

# Oracle8*i*

for AIX-Based Systems 管理者リファレンス

リリース 8.1.7

2001 年 1 月

部品番号 : J02424-01

ORACLE®

---

Oracle8i for AIX-Based Systems 管理者リファレンス , リリース 8.1.7

部品番号 : J02424-01

原本名 : Oracle8i Administrator's Reference Release3(8.1.7) for AIX-Based Systems

原本部品番号 : A85348-01

Copyright © 1996, 2000, Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

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

当プログラムのリバース・エンジニアリング等は禁止されています。

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

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

危険な用途への使用について

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

<b>はじめに</b> .....	vii
用途 .....	vii
対象読者 .....	vii
Oracle8i Workgroup Server および Oracle8i Enterprise Edition .....	vii
このマニュアルで使用する表記規則 .....	viii
コマンドの構文 .....	viii
インストールしたドキュメントの表示 .....	ix
Oracle Information Navigator .....	ix
参照マニュアル .....	x
Oracle サービスおよびサポート .....	x
 <b>1 Oracle8i の管理</b>	
概要 .....	1-2
<b>環境変数</b> .....	1-2
Oracle8i 環境変数 .....	1-3
UNIX 環境変数 .....	1-5
共通の環境設定 .....	1-7
現行セッションでの環境変数値の設定およびエクスポート .....	1-8
システム時刻の設定 .....	1-8
<b>ネットワーク実行ファイル</b> .....	1-9
<b>システム・グローバル領域</b> .....	1-10
SGA のサイズの計算 .....	1-10
<b>Oracle8i のメモリー要件および使用量</b> .....	1-11
<b>サーバー・リソースの制限</b> .....	1-12
<b>データベースの制限</b> .....	1-12
<b>特殊アカウントおよび特殊グループ</b> .....	1-13

<b>セキュリティ</b> .....	1-14
グループとセキュリティ .....	1-15
Server Manager コマンドのセキュリティ .....	1-15
データベース・ファイルのセキュリティ .....	1-16
リモート・パスワード .....	1-16
initSID.ora ファイルのカスタマイズ .....	1-18
<b>埋込み PL/SQL ゲートウェイ</b> .....	1-20
概要 .....	1-20
埋込み PL/SQL ゲートウェイのインストール .....	1-21
<b>Oracle HTTP Server</b> .....	1-21
Oracle HTTP Server のステータス .....	1-22
Oracle HTTP Server のログ・ファイル .....	1-23
<b>デモンストレーション・ファイル</b> .....	1-23
SQL*Loader のデモンストレーション .....	1-23
SQL*Loader の管理 .....	1-24
PL/SQL デモンストレーション .....	1-25
データベースの例 .....	1-27

## 2 Oracle8i for AIX-Based Systems のチューニング

<b>AIX-Based Systems のパフォーマンス・ツール</b> .....	2-2
AIX-Based Systems の Base Operation System ツール .....	2-2
AIX-Based Systems の Performance Toolbox .....	2-2
AIX-Based Systems System Management Interface Tool .....	2-3
<b>Oracle のパフォーマンス・ツール</b> .....	2-4
Utlbstat/Utlestat による統計収集 .....	2-4
動的パフォーマンス表 .....	2-4
SQL トレース .....	2-4
Oracle Enterprise Manager Performance Tuning Pack .....	2-5
<b>Oracle8i のチューニング</b> .....	2-6
Oracle8i のチューニング手順 .....	2-6
<b>メモリーおよびページング</b> .....	2-7
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 .....	2-7
AIX-Based Systems でのファイル・バッファ・キャッシュのチューニング .....	2-8
十分なページング領域の割当て .....	2-9
ページングの制御 .....	2-9

データベース・ブロック・サイズの設定 .....	2-10
アーカイバ・バッファのチューニング .....	2-10
SGA サイズのチューニング .....	2-11
I/O バッファおよび SQL*Loader .....	2-11
<b>ディスク I/O の問題</b> .....	2-12
索引を表から分離する .....	2-12
REDO ログを専用ディスク・デバイスに配置する .....	2-12
ホット・ファイルを他のディスクに移動する .....	2-12
ホット・ファイルへの I/O を削減する .....	2-12
データベースの過度の断片化チェック .....	2-13
AIX 論理ボリューム・マネージャ .....	2-14
異なるデータベース・ファイルの同一ディスクへの配置の回避 .....	2-15
ジャーナル・ファイル・システム (JFS) と RAW パーティション .....	2-15
非同期 I/O の使用 .....	2-17
I/O スレーブ .....	2-18
DB_FILE_MULTIBLOCK_READ_COUNT の使用 .....	2-19
RAID 機能の使用 .....	2-19
後書きの使用 .....	2-19
順次ファイルの先読みのチューニング .....	2-20
ディスク I/O の歩調合せ .....	2-20
ディスク形状の考慮事項 .....	2-21
リモート I/O の最小化 .....	2-21
VSD キャッシュ・バッファを使用しない .....	2-21
<b>CPU のスケジューリングおよびプロセスの優先順位</b> .....	2-22
プロセスのランタイム・スライスの変更 .....	2-22
SMP マシンでのプロセッサ・バインディングの使用 .....	2-22
<b>その他のチューニングのヒント</b> .....	2-25
UDP のチューニング .....	2-25
RAW デバイスのバックアップ .....	2-25
パラレル・リカバリの使用 .....	2-26

### 3 SQL\*Plus の管理

<b>SQL*Plus の管理</b> .....	3-2
設定ファイルの使用 .....	3-2
PRODUCT_USER_PROFILE 表の使用 .....	3-2

デモンストレーション表の使用 .....	3-3
ヘルプ機能の使用 .....	3-4
<b>SQL*Plus の使用</b> .....	3-5
SQL*Plus からのシステム・エディタの使用 .....	3-5
SQL*Plus からのオペレーティング・システム・コマンドの実行 .....	3-6
SQL*Plus への割込み .....	3-6
SPOOL コマンドの使用 .....	3-6
<b>SQL*Plus の制限事項</b> .....	3-6
ウィンドウのサイズ変更 .....	3-6
リターン・コード .....	3-6

## 4 Oracle Message Broker の構成

JDK のパスの更新 .....	4-2
ディレクトリのインストールおよび構成 .....	4-2
ディレクトリ構成 .....	4-3
<b>Oracle Advanced Queuing の構成（オプション）</b> .....	4-9
Oracle Internet Directory の停止 .....	4-9
データベース・サーバーおよびリスナーの停止 .....	4-10
データベース・パラメータの構成 .....	4-10
データベース・サーバーおよびリスナーの再起動 .....	4-12
Oracle Internet Directory の起動 .....	4-13
AQ の初期化 .....	4-13
AQ キュー・テーブルの削除 .....	4-15
<b>MQSeries のインストール・タスク（オプション）</b> .....	4-15
<b>TIBCO のインストール・タスク（オプション）</b> .....	4-15
<b>Asynchronous Component Invocation のインストール・タスク（オプション）</b> .....	4-16
<b>OMB インスタンスの作成および Oracle Message Broker の使用</b> .....	4-16

## 5 Oracle プリコンパイラおよび Oracle コール・インタフェースの使用

<b>Oracle プリコンパイラの概要</b> .....	5-2
プリコンパイラ構成ファイル .....	5-2
すべてのプリコンパイラに共通の問題 .....	5-3
静的および動的リンク .....	5-3
Oracle 共有ライブラリ .....	5-4
<b>Pro*C/C++ プリコンパイラ</b> .....	5-4

Pro*C/C++ デモ・プログラム .....	5-5
Pro*C/C++ ユーザー・プログラム .....	5-6
<b>Pro*COBOL プリコンパイラ .....</b>	<b>5-7</b>
<b>Pro*COBOL の環境変数の設定 .....</b>	<b>5-7</b>
Pro*COBOL デモ・プログラム .....	5-8
Pro*COBOL ユーザー・プログラム .....	5-9
FORMAT プリコンパイラ・オプション .....	5-10
<b>Pro*FORTRAN プリコンパイラ .....</b>	<b>5-10</b>
Pro*FORTRAN デモ・プログラム .....	5-10
Pro*FORTRAN ユーザー・プログラム .....	5-11
<b>SQL*Module for Ada の使用 .....</b>	<b>5-12</b>
SQL*Module for Ada デモ・プログラムの使用 .....	5-12
SQL*Module Ada ユーザー・プログラムの作成 .....	5-13
<b>Oracle コール・インタフェース .....</b>	<b>5-13</b>
OCI デモ・プログラム .....	5-13
OCI ユーザー・プログラム .....	5-14
カスタム Make ファイル .....	5-15
未定義シンボルの修正 .....	5-15
マルチスレッド・アプリケーション .....	5-16
シグナル・ハンドラの使用方法 .....	5-16
<b>XA 機能 .....</b>	<b>5-19</b>

## 6 Net8 の構成

<b>主な Net8 製品および特徴 .....</b>	<b>6-2</b>
Net8 ファイルおよびユーティリティ .....	6-2
Oracle Connection Manager .....	6-4
サーバー・モデル .....	6-4
Oracle Names .....	6-5
Net8 Assistant .....	6-5
<b>Net8 プロトコルのサポート .....</b>	<b>6-6</b>
ADDRESS の指定 .....	6-6
<b>BEQ プロトコル .....</b>	<b>6-7</b>
BEQ ADDRESS の指定 .....	6-7
<b>IPC プロトコル .....</b>	<b>6-8</b>
IPC ADDRESS の指定 .....	6-9
<b>RAW プロトコル .....</b>	<b>6-9</b>

TCP/IP プロトコル .....	6-10
APPC/LU6.2 プロトコル .....	6-11
APPC/LU6.2 の ADDRESS 指定 .....	6-11
Oracle Enterprise Manager (OEM) .....	6-12
Oracle SNMP での Oracle Intelligent Agent .....	6-13
Oracle Advanced Security .....	6-14
参照ドキュメント .....	6-14

## 7 Oracle データ・オプション・デモの実行

Oracle8i interMedia .....	7-2
Text .....	7-2
Audio、Video および Image .....	7-2
Locator .....	7-6
Oracle8i Time Series デモ .....	7-8
Oracle8i Visual Information Retrieval .....	7-9
Oracle8i Spatial .....	7-10

## A Optimal Flexible Architecture

OFA の概要 .....	A-2
OFA に準拠したデータベースの特長 .....	A-2
UNIX での OFA の実装 .....	A-4
マウント・ポイントの使用 .....	A-4
ディレクトリの名前 .....	A-5
ファイルの名前 .....	A-7
表領域の名前 .....	A-9
OFA 構造に基づいた Oracle ファイル .....	A-10
OFA ファイルのマッピング .....	A-10
RAW デバイスのサイズ .....	A-12
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 .....	A-13
ディレクトリ構造 .....	A-14

## 索引



---

# はじめに

## 用途

このマニュアルおよび『Oracle8i for AIX-Based Systems インストレーション・ガイド』は、AIX-Based Systems システムでの Oracle8i リリース 8.1.7 のインストールおよび構成方法を記載しています。このマニュアルの内容は、Oracle8i 関連マニュアル・セットの製品情報を補足しています。

## 対象読者

このマニュアルは、AIX-Based Systems システムで Oracle8i リリース 8.1.7 を管理する方を対象としています。

## Oracle8i Workgroup Server および Oracle8i Enterprise Edition

特に指示がない限り、このマニュアルで説明している特長や機能は、Oracle8i Workgroup Server および Oracle8i Enterprise Edition に共通です。

## このマニュアルで使用する表記規則

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。
大カッコ [ ]	大カッコで囲まれた語は、キーの名前を表します（たとえば、[Return] を押す）。ただし、コマンド構文で大カッコを使用する場合は意味が異なります。
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。

## コマンドの構文

UNIX のコマンド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 \$ 文字はプロンプトで、入力する必要はありません。UNIX では大文字と小文字が区別されるため、このマニュアルの表記規則は、他の Oracle マニュアルで使用されている表記規則と多少異なります。

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、またはバックスラッシュを付けずに 1 行で入力します。 <code>dd if=/dev/rdisk/c0t1d0s6 of=/dev/rst0 bs=10b \</code> <code>count=10000</code>
中カッコ { }	中カッコは、必ず選択する項目を表します。 <code>.DEFINE {macro1}</code>
大カッコ [ ]	大カッコは、任意に選択する項目を表します。 <code>cvtcrt termname [outfile]</code>  ただし、本文で大カッコを使用する場合は意味が異なります。
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。 <code>CHKVAL fieldname value1 value2 ... valueN</code>
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。 <code>library_name</code>
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。 <code>SIZE filesize [K M]</code>

# インストールしたドキュメントの表示

Oracle8i AIX-Based Systems のドキュメントには、このマニュアルおよび『Oracle8i for AIX-Based Systems インストレーション・ガイド』があります。ドキュメントは、HTML および PDF（Adobe Portable Document Format を表示するには Acrobat Reader が必要）形式でインストールできます。AIX 固有のドキュメント・ファイルは、Oracle8i CD-ROM からインストールできます。Oracle8i 関連ドキュメント・ファイルは、CD-ROM 版の Oracle8i 関連マニュアル・セットからインストールできます。ドキュメント・ファイルの位置は、次の規則に従って決定されます。

- 環境変数に ORACLE\_DOC が定義されている場合、定義されたディレクトリにファイルがインストールされます。
- ORACLE\_DOC は定義されていないが、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。
- 環境変数に ORACLE\_DOC または ORACLE\_BASE のどちら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、index.htm または products.htm のいずれかをブラウザで表示します（後者は、フレーム表示可能なブラウザが不要です）。紙マニュアルが必要な場合は、PDF ファイルを印刷してください。

## Oracle 製品のドキュメント

Oracle8i 製品のドキュメントは、CD-ROM 版の Oracle8i 関連マニュアル・セットにあります。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。

## Oracle Information Navigator

Oracle Information Navigator は、Oracle オンライン・ドキュメントとともに提供される Java ベースの検索およびナビゲーション・ユーティリティです。Java 対応のブラウザを使用して、CD-ROM の最上位層にある index.htm ファイルをオープンすると、Information Navigator が自動的に起動されます。Information Navigator は Oracle 製品ドキュメントに使用でき、CD-ROM またはインストールしたファイルのどちらからでもドキュメントを読むことができます。

## 参照マニュアル

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、インストールを始める前に、『Oracle8i 概要』の第 1 章をお読みください。

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。

- 『Oracle8i for AIX-Based Systems インストレーション・ガイド』
- 『Oracle8i 管理者ガイド』
- 『Oracle8i Net8 管理者ガイド』
- 『Oracle8i パフォーマンスのための設計およびチューニング』

前回のリリースの Oracle Server からの移行およびアップグレードの詳細は、『Oracle8i 移行ガイド』を参照してください。

Oracle Workflow のインストールについては、『Oracle Workflow インストレーション補足』を参照してください。

Oracle Internet Directory のインストールについては、『Oracle Internet Directory 管理者ガイド』を参照してください。

## Oracle サービスおよびサポート

オラクル社は、企業システム・ソリューションをサポートする、Oracle Education コース、Oracle Consulting サービス、Oracle Worldwide Customer Support などの幅広いサービスを提供しています。Oracle 製品およびグローバル・サービスの詳細は、インターネットで次の URL を参照してください。

<http://www.oracle.com>

ここでは、選択されたサービスの URL を示します。

### Oracle サポート・サービス

テクニカル・サポートの情報は、次の URL に示されています。

<http://www.oracle.com/support>

テンプレートをご使用になると、電話をかける前に、問題に関する情報を用意することができます。CSI 番号（該当する場合）、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）も必要です。

## **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.com/support/>

## **教育およびトレーニング**

トレーニング情報およびスケジュールには、次の URL から参照できます。

<http://education.oracle.com>



---

# Oracle8i の管理

この章では、AIX-Based Systems で Oracle8i 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
- ネットワーク実行ファイル
- システム・グローバル領域
- Oracle8i のメモリー要件および使用量
- サーバー・リソースの制限
- データベースの制限
- 特殊アカウントおよび特殊グループ
- セキュリティ
- 埋込み PL/SQL ゲートウェイ
- Oracle HTTP Server
- デモンストレーション・ファイル

## 概要

Oracle8i で作業するには、環境変数、パラメータ、メモリーおよびユーザー設定を構築する必要があります。この章では、AIX-Based Systems の様々な設定について説明します。

Oracle8i のファイルおよびプログラムでは、疑問符「?」は ORACLE\_HOME の値を表します。たとえば、Oracle8i では、次の SQL 文中の疑問符を ORACLE\_HOME のフルパス名として使用します。

```
alter tablespace TEMP add datafile '?/dbs/dbs2.dbf' size 2M
```

アットマーク「@」は、\$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。

```
alter tablespace tablespace_name add datafile 'dbsfile@.dbf'
```

## 環境変数

この項では、通常使用される Oracle8i および UNIX 環境変数について説明します。

現行の環境変数の値を表示するには、echo コマンドを使用します。たとえば、ORACLE\_SID の値を表示するには、次のように入力します。

```
$ echo ORACLE_SID
```

これらの環境変数のいくつかは、Oracle8i をインストールする前に設定しておく必要があります。このような環境変数については、『Oracle8i for AIX-Based Systems インストレーション・ガイド』を参照してください。



## Oracle8i 環境変数

表 1-1 に、Oracle8i 環境変数の機能、構文および例を示します。

**表 1-1 UNIX における Oracle8i 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 または 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は、キャラクタ・セットを自動的に変換します。  設定値の詳細は、『Oracle8i NLS ガイド』を参照してください。
	構文	<code>language_territory.characterset</code>
	例	<code>french_france.we8dec</code>
ORA_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示します。
	例	<code>\$ORACLE_HOME/ocommon/nls/admin/data</code>
ORACLE_BASE	機能	OFA 準拠のデータベースに対応した、Oracle ディレクトリ構造のベースを指定します。
	構文	<code>directory_path</code>
	例	<code>/u01/app/oracle</code>
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	<code>directory_path</code>
	例	<code>\$ORACLE_BASE/product/8.1.7</code>
ORACLE_PATH	機能	*.sql、*.frm、*.rpt などの Oracle アプリケーションが使用するファイルの検索パス名を指定します。そのファイルのフルパス名が指定されていない場合、またはそのファイルが現行のディレクトリにない場合、Oracle アプリケーションは、ファイルを見つけるために、ORACLE_PATH を使用します。
	構文	ディレクトリをコロンで区切ったリスト <code>directory1:directory2:directory3</code>
	例	<code>/u01/oracle/adhoc/8.1.7/bin:.</code>

**注意:** 最後にピリオドを付けることによって、検索パスに現行の作業ディレクトリが追加されます。

表 1-1 UNIX における Oracle8i 環境変数（続き）

環境変数	項目	定義
ORACLE_SID	機能	Oracle システム識別子（Oracle System Identifier: SID）を指定します。
	構文	英字で始まる英数字列を指定します。4 文字以内で指定することをお勧めします。詳細は、『Oracle8i for AIX-Based Systems インストール・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 Bourne シェル・スクリプトのトレース状態を切り替えます。T を設定すると、多くの Oracle シェル・スクリプトは set -x フラグがオンの状態で実行されます。
	許容値	T または T 以外
ORAENV_ASK	機能	coraenv または oraenv が、ORACLE_SID または ORACLE_HOME 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
	許容値	NO または NO 以外
SQLPATH	機能	SQL*Plus が login.sql ファイルを検索するディレクトリまたはディレクトリのリストを設定します。
	構文	ディレクトリをコロンで区切ったリスト directory1:directory2:directory3
	例	/home:/home/oracle:/u01/oracle
TNS_ADMIN	機能	Net8 構成ファイルがあるディレクトリを設定します。
	構文	directory_path
	許容値	任意のディレクトリ。詳細は、『Oracle8i for AIX-Based Systems インストール・ガイド』を参照してください。
	例	\$ORACLE_HOME/network/admin
TWO_TASK	機能	tnsnames.ora ファイルで定義されている、Net8 接続文字列記述子のデフォルトの別名を設定します。
	構文	使用可能なネットワーク別名
	許容値	tnsnames.ora ファイルで定義されている有効な Oracle Net8 別名
	例	PRODDB_TCP

**注意：** 環境変数に、Oracle Server プロセスと同じ名前（arch、pmon、dbwr など）は設定しないでください。

## UNIX 環境変数

表 1-2 に、Oracle8i 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 Oracle8i で使用する UNIX 環境変数

環境変数	項目	定義
ADA_PATH	機能	Ada コンパイラがあるディレクトリを指定します。
CLASSPATH	機能	Java 機能により使用されます。この環境変数は、Java アプリケーションによって異なります。デフォルトの設定はありません。また、CLASSPATH には、JRE_Location、\$ORACLE_HOME/product/jlib を含める必要があります。 <b>注意：</b> JRE_Location は、\$ORACLE_HOME/JRE のように定義されます。 詳細は、ご使用の Java アプリケーションの製品ドキュメントを参照してください。
	構文	<i>directory_path</i>
	例	\$ORACLE_HOME/JRE、\$ORACLE_HOME/product/jlib <b>注意：</b> JRE_Location は、\$ORACLE_HOME/JRE のように定義されます。
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、ベンダーの提供する X-Window のドキュメントを参照してください。
	構文	<i>hostname:display</i> <i>hostname</i> はマシン名（IP アドレスまたは別名）、 <i>display</i> はモニター番号です。モニターが 1 つの場合、番号は 0 になります。
	例	135.287.222.12:0 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。
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『Oracle8i for AIX-Based Systems インストレーション・ガイド』を参照してください。

表 1-2 Oracle8iで使用する UNIX 環境変数（続き）

環境変数	項目	定義
LD_LIBRARY_PATH	機能	AIX での Java アプリケーションの実行時に使用されます。
	構文	ディレクトリをコロンで区切ったリスト <i>directory:directory:directory</i>
	例	<i>/usr/lpp/J1.1.8/:\$ORACLE_HOME/lib</i>
LIBPATH	機能	共有オブジェクト・ライブラリを検索するために、共有ライブラリ・ローダーにより、実行時に AIX 上で使用されます。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 <i>directory:directory:directory</i>
	例	<i>/usr/dt/lib:\$ORACLE_HOME/lib</i>
PATH	機能	シェルが実行プログラムの場所を見つけるために使用します。 \$ORACLE_HOME/bin 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 <i>directory:directory:directory</i>
	例	<i>/bin:/usr/bin:/usr/lbin: /usr/bin/X11:\$ORACLE_HOME/bin:\$HOME/bin:.</i> <b>注意:</b> 最後にピリオドを付けることによって、検索パスに現行の作業ディレクトリが追加されます。
PRINTER	機能	AIX システムのデフォルト・プリンタを選択します。
	構文	<i>printer_name</i>
	例	<i>docqms</i>
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	シェルのパス名
	許容値	<i>/bin/sh、/bin/csh、/bin/ksh</i> または AIX-Based Systems に付属のその他のコマンド・インタプリタ
	例	<i>/bin/sh</i>
TERM	機能	Oracle Toolkit II キャラクタ・モード・ツールおよび UNIX ツールが端末タイプを判断する場合に使用します。
	例	<i>vt100</i>
TMPDIR	機能	テンポラリ・ディスク・ファイル用のデフォルト・ディレクトリを指定します。設定すると、テンポラリ・ファイルを作成するツールは、このディレクトリにテンポラリ・ファイルを作成します。

表 1-2 Oracle8iで使用する UNIX 環境変数（続き）

環境変数	項目	定義
XENVIRONMENT	構文	<i>directory_path</i>
	例	/u02/oracle/tmp
	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

Oracle8i では、データベース管理者（DBA）がすべてのユーザーに共通の環境を設定できます。共通の環境を設定することによって、システム管理者とデータベース管理者が Oracle8i システムを変更しやすくなります。

### oraenv コマンド・ファイル

oraenv コマンド・ファイルは、インストール時に作成されます。このファイルには、Oracle 環境変数の値および次の情報が含まれています。

- 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
- Oracle8i データベース間を移動するメカニズム

たとえば、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 によって、それぞれのユーザー・プロファイルが oraenv コマンド・ファイルを呼び出します。

---

**注意：** C シェルの場合は、oraenv のかわりに coraenv コマンドを使用します。

---

### ローカル bin ディレクトリ

oraenv および dbhome を、Oracle ソフトウェアのホーム・ディレクトリではなくローカル bin ディレクトリに入れると、すべてのユーザーがそれらのファイルにアクセスできるようになります。また、これによって、パスが別の ORACLE\_HOME を示すように変更した場合でも、oraenv は引き続き動作することが保証されます。ローカル bin ディレクトリは、次のインストールで実行される root.sh で指定します。AIX の場合、ローカル bin ディレクトリのデフォルト位置は、/usr/lbin です。

## データベース間の移動

あるデータベースまたはインスタンスから別のデータベースまたはインスタンスへ移動するには、oraenv ルーチン呼び出して、プロンプトに対して移動先のデータベースの *sid* を指定します。oraenv コマンド・ファイルのフルパス名を必ず入力してください。次に例を示します。

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

## 現行セッションでの環境変数値の設定およびエクスポート

コマンド `env` は、環境にエクスポートされたものを表示するために使用してください。Bourne シェルおよび Korn シェルでは、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
```

この場合、*test* は環境変数 `ORACLE_SID` に設定する値です。

## 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・ゾーンにしたりできます。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` の値を調べたりする際に使用します。

できるだけ、`TZ` 値は変更しないでください。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た日が変わってしまいます。その場合、`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。

## ネットワーク実行ファイル

\$ORACLE\_HOME/bin ディレクトリ内のシェル・スクリプトの再リンクによって、製品実行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・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。

再リンク・スクリプトは、ORACLE\_HOME にインストールされているものに基づいて Oracle 製品実行ファイルの再リンクを手動で実行します。

再リンクするには、次のように入力します。

```
$ relink parameter
```

**表 1-3 再リンク・スクリプト・パラメータ**

パラメータ	値
all	インストールされたものすべて
oracle	Oracle データベースの実行ファイルのみ
network	net_client、net_server、nau、cman、cnames
client	net_client、otrace、plssql、client_sharedlib
interMedia	ctx、ordimg、ordaud、ordvir、md
precomp	インストールされたすべてのプリコンパイラ
utilities	ユーティリティ
oemagent	oemagent、odg

---

---

**注意：** データベースを再リンクする場合は、Oracle Intelligent Agent および ORACLE\_HOME にあるその他の Oracle プログラムを停止してください。

---

---

## システム・グローバル領域

システム・グローバル領域（SGA）は、共有メモリー内に常駐する Oracle 構造体です。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、十分な共有メモリーが必要です。

AIX の場合、1 つの共有メモリー・セグメントの最大サイズは、256MB です。プロセスは、連続した共有メモリー・セグメントを 10 個まで割り当てることができます。このため、AIX での最大 SGA サイズは 2.5GB です。

---

---

**注意：** Oracle8i の実行中には、環境変数 EXTSHM を設定しないでください。EXTSHM が設定されていると、Oracle8i は動作しません。

---

---

SGA のサイズは、次の `init.ora` パラメータによって決まります。

- DB\_BLOCK\_BUFFERS
- DB\_BLOCK\_SIZE
- JAVA\_POOL\_SIZE
- LARGE\_POOL\_SIZE
- SHARED\_POOL\_SIZE
- SORT\_AREA\_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、マシン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 のサイズの計算

次のいずれかの方法で、SGA サイズを判断できます。

- インスタンスあたりの SGA の適切なサイズは、次の式で計算します。

```
(DB_BLOCK_BUFFERS × DB_BLOCK_SIZE)
+ SORT_AREA_SIZE
+ SHARED_POOL_SIZE
+ LOG_BUFFER
+ LARGE_POOL_SIZE
+ JAVA_POOL_SIZE
```

- 実行中のデータベースの SGA サイズを表示するには、SQL\*Plus の `show sga` コマンドを使用します。結果はバイト単位で表示されます。
- データベース・システムを起動するときに、SGA のサイズを決定します。ヘッダー「Total System Global Area」の横に SGA サイズが表示されます。



## Oracle8i のメモリー要件および使用量

システム上のユーザー数を決定するために、メモリー要件および使用量を計算します。これは、物理メモリー要件およびスワップ領域要件を決定する場合にも役立ちます。次の式を使用して、メモリー要件を見積もります。

$$\begin{aligned} & \text{<Oracle 実行ファイルのテキストのサイズ>} \\ + & \text{<SGA のサイズ>} \\ + & n \times ( \text{<Tool 実行ファイルのプライベート・データ・セクションのサイズ>} \\ & \quad + \text{<Oracle 実行ファイルの未初期化データ・セクションのサイズ>} \\ & \quad + \text{<スタック用に 8192 バイト>} \\ & \quad + \text{<プロセス・ユーザー領域用に 2048 バイト>} ) \end{aligned}$$

この場合、 $n$  はバックグラウンド・プロセスの数を示します。

各クライアント / サーバー接続に対して、次の式を使用して仮想メモリー要件を見積もります。

$$\begin{aligned} & \text{<Oracle 実行ファイルのデータ・セクションのサイズ>} \\ + & \text{<Oracle 実行ファイルの未初期化データ・セクションのサイズ>} \\ + & \text{<スタック用に 8192 バイト>} \\ + & \text{<プロセス・ユーザー領域用に 2048 バイト>} \\ + & \text{<アプリケーションに必要なカーソル領域>} \end{aligned}$$

実行ファイルのテキスト・サイズ、プライベート・データ・セクションのサイズ、および未初期化データ・セクションのサイズ（または DSS）を見積もるには、`size` コマンドを使用します。Oracle 実行ファイル・テキストは共有されているため、プログラムが何度起動されても、プログラム・テキストがカウントされるのは 1 回のみです。

実行中のデータベースにユーザーが接続している状態で、実際の Oracle 物理メモリーの使用量を計算するには、`ps -elf` コマンドを使用します。すべてのフロント・エンド、サーバー、およびバックグラウンドの Oracle プロセス・エントリが検索されます。エントリごとに、「SZ」列の値を合計します。

**参照：** `ps` コマンドで使用できるスイッチのリストについては、AIX-Based Systems の man ページまたはドキュメントを参照してください。

`ps` コマンドは、プロセス・サイズをページで返します。システムのページ・サイズは、アーキテクチャによって異なります。`pagesize` コマンドを使用して、サイズが 4096 バイトか、または 8192 バイトかを確認します。各プロセスについて、SZ 値とページ・サイズを掛けます。

最後に、Oracle 実行ファイルのテキスト・サイズと、システムで実行中のその他のすべての Oracle Tool 実行ファイルのテキスト・サイズを、その小計に加算します。実行ファイルが何度起動されても、実行ファイルのサイズがカウントされるのは 1 回のみです。

# サーバー・リソースの制限

AIX では、親プロセスからリソースの制限を継承します。この制限は、ユーザー・プロセスに対して実行される Oracle8i のシャドウ・プロセスに適用されます。AIX のデフォルトのリソース制限は、Oracle8i のどのシャドウ・プロセスまたはバックグラウンド・プロセスに対しても十分高い値です。ただし、この制限を低くすると、Oracle8i システムに影響が出る場合があります。

ファイル・システム上で 1GB を超える Oracle データ・ファイルを使用する場合には、すべてのユーザーのデフォルトのファイル・サイズ制限値を増やす必要があります。詳細は、『Oracle8i for AIX-Based Systems インストレーション・ガイド』の「サイズが大きなファイル」を参照してください。

Oracle ユーザー用に設定したディスクの割当て制限によって、Oracle8i システムの操作性が悪くなる場合があります。ディスクの割当て制限を設定する前に、Oracle8i のデータベース管理者および AIX のシステム管理者に問い合せてください。

# データベースの制限

表 1-4 に、CREATE DATABASE 文または CREATE CONTROL FILE 文のパラメータの最大値およびデフォルト値を示します。

**注意：** それらのパラメータ間の相互依存によって、正当な値に影響を与える場合があります。

表 1-4 CREATE DATABASE 文および CREATE CONTROL FILE 文のパラメータ

パラメータ	デフォルト値	最大値
MAXDATAFILES	30	16383
MAXINSTANCES	1	7
MAXLOGFILES	16	16512
MAXLOGMEMBERS	2	5
MAXLOGHISTORY	100	31676

Oracle 固有のファイル・サイズ制限の詳細は、表 1-5 を参照してください。

**表 1-5 Oracle 固有のファイル・サイズ制限**

ファイル・タイプ	最大サイズ
db_block_size = 2048 の場合のデータ・ファイル	8,589,932,544
db_block_size = 4096 の場合のデータ・ファイル	17,179,865,088
db_block_size = 8192 の場合のデータ・ファイル	34,359,730,176
db_block_size = 16384 の場合のデータ・ファイル	68,719,460,352
インポート / エクスポート・ファイル	2,147,483,647
SQL*Loader	2,147,483,647

## 特殊アカウントおよび特殊グループ

Oracle Server では、特殊アカウントが必要です。表 1-6 に特殊 UNIX アカウント、表 1-7 に特殊 Oracle Server アカウント、表 1-8 に特殊グループ・アカウントを示します。

**表 1-6 特殊 UNIX アカウント**

oracle	Oracle ソフトウェア所有者は、Oracle8i ソフトウェアを所有するアカウントを示します。このメンテナンス・アカウントには、データベースを作成、起動、停止、または INTERNAL として接続するための DBA 権限が必要です。Oracle ソフトウェア所有者は、スーパーユーザーであってはけません。
root	root ユーザーは、最高の権限（スーパーユーザー権限）が与えられた特殊 UNIX アカウントです。このアカウントを使用することによって、UNIX カーネルを構成したり、ネットワーク・ソフトウェアを構成およびインストールしたりできます。また、ユーザー・アカウントおよびグループを作成できます。

**表 1-7 特殊 Oracle Server アカウント**

SYS	インストール時に自動的に作成され、DBA 権限が与えられた標準的な Oracle8i アカウントです。SYS アカウントは、データ・ディクショナリのすべてのベース表を所有します。このアカウントは DBA が使用します。
SYSTEM	インストール時に自動的に作成され、DBA 権限が与えられた標準的な Oracle8i アカウントです。SYSTEM ユーザーで追加の表またはビューを作成できます。DBA は SYSTEM としてログインし、データベースを監視またはメンテナンスします。

表 1-8 特殊グループ・アカウント

dba グループ	Oracle ソフトウェア所有者は、dba グループの唯一の必須メンバーです。dba グループには、他の UNIX ユーザーを追加できます。このグループのメンバーは、SQL*Plus の特殊権限機能にアクセスできます。アカウントが dba グループのメンバーでない場合、INTERNAL として接続するためにパスワードを入力するか、または SQL*Plus のその他の管理機能へのアクセス権限を取得する必要があります。デフォルトの OSDBA グループは、dba です。
oinstall グループ	ORACLE_HOME では、Oracle8i をインストールするすべてのユーザーは同じ UNIX グループに属している必要があります。マシン上のすべての ORACLE_HOME は、OUI Inventory を共有します。また、グループの書き込み権限もあります。主グループとして oinstall でインストールすることをお勧めします。
oper グループ	オプションで作成可能な UNIX グループです。メンバーには、データベースの OPERATOR 権限が与えられます。DBA 権限の一部を制限したものが OPERATOR 権限です。
root グループ	root グループのメンバーになれるのは root ユーザーのみです。
hagsuser グループ	Oracle Parallel Server の場合、Oracle ユーザーは、hagsuser グループのメンバーである必要があります。

## セキュリティ

Oracle8i では、UNIX オペレーティング・システムのいくつかの機能を使用し、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。

Oracle8i 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、『Oracle8i 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、セキュリティを高めるために、すべてのユーザー（UNIX 用語では、other）が実行できるものと、DBA のみが実行できるものの 2 つのグループに分かれます。次のようにして、セキュリティを高めることをお勧めします。

- oinstall という名前のグループを作成します。oinstall グループは、OUI oraInventory を所有し、Oracle8i システムのインストールおよびアップグレードを担当します。グループ名は任意に指定できますが、このマニュアルでは oinstall グループとします。すべての oracle アカントは、oinstall グループをプライマリ・グループとする必要があります。
- Oracle8i をインストールする前に、データベース管理者の UNIX グループ（特殊なデータベース権限を保持する）を作成します。グループ名は任意に指定できますが、このマニュアルでは dba グループとします。

oracle アカントのプライマリ・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 アカントのみです。

## Server Manager コマンドのセキュリティ

Server Manager へのアクセスを制限することをお勧めします。STARTUP、SHUTDOWN および CONNECT INTERNAL のシステム権限にアクセスできるのは、Oracle ソフトウェア所有者および dba グループのメンバーのみにしてください。

---

---

**警告：** システム権限が与えられた文は、正しく使用しないとデータベースを破壊する可能性があります。dba グループ以外のユーザーでも、パスワードを知っていれば、INTERNAL として接続できるので注意してください。

---

---

## データベース・ファイルのセキュリティ

Oracle8i のインストールに使用するユーザー ID は、データベース・ファイルの所有者である必要があります。デフォルトのユーザー ID は、Oracle ソフトウェア所有者です。これらのファイルの認可を、読取りおよび書込み権限は所有者のみに、読取り専用権限は group や other ユーザーに与えられるように設定します。

Oracle ソフトウェア所有者は、データベース・ファイルを含むディレクトリを所有する必要があります。セキュリティを高めるために、group および other ユーザーの読取り権限を取り消してください。

保護されているデータベース・ファイルにアクセスするには、Oracle プログラムでそのセット・ユーザー ID (setuid) ビットをオンにする必要があります。

Oracle Universal Installer は、Oracle 実行ファイルの権限を自動的に次のように設定します。

```
-rwsr-s--x 1 oracle dba 443578 Mar 10 23:03 oracle
```

ユーザーの実行フィールドにある「s」は、Oracle プログラムを実行する場合に、そのプログラムを起動した実際のユーザー ID に関係なく、そのプログラムの事実上のユーザー ID は、oracle になることを示しています。

これを手動で設定する必要がある場合は、次のように入力します。

```
$ chmod 6751 $ORACLE_HOME/bin/oracle
```

## リモート・パスワード

オペレーティング・システム・アカウントがなくても、パーソナル・コンピュータ (PC) などのリモート・マシンからデータベースを管理できます。この場合、ユーザーは、orapwd ユーティリティで作成および管理される Oracle8i パスワード・ファイルを使用して検証されます。オペレーティング・システム・アカウントをサポートするシステムでは、パスワード・ファイル検証も行えます。

ローカル・パスワード・ファイルは \$ORACLE\_HOME/dbs ディレクトリにあり、1つのデータベースのユーザー名およびパスワード情報が入っています。1つのマシンに複数の \$ORACLE\_HOME ディレクトリがある場合、それぞれにパスワード・ファイルがあります。データベースで、パスワード・ファイルを使用可能にするには、initSID.ora パラメータ remote\_login\_passwordfile を EXCLUSIVE に設定します。

### リモート PC からデータベースへのアクセス

Oracle8i パスワード・ファイルがあれば、DBA 権限を持つネットワーク上の PC ユーザーは、このデータベースに INTERNAL としてアクセスできます。権限を持つユーザーがデータベースに対して DBA 機能を実行する場合は、SQL\*Plus コマンドに dba ユーザーのパスワードを追加して入力します。次に例を示します。

```
SQL> connect internal/dba_password@alias as {sysdba|sysoper}
```

## リモート認証

表 1-9 に、安全性の低いプロトコルを使用したリモート接続の動作を制御する `initnsid.ora` ファイルのパラメータを示します。

表 1-9 リモート接続を制御するパラメータ

REMOTE_OS_AUTHENT	ops\$ 接続を使用可能または使用禁止にします。
OS_AUTHENT_PREFIX	ops\$ アカウントによって使用されます。
REMOTE_OS_ROLES	リモート接続でロールを使用可能または使用禁止にします。

**参照：** リソース制限の詳細は、オペレーティング・システムのドキュメントの「`getrlimit(2)`」を参照してください。

## orapwd の実行

`orapwd` ユーティリティは `$ORACLE_HOME/bin` にあり、Oracle ソフトウェア所有者が実行します。`orapwd` のコマンド構文は、次のとおりです。

```
$ orapwd file=filename password=password entries=max_users
```

表 1-10 に、`orapwd` の構文を示します。

表 1-10 orapwd の実行構文

<i>filename</i>	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 <code>orapwsid</code> で、フルパス名を指定します。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。
<i>password</i>	INTERNAL および SYS 用に指定する初期パスワードです。このパスワードは、データベースの作成後に <code>ALTER USER</code> 文を使用して変更できます。このパラメータの入力は必須です。
<i>max_users</i>	データベースに SYSDBA または SYSOPER として接続できるユーザーの最大数です。このパスワード・ファイルを EXCLUSIVE にする必要がある場合のみ、このパラメータの入力は必須です。 <i>max_users</i> には、必要であると予想される数より大きい値を設定してください。ユーザー数がこの値より多くなる場合、新しいパスワード・ファイルを作成する必要があります。

## orapwd の例

```
$ orapwd file=/u01/app/oracle/product/8.1.7/dbs/orapwV817  
password=V817pw entries=30
```

**参照：** セキュリティおよびパスワードの詳細は、『Oracle8i 管理者ガイド』を参照してください。

initSID.ora ファイルのカスタマイズ

デフォルトの initSID.ora ファイルは、Oracle8i ソフトウェアとともに提供されます。Oracle Universal Installer (OUI) は、\$ORACLE\_BASE/admin/SID/pfile ディレクトリ中にファイルを作成します。このファイルを変更することによって、Oracle8i のインストールをカスタマイズできます。initSID.ora のサンプル・ファイルは、\$ORACLE\_HOME/dbs ディレクトリにあります。

表 1-11 に、AIX での初期化パラメータのデフォルト値を示します。initSID.ora ファイルのこれらのパラメータに別の値を設定しなければ、すべての Oracle8i インスタンスでこれらの値が使用されます。initSID.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設定してください。

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

表 1-11 デフォルト初期化パラメータ

パラメータ	デフォルト値	範囲
BACKGROUND_DUMP_DEST	\$ORACLE_HOME/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	\$ORACLE_HOME/dbs/ctrlSID.dbf (SID は ORACLE_SID を表す)	有効なファイル名
CREATE_BITMAP_AREA_SIZE	8388608	65536 ～無制限
DB_BLOCK_BUFFERS	12288	50 ～無制限
DB_BLOCK_SIZE	4096	2K ～ 16K
DB_FILE_DIRECT_IO_COUNT	64	0 - 1048576
DB_FILE_MULTIBLOCK_READ_COUNT	8	1 ～ 128。ただし、値が次のいずれか小さい方の値を超えてはなりません。 <ul style="list-style-type: none"><li>DB_BLOCK_BUFFERS/4</li><li>1048576/DB_BLOCK_SIZE</li></ul>
DB_FILES	200	1 - 2000000
DISTRIBUTED_TRANSACTIONS	TRANSACTIONS/4	0 ～無制限
HASH_AREA_SIZE	SORT_AREA_SIZE 値の 2 倍	0 ～無制限



表 1-11 デフォルト初期化パラメータ（続き）

パラメータ	デフォルト値	範囲
HASH_MULTIBLOCK_IO_COUNT	0（セルフ・チューニング）	0 ～ 127。ただし、値が次のいずれか小さい方の値を超えてはなりません。 <ul style="list-style-type: none"> <li>■ DB_BLOCK_BUFFERS/4</li> <li>■ 1048576/DB_BLOCK_SIZE</li> </ul>
JAVA_POOL_SIZE	20000000	
LOCK_SGA	FALSE	TRUE または FALSE
LOG_ARCHIVE_BUFFER_SIZE	64	
LOG_ARCHIVE_BUFFERS	4	1 - 128
LOG_ARCHIVE_DEST	デフォルト値なし	有効な任意のデフォルト名
LOG_ARCHIVE_FORMAT	%t_%s.dbf	有効な任意のファイル名
LOG_BUFFER	524288	66560 ～ 無制限
LOG_CHECKPOINT_INTERVAL	0	0 ～ 無制限
LOG_SMALL_ENTRY_MAX_SIZE	80	
MTS_LISTENER_ADDRESS	有効なデフォルト値なし	有効な任意のディレクトリ名
MTS_MAX_DISPATCHERS	5	MTS_DISPATCHERS 値と PROCESSES 値の間の任意の数値
MTS_MAX_SERVERS	デフォルトは、MTS_SERVERS の 2 倍（MTS_SERVERS が 20 未満の場合）、または 20（それ以外の場合）	MTS_SERVERS 値と PROCESSES 値の間の任意の数値
MTS_SERVERS	デフォルトは、1（MTS_DISPATCHERS が指定されている場合）または 0（それ以外の場合）	1 ～ PROCESSES 値の間の任意の数値
NLS_LANGUAGE	AMERICAN	有効な任意の言語名
NLS_TERRITORY	AMERICA	有効な任意の地域名
OBJECT_CACHE_MAX_SIZE_PERCENT	10	0 ～ 無制限
OBJECT_CACHE_OPTIMAL_SIZE	102400	102400 ～ 無制限
OPEN_CURSORS	50	1 ～ 無制限
OS_AUTH_PREFIX	ops\$	任意の文字列
PROCESSES	30 (PARALLEL_AUTOMATIC_TUNING が設定されていない場合)	6 ～ 無制限

表 1-11 デフォルト初期化パラメータ（続き）

パラメータ	デフォルト値	範囲
SHARED_POOL_SIZE	8000K	300000 ～無制限
SORT_AREA_SIZE	65536	0 ～無制限
USER_DUMP_DEST	\$ORACLE_HOME/rdbms/log	有効なディレクトリ名

**参照：** 初期化パラメータの詳細は、『Oracle8i リファレンス・マニュアル』、『Oracle8i 管理者ガイド』および『Oracle8i パフォーマンスのための設計およびチューニング』を参照してください。

## 埋込み PL/SQL ゲートウェイ

埋込み PL/SQL ゲートウェイは、PL/SQL ベースのデータベース・アプリケーションを Web 上で展開するための固有のサポートを提供するために Oracle8i サーバーに埋め込まれたゲートウェイです。埋込み PL/SQL ゲートウェイは、Oracle Servlet Engine (OSE) サーブレットとして実装されており、OSE と mod\_ose (OSE をサポートする Apache モジュール) の両方の存在および構成に依存します。次に、ゲートウェイのインストールおよび構成方法について説明します。

## 概要

Apache モジュール mod\_ose および mod\_plsql は、PL/SQL ベースの Web アプリケーションをサポートしています。

mod\_ose モジュールは、Oracle8i インスタンスで実行する OSE の要求ルーターとして機能します。mod\_ose のルーティング機能によって、ステートフル OSE アプリケーションはステートフルな要求をルーティングし、中間層を介して指定された OSE/Oracle8i インスタンスに戻すことができます。埋込み PL/SQL ゲートウェイは、Oracle8i サーバー上で実行する OSE サーブレットとして実装されているため、ステートフルおよびステートレスな PL/SQL Web アプリケーションに対応できます。ステートフル PL/SQL Web アプリケーションでは、そのデータベース・セッション（たとえば、パッケージおよびトランザクション）の状態がすべて、要求間で保存されます。

mod\_plsql モジュールは、PL/SQL ゲートウェイに与えられた名前です。PL/SQL ゲートウェイは、中間層サーバーにある Apache モジュール内で実行し、バックエンド Oracle Server の PL/SQL プロシージャを、OCI を使用して実行します。現在、mod\_plsql モジュールでは、ステートレス PL/SQL Web アプリケーションのみをサポートしています。

**参照：** PL/SQL ベースの Web アプリケーションの開発については、PL/SQL ゲートウェイの共通ドキュメントである、『Oracle Internet Application Server 8i mod\_PL/SQL の使用』を参照してください。

## 埋込み PL/SQL ゲートウェイのインストール

すべての OSE サブレットと同様、埋込み PL/SQL ゲートウェイはロードおよび公開される必要があります。次に、埋込み PL/SQL ゲートウェイ・サブレットをロードおよび公開する方法について示します。

1. サブレットをロードするには、sys として SQL\*Plus に接続し、SQL スクリプト `initplgs` を実行します。

```
$ SQL>@rdbms/admin/initplgs.sql
```

2. 埋込み PL/SQL ゲートウェイ・サブレットの名前は、`oracle.plsql.web.PLSQLGatewayServlet` です。サブレットを公開するには、次のコマンドを実行します。

```
% $ORACLE_HOME/bin/sess_sh -s http://<OSE machine name>:<OSE port \
number> -u sys/change_on_install -c "publishservlet -virtualpath \
pls/* /webdomains/contexts/default plsGatewayway \
SYS:oracle.plsql.web.PLSQLGatewayServlet"
```

これによって、ゲートウェイ・サブレットがデフォルト・コンテキストとともに、`plsGateway` として公開されます。サブレットは、仮想パス `/pls` を使用して接続できません。ゲートウェイ・サブレットにアクセスする URL の例は、次のとおりです。

```
http://dlsun240/pls/dadname/hello_world
```

**参照：** サブレットの使用および公開については、『Oracle8i Oracle Servlet Engine ユーザーズ・ガイド』を参照してください。

### Oracle PL/SQL 埋込みゲートウェイの構成

Apache Server または `mod_ose` の構成手順および埋込み PL/SQL ゲートウェイについては、このマニュアルでは説明していません。

**参照：** Apache Server または `mod_ose` の構成については、『Oracle8i Oracle Servlet Engine ユーザーズ・ガイド』を参照してください。

## Oracle HTTP Server

Oracle HTTP Server は、Apache ベースです。サーバーの管理タスクは、サーバーが実行されているローカル・マシンに接続している必要があります。場合によっては、`root` での接続が必要です。

Oracle HTTP Server は、ポート 7777 で自動的に起動します。バイナリのサーバー名は、`httpd` です。次のコマンドを使用してサーバーが実行中であることを確認してください。

```
ps -edaf | grep httpd
```

## Oracle HTTP Server の開始および停止

構成に変更が加えられた場合、サーバーを再起動する必要があります。使用可能な SSL を使用してサーバーを再起動するには、root ユーザーである必要があります。次のようにサーバーを再起動します。

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

**注意：** 使用可能な SSL を使用してサーバーを再起動すると、デフォルト・ポートは 80 および 443 になります。

## 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、Apache のオンライン・ドキュメント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、ご使用のインターネット・ブラウザで、次のいずれかの URL をロードします。

- サーバーに使用可能な SSL がない場合は、次の URL をロードします。

`http://<ServerName>:7777/`

- サーバーに使用可能な SSL がある場合は、次の URL をロードします。

`http://<ServerName>/`

`ServerName` が Apache Server で構成されている場合、構成ファイルは、`httpd.conf` です。次のコマンドを使用して、構成ファイルの位置を確認します。

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

## Oracle HTTP Server のステータス

複数のステータス・ページが使用可能です。デフォルトのサーバー構成ファイルでは、セキュリティのため、サーバー・ステータスが使用禁止になっています。これらを使用可能にするには、適切な構成ファイルを編集し、サーバーを再起動します。

構成ファイル `$ORACLE_HOME/Apache/Apache/conf/httpd.conf` の場合は、次の URL を使用します。

```
http://<ServerName>/server-status
http://<ServerName>/server-info
http://<ServerName>/perl-status
```

構成ファイル `$ORACLE_HOME/Apache/Jserv/etc/conf/jserv.conf` の場合は、次の URL を使用します。

```
http://<ServerName>/jserv
```

## 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。サーバーが正常に動作していることを確認するために、定期的にログ・ファイルをチェックすることは重要です。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・レベルは、適切な構成ファイルを編集し、サーバーを再起動することによって変更できます。

```
$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
```

## デモンストレーション・ファイル

### SQL\*Loader のデモンストレーション

デモンストレーション・ファイルは、Oracle8i に付属しています。SQL\*Loader のデモンストレーションは、次の順番で実行してください。

**表 1-12 SQL\*Loader のデモンストレーション・ファイル**

ulcase1	ulcase3	ulcase5	ulcase7
ulcase2	ulcase4	ulcase6	

### デモンストレーションの作成と実行

デモンストレーションは、ユーザー scott/tiger として実行する必要があります。次の要件を確認してください。

- ユーザー scott/tiger に CONNECT 権限と RESOURCE 権限がある
  - 空の EMP 表および DEPT 表がある
1. 実行するデモンストレーションの ulcasen.sql スクリプトを実行します。
 

```
$ sqlplus scott/tiger @ulcasen.sql
```
  2. デモンストレーション・データをオブジェクトにロードします。
 

```
$ sqlldr scott/tiger ulcasen.ctl
```
- ulcase2 デモンストレーションに対して、ulcase2.sql スクリプトを実行する必要はありません。
  - ulcase6 デモンストレーションに対して、ulcase6.sql スクリプトを実行し、コマンドラインに次のように入力します。

```
$ sqlldr scott/tiger ulcase6 DIRECT=true
```

- ulcase7 デモンストレーションに対して、ulcase7s.sql スクリプトを実行し、コマンドラインに次のように入力します。

```
$ sqlldr scott/tiger ulcase7
```

デモンストレーションを実行した後、ulcase7e.sql を実行して、挿入トリガーおよびグローバル変数パッケージを削除します。

## SQL\*Loader の管理

Oracle8i には、SQL\*Loader 機能が組み込まれています。デモンストレーションおよびメッセージ・ファイルは、rdbms ディレクトリにあります。

SQL\*Loader の制御ファイルには、次のファイル処理オプションの文字列が追加されています。デフォルトは str で、引数は指定されていません。

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

表 1-13 ファイル処理オプション

str	改行文字で終了するレコードを、一度に 1 レコードずつ読み込むストリームを指定します。(デフォルト)
fix	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	ファイルが n バイトの可変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。レコードの最初の 5 文字を指定します。

ファイル処理オプションを選択しないと、情報はデフォルトでレコード・ストリーム (str) として処理されます。fix モードではレコード終了記号をスキャンする必要がないため、デフォルトの str モードより高いパフォーマンスが得られます。

### 固定長レコードでの改行

各レコードが改行で終了する固定長レコードを含むファイルを読むために fix オプションを使用する場合、レコード長に改行文字の長さ (1 文字) を含めて、SQL\*Loader に指定してください。

次に例を示します。

```
AAA [ 改行 ]  
BBB [ 改行 ]  
CCC [ 改行 ]
```

改行文字も含めるため、`fix 3`ではなく`fix 4`を指定します。

固定長レコードのファイルの最後のレコードが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。

---

---

## 改行の削除

改行をロードせずに固定長レコードから改行を削除するには、制御ファイルで `position (x:y)` 関数を使用します。たとえば、制御ファイルに次のように入力します。

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

これにより、各固定長レコードの 4 桁目にある改行は削除されます。

## PL/SQL デモンストレーション

PL/SQL には、ロードできるサンプル・プログラムが多数あります。デモンストレーション・ファイルおよびメッセージ・ファイルは、`rdbms` ディレクトリにあります。サンプル・プログラムが機能するには、Oracle8i 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。

1. SQL\*Plus を起動し、ユーザー / パスワードを `scott/tiger` で接続します。

```
$ cd $ORACLE_HOME/plsql/demo
$ sqlplus scott/tiger
```

2. デモンストレーションをロードするために、SQL\*Plus から `exampbld.sql` を起動します。

```
SQL> @exampbld
```

---

---

**注意：** デモンストレーションの作成は、必要な権限を持つ `oracle` アカウントで行ってください。また、デモンストレーションの実行も、同じアカウントで行ってください。

---

---

表 1-14 に、カーネルのデモンストレーションを示します。

表 1-14 カーネルのデモンストレーション

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 デモンストレーションを実行するには、そのデモンストレーションを作成するとき  
に使用したユーザー / パスワードで SQL\*Plus を起動し、データベースに接続します。デモ  
ンストレーション名の前にアットマーク「@」または start を入力し、デモンストレーショ  
ンを起動します。たとえば、examp1 のデモンストレーションを起動するには、次のように  
入力します。

```
$ sqlplus scott/tiger
SQL> @examp1
```

プリコンパイラ PL/SQL デモンストレーションを作成するには、次のように入力します。

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

表 1-15 に、プリコンパイラのデモンストレーションを示します。

表 1-15 プリコンパイラのデモンストレーション

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

単一のデモンストレーションを作成する場合は、make コマンドにそのデモンストレーショ  
ン名を引数として指定します。たとえば、examp9.pc 実行ファイルを作成する場合、次の  
ように入力します。

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

現在のシェルから examp9 デモンストレーションを起動するには、次のように入力します。

```
$ ./examp9
```

extproc デモンストレーションを実行するには、まず、ファイル tnsnames.ora に次の行  
を追加します。

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=plsf)) (CONNECT_DATA=(SID=extproc)))
```



次に、listener.ora ファイルに次の行を追加します。

```
SC=( (SID_NAME=extproc) (ORACLE_HOME=/u01/app/oracle/product/8.1.7) (PROGRAM=extproc) )
```

SQL\*Plus セッションで次のように入力します。

```
SQL> connect system/manager
Connected.
SQL> grant create library to scott;
Grant succeeded.
SQL> connect scott/tiger
Connected.
SQL> create library demolib as
'$ORACLE_HOME/plsql/demo/extproc.so';
Library created.
```

最後に、テストを実行します。

```
SQL> connect scott/tiger
Connected.
SQL> @extproc
```

## データベースの例

次の例では、ローカル bin ディレクトリが /usr/sbin で、本番データベースが PROD であることを想定しています。起動時に ORACLE\_SID を入力するプロンプトが表示されないようにするには、環境変数 ORAENV\_ASK を NO に設定します。

---

**注意：** ORACLE\_SID を入力するプロンプトが起動時に表示されないようにするには、環境変数 ORAENV\_ASK を NO に設定します。

---

Oracle Database Configuration Assistant を使用するかわりに、データベースを手動で作成した場合、システム構成が /var/opt/oracle/oratab ファイルに反映されていることを確認してください。

各サーバーのインスタンスに、エントリを次の形式で追加します。

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

Y または N は、dbstart および dbshut スクリプトをアクティブにするかどうかを示します。Oracle Database Configuration Assistant は、作成する各データベースにエントリを自動的に追加します。

## シングル・インスタンスの例

Bourne シェルまたは Korn シェルの場合、`.profile` ファイルに次の行があります。

```
. local_bin_directory/oraenv
```

この行を次の行に置き換えます。

```
PATH=${PATH}:/usr/lbin
ORACLE_SID=PROD
export PATH ORACLE_SID
ORAENV_ASK=NO
. oraenv
ORAENV_ASK=
```

C シェルの場合、`.cshrc` ファイルに次の行があります。

```
source local_bin_directory/coraenv
with the lines:
setenv PATH ${PATH}:/usr/lbin
setenv ORACLE_SID PROD
set ORAENV_ASK = NO
source /usr/lbin/coraenv
unset ORAENV_ASK
```

## 複数インスタンスの例

複数インスタンスの場合は、起動時に `sid` を定義します。

Bourne シェルまたは Korn シェルの場合、次のように入力します。

```
#!/usr/bin/sh
echo "The SIDs on this machine are:"
cat /etc/oratab | awk -F: '{print $1}' | grep -v "#"
ORAENV_ASK="YES"
. /usr/lbin/oraenv
```

C シェルの場合、次のように入力します。

```
#!/usr/bin/csh
echo "The SIDs on this machine are:"
cat /etc/oratab | awk -F: '{print $1}' | grep -v "#"
set ORAENV_ASK="YES"
source /usr/lbin/coraenv
```

---

# Oracle8i for AIX-Based Systems の チューニング

チューニングを行うことで、Oracle8i for AIX-Based Systems システムのパフォーマンスおよびスケーラビリティを改善できます。ここで説明する情報とともに、『Oracle8i パフォーマンスのための設計およびチューニング』（一般的なチューニング方法を説明）、および IBM 関連のドキュメントを参照してください。

- [AIX-Based Systems のパフォーマンス・ツール](#)
- [Oracle のパフォーマンス・ツール](#)
- [Oracle8i のチューニング](#)
- [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](#)
- [その他のチューニングのヒント](#)

## AIX-Based Systems のパフォーマンス・ツール

### AIX-Based Systems の Base Operation System ツール

AIX-Based Systems Base Operation System (BOS) に含まれるパフォーマンス・ツールには、歴史的に UNIX システムの一部となったものと、AIX-Based Systems の実装固有の機能を管理するうえで必要なものがあります。最も重要な BOS 機能を、次に示します。

<code>iostat</code>	CPU および I/O 統計をレポートします。
<code>lsattr</code>	デバイスの属性を表示します。
<code>lslv</code>	論理ボリューム、または物理ボリュームの論理ボリューム割当てに関する情報を表示します。
<code>netstat</code>	ネットワーク関連のデータ構造の内容を表示します。
<code>nfsstat</code>	Network File System (NFS) および Remote Procedure Call (RPC) のアクティビティに関する統計を表示します。
<code>nice</code>	プロセスの最初の優先順位を変更します。
<code>no</code>	ネットワーク・オプションを表示または設定します。
<code>ps</code>	1 つ以上のプロセスの状態を表示します。
<code>reorgvg</code>	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。
<code>sar</code>	システム・アクティビティ情報を集めて、レポートおよび記録します。
<code>time</code>	実行経過時間、ユーザーおよびシステムの CPU 処理時間を出力します。
<code>trace</code>	選択されたシステム・イベントを記録およびレポートします。
<code>vmstat</code>	仮想記憶アクティビティおよび他のシステム統計をレポートします。

### AIX-Based Systems の Performance Toolbox

AIX-Based Systems の 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し、チューニングするためのツールが含まれます。PTX は、PTX Manager および PTX Agent という 2 つの主なコンポーネントで構成されています。PTX Manager は、`xmperf` というプログラムを使用して、構成内の様々なシステムのデータを収集および表示します。PTX Agent は、`xmserd` というプログラムを使用して、データを収集し、PTX Manager に送信します。PTX Agent は、Performance Aide for AIX-Based Systems という名前の独立した製品としても入手可能です。

PTX と Performance Aide のどちらにも、次に示す個別の監視およびチューニング・ツールが含まれます。

fdpr	特定のワークロードにあわせて、実行プログラムを最適化します。
filemon	トレース機能を使用して、AIX-Based Systems ファイル・システムのアクティビティを監視およびレポートします。
fileplace	論理または物理ボリューム内のファイル・ブロックの配置状況を表示します。
lockstat	カーネル・ロックの競合に関する統計を表示します。
lvedit	対話形式で、論理ボリュームをボリューム・グループ内に容易に配置できるようにします。
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 使用状況をレポートします。
rmss	パフォーマンスをテストするために、様々なサイズのメモリーでシステムをシミュレートします。
svmon	仮想メモリーの使用状況を取得および分析します。
syscalls	システム・コールを記録およびカウントします。
tprof	トレース機能を使用して、CPU の使用状況をモジュールおよびソース・コード文レベルでレポートします。
BigFoot	プロセスのメモリー・アクセス・パターンをレポートします。
stem	既存の実行ファイルのサブルーチン・レベルでの入口または出口として使用できます。

**参照：** PTX 情報の詳細は、『AIX Performance Toolbox/6000 User's Guide 1.2 and 2.1』を参照してください。

『AIX Version 4.x Commands Reference』にも、ツール・コマンドの構文の説明があります。

## AIX-Based Systems System Management Interface Tool

AIX-Based Systems 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、メニュー操作可能なインタフェースを提供します。smit を使用すると、大量のツールの中から目的のツールを容易に見つけることができるため、実行する作業のみに集中できます。

## Oracle のパフォーマンス・ツール

ここでは、オラクル社が提供するパフォーマンス監視ツールについて説明します。

### Utlbstat/Utlestat による統計収集

utlbstat.sql および utlestat.sql スクリプトは、`$ORACLE_HOME/rdbms/admin` にあります。通常のワークロード期間の開始時に、SQL\*Plus から utlbstat を実行し、終了時に utlestat を実行します。これにより、utlestat スクリプトが実行されたディレクトリに report.txt ファイルが作成されます。このレポートには、測定期間内のパフォーマンス統計が含まれます。この統計は、データベースの競合検出およびチューニングに役立ちます。初期化パラメータ `timed_statistics` を `true` に設定しない限り、タイミング統計はすべて 0 として表示されます。

**参照：** 統計の詳細は、『Oracle8i パフォーマンスのための設計およびチューニング』を参照してください。

### 動的パフォーマンス表

Oracle8i は、リソースの競合、ボトルネックおよびパフォーマンス上の問題の検出に役立つ、多数の動的サーバー・パフォーマンス表 / ビューを提供しています。頻繁に使用されるものを次に示します。

- V\$system\_event
- V\$resource\_limit
- V\$waitstat
- V\$session\_wait
- V\$rowcache
- V\$ping
- V\$file\_ping
- V\$lock\_activity

### SQL トレース

初期化パラメータ `SQL_TRACE=TRUE` を設定することにより、インスタンスのすべてのセッションで SQL トレースをオンにできます。コマンド `ALTER SESSION SET SQL_TRACE=TRUE` を発行することにより、セッション・レベルで SQL トレースをオンにすることもできます。これにより、影響を受けたセッションごとに、初期化パラメータ `USER_DUMP_DEST` に指定されたディレクトリにトレース・ファイルが作成されます。このパラメータのデフォルト値は、`$ORACLE_HOME/rdbms/log` です。

Oracle の提供する TKPROF コマンドを実行することで、生成されたトレース・ファイルを分析できます。

**参照：** SQL トレース機能および TKPROF の詳細は、『Oracle8i パフォーマンスのための設計およびチューニング』を参照してください。

## Oracle Enterprise Manager Performance Tuning Pack

Oracle Enterprise Manager リリース 2.2 では、システムのパフォーマンスを監視およびチューニングするための次のパッケージを提供します。

### Oracle Diagnostics Pack

Oracle Diagnostics Pack は、次のパッケージで構成されます。

- Oracle Performance Manager
- Oracle Capacity Planner
- Oracle Trace Manager and Viewer
- Oracle TopSessions
- Oracle Lock Manager

#### Oracle Performance Manager

パフォーマンスをリアルタイムに監視し、Oracle データベース・サーバーおよびオペレーティング・システムのパフォーマンス・データをグラフィカルで、かつ動的なビューで提供します。Oracle データベース・カートリッジおよびオペレーティング・システム・カートリッジが、ホストにインストールされている必要があります。

#### Oracle Capacity Planner

システムおよびデータベース管理者が、データベースおよびオペレーティング・システムのパフォーマンス統計を収集し、そのデータを履歴データベースに記録できるようにします。その後、Oracle Capacity Planner を使用して履歴データを分析し、将来の容量計画を立てることができます。

Oracle データベース・カートリッジおよびオペレーティング・システム・カートリッジに加え、エージェント・データ収集サービスも、オペレーティング・システム・データの収集対象となるすべてのホストにインストールする必要があります。

#### Oracle Trace Manager and Viewer

グラフィカルな Oracle Trace Manager アプリケーションを使用して、Oracle Trace API を備えた製品に対して、Oracle Trace コレクションの作成、スケジュールおよび管理を実行できます。ホストに Oracle Intelligent Agent がインストール済であること、および Oracle Trace Collection Services が使用可能な状態にしておくことが必要です。

#### Oracle TopSessions

Oracle セッションの上位 n 個のパフォーマンス・アクティビティに注目したビューを提供します。

#### Oracle Lock Manager

ロックによりブロックされたセッションの表示および解決に使用します。

## Oracle Tuning Pack

Oracle Tuning Pack は、次のパッケージで構成されます。

- Oracle Expert
- Oracle Tablespace Manager
- Oracle SQL Analyze

### Oracle Expert

データベースの初期構成および既存のデータベースのパフォーマンス特性を収集および評価することにより、データベース環境のパフォーマンスの最適化を支援します。

### Oracle Tablespace Manager

表領域の詳細情報を提供し、表の最適化および未使用領域の再割当てを実行します。

### Oracle SQL Analyze

データベース環境およびスキーマ・オブジェクト情報の収集、SQL パフォーマンスの分析、異なるオプティマイザ手法の識別と比較、および最適なパフォーマンスを得るための文を編集する（場合によっては自動）ためのツールを提供します。

## Oracle8i のチューニング

Oracle8i は、高度に最適化できるソフトウェア製品です。Oracle for AIX-Based Systems アプリケーションの適用範囲が広がり、複雑化するなかでチューニングを頻繁に実行することにより、システムのパフォーマンスを最適化し、データのボトルネックを防ぎます。

## Oracle8i のチューニング手順

Oracle8i のパフォーマンスを最大限に引き出すために実行可能なタスクは多数あります。あるチューニング操作での決定が、後続のチューニング手順に影響を与える可能性があるため、タスクを次の手順で実行してください。

1. アプリケーション設計のチューニング
2. データ・アクセスのチューニング
3. メモリー管理のチューニング
4. ディスク I/O のチューニング
5. CPU の使用状況のチューニング
6. Oracle リソースの競合のチューニング
7. Oracle Parallel Server での Oracle リソースの競合のチューニング

ここでは、メモリー、ディスク I/O および CPU の使用状況を扱います。



**参照：** 他のタスクに関する詳細は、『Oracle8i パフォーマンスのための設計およびチューニング』、『AIX Versions 4.x Performance Tuning Guide』および『AIX Version 4.x System Management Guide』の「Operating Systems and Devices」を参照してください。

## メモリーおよびページング

メモリーの競合は、プロセスが、使用可能な範囲以上のメモリーを要求する場合に発生します。容量の不足に対処するために、システムは、メモリーとディスクとの間でプログラムおよびデータのページングを行います。

### 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

過度のページング・アクティビティは、パフォーマンスを大幅に低下させます。このため、JFS ファイル上に作成されたデータベースでは、その影響は重大なものになる可能性があります。このような状況では、大量の SGA データ・バッファも、最も頻繁に参照されるデータを含む JFS バッファと同様の影響を受ける可能性があります。ファイル・システムのバッファ・キャッシュ・マネージャの動作が、パフォーマンスに著しい影響を与える場合があります。これが I/O ボトルネックを引き起こすと、システム全体のスループットが低下します。

AIX-Based Systems では、バッファ・キャッシュ・ページング・アクティビティのチューニングは可能ですが、頻繁にならないように、慎重に行う必要があります。

AIX-Based Systems では、チューニング可能な 4 つのシステム・パラメータが存在します。

minfree	空きリスト・サイズの最小値。このサイズ以下になると、ページ・スティーリングにより空きリストの補充が行われます。
maxfree	空きリスト・サイズの最大値。このサイズを超えると、ページ・スティーリングが終了します。
minperm	ファイル I/O の最小バッファ・ページ数。
maxperm	ファイル I/O の最大バッファ・ページ数。

vmtune コマンドで前述の値を変更するには、root 権限が必要です。AIX-Based Systems の vmtune ユーティリティは、オペレーティング・システムのバージョンに固有なものです。異なるリリースで実行可能な vmtune コマンドを実行すると、オペレーティング・システムに障害が発生する可能性があります。

## minfree および maxfree のチューニング

迅速な応答を必要とするプログラムの作業セット・サイズを基にして、minfree を設定できます。基本的に、ロードされたプログラムが空きリストを補充せずに済むように、空きリスト内に十分なページを保持する必要があります。プログラム用の作業セット・サイズは、vmstat コマンドを使用して設定できます。maxfree は、8 または maxpgahead の値だけ minfree よりも大きい必要があります。たとえば、maxpgahead が 16 で、minfree を 128 に設定する必要がある場合、次のコマンドを使用して minfree を 128 に、maxfree を 144 に設定できます。

```
/usr/lpp/bos/samples/vmtune -f 128 -F 144
```

## AIX-Based Systems でのファイル・バッファ・キャッシュのチューニング

AIX-Based Systems のバッファ・キャッシュの目的は、JFS ファイル使用時の、ディスクへのアクセスの頻度を減らすことです。このキャッシュ・サイズが小さすぎると、ディスクの使用量が増えて、1 つまたは複数のディスクがいっぱいになる可能性があります。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：** このパラメータ値を増やした場合の影響については、「[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-Based Systems のバッファ・キャッシュは、minperm および maxperm パラメータを調整することで構成できます。一般に、バッファ・ヒット率が低い (90% 未満、sar -b を参照) 場合は、minperm の値を増やします。バッファ・ヒット率の高さが重要ではない場合、minperm の値を減らすことで、利用可能な物理メモリーを増やせます。この項目の詳細は、AIX-Based Systems のドキュメントを参照してください。

パフォーマンスの向上は、マルチプログラミングの度合いおよびワークロードの I/O 特性に依存するため、簡単に定量化できるものではありません。

## minperm および maxperm のチューニング

AIX-Based Systems では、ファイルで使用するページ・フレームの割合を管理者がゆるやかに制御できますが、一方で、次のガイドラインに沿って minperm 値および maxperm 値を調整することにより、計算可能 (作業用またはプログラム・テキスト) セグメントで使用するページ・フレームを制御することもできます。

- 実メモリーを占めるファイル・ページの割合 (パーセンテージ) が minperm 値を下回ると、ページ置換アルゴリズムにより、再ページング率に関係なく、ファイルと計算可能ページの両方が奪います。
- 実メモリーを占めるファイル・ページの割合 (パーセンテージ) が maxperm 値を超えると、ページ置換アルゴリズムにより、ファイル・ページのみが奪います。
- 実メモリーを占めるファイル・ページの割合 (パーセンテージ) が minperm 値と maxperm 値の間にある場合、VMM は通常ファイル・ページのみを奪います。ただし、ファイル・ページの再ページング率が計算用ページの再ページング率より高い場合、計算可能ページも奪います。

次のアルゴリズムにより、デフォルト値が算出されます。

- $\text{minperm (ページ)} = ((\text{ページ} \cdot \text{フレームの数}) - 1024) \times 0.2$
- $\text{maxperm (ページ)} = ((\text{ページ} \cdot \text{フレームの数}) - 1024) \times 0.8$

次のコマンドを実行すると、minperm が総ページ・フレーム数の 5% に、maxperm が総ページ・フレーム数の 20% に変更されます（デフォルト値は、minperm が 20%、maxperm が 80%）。

```
vm tune -p 5 -P 20
```

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 を低い値（たとえば、それぞれを 5% と 20% に）に設定できます。これは、AIX-Based Systems のファイル・バッファ・キャッシュが RAW デバイスでは使用されないためです。このため、Oracