i* Real Application Clusters オプションを使用可能にしておく必要があります。Oracle9*i* Real Application Clusters オプションを使用可能にするには、Oracle Universal Installer を使用するか、次のコマンドを入力します。

```
$ cd $ORACLE_HOME/rdbms/lib
$ make -f ins_rdbms.mk rac_on
$ make -f ins_rdbms.mk ioracle
```

Oracle IPC ルーチンで UDP プロトコルを使用するには、Oracle 実行可能ファイルを再リンクする必要があります。次の手順を実行する前に、クラスタ内のインスタンスをすべてシャットダウンしてください。

UDP IPC を使用可能にするには、次のコマンドを入力します。

```
$ cd $ORACLE_HOME/rdbms/lib
$ make -f ins_rdbms.mk ipc_udp
$ make -f ins_rdbms.mk ioracle
```

UDP IPC を無効にし、Oracle9*i* Real Application Clusters のデフォルトの実装に戻るには、次のコマンドを入力します。

```
$ cd $ORACLE_HOME/rdbms/lib
$ make -f ins_rdbms.mk rac_on
$ make -f ins_rdbms.mk ioracle
```

CLUSTER_INTERCONNECTS 初期化パラメータ（旧 TRU64_IPC_NET）

Tru64 UNIX の Oracle9*i* リリース 2（9.2.0.1.0）では、TRU64_IPC_NET パラメータが CLUSTER_INTERCONNECTS パラメータに置き換えられています。CLUSTER_INTERCONNECTS パラメータには、デバイス名ではなく、インターコネクトの IP アドレスが必要です。CLUSTER_INTERCONNECTS パラメータを使用すると、システムに複数の IP アドレスを指定できます。Oracle9*i* Real Application Clusters の通信量は、指定されたすべての IP アドレス間で分散されます。

CLUSTER_INTERCONNECTS パラメータは、Oracle9*i* Real Application Clusters および UDP IPC が使用可能になっている場合にのみ有効です。これらを使用可能にすると、Oracle GCS、GES、IPQ などのすべての IPC 通信量にインターコネクトを指定できます。

CLUSTER_INTERCONNECTS パラメータは、Memory Channel インターコネクトがオーバーロードされているときに使用してください。CLUSTER_INTERCONNECTS パラメータを設定して、Oracle GCS、GES および IPQ 通信量を別のインターコネクトに分散すると、クラスタ全体の安定性とパフォーマンスが向上する場合があります。たとえば、すべての GCS、GES および IPQ IPC 通信量に対して、IP アドレスが 129.34.137.212 であるファイバー分散データ・インタフェース（FDDI）・ネットワーク・コントローラを使用するには、CLUSTER_INTERCONNECTS パラメータを次のように設定します。

```
CLUSTER_INTERCONNECTS=129.34.137.212
```

デバイスの IP アドレスを表示する場合は、/usr/sbin/ifconfig -a コマンドを使用します。このコマンドを実行すると、デバイス名と IP アドレスとのマップが表示されます。デバイスの IP アドレスを調べるには、次のコマンドを入力します。

```
$ /usr/sbin/ifconfig -a
fta0: flags=c63<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,SIMPLEX>
      inet 129.34.137.212 netmask fffffc00 broadcast 129.34.139.255 ipmtu 1500

lo0:  flags=100c89<UP,LOOPBACK,NOARP,MULTICAST,SIMPLEX,NOCHECKSUM>
      inet 127.0.0.1 netmask ff000000 ipmtu 4096

mc0:  flags=1100063<UP,BROADCAST,NOTRAILERS,RUNNING,NOCHECKSUM,CLUIF>
      inet 10.0.0.1 netmask ffffff00 broadcast 10.0.0.255 ipmtu 7000

sl0:  flags=10<POINTOPOINT>

tun0: flags=80<NOARP>
```

この例では、デバイス fta0: の IP アドレスは 129.34.137.212 で、デバイス mc0: の IP アドレスは 10.0.0.1 です。

CLUSTER_INTERCONNECTS 初期化パラメータを使用するときは、次の重要事項を覚えておいてください。

- CLUSTER_INTERCONNECTS パラメータは、UDP が IPC の実装として使用可能になっているときにのみ使用します。

- 同一データベースが複数のノード上に配置されている場合、その複数のインスタンスに指定されている IP アドレスは、同じネットワークに接続するネットワーク・アダプタに属している必要があります。この規則に従わない場合は、ノード間通信量がブリッジやルーターを通過し、2 つのノード間でパスが認識されないことがあります。
- `CLUSTER_INTERCONNECTS` パラメータは、インスタンス固有の初期化パラメータ・ファイルに指定します。同じネットワークに接続された異なるノード上のデバイスには異なる IP アドレスが指定されているため、`CLUSTER_INTERCONNECTS` パラメータを共通の初期化パラメータ・ファイルに指定しないでください。
- このパラメータに複数の IP アドレスを指定する場合は、同じデータベースのすべてのインスタンスに対して同じ順序でそれらの IP アドレスを列挙します。たとえば、ノード 1 にあるインスタンス 1 のパラメータには、alt0、fta0、mc0 の各デバイスの IP アドレスをこの順序で列挙し、ノード 2 にあるインスタンス 2 のパラメータには、同等のネットワーク・アダプタの IP アドレスを同じ順序で列挙する必要があります。
- 指定したインターコネクト・デバイスまたは IP アドレスが正しくなかったり、システムに存在しない場合、Oracle9*i* ではデフォルトのクラスタ・インターコネクト・デバイスが使用されます。Tru64 UNIX V5.1 では、デフォルトのデバイスは `mc0` です。Tru64 UNIX V5.1A 以上では、デフォルトのデバイスは `ics0` です。

  Oracle9*i* では、どのデバイスが使用されるかは確認しません。使用されるクラスタ・インターコネクト・デバイスの IP アドレスを調べるには、次の手順を実行します。

  1. 次のコマンドを入力します。

     ```
     $ /usr/sbin/clu_get_info
     ```

  2. このコマンドの出力で、`Hostname` パラメータに格納されているクラスタ・インターコネクトの IP 名と、`Cluster interconnect IP address` パラメータに格納されているクラスタ・インターコネクトの IP アドレスを確認します。次の例では、クラスタ・インターコネクトの IP 名は `server1` で、そのアドレスは `10.0.0.1` です。

     ```
     Information on each cluster member

     Cluster memberid = 1
     Hostname = server1.employee.records
     Cluster interconnect IP name = server1-mc0
     Cluster interconnect IP address = 10.0.0.1
     Member state = UP
     Member base O/S version = Compaq Tru64 UNIX V5.1 (Rev. 732)
     Member cluster version = TruCluster Server V5.1 (Rev. 389)
     Member running version = INSTALLED
     Member name = server1
     Member votes = 1
     csid = 0x20002
     ```

非同期 I/O のチューニング

Oracle9*i* for Compaq Tru64 UNIX 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薦めします。非同期 I/O を使用可能にするには、`DISK_ASYNCH_IO` パラメータを `TRUE` に設定します。

Oracle9*i* では、AdvFS ファイル・システム、クラスタ・ファイル・システム（CFS）または 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、いくつかのオペレーティング・システム・パラメータをチューニングする必要があります。

aio_task_max_num パラメータ

単一インスタンスの `aio_task_max_num` オペレーティング・システム・パラメータは、次の値よりも大きく設定してください。

- DBWR I/O 操作の最大数
- `DB_FILE_MULTIBLOCK_READ_COUNT` 初期化パラメータの値

DBWR I/O 操作の最大数は、デフォルトで 8192 です。

`aio_task_max_num` パラメータの設定は、単一ノードで複数の Oracle9*i* 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 以上に設定する必要があります。

`aio_task_max_num` オペレーティング・システム・パラメータを設定しない場合は、Oracle9*i* 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・ファイルとトレース・ファイルに格納されます。

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

ここでは、ダイレクト I/O およびコンカレント・ダイレクト I/O のサポートについて説明します。

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、Oracle9*i* には次の要件があります。

- Tru64 UNIX V5.1 以上（適切なパッチキットを適用）。

  参照： Tru64 のパッチキットについては、『Oracle9*i* for UNIX Systems インストレーション・ガイド』を参照してください。

- Oracle データ・ファイルが Tru64 UNIX AdvFS ファイル・システム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、Oracle9*i* インスタンスが動作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。ファイバー・チャネルにも接続する必要があります。物理的に接続しない I/O は、別のノードで処理する必要があります。

Tru64 UNIX V5.1 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。

クラスタ化されたシステム

Tru64 UNIX V5.1 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とんど継承します。

Tru64 UNIX V5.1 クラスタ化システム

CFS ファイル・システムは、現在ではコンカレント・ダイレクト I/O モデルに対応しているため、Tru64 UNIX V5.1 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。

ファイルのメタデータを変更すると（アクセスまたは変更日の延長、クローズ、変更など）、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する可能性があります。インターコネクトが飽和すると、CFS ファイル・システム上での 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 デバイスより低下することがあります。

複数インスタンスの要件（Oracle9*i* Real Application Clusters）

Oracle9*i* Real Application Clusters では、Oracle データ・ファイルを Tru64 AdvFS ファイル・システムに格納する必要があります。AdvFS ファイル・システムを使用するディスクは、Oracle インスタンスを実行しているすべてのコンピュータに物理的に接続する必要があります。ファイバー・チャネルにも接続する必要があります。物理的に接続しない I/O は、別のノードで処理する必要があります。

データベースがアーカイブ・モードで動作し、アーカイブ・ログがディスクに書き込まれる場合、書込み先の AdvFS ドメインは、REDO ログを保管しているインスタンスのノードにする必要があります。たとえば、3 つのノードで構成されるクラスタで、ノード（nodea、nodeb、nodec）ごとにインスタンスが 1 つずつ動作している場合、アーカイブ・ログが書き込まれる AdvFS ドメインは 3 つ（arcnodea、arcnodeb、arcnodec）です。これらのドメイン名の場所は、nodea、nodeb および nodec として、各インスタンスの `LOG_ARCHIVE_DEST` 初期化パラメータに指定する必要があります。

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・システムで動作している Oracle9*i* データベースは、RAW デバイスで動作している Oracle9*i* データベース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 Oracle9*i* データベースは、ダイレクト I/O サポートが無効になっている Oracle9*i* データベースと同等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、次の作業負荷属性によってパフォーマンスが低下する可能性があります。

- 読取りと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利用されるため、Oracle データ・ブロックが SGA にキャッシュされない
- UNIX バッファ・キャッシュ（UBC）のサイズが数メガバイト以上ある
- 全表スキャンによって、同じ表セットが繰り返しスキャンされる
- 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、前述のほとんどの属性を持つ作業負荷は 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。つまり、データのすべてをディスクから読み取る場合（ダイレクト I/O が有効になっている場合）に比べて、問合せがより早く返されます。

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取る問合せでも、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大きくなります。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属性を持っている場合は、ダイレクト I/O サポートを無効にすると、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。データを読み取るだけの問合せもあれば、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。

Oracle9*i* リリース 2（9.2.0.1.0）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。Oracle9*i* 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ない）は、Oracle9*i* リリース 2（9.2.0.1.0）では削除されています。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 で解釈される `FILESYSTEMIO_OPTIONS` 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。

参照： FILESYSTEMIO_OPTIONS 初期化パラメータの詳細は、『Oracle9*i* データベース・リファレンス』を参照してください。

DISK_ASYNCH_IO 初期化パラメータは、ファイル・システムまたは RAW デバイスのどちら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_ASYNCH_IO 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 パラメータの値に関係なく同期化されます。DISK_ASYNCH_IO パラメータは、デフォルトで TRUE に設定されています。

ファイルの断片化の防止

ダイレクト I/O とファイル割当てとの相互作用のため、ダイレクト I/O サポートを有効にして作成されたファイルは、細かく断片化される可能性があります。ファイルが細かく断片化されていると、特にバックアップやリカバリ時に、パフォーマンスが低下し、I/O エラーの発生につながる可能性があります。Oracle9*i* リリース 2（9.2.0.1.0）では、ファイルの作成や拡張時にダイレクト I/O サポートを一時的に無効にする方法で、この問題を解決しています。ダイレクト I/O サポートが有効になっている場合は、ファイルの作成 / サイズ変更操作が完了した後で、ダイレクト I/O サポートを有効にして、新しいファイルまたは拡張されたファイルが再オープンされます。ファイルのサイズ変更操作では、新しいファイル・サイズが現在のファイル・サイズよりも小さければ、ダイレクト I/O サポートは一時的に無効になりません。この場合、ファイル（エクステント）がすでに存在するため、断片化は行われません。

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、定期的に行われます。これらのタイミング機能は、Tru64 カーネルを呼び出すため、Oracle9*i* のパフォーマンスに影響する可能性があります。Tru64 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向上させることができます。

リアルタイム・クロックへのアクセスを有効にするには、次の手順で行います。

1. root としてログインします。
2. 次のコマンドを入力します。

```
# mknod /dev/timedev c 15 0
# chmod +r /dev/timedev
```

   Tru64 UNIX V5.1 以上が稼動するクラスタ化システムの場合は、これらのコマンドを各クラスタで入力します。Tru64 UNIX V5.0 以前が稼動するクラスタ化システムの場合は、これらのコマンドを各ノードで入力します。

   注意： 特殊ファイル /dev/timedev は、リブート後もシステム上に残ります。

3. Oracle9*i* インスタンスを再起動します。

   インスタンスの起動時にのみ、/dev/timedev ファイルが存在するかどうかがチェックされます。

この機能は、クラスタ内のすべてのインスタンス（すべてのノード）で有効にすることをお薦めします。

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お使いのシステムに関する専門知識が必要です。

Tru64 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、次の手順で行います。

1. Oracle9*i* Real Application Clusters 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てください。ただし、お使いのプラットフォームが、オラクル社が認定するクラスタ・ファイル・システムに対応している場合は、Oracle9*i* Real Application Clusters に必要なファイルをそのクラスタ・ファイル・システムに直接格納できます。

2. 空きディスク・パーティションの名前を調べます。

   空きディスク・パーティションは、Tru64 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ものです。

   - `/usr/sbin/mount` コマンドの実行時に、リストに表示されません。
   - スワップ・デバイスとして使用されません。
   - スワップ・パーティションをオーバーラップしません。
   - 他の Tru64 アプリケーション（Oracle9*i* サーバーの他のインスタンスなど）によって使用されません。
   - Tru64 ファイル・システムをオーバーラップしません。
   - 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ません。

   パーティションが空いているかどうかを確認するには、デバイス上のパーティションの開始位置とサイズからなる詳細なマップを出力し、空き領域を調べます。パーティションの中には、現在マウントされず、`/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 で開始していないことを確認してください。

   ---

3. Oracle9*i* サーバーで使用する RAW デバイスを設定します。

   まず、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 コマンドを入力して、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 -1a
```

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・ファイルやキャラクタ・ファイルの所有権を変更します。たとえば、次のように入力します。

```
# chown oracle /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だけ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、次のコマンドを入力します。

```
$ ls -Ll datafile
```

次のメッセージが表示されます。

```
crwxrwxrwx oracle dba datafile
```

注意： このシンボリック・リンクは、クラスタのノードごと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ます。

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 SQL コマンドを入力します。

注意： RAW パーティションに作成した 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より、64KB と 1 Oracle ブロック・サイズを足し合せた値以上小さくなければなりません。

```
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 データベースの表領域に追加するには、次のコマンドを入力します。

```
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 デバイスを設定することもできます。

SPIKE 最適化ツール

SPIKE 最適化ツール（Spike）は、Tru64 バイナリのパフォーマンスを向上させるパフォーマンス最適化ツールの 1 つです。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、Oracle9*i* サーバーのパフォーマンスが最大 23 パーセント向上しました。

SPIKE については、Tru64 のドキュメントを参照するか、次のどちらかのコマンドを入力してください。

- `man spike`
- `spike`

Oracle9*i* には、SPIKE V5.1（1.2.2.31.2.4 ADK、2001 年 2 月 22 日）以上が必要です。

注意： V5.1（1.2.2.31.2.4 ADK、2001 年 2 月 22 日）より前のバージョンの SPIKE を使用している場合は、Compaq 社に連絡してパッチキットを入手してください。

SPIKE のバージョンを調べるには、次のコマンドを入力します。

```
$ spike -V
```

Compaq 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 Oracle9*i* 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、「はじめに」の「Oracle サービスおよびサポート」を参照してください。

SPIKE の使用

ここでは、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 D-2 に、SPIKE の実行に必要なシステム・リソースを示します。

表 D-2 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
sysconfigtab ファイルの max-per-proc-address-space パラメータ	1024MB
sysconfigtab ファイルの max-per-proc-data-space パラメータ	1024MB
sysconfigtab ファイルの vm-maxvas パラメータ	1024MB

/etc/sysconfigtab ファイルのこれらのパラメータ値を設定するには、次の行を編集します。

```
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、次のように入力します。

```
% 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・コマンドを入力します。

```
% 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。しかし、Oracle9*i* に対して、spike コマンドを使ったすべての最適化を使用できるわけではありません。Oracle9*i* では、次の SPIKE 最適化フラグの実行が証明されています。

```
-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 Oracle9*i* 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。Oracle9*i* 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。Oracle9*i* が Oracle9*i* バイナリで証明されていない最適化を検出した場合、または Oracle9*i* バイナリを以前に OM（Compaq 社製の SPIKE の先行モデル）を使って最適化している場合は、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が失敗した場合は、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： Oracle9*i* リリース 2（9.2.0.1.0）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 を使って実行可能ファイルを最適化するには、次のどちら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、いくつかの手順を設定するだけで、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 を実行すると、静的 SPIKE のすべての最適化以外に、作業負荷関連の最適化がいくつか実行されます。フィードバック付き SPIKE では、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、スパイクされた 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薦めします。

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、操作は簡単ですが、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 によ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 に比べて複数の手順と多くの操作を必要とします。しかし、パフォーマンスを重視する場合は、余分の工数はかかりますが、この方法で最適化することをお薦めします。

フィードバック付き SPIKE を実行するには、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ルメントします。

   $ pixie -output oracle.pixie -dirname *dir* -pids *oracle_image*

   oracle_image は、元のイメージです。*dir* は、インストルメントされた実行可能プログラムがプロファイル中のデータ・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、後で必要となる `oracle.Addrs` 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、ピクシー・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。Oracle9*i* サーバーを使用しているときは、oracle.Counts.*pid* ファイルがいくつか作成されます。*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. 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。次のコマンドを入力してください。

```
$ ln -s oracle.orig oracle
```

7. 特定の oracle.Counts.*pid* ファイルを作業負荷の代表として見なすことができる場合は、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利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、いくつかの 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、次のコマンドを入力します。

```
$ 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・ディレクトリで利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てください。

```
$ 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て、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。

E

Oracle *inter*Media、Oracle Text、および Oracle Spatial のデモの実行

この付録では、Oracle9*i* *inter*Media、Oracle Text、および Oracle Spatial の各デモの実行について説明します。この章では、次の項目について説明します。

- Oracle interMedia
- Oracle Text
- Oracle Spatial

Oracle *inter*Media

Oracle *inter*Media は、いくつかのサービスを統合したスイート製品です。イメージ、オーディオ、ビデオ・データを格納、管理、および検索するためのサービスを提供し、Oracle9*i* の機能を拡張します。また、ロケーション・サービスや、マルチメディア・データの注釈サービスも提供します。Oracle *inter*Media は、次のコンポーネントから構成されます。

- Oracle interMedia Audio，Image，Video
- Oracle interMedia Annotator
- Locator

Oracle *inter*Media Audio， Image， Video

Oracle *inter*Media には、多数のスクリプトやサンプル・プログラムが含まれています。各スクリプトやサンプル・プログラムの詳細は、表 E-1 に示すディレクトリの README ファイルを参照してください。

表 E-1 Oracle *inter*Media のデモ・プログラム

Oracle *inter*Media のデモ	ディレクトリ
ORDAudio SQL	`$ORACLE_HOME/ord/aud/demo/`
ORDAudio Java	`$ORACLE_HOME/ord/aud/demo/java/`
ORDDoc SQL	`$ORACLE_HOME/ord/doc/demo/`
ORDDoc Java	`$ORACLE_HOME/ord/doc/demo/java/`
ORDImage OCI	`$ORACLE_HOME/ord/img/demo/`
ORDImage Java	`$ORACLE_HOME/ord/img/demo/java/`
ORDVideo SQL	`$ORACLE_HOME/ord/vid/demo/`
ORDVideo Java	`$ORACLE_HOME/ord/vid/demo/java/`
Code Wizard for PL/SQL Gateway	`$ORACLE_HOME/ord/http/demo/plsgwycw`
PL/SQL Web Toolkit Demonstration	`$ORACLE_HOME/ord/http/demo/plsqlwtk`
Java Servlet Photo Album	`$ORACLE_HOME/ord/http/demo/servlet`
Java Server Pages Photo Album	`$ORACLE_HOME/ord/http/demo/jsp`
Relational Interface	`$ORACLE_HOME/ord/im/demo/relintfc`

Oracle *inter*Media Annotator

Oracle *inter*Media Annotator には、変更および実行できるデモがいくつか含まれています。これらのデモの保存場所については、`$ORACLE_HOME/ord/Annotator/README.txt` ファイルを参照してください。

Locator

Oracle *inter*Media Locator には、変更および実行できるデモがいくつか含まれています。これらのデモは、`$ORACLE_HOME/md/demo` ディレクトリに入っています。

Oracle Text

Oracle Text のコード・サンプルについては、`$ORACLE_HOME/ctx/sample/api/index.html` ファイルを参照してください。

> **参照：** Oracle Text の詳細は、『Oracle Text リファレンス』を参照してください。

Oracle Spatial

Oracle Spatial については、『Oracle Spatial User's Guide and Reference』を参照してください。

F

Oracle Cluster Management Software for Linux

Oracle Cluster Management Software（OCMS）は、Oracle9*i* for Linux System で利用できます。この付録では、次の項目について説明します。

- 概要
- Watchdog デーモン
- Cluster Manager
- OCMS の起動
- Watchdog デーモンおよび Cluster Manager の起動オプション

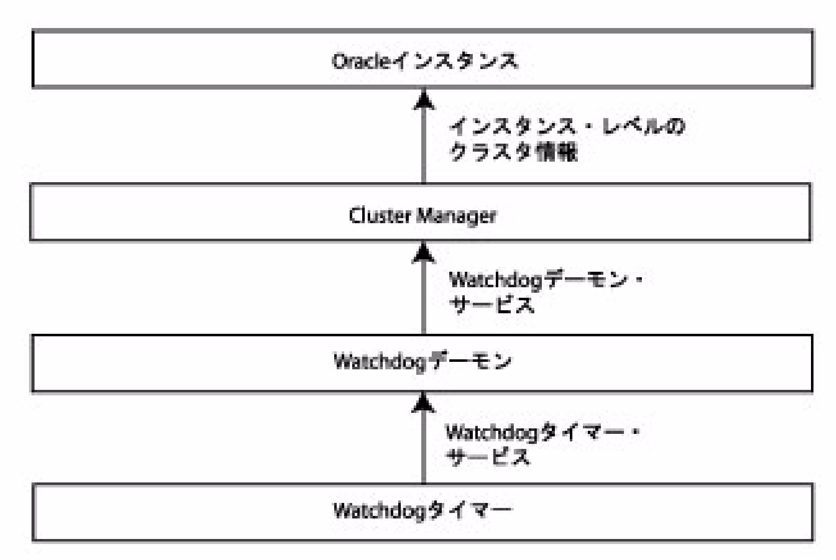
概要

OCMS は、Oracle9*i* Enterprise Edition for Linux に組み込まれています。OCMS を使って、クラスタ・メンバーの管理、クラスタのグローバル・ビュー、ノードの監視、およびクラスタの再構成を行います。OCMS は、Linux 上の Oracle9*i* Real Application Clusters のコンポーネントの 1 つで、Oracle9*i* Real Application Clusters を選択すると自動的にインストールされます。OCMS は、次のコンポーネントから構成されます。

- Watchdog デーモン
- Cluster Manager

図 F-1 に、Watchdog デーモン・サービスと Cluster Manager の関係を示します。

図 F-1 Oracle インスタンスと OCMS のコンポーネント



Watchdog デーモン

Watchdog デーモン（`watchdogd`）は、ソフトウェアとして実装された Watchdog タイマーを使用して選択されたシステム・リソースを監視し、データベースの破損を防ぎます。

Watchdog タイマーは、Linux カーネルの機能の 1 つです。Watchdog デーモンは、Oracle9*i* Real Application Clusters の一部です。

Watchdog デーモンは、Cluster Manager を監視し、定義された時間間隔で通知を Watchdog タイマーに渡します。Watchdog タイマーの一部の動作は、Linux カーネルの CONFIG_WATCHDOG_NOWAYOUT 構成パラメータによって管理されています。

Oracle9*i* Real Application Clusters を使用している場合は、CONFIG_WATCHDOG_NOWAYOUT 構成パラメータの値を `Y` に設定する必要があります。Watchdog タイマーは、Oracle インスタンスまたは Cluster Manager の障害を検出すると、そのインスタンスをリセットしてデータベースの破損を回避します。

CONFIG_WATCHDOG_NOWAYOUT パラメータの設定方法については、Linux カーネル・ソース・コードの `/usr/src/linux/Documentation/configure.help` ファイルを参照してください。Watchdog デバイスの詳細は、Linux カーネル・ソース・コードの `/usr/src/linux/Documentation/watchdog.txt` ファイルを参照してください。

Cluster Manager

Cluster Manager は、クラスタのノードおよび Oracle インスタンスの状態を管理します。Cluster Manager のプロセスは、Oracle9*i* Real Applications Clusters の各ノードで実行されます。各ノードには、Cluster Manager が 1 つずつ割り当てられます。Oracle9*i* Real Application Clusters では、ノードごとの Oracle インスタンスの数に制限はありません。Cluster Manager は、ノード間で次の通信チャネルを使用します。

- プライベート・ネットワーク
- 共有ディスク上の定数パーティション

クラスタを通常に操作している場合、クラスタの各ノードの Cluster Manager は、プライベート・ネットワークで送信されたハートビート・メッセージを介して相互に通信を行います。定数パーティションは、ハートビート・メッセージが失敗した場合の緊急通信チャネルとして使用されます。ハートビート・メッセージは、次の理由で失敗する可能性があります。

- ノード上の Cluster Manager が終了する
- プライベート・ネットワークに障害が発生する
- ノードにかかる負荷が非常に大きい

Cluster Manager は、定数パーティションを使用して、障害の理由を調べます。各ノードの Cluster Manager は、定数パーティション上の特定のブロックを定期的に更新します。他のノードでは、各ブロックのタイムスタンプを確認します。ノードからのメッセージが着信し

ていないときでも、対応する定数パーティションに現在のタイムスタンプが記録されている場合、このノードと他のノード間のネットワーク・パスに障害が発生しています。

各 Oracle インスタンスは、ローカルの Cluster Manager に登録されます。Cluster Manager は、ローカルの Oracle インスタンスの状態を監視し、この情報を他のノードの Cluster Manager に送信します。いずれかのノードで Oracle インスタンスが失敗すると、次のイベントが発生します。

1. Oracle インスタンスが失敗したノードの Cluster Manager は、その障害について Watchdog デーモンに通知します。
2. Watchdog デーモンは、Watchdog タイマーに対して、障害の発生したノードをリセットするように要求します。
3. Watchdog タイマーはそのノードをリセットします。
4. 残りのノードの Cluster Manager は、障害が発生したノードがクラスタから削除されたことをローカルの Oracle インスタンスに通知します。
5. 残りのノードの Oracle インスタンスは、Oracle9*i* Real Application Clusters の再構成プロシージャを起動します。

Oracle インスタンスが失敗した場合は、ノードをリセットする必要があります。これによって次のことが保証されます。

- Oracle インスタンスが失敗した後に、障害のあるノードから共有ディスクに物理的な I/O 要求が発行されることがありません。
- 残りのノードは、共有ディスクのデータを破損しないで、クラスタの再構成プロシージャを起動できます。

> **参照：** Cluster Manager の詳細は、F-10 ページの「クラスタ再構成のタイミングの設定」および F-11 ページの「Watchdog デーモンおよび Cluster Manager の起動オプション」を参照してください。

OCMS の起動

ここでは、OCMS の起動方法について説明します。

- Watchdog デーモンの起動
- Cluster Manager の構成
- Cluster Manager の起動
- クラスタ再構成のタイミングの設定

注意： オラクル社では、`$ORACLE_HOME/oracm/bin/ocmstart.sh` という起動スクリプト・サンプルを用意しています。`root` ユーザーとして、このスクリプトを実行してください。また、『Oracle9*i* for UNIX Systems インストレーション・ガイド』の説明に従って環境変数 ORACLE_HOME および PATH が設定されていることを確認してください。Watchdog デーモンおよび Cluster Manager の起動方法について理解すれば、このスクリプトを使用して起動プロセスを自動化できます。

Watchdog デーモンの起動

Watchdog デーモンを起動するには、次のコマンドを入力します。

```
$ su root
# cd $ORALE_HOME/oracm/bin
# watchdogd
```

注意： Watchdog デーモンは、必ず `root` ユーザーとして起動してください。

Watchdog デーモンのデフォルトのログ・ファイルは、`$ORACLE_HOME/oracm/log/wdd.log` です。

Watchdog デーモンには構成ファイルがありません。表 F-1 に、Watchdog デーモンの起動時に使用できる引数を示します。

表 F-1 watchdogd デーモンの引数

引数	有効な値	デフォルト値	説明
-l *number*	0 または 1	1	値が 0 の場合、監視用のリソースは登録されません。この引数は、システム構成の問題をデバッグするときに使用します。 値が 1 の場合、監視用の Cluster Manager が登録されます。通常の操作では、このオプションを使用することをお薦めします。
-m *number*	5000 〜 180000ms	5000	Watchdog デーモンは、この値で指定した時間内にすべてのクライアント（oracm スレッド）からハートビート・メッセージを受信しなければなりません。この時間内にクライアントがハートビート・メッセージを送信できない場合、Watchdog デーモンはカーネルの Watchdog タイマーにハートビート・メッセージを送信するのを停止し、システムがリセットされます。
-d *string*		/dev/watchdog	Watchdog タイマーのデバイス・ファイルのパス。
-e *string*		$ORACLE_HOME/oracm/log/wdd.log	Watchdog デーモンのログ・ファイルのファイル名。

Cluster Manager の構成

OCMS を起動する前に、クラスタの各ノードで Cluster Manager 構成ファイル `$ORACLE_HOME/oracm/admin/cmcfg.ora` を作成する必要があります。このファイルには、次のパラメータを指定してください。

- `PublicNodeNames`
- `PrivateNodeNames`
- `CmDiskFile`
- `WatchdogTimerMargin`
- `HostName`

`cmcfg.ora` ファイルを作成する前に、クラスタの各ノードの `/etc/hosts` ファイルにパブリック・ネットワークのエントリ（パブリック名）とプライベート・ネットワークのエントリ（各ノードのプライベート名）が含まれていることを確認します。プライベート・ネットワークは、Oracle9*i* Real Application Clusters のノード間通信に使用されます。`CmDiskFile` パラメータには、Cluster Manager の定数パーティションの保存場所を定義します。クラスタの各ノードの `CmDiskFile` パラメータには、同じ定数パーティションを指定する必要があります。

次の例は、4 つのノードで構成されるクラスタのうち、最初のノードの `cmcfg.ora` ファイルを示しています。

```
PublicNodeNames=pubnode1 pubnode2 pubnode3 pubnode4
PrivateNodeNames=prinode1 prinode2 prinode3 prinode4
CmDiskFile=/dev/raw1
WatchdogTimerMargin=1000
HostName=prinode1
```

表 F-2 に、cmcfg.ora ファイルで構成できる Cluster Manager パラメータをすべて示します。

表 F-2 cmcfg.ora ファイルの Cluster Manager パラメータ

パラメータ	有効な値	デフォルト値	説明
CmDiskFile	ディレクトリ・パス（最大 256 文字）	デフォルト値はありません。この値は明示的に設定する必要があります。	定数パーティションのパス名を指定します。
MissCount	2 ～ 1000	5	リモート・ノードからハートビートが着信するのを Cluster Manager が待機する時間を指定します。この時間が経過すると、そのノードを非アクティブであると宣言します。待機時間（秒）は、MissCount パラメータ値の 3 倍になります。
PublicNodeNames	ホスト名のリスト（最大 4096 文字）	デフォルト値はありません。	パブリック・ネットワークのすべてのホスト名を空白で区切って指定します。ホスト名はノードごとに同じ順序で列挙してください。
PrivateNodeNames	ホスト名のリスト（最大 4096 文字）	デフォルト値はありません。	プライベート・ネットワークのすべてのホスト名を空白で区切って示します。ホスト名はノードごとに同じ順序で列挙してください。
HostName	ホスト名（最大 256 文字）	デフォルト値はありません。	プライベート・ネットワークのローカル・ホスト名を示します。この名前は /etc/hosts ファイルに定義してください。
ServiceName	サービス（最大 256 文字）	CMSrvr	Cluster Manager 間の通信に使用されるサービス名を指定します。Cluster Manager は、/etc/services ファイルにそのサービス名を検出できなかった場合、ServicePort パラメータに指定されているポートを使用します。 このリリースでは、ServiceName は固定値パラメータです。Cluster Manager が使用する代替ポートを選択する必要がある場合は、ServicePort パラメータを使用してください。

表 F-2 cmcfg.ora ファイルの Cluster Manager パラメータ （続き）

パラメータ	有効な値	デフォルト値	説明
ServicePort	任意の有効なポート番号	9998	ServiceName パラメータにサービスが指定されていないときに、Cluster Manager 間の通信に使用されるポートの番号を指定します。
WatchdogTimerMargin	1000 〜 180000ms	デフォルト値はありません。	Linux の softdog の起動時に指定される soft_margin パラメータの値と同じです。soft_margin パラメータの値は秒単位で指定しますが、WatchdogTimerMargin パラメータの値はミリ秒単位で指定します。 このパラメータは、ローカル・ノードの Cluster Manager がいずれかのノードで発生した Oracle インスタンスの障害または結合を検出してから、ローカル・ノードの Oracle インスタンスにクラスタの再構成を報告するまでの時間を計算する計算式の一部です。この計算式については、F-10 ページの「クラスタ再構成のタイミングの設定」を参照してください。
WatchdogSafetyMargin	1000 〜 180000ms	5000ms	Cluster Manager がリモート・ノードの障害を検出してから、クラスタの再構成が開始されるまでの時間を指定します。 このパラメータは、ローカル・ノードの Cluster Manager がいずれかのノードで発生した Oracle インスタンスの障害または結合を検出してから、ローカル・ノードの Oracle インスタンスにクラスタの再構成を報告するまでの時間を計算する計算式の一部です。この計算式については、F-10 ページの「クラスタ再構成のタイミングの設定」を参照してください。

Cluster Manager の起動

Cluster Manager を起動するには、次の手順で行います。

1. Watchdog デーモンが動作していることを確認します。
2. cmcfg.ora ファイルの PublicNodeNames および PrivateNodeNames パラメータに指定したホスト名が /etc/hosts ファイルに指定されていることを確認します。
3. root ユーザーとして、oracm をバックグラウンド・プロセスとして起動します。出力をログ・ファイルにリダイレクトします。たとえば、次のように入力します。

```
$ su root
# cd $ORACLE_HOME/oracm/bin
# oracm </dev/null >$ORACLE_HOME/oracm/log/cm.out 2>&1 &
```

この例では、出力メッセージおよびエラー・メッセージがすべて、$ORACLE_HOME/oracm/log/cm.out ファイルに書き込まれます。

oracm プロセスでは、複数のスレッドが生成されます。それらのスレッドをすべて表示するには、ps -elf コマンドを入力します。

表 F-3 に、oracm 実行可能プログラムの引数を示します。

表 F-3 oracm 実行可能プログラムの引数

引数	値	デフォルト値	説明
/a:*action*	0,1	0	LMON プロセスなど、共有ディスクへの書込みを行う Oracle プロセスが異常終了したときに、適用する処置を指定します。 action が 0（デフォルト）に設定されている場合は、何の処置も適用されません。action が 1 に設定されている場合、Cluster Manager はそのノードが完全に停止するように Watchdog デーモンに要求します。
/l:*filename*	すべて	/$ORACLE_HOME/oracm/log/cm.log	Cluster Manager のログ・ファイルのパス名を示します。パス名の長さは、最大 192 文字です。
/?	なし	なし	oracm 実行可能プログラムの引数のヘルプを表示します。この引数を指定した場合、Cluster Manager は起動しません。
/m	すべて	25000000	oracm ログ・ファイルのサイズをバイト単位で示します。

クラスタ再構成のタイミングの設定

ノードに障害が発生したときにデータベースの破損を回避するには、Oracle9*i* Real Application Clusters の再構成が開始するまでの遅延時間を指定します。この遅延時間を指定しないと、障害が発生したノードおよびリカバリを実行しているノードから同一データ・ブロックに対して同時にアクセスした場合、データベースが破損する可能性があります。遅延時間の長さは、次の値の合計として定義します。

- `WatchdogTimerMargin` パラメータの値
- `WatchdogSafetyMargin` パラメータの値
- Watchdog デーモンの `-m` コマンドライン引数の値

> **参照：** `WatchdogTimerMargin` および `WatchdogSafetyMargin` の各パラメータの詳細は、F-7 ページの表 F-2 を参照してください。Watchdog デーモンの `-m` コマンドライン引数の詳細は、F-6 ページの表 F-1 を参照してください。

Linux カーネルの `soft_margin` パラメータおよび `Cluster Manager` パラメータのデフォルト値を使用する場合、障害が検出されてからクラスタの再構成が開始されるまでの時間は 70 秒となります。ほとんどの作業負荷では、この時間を大幅に減らしてもかまいません。次の例は、再構成までの遅延時間を 70 秒から 20 秒に減らす方法を示しています。

- `WatchdogTimerMargin`（`soft_margin`）パラメータの値を 10 秒に設定します。
- `WatchdogSafetyMargin` パラメータの値をデフォルト値（5000ms）のままにします。
- Watchdog デーモンの `-m` コマンド行引数の値をデフォルト値（5000ms）のままにします。

`WatchdogTimerMargin`（`soft_margin`）および `WatchdogSafetyMargin` の値を変更するには、次の手順で行います。

1. Oracle インスタンスを停止します。
2. `soft_margin` に新しい値を指定して `softdog` モジュールを再ロードします。たとえば、次のように入力します。

   ```
   #/sbin/insmod softdog soft_margin=10
   ```

3. `$ORACLE_HOME/oracm/admin/cmcfg.ora` ファイルに入っている `WatchdogTimerMargin` の値を変更します。たとえば、次の行を編集します。

   ```
   WatchdogTimerMargin=50000
   ```

4. `-m` コマンドライン引数を 5000 に設定して `watchdogd` を再起動します。
5. `oracm` 実行可能プログラムを再起動します。
6. Oracle インスタンスを再起動します。

Watchdog デーモンおよび Cluster Manager の起動オプション

OCMS では、Oracle インスタンスが失敗し、Cluster Manager スレッドが正しく動作しない場合、ノードを完全にリセットする方法でノードを保護しています。この方法では、データベースが破損することはありません。

ただし、Oracle インスタンスが失敗した場合でも、ノードのリセットが必要でない場合もあります。Oracle インスタンスが同期 I/O を使用している場合、ノードのリセットは不要です。また、Oralce インスタンスが非同期 I/O を使用している場合でも、Linux カーネルにおける非同期 I/O の実装方法によっては、ノードのリセットが不要になることがあります。

次のコマンドの `/a:`*`action`* フラグには、Oracle プロセスが失敗したときの OCMS の動作を定義します。

```
$ oracm /a:[action]
```

この例の *`action`* 引数が 0 に設定されている場合は、ノードはリセットされません。

デフォルトでは、`watchdog` デーモンは `-l 1` オプションで起動し、`oracm` プロセスは `/a:0` オプションで起動します。これらのデフォルト値を使用すると、`oracm` または `watchdogd` プロセスが終了した場合にのみノードがリセットされます。ディスクへの書込みを行う Oracle プロセスが終了した場合は、ノードのリセットは行われません。ノードのリセットが不要であることが保証されている Linux カーネルを使用しているときは、この方法でもかまいません。

前述の例の *`action`* 引数が 1 に設定されている場合は、`oracm` コマンド、`watchdogd` デーモン、またはディスクへの書込みを行う Oracle プロセスが終了したときに、ノードがリセットされます。これらの状況では、Oracle インスタンスに対して SHUTDOWN ABORT コマンドを実行すると、ノードがリセットされ、そのノードで動作している Oracle インスタンスがすべて終了します。

G

Optimal Flexible Architecture

この付録では、Optimal Flexible Architecture（OFA）標準について説明します。OFA 標準は、高速で、メンテナンスの必要性を最小限に抑えた信頼性の高いデータベース構成を実現するためのガイドラインです。この付録では、次の項目について説明します。

- Optimal Flexible Architecture
- UNIX で適用される Optimal Flexible Architecture

Optimal Flexible Architecture

Oracle9*i* のインストールおよび構成時には、OFA 標準を適用するようにしてください。

OFA には、次の目的があります。

- デバイスのボトルネックやパフォーマンスの低下が発生しないように、ディスク上の大量で複雑なソフトウェアおよびデータを編成します。
- データの品質低下を招く恐れのある管理タスク（ソフトウェアやデータのバックアップなど）を簡略化します。
- 複数の Oracle データベース間の切替えを簡略化します。
- データベースの拡張を適切に管理します。
- データ・ディクショナリの空き領域の断片化を回避し、その他の領域で起きる断片化を分離して、リソースの競合を最小限に抑えます。

OFA に準拠したデータベースの特長

この項では、OFA 標準に準拠しているデータベースの特長について説明します。

ファイル・システムの編成

ファイル・システムは、次のような操作を簡単に管理できるように編成されています。

- 既存のデータベースへのデータの追加
- ユーザーの追加
- データベースの作成
- ハードウェアの追加

I/O 負荷の分散

I/O 負荷を十分な数のディスク・ドライブに分散させるため、パフォーマンスのボトルネックが発生しません。

ハードウェア・サポート

ほとんどの場合、OFA 標準を活用するために新しいハードウェアに出資する必要はありません。

ドライブの障害に対する予防措置

複数ドライブにアプリケーションを分散することにより、ドライブに障害が発生した場合でも、アプリケーションへの影響を最小限に抑えられます。

ホーム・ディレクトリの分散

次のものを、複数のディスク・ドライブに分散できます。

- 複数のホーム・ディレクトリ
- 各ホーム・ディレクトリの内容

ログイン・ホーム・ディレクトリの整合性

ログイン・ホーム・ディレクトリを参照するプログラムを修正しなくても、それらを追加、移動または削除できます。

UNIX ディレクトリ・サブツリーの独立性

ファイルのカテゴリは、独立した UNIX ディレクトリ・サブツリーに分類されています。そのため、あるカテゴリのファイルでの操作によって、別のカテゴリのファイルが受ける影響を最小限に抑えることができます。

アプリケーション・ソフトウェアの同時実行サポート

複数バージョンのアプリケーション・ソフトウェアを同時に実行できます。これによって、ユーザーは旧バージョンを削除しなくても、新しいリリースのアプリケーションを使用できます。アップグレード後に新しいバージョンへ移行する作業は、管理者にとっては簡単な作業で、ユーザーがその作業を意識する必要はありません。

各データベースの管理情報の区別

データベースごとに管理情報を区別できるため、管理データをわかりやすい構造で編成および保存できます。

データベース・ファイルのネーミング規則

データベース・ファイルには、次のネーミング規則があります。

- 他のファイルと区別しやすい名前にします。
- 2 つのデータベースに属するファイルが、どちらのデータベースに属するかを区別できる名前にします。
- 制御ファイル、REDO ログ・ファイルおよびデータ・ファイルであることが識別できる名前にします。
- データ・ファイルと表領域の関係が明確に識別できる名前にします。

表領域の内容の分割

表領域の内容が分割されているため、次のメリットがあります。

- 表領域内の空き領域の断片化を最小限に抑えられます。

- I/O 要求の競合を最小限に抑えられます。
- 管理面の柔軟性が高くなります。

すべてのドライブに分散されている I/O 負荷のチューニング

Oracle データを RAW デバイスに保存しているドライブを含め、すべてのドライブへの I/O 負荷をチューニングします。

Oracle9*i* Real Application Clusters に対する OFA のその他の利点

Oracle9*i* Real Application Clusters のインストール

- 管理データを一元管理し、すべてのデータベース管理者が利用できるようにします。
- 特定のインスタンスの管理データをファイル名で識別できます。

UNIX で適用される Optimal Flexible Architecture

この項では、OFA 標準で推奨されているネーミング方法について説明します。

マウント・ポイント

この項では、マウント・ポイントのネーミング規則について説明します。

マウント・ポイントの作成

Oracle9*i* のインストールには、マウント・ポイントが 2 つ以上（1 つはソフトウェア用、残りはデータベース・ファイル用）必要です。Oracle9*i* に OFA 推奨事項を十分に適用するためには、マウント・ポイントが 4 つ以上（1 つはソフトウェア用、残りはデータベース・ファイル用）必要です。

マウント・ポイントの構文

すべてのマウント・ポイントは、/*pm* という構文で名前を付けます。*p* は文字列定数、*m* は各マウント・ポイントを区別するための一意の固定長キー（通常、2 桁の数字）です。たとえば、`/u01` および `/u02`、`/disk01` および `/disk02` などです。

超大規模データベース（VLDB）のマウント・ポイントの名前

1 つのアプリケーションが使用するデータベース・ファイルが各ディスク・ドライブにあり、I/O のボトルネックが発生しないようにデータベースごとに十分な数のドライブを備えている場合は、/*pm*/*q*/*dm* という構文で、マウント・ポイントに名前を付けます。表 G-1 に、この構文で使用する変数を示します。

表 G-1 マウント・ポイントの名前の構文

変数	説明
pm	マウント・ポイントの名前
q	Oracle データがこのディレクトリに保存されていることを示す文字列
dm	初期化パラメータ DB_NAME の値（単一インスタンス・データベースのインスタンス *sid* と同じ）

たとえば、/u01/oradata/test および /u02/oradata/test という名前のマウント・ポイントは、Oracle テスト・データベースに 2 つのドライブを割り当てます。

ディレクトリの名前

この項では、OFA に準拠したディレクトリのネーミング規則について説明します。

ホーム・ディレクトリの構文

/*pm*/*h*/*u* という構文で、ホーム・ディレクトリに名前を付けます。表 G-2 に、この構文で使用する変数を示します。

表 G-2 ホーム・ディレクトリの名前の構文

変数	説明
pm	マウント・ポイントの名前
h	標準のディレクトリ名
u	ディレクトリの所有者の名前

たとえば、/u01/app/oracle は、Oracle サーバー・ソフトウェア所有者のホーム・ディレクトリ（Oracle ベース・ディレクトリとも呼ばれ、インストーラによって使用されるデフォルトのディレクトリ）で、/u01/app/applmgr は、Oracle アプリケーション・ソフトウェア所有者のホーム・ディレクトリです。

ホーム・ディレクトリを UNIX ファイル・システム内の同一レベルに置くと便利です。こうすると、別のマウント・ポイントにあるアプリケーション所有者のログイン・ホーム・ディレクトリの集まりを、決まった形式（たとえば、/*/app/*）で参照できるためです。

パス名の参照

明示的なパス名は、そのパス名を特別に保存するファイル（パスワード・ファイル /etc/passwd、Oracle の oratab ファイルなど）で参照します。グループのメンバーは、/etc/group ファイルで参照します。

ソフトウェア・ディレクトリ

複数バージョンのアプリケーション・ソフトウェアを同時に実行するための OFA 要件を満たすために、各バージョンの Oracle9*i* ソフトウェアを `/pm/h/u/product/v` という構文の 1 つのディレクトリに保存します。

表 G-3 に、この構文で使用する変数を示します。

表 G-3 Oracle9*i* ソフトウェアを保存するディレクトリ名の構文

変数	説明
pm	マウント・ポイントの名前
h	標準のディレクトリ名
u	ディレクトリの所有者の名前
v	ソフトウェアのバージョン

たとえば、`/u01/app/oracle/product/9.2.0.1.0` は、Oracle9*i* の親ディレクトリを示します。環境変数 ORACLE_HOME は、この値に設定します。

サブディレクトリの名前

管理データの編成を簡略化するために、データベース固有の管理ファイルを `/h/admin/d/a/` というサブディレクトリに保存しておくことをお薦めします。*h* は Oracle ソフトウェア所有者のホーム・ディレクトリ、*d* はデータベース名（DB_NAME）、*a* は各データベース管理ファイルのサブディレクトリを表しています。表 G-4 に、データベース管理ファイルのサブディレクトリを示します。

表 G-4 データベース管理ファイルのサブディレクトリ

サブディレクトリ	説明
`adhoc`	特定のデータベースの非定型 SQL スクリプト
`arch`	アーカイブ REDO ログ・ファイル
`adump`	監査ファイル （`AUDIT_FILE_DEST` 初期化パラメータを adump に設定してください。このサブディレクトリは定期的に整理してください）。
`bdump`	バックグラウンド・プロセスのトレース・ファイル
`cdump`	コア・ダンプ・ファイル
`create`	データベースを作成するプログラム
`exp`	データベース・エクスポート・ファイル
`logbook`	データベースの状態および履歴を記録するファイル
`pfile`	インスタンス・パラメータ・ファイル

表 G-4 データベース管理ファイルのサブディレクトリ（続き）

サブディレクトリ	説明
udump	ユーザー SQL トレース・ファイル

たとえば、adhoc サブディレクトリが sab というデータベースの一部である場合、このサブディレクトリのパス名は、/u01/app/oracle/admin/sab/adhoc/ となります。

データベース・ファイルの名前

次のネーミング規則に従ってデータベース・ファイルに名前を付けることによって、これらの識別が簡単になります。

ファイル・タイプ	ファイルのネーミング規則
制御ファイル	/*pm*/*q*/*d*/control.ctl
REDO ログ・ファイル	/*pm*/*q*/*d*/redo*n*.log
データ・ファイル	/*pm*/*q*/*d*/*tn*.dbf

次の表に、この構文で使用する変数を示します。

変数	説明
pm	マウント・ポイント名（この付録の前の方に説明されています）
q	Oracle データを他の Oracle ファイルと区別するための文字列（通常、ORACLE または oradata）
d	初期化パラメータ DB_NAME の値（単一インスタンス・データベースのインスタンス *sid* と同じ）
t	Oracle 表領域名
n	2 桁の数字

注意： /*pm*/*q*/*d* に、データベース *d* に対応付けられた制御ファイル、REDO ログ・ファイル、データ・ファイル以外のファイルを保存しないでください。

たとえば、この規則に従って /u03/oradata/sab/system01.dbf という名前のデータ・ファイルを作成すると、そのファイルが属しているデータベースを簡単に判断できます。

要件に応じた個々のセグメント

異なる表領域間で、存在期間、I/O 要求の必要性およびバックアップの頻度別に、セグメントをいくつかのグループに分類します。

表 G-5 に、Database Configuration Assistant が Oracle データベースごとに作成する特殊な表領域を示します。データベースを手動で作成した場合は、必要な表領域を作成する必要があります。アプリケーション・セグメントに必要な表領域以外に、これらの表領域も必要です。

> **参照：** 表領域を手動で作成する方法については、『Oracle9*i* データベース管理者ガイド』を参照してください。

表 G-5 特殊な表領域

表領域	必須	説明
CWMLITE	いいえ	OLAP カタログ・メタデータ・リポジトリ（CWMLite）
DEMO	いいえ	デモ・スキーマ
DRSYS	いいえ	Oracle Text セグメント
INDX	いいえ	USERS 表領域のデータに対応付けられた索引
OEM_REPOSITORY	いいえ	Oracle Enterprise Manager のリポジトリ
RBS	はい	ロールバック・セグメント
SYSTEM	はい	データ・ディクショナリ・セグメント
TEMP	はい	テンポラリ・セグメント
USERS	いいえ	その他のユーザー・セグメント
XDB	いいえ	XDB 表領域には、SQL を介して、または HTTP や WebDAV などのプロトコルを介して Oracle XML DB リポジトリに格納されるデータが入ります。

これらの特殊な表領域を作成すると、データ・ディクショナリ・セグメントが削除されることがなく、削除できる他のセグメントが SYSTEM 表領域に残ることもありません。これによって、表領域の空き領域が断片化されるため、SYSTEM 表領域を再作成する必要がありません。

アプリケーション・データが保存されている表領域にロールバック・セグメントが保存されることはないため、管理者は、アプリケーションの表領域をオフラインにしてメンテナンスを行うことができます。セグメントは種類別に分割されているので、管理者は、複雑なツールを使用しなくてもデータ拡張の記録および見積りができます。

表領域の名前

表領域の名前は、8 文字以下で指定します。Oracle9*i* では表領域に 30 文字の名前を付けることができますが、UNIX ファイル名としては、14 文字までに制限されています。データ・ファイル名は、*tn*.dbf という構文で指定することをお薦めします。この場合、*t* は表領域名、*n* は 2 桁の数字からなる文字列です。拡張子と 2 桁の数字からなる文字列の合計が 6 文字になるため、表領域名に使用できるのは、8 文字までになります。

データ・ファイルとそれを使用する表領域との関連性がわかるような名前にします。たとえば、General Ledger データと索引を保存するための表領域には、それぞれ GLD および GLX という名前を指定します。

注意： 表領域の名前に「tablespace」という語を連想させる文字列を付ける必要はありません。表領域は、前後関係から識別できるため、名前によってタイプに関する情報を示す必要はありません。

OFA 構造に基づいた Oracle ファイル

表 G-6 に、ファイルのクラスを識別するための構文を示します。

表 G-6 ファイルのクラスを識別するためのディレクトリ構造の構文

ディレクトリ構造の構文	説明
`/u[0-9][0-9]`	ユーザー・データ・ディレクトリ
`/*/home/*`	ユーザー・ホーム・ディレクトリ
`/*/app/*`	ユーザー・アプリケーション・ソフトウェア・ディレクトリ
`/*/app/applmgr`	Oracle アプリケーション・ソフトウェアのサブツリー
`/*/app/oracle/product`	Oracle ソフトウェアのサブツリー
`/*/app/oracle/product/9.2.0.1.0`	Oracle9*i* リリース 2（9.2.0.1.0）の配布ファイル
`/*/app/oracle/admin/sab`	sab データベースの管理サブツリー
`/*/app/oracle/admin/sab/arch/*`	sab データベースのアーカイブ・ログ・ファイル
`/*/oradata`	Oracle データのディレクトリ
`/*/oradata/sab/*`	sab データベース・ファイル
`/*/oradata/sab/*.log`	sab データベースの REDO ログ・ファイル

OFA ファイルのマッピング

表 G-7 に、ファイルのマウント・ポイント、アプリケーション、データベース、表領域などが含まれる、OFA に準拠したサンプル・データベースのファイル・マッピングの階層を示します。ファイル名によって、ファイルの種類（制御、ログまたはデータ）がわかります。

表 G-7 OFA インストールの階層型ファイル・マッピング

ディレクトリ	説明
/	ルート・マウント・ポイント
/u01/	ユーザー・データのマウント・ポイント 1
/u01/app/	アプリケーション・ソフトウェアのサブツリー
/u01/app/oracle/	oracle ソフトウェア・ユーザーのホーム・ディレクトリ
/u01/app/oracle/admin/	データベース管理ファイルのサブツリー
/u01/app/oracle/admin/TAR	サポート・ログ・ファイルのサブツリー
/u01/app/oracle/admin/*db_name1*/	*db_name1* データベースの admin サブツリー
/u01/app/oracle/admin/*db_name2*/	*db_name2* データベースの admin サブツリー
/u01/app/oracle/doc/	オンライン・マニュアル
/u01/app/oracle/product/	配布ファイル
/u01/app/oracle/product/8.1.6/	リリース 8.1.6 インスタンスの Oracle ホーム・ディレクトリ
/u01/app/oracle/product/8.1.7/	リリース 8.1.7 インスタンスの Oracle ホーム・ディレクトリ
/u01/app/oracle/product/9.2.0.1.0	リリース 2（9.2.0.1.0）インスタンスの Oracle ホーム・ディレクトリ
/u01/app/ltb/	ユーザーのホーム・ディレクトリ
/u01/app/sbm/	ユーザーのホーム・ディレクトリ
/u01/oradata/	Oracle データのサブツリー
/u01/oradata/*db_name1*/	*db_name1* データベース・ファイルのサブツリー
/u01/oradata/*db_name2*/	*db_name2* データベース・ファイルのサブツリー
/u02/	ユーザー・データのマウント・ポイント 2
/u02/home/	ログイン・ホーム・ディレクトリのサブツリー

表 G-7 OFA インストールの階層型ファイル・マッピング（続き）

ディレクトリ	説明
/u02/home/cvm/	ユーザーのホーム・ディレクトリ
/u02/home/vrm/	ユーザーのホーム・ディレクトリ
/u02/oradata/	Oracle データのサブツリー
/u02/oradata/*db_name1*/	*db_name1* データベース・ファイルのサブツリー
/u02/oradata/*db_name2*/	*db_name2* データベース・ファイルのサブツリー
/u03/	ユーザー・データのマウント・ポイント 3
/u03/oradata/	Oracle データのサブツリー
/u03/oradata/*db_name1*/	*db_name1* データベース・ファイルのサブツリー
/u03/oradata/*db_name2*/	*db_name2* データベース・ファイルのサブツリー

複数インスタンスにおける OFA に準拠したデータベースのファイル・マッピング

Oracle9*i* Real Application Clusters を使用する場合、そのクラスタに対して Oracle 管理ホーム・ディレクトリとして機能するノードを 1 つ選択します。管理ホーム・ディレクトリには、管理サブツリーが含まれます。~/admin/*d*/ ディレクトリにある bdump、cdump、logbook、pfile および udump ディレクトリのデータベースにアクセスする各インスタンスにサブディレクトリを作成します。管理ホームの admin ディレクトリは、すべてのインスタンスの admin ディレクトリにマウントする必要があります。表 G-10 に、ディレクトリ構造のサンプルを示します。

表 G-8 二重インスタンスを持つ Oracle9*i* Real Application Clusters の管理ディレクトリ構造

ディレクトリ・パス	説明
/u01/app/oracle/admin/sab/	sab データベース用の管理ディレクトリ
/u01/adhoc/	スクリプト用のディレクトリ
/u01/arch/	すべてのインスタンス用のログ・アーカイブ先
/u01/arch/redo001.arc	アーカイブ REDO ログ・ファイル
/u01/bdump/	バックグラウンド・ダンプ・ファイルのディレクトリ

表 G-8 二重インスタンスを持つ Oracle9*i* Real Application Clusters の管理ディレクトリ構造（続き）

ディレクトリ・パス	説明
/u01/bdump/inst1/	inst1 インスタンスのバックグラウンド・ダンプ先
/u01/bdump/inst2/	inst2 インスタンスのバックグラウンド・ダンプ先
/u01/cdump/	コア・ダンプ・ファイル用のディレクトリ
/u01/cdump/inst1/	inst1 インスタンスのコア・ダンプ先
/u01/cdump/inst2/	inst2 インスタンスのコア・ダンプ先
/u01/create/	作成スクリプト用のディレクトリ
/u01/create/1-rdbms.sql	*inst* データベース作成用の SQL スクリプト
/u01/exp/	エクスポート用のディレクトリ
/u01/exp/20000120full.dmp	2000 年 1 月 20 日のフル・エクスポート・ダンプ・ファイル
/u01/exp/export/	エクスポート・ファイル用のディレクトリ
/u01/exp/import/	インポート・ファイル用のディレクトリ
/u01/logbook/	ログブック・エントリ用のディレクトリ
/u01/logbook/inst1/	inst1 インスタンス・レポート用のディレクトリ
/u01/logbook/inst1/params.1st	inst1 インスタンスの V$PARAMETER レポート
/u01/logbook/inst2/	inst2 インスタンス・レポート用のディレクトリ
/u01/logbook/inst2/params.1st	inst2 インスタンスの V$PARAMETER レポート
/u01/logbook/user.1st	DBA_USERS レポート
/u01/pfile/	インスタンス・パラメータ・ファイル用のディレクトリ
/u01/pfile/inst1/	inst1 インスタンス・パラメータ用のディレクトリ
/u01/pfile/inst1/initinst1.ora	inst1 インスタンスのインスタンス・パラメータ
/u01/pfile/inst2/	inst2 インスタンス・パラメータ用のディレクトリ
/u01/pfile/inst2/initinst2.ora	inst2 インスタンスのインスタンス・パラメータ
/u01/udump/	ユーザー・ダンプ・ファイル用のディレクトリ
/u01/udump/inst1/	inst1 インスタンスのユーザー・ダンプ先
/u01/udump/inst2/	inst2 インスタンスのユーザー・ダンプ先

ディレクトリ構造

次の項では、OFA に準拠したインストールのディレクトリ構造について説明します。

Oracle ベース・ディレクトリ

Oracle ベース・ディレクトリは、Oracle ディレクトリ構造のルート・ディレクトリです。Oracle Universal Installer を使用して OFA に準拠したデータベースをインストールするときに、デフォルトの Oracle ベース・ディレクトリは /*pm*/app/oracle に設定されます。表 G-9 に、Oracle ベース・ディレクトリ構造の詳細を示します。

表 G-9 Oracle ベース・ディレクトリ構造および内容

ディレクトリ	説明
admin	管理ファイル
doc	オンライン・マニュアル
jre	Java Runtime Environment（JRE）
oradata	Oracle データ・ファイル
oui	Oracle Universal Installer（OUI）
product	Oracle ソフトウェア

Oracle ホーム・ディレクトリ

OFA に準拠した Oracle サーバーをインストールする場合、Oracle ホーム・ディレクトリは、/*pm*/app/oracle/product/*release_number* となります。表 G-10 に、Oracle ホーム・ディレクトリ構造の詳細を示します。UNIX の場合、Oracle ホーム・ディレクトリには、表 G-10 に示すサブディレクトリ、およびインストールした各 Oracle 製品用のサブディレクトリが含まれます。

表 G-10 Oracle ホーム・ディレクトリ構造および内容（一部）

ディレクトリ	説明
Apache	
BC4J	
JRE	
assistants	構成アシスタント
bin	すべての製品のバイナリ・ファイル
classes	
ctx	*inter*Media Text オプション
cwmlite	

表 G-10 Oracle ホーム・ディレクトリ構造および内容（一部）（続き）

ディレクトリ	説明
dbs	初期化ファイル
dcommon	
demo	
doc	
ds	
hs	
install	インストール関連のファイル
jar	
javavm	
jdbc	
jsp	
ldap	
lib	Oracle 製品ライブラリ
md	Spatial オプション
network	Oracle Net Services ファイル
oci	
ocommon	すべての製品に共通のファイル
oem_webstage	
olap	
oradata	
ord	データ・オプション
otrace	Oracle TRACE ファイル
owra	
plsql	PL/SQL ファイル
precomp	プリコンパイラ・ファイル
rdbms	データベースに必要なサーバー・ファイルおよびライブラリ
relnotes	
slax	SLAX パーサー・ファイル
sqli	

表 G-10 Oracle ホーム・ディレクトリ構造および内容（一部）（続き）

ディレクトリ	説明
sqlplus	SQL*Plus ファイル
syndication	
sysman	
ultrasearch	
wug	
xdk	

製品のサブディレクトリの例

表 G-11 に、製品のサブディレクトリおよびそれらの内容の例を示します。

表 G-11 製品のサブディレクトリの例

ディレクトリ	説明
rdbms	admin、doc、install、lib、log、mesg
sqlplus	admin、demo、doc、install、lib、mesg

製品のサブディレクトリの内容

表 G-12 に、製品のサブディレクトリ rdbms および sqlplus に含まれるサブディレクトリを示します。

表 G-12 製品のサブディレクトリの内容

ディレクトリ	説明
admin	管理 SQL およびシェル・スクリプト（catalog.sql、catexp.sql、demo.sql など）
admin/*	他の製品専用ディレクトリ
admin/resource	リソース・ファイル
admin/terminal	ランタイム端末ファイル
demo	デモ用のスクリプトおよびデータ・ファイル
doc	README ファイル
install	製品のインストール・スクリプト
jlib	製品の Java クラス
lib	製品ライブラリおよび配布された Make ファイル

表 G-12 製品のサブディレクトリの内容 （続き）

ディレクトリ	説明
log	トレース・ファイルおよびログ・ファイル（orasrv.log、*.trc ファイルなど）
mesg	英語メッセージ・ファイルおよびバイナリ・ファイル（oraus.msg、oraus.msb など）

admin ディレクトリのファイルのネーミング規則

表 G-13 に、$ORACLE_HOME/rdbms/admin ディレクトリ内にある SQL スクリプトを示します。

表 G-13 admin ディレクトリのファイルのネーミング規則

ファイル	説明
cat*.sql	カタログおよびデータ・ディクショナリ表およびビューを作成します。インストール時に次のファイルが自動的に実行されます。 ■ catalog.sql（すべてのインストール用） ■ catproc.sql（すべてのインストール用） ■ catclust.sql（Oracle9*i* Real Application Clusters オプションのインストール用） ■ catrep.sql（すべてのインストール用） catproc.sql ファイルは、標準の PL/SQL パッケージ（DBMS_SQL、DBMS_OUTPUT など）を作成するためにスクリプトを順に実行します。
d*.sql	ダウングレード・スクリプト
dbms*.sql	その他のデータベース・パッケージ
u*.sql	アップグレード・スクリプト
utl*.sql	データベース・ユーティリティ用の表およびビューを作成するスクリプト

ファイル名の拡張子

表 G-14 に、ファイル名の拡張子を示します。

表 G-14 ファイル名の拡張子

拡張子	説明
.a	オブジェクト・ファイル・ライブラリ（Ada ランタイム・ライブラリ）
.aud	Oracle 監査ファイル
.bdf	X11 フォント記述ファイル
.bmp	X11 ビットマップ・ファイル
.c	C ソース・ファイル
.ctl	SQL*Loader 制御ファイル、Oracle サーバー制御ファイル
.dat	SQL*Loader データ・ファイル
.dbf	Oracle サーバー表領域ファイル
.dmp	エクスポート・ファイル
.doc	ASCII テキスト・ファイル
.env	環境設定用のシェル・スクリプト・ファイル
.h	C ヘッダー・ファイル（sr.h は SQL*Report Writer のヘルプ・ファイル）
.jar	Java クラス・アーカイブ
.l	UNIX man ページ
.lis	SQL*Plus スクリプトの出力
.log	インストール・ログ・ファイル（Oracle サーバー REDO ログ・ファイル）
.mk	Make ファイル
.msb	多言語オプションのメッセージ・ファイル（バイナリ）
.msg	多言語オプションのメッセージ・ファイル（テキスト）
.o	オブジェクト・モジュール
.ora	Oracle 構成ファイル
.orc	インストール・プロトタイプ・ファイル
.pc	Pro*C ソース・ファイル
.pco	Pro*COBOL ソース・ファイル
.ppd	プリンタ・ドライバ・ファイル
.sh	Bourne シェル・スクリプト・ファイル

表 G-14 ファイル名の拡張子（続き）

拡張子	説明
`.sql`	SQL スクリプト・ファイル
`.sys`	Bourne シェル・スクリプト・ファイル
`.tab`	SQL スクリプト・ファイル
`.trc`	トレース・ファイル
`.utd`	ユニフォーム端末定義ファイル
`.zip`	Zip ファイル

索引

記号

@ 略称，1-2

A

AIX のツール
- PTX Agent，2-8
- PTX Manager，2-8

B

Bourne シェル・スクリプトのトレース，1-4
BUFFER
- インポート用のパラメータ，A-7

C

C
- Pro*C/C++，4-8

Cluster Manager
- 起動，F-9
- 起動オプション，F-11
- 構成，F-6

CONFIG_WATCHDOG_NOWAYOUT パラメータ，F-3
coraenv，1-4
CREATE CONTROLFILE パラメータ，1-17
CREATE DATABASE パラメータ，1-17

D

DB_BLOCK_SIZE 初期化パラメータ，1-13
DB_CACHE_SIZE 初期化パラメータ，1-13
dba グループ，1-18
dbhome ファイル，1-9
DISM，1-14

F

FORMAT プリコンパイラ，4-14
- Pro*COBOL，4-15

free コマンド，2-6

G

Glance/UX，2-9
glogin.sql ファイル，3-2

H

HP-UX ツール，2-9，2-10
HP の非同期フラグ，B-7
HTTP Server のログ・ファイル，1-26

I

initsid.ora ファイル，1-22
Intelligent Agent
- SNMP での構成，5-9

*inter*Media
- Annotator，E-3
- Java Servlet Photo Album デモ，E-2
- JavaServer Pages Photo Album デモ，E-2
- ORDAudio Java デモ，E-2
- ORDAudio SQL デモ，E-2
- ORDDoc Java デモ，E-2
- ORDDoc SQL デモ，E-2
- ORDImage Java デモ，E-2
- ORDImage OCI デモ，E-2

ORDVideo Java デモ，E-2
ORDVideo SQL デモ，E-2
I/O
AIX での非同期，A-11
HP での非同期，B-4
Tru64 での非同期，D-11
チューニング，2-13
iostat コマンド，2-5
I/O スレーブ，A-13
IPC プロトコル，5-5
ireclen，4-4
ISM，1-14

J

JAVA_POOL_SIZE 初期化パラメータ，1-13

L

LARGE_POOL_SIZE 初期化パラメータ，1-13
Locator，E-3
LOG_BUFFERS 初期化パラメータ，1-13
login.sql ファイル，3-2
lsps コマンド，2-6

M

max_async_req パラメータ，D-7
max_objs パラメータ，D-7
max_sessions パラメータ，D-7
MAXDATAFILES パラメータ，1-17
maxfree パラメータ，A-2
MAXINSTANCES パラメータ，1-17
MAXLOGFILES パラメータ，1-17
MAXLOGHISTORY パラメータ，1-17
MAXLOGMEMBERS パラメータ，1-17
maxperm パラメータ，A-2，A-4
MicroFocus COBOL コンパイラ，4-11
minfree パラメータ，A-2
minperm パラメータ，A-2，A-4
MLOCK 権限，B-4
mpstat コマンド，2-6
msg_size パラメータ，D-7

N

NUMA
「指定配置最適化」を参照

O

OCMS
Watchdog デーモン，F-3
oinstall グループ，1-19
oper グループ，1-19
Optimal Flexible Architecture
OFA の特長，G-2
ファイル・マッピング，G-2，G-10
Oracle Advanced Security，5-11
Oracle Enterprise Manager Intelligent Agent，5-8
Oracle HTTP Server，1-25
Oracle Net Services
IPC プロトコル，5-5
Oracle Advanced Security，5-11
Oracle Enterprise Manager Intelligent Agent，5-8
TCP/IP プロトコル，5-7
プロトコル，5-5
プロトコル・サポート，5-5
Oracle Net Services 構成ファイル，5-2
Oracle Net 構成ファイル，5-2
Oracle Text，E-3
Oracle9*i* Spatial，E-3
Oracle Call Interface，4-19
デモ・プログラム，4-20
oracle アカウント，1-20
Oracle 環境変数
EPC_DISABLED，1-3
NLS_LANG，1-3
ORA_NLS，1-3
ORA_NLS33，1-3
ORA_TZFILE，1-3
ORACLE_BASE，1-3
ORACLE_HOME，1-4
ORACLE_PATH，1-4
ORACLE_SID，1-4
ORACLE_TRACE，1-4
Oracle9*i* 変数，1-2
ORAENV_ASK，1-4
SQLPATH，1-4，1-5
TWO_TASK，1-5
Oracle システム識別子，1-4
Oracle プリコンパイラと OCI のリンクおよび Make ファイル
カスタム Make ファイル，4-22

未定義シンボル，4-23
Oracle ランタイム・システム
Pro*COBOL，4-13
oracm，F-9
oradism コマンド，1-15
ORAENV_ASK，1-4
oraenv ファイル
説明，1-8
データベース間の移動，1-9
ORAINVENTORY，1-19
orapwd コマンド，1-21
orapwd ユーティリティ，1-21
oreclen，4-4
OS_AUTHENT_PREFIX パラメータ，1-20
OSDBA，1-18
OSOPER，1-19

P

PL/SQL デモ，1-28
ロード，1-28
Polycenter Advanced ファイル・システム，2-13
Pro*C/C++
Make ファイル，4-9
シグナル，4-25
使用，4-8
デモ・プログラム，4-9
Pro*COBOL，4-11
FORMAT プリコンパイラ，4-14，4-15
Oracle ランタイム・システム，4-13
デモ・プログラム，4-13
ネーミングの違い，4-11
ユーザー・プログラム，4-14
PRODUCT_USER_PROFILE 表，3-3
Programmer's Analysis Kit（HP PAK），2-10
PTX Agent，2-8

R

rad_gh_regions パラメータ，D-4
RAW デバイス，2-16
Oracle9*i* Real Application Clusters のインストール，2-16
RAW ディスク・パーティションの可用性，2-16，2-17
Tru64，D-17
ガイドライン，2-16
設定，2-16
バッファ・キャッシュ・サイズ，2-15
RAW デバイスの設定，2-18
rdg_max_auto_msg_wires パラメータ，D-7
Reliable Data Gram，D-7
root.sh スクリプト，1-9

S

sar コマンド，2-4，2-12
SCHED_NOAGE パラメータ，B-3
SCOTT アカウント，1-22
SGA，1-13
確認，1-14
SGA_MAX_SIZE パラメータ，1-14
SHARED_POOL_SIZE 初期化パラメータ，1-13
shm_allocate_striped パラメータ，D-4
SHM_MAX パラメータ，1-13
SHM_SEG パラメータ，1-13
SHMMAX パラメータ，1-13
SHMSEG パラメータ，1-13
SIGCLD 2 タスク・シグナル，4-24
SIGIO 2 タスク・シグナル，4-24
SIGINT 2 タスク・シグナル，4-24
SIGPIPE 2 タスク・シグナル，4-24
SIGTERM 2 タスク・シグナル，4-24
SIGURG 2 タスク・シグナル，4-24
Simple Network Management Protocol（SNMP），5-9
SNMP
および Intelligent Agent，5-9
snmpd 実行可能ファイル，5-10
SNMP および Intelligent Agent，5-9
Spatial のデモ，E-3
SPIKE 最適化ツール，D-19
SPOOL コマンド
SQL*Plus，3-14
使用，3-14
SQL*Loader，A-7
SQL*Module for Ada
使用，4-18
ユーザー・プログラム，4-19
SQL*Plus
PRODUCT_USER_PROFILE 表，3-3
Site Profile，3-2
SPOOL コマンド，3-14
User Profile，3-2
オペレーティング・システム・コマンドの実行，

3-14
コマンドライン SQL*Plus の使用，3-13
コマンドライン・ヘルプ，3-5
システム・エディタ，3-13
制限事項，3-15
設定ファイル，3-2
デフォルト・エディタ，3-13
デモンストレーション表，3-4
割込み，3-14
SQL*Plus への割込み，3-14
SSL，1-25
ssm_threshold パラメータ，D-4
start_peer スクリプト，5-10
swapinfo コマンド，2-6
swapon コマンド，2-6
swap コマンド，2-6
SYSDATE，1-10
SYSTEM，1-22
SYS アカウント，1-22

T

TCP/IP プロトコル，5-7
TIMED_STATISTICS パラメータ，D-16

U

UDP IPC，D-8
UDP のチューニング，A-21
Unified ファイル・システム，2-13
UNIX System V ファイル・システム，2-13
UNIX 環境変数
CLASSPATH，1-6
DISPLAY，1-6
HOME，1-6
LANG，1-6
LANGUAGE，1-6
LD_LIBRARY_PATH，1-7
LD_OPTIONS，1-6
LDPATH，1-7
LIBPATH，1-7
LPDEST，1-7
PATH，1-7
PRINTER，1-7
SHELL，1-7
SHLIB_PATH，1-8
TMPDIR，1-8
XENVIRONMENT，1-8
UNIX グループ
dba，1-18
oinstall，1-19
oper，1-19
User Profile
SQL*Plus，3-2

V

Veritas ファイル・システム，2-13
vmstat コマンド，2-3，2-12
vmtune コマンド，A-2

W

Watchdog デーモン
起動，F-5
起動オプション，F-11
説明，F-3

X

XA 機能，4-27

あ

アカウント
SCOTT，1-22
SYS，1-22
SYSTEM，1-22

い

インストール
Oracle Internet Directory，xvi
Oracle Workflow，xvi
インポート
BUFFER パラメータ，A-7

う

受取りルーチン，4-25
例，4-25

お

オペレーティング・システム・コマンドの実行，3-14

か

拡張ファイル・システム，2-13
カプセル化機能の構成，5-9
環境変数，4-11
  MicroFocus COBOL コンパイラ，4-11
  ORACLE_DOC，xvi
  TNS_ADMIN，5-2
環境変数 ADA_PATH，1-6
環境変数 CLASS_PATH，1-6
環境変数 DISPLAY，1-6
環境変数 EPC_DISABLED，1-3
環境変数 HOME，1-6
環境変数 LANG，1-6
環境変数 LANGUAGE，1-6
環境変数 LD_LIBRARY_PATH，1-7
環境変数 LDPATH，1-7
環境変数 LIBPATH，1-7
環境変数 LP_DEST，1-6
環境変数 LPDEST，1-7
環境変数 NLS_LANG，1-3
環境変数 ORA_NLS，1-3
環境変数 ORA_TZFILE，1-3
環境変数 ORACLE_BASE，1-3
環境変数 ORACLE_DOC，xvi
環境変数 ORACLE_HOME，1-4
環境変数 ORACLE_PATH，1-4
環境変数 ORACLE_SID，1-2，1-4
環境変数 PATH，1-7
環境変数 PRINTER，1-7
環境変数 SHELL，1-7
環境変数 SHLIB_PATH，1-8
環境変数 SQLPATH，1-4
環境変数 TMPDIR，1-8
環境変数 TWO_TASK，1-5
環境変数 XENVIRONMENT，1-8
関連ドキュメント，xvi

き

キャッシュ・サイズ，2-15
共通の環境
  設定，1-8
緊密共有メモリー，1-14

く

クラスタ・ファイル・システム，2-13

け

軽量タイマー，B-4
言語，1-3

こ

構成
  カプセル化機能，5-9
  マスター・エージェント，5-9
構成ファイル
  Oracle Net，5-2
  Oracle Net Services，5-2
  プリコンパイラ，4-2
コマンド・インタプリタ，1-7
コマンドライン SQL*Plus の使用，3-13
コマンドライン SQL の管理，3-2

さ

再リンク，1-12

し

シグナル・ハンドラ
  使用，4-24
シグナル・ルーチン，4-25
  例，4-25
システム・エディタ
  SQL*Plus，3-13
システム時刻，1-10
指定配置最適化，D-2
ジャーナル・ファイル・システム，2-13
シャドウ・プロセス，1-19
初期化パラメータ
  BACKGROUND_DUMP_DEST，1-23，2-19
  BITMAP_MERGE_AREA_SIZE，1-23
  COMMIT_POINT_STRENGTH，1-23
  CONTROL_FILES，1-23
  CREATE_BITMAP_AREA_SIZE，1-23
  DB_BLOCK_SIZE，1-13，1-23

DB_CACHE_SIZE，1-13
DB_FILE_DIRECT_IO_COUNT，1-23
DB_FILE_MULTIBLOCK_READ_COUNT，1-23
HASH_AREA_SIZE，1-23
HASH_MULTIBLOCK_IO_COUNT，1-23
JAVA_POOL_SIZE，1-13，1-23
LARGE_POOL_SIZE，1-13
LOCK_SGA，1-23
LOG_ARCHIVE_DEST，1-23
LOG_ARCHIVE_FORMAT，1-23
LOG_BUFFER，1-24
LOG_BUFFERS，1-13
LOG_CHECKPOINT_INTERVAL，1-24
MAX_DISPATCHERS，1-24
MAX_DUMP_FILE，2-18
MAX_SERVERS，1-24
NLS_LANGUAGE，1-24
NLS_TERRITORY，1-24
OBJECT_CACHE_MAX_SIZE_PERCENT，1-24
OBJECT_CACHE_OPTIMAL_SIZE，1-24
OPEN_CURSORS，1-24
OS_AUTHENT_PREFIX，1-24
PROCESSES，1-24
SHARED_POOL_SIZE，1-13，1-24
SHARED_SERVERS，1-24
SORT_AREA_SIZE，1-24
USER_DUMP_DEST，2-18

す

スクリプト
start_peer，5-10
スレッドのサポート，4-24
スワップ領域，2-10
チューニング，2-10

せ

制限事項（SQL*Plus），3-15
ウィンドウのサイズ変更，3-15
リターン・コード，3-15
セキュリティ，1-19，1-20
2 タスク構造，1-19
UNIX の機能，1-19
グループ・アカウント，1-19
ファイル所有権，1-19
設定ファイル
SQL*Plus，3-2

ち

チューニング，2-10
I/O ボトルネック，2-13
UDP，A-21
アーカイブ・バッファ，A-6
ディスク I/O，2-13
トレース・ファイルおよびアラート・ファイル，2-15，2-18
メモリー管理，2-10
リソースの競合，2-15
チューニング・ツール
AIX システム管理インタフェース・ツール，2-7
free，2-6
Glance/UX，2-9
iostat，2-5
lsps，2-6
mpstat，2-6
Programmer's Analysis Kit（HP PAK），2-10
PTX Agent，2-8
PTX Manager，2-8
sar，2-4
swap，2-6
swapinfo，2-6
swapon，2-6
vmstat，2-3

つ

ツール，2-3

て

ディスク
パフォーマンスの監視，2-14
ディスク I/O
I/O スレーブ，A-13
チューニング，2-13
ファイル・システム・タイプ，2-13
データベース
ブロック・サイズ，A-6
データベースの変更，1-9
データベース・ファイル，1-20
デバッガ・プログラム，4-4
デモ

PL/SQL，1-28
SQL*Loader，1-26
プリコンパイラ，1-30
プロシージャ・オプション、PL/SQL，1-28
デモ・プログラム
Oracle Call Interface，4-20
Pro*C/C++，4-9
Pro*COBOL，4-13
Pro*FORTRAN，4-16
SQL*Module for Ada，4-18
デモンストレーション表
SQL*Plus，3-4
削除，3-4
手動で作成，3-4

と

ドキュメント
以前のリリースからの移行およびアップグレード，xvi
インストール位置，xvi
管理およびチューニング，xvi
関連，xvi
ナビゲーション・ファイル，xvi
表示，xvi
特殊アカウント，1-18
トレース・ファイルおよびアラート・ファイル
アラート・ファイル，2-19
使用，2-15，2-18
トレース・ファイル名，2-18

は

配布ソフトウェア，1-4
バッファ・キャッシュ
チューニング，A-2
バッファ・キャッシュ・サイズ，2-15
チューニング，2-15
バッファ・マネージャ，2-10
パラメータ
CONFIG_WATCHDOG_NOWAYOUT，F-3
CREATE CONTROLFILE，1-17
CREATE DATABASE，1-17
max_async_req，D-7
max_objs，D-7
max_sessions，D-7
MAXDATAFILES，1-17
maxfree，A-2
MAXLOGFILES，1-17
MAXLOGHISTORY，1-17
MAXLOGMEMBERS，1-17
maxperm，A-2，A-4
minfree，A-2
minperm，A-2，A-4
msg_size，D-7
OS_AUTHENT_PREFIX，1-20
rad_gh_regions，D-4
rdg_max_auto_msg_wires，D-7
SCHED_NOAGE，B-3
SGA_MAX_SIZE，1-14
shm_allocate_striped，D-4
SHM_MAX，1-13
SHM_SEG，1-13
SHMMAX，1-13
SHMSEG，1-13
TIMED_STATISTICS，D-16

ふ

ファイル
dbhome，1-9
root.sh，1-9
トレース・ファイル，2-18
ファイル・システム，2-13
複数のシグナル・ハンドラ，4-25
プリコンパイラ
ireclen および oreclen の値，4-4
大文字から小文字への変換，4-4
概要，4-2
シグナル，4-25
デモの実行，1-30
ベンダー提供のデバッガ・プログラム，4-4
プリコンパイラ構成ファイル，4-2
ブロック・サイズ，A-6
チューニング，2-12
プロトコル，5-5

へ

ページアウト・アクティビティ，2-12
ページング，2-12
過剰な，A-2，A-5
不十分，A-5
ページング領域，2-10

チューニング，2-10，2-12

ま

マスター・エージェントの構成，5-9

め

メモリー
競合，A-2
チューニング，2-10
メモリー管理，2-10
スワップ領域，2-10
ページングの制御，2-12

ゆ

ユーザー・プログラム
Pro*COBOL，4-14
SQL*Module for Ada，4-19
ユーザー割込みハンドラ，4-25

り

リアルタイム・クロック，D-16
リソースの競合
チューニング，2-15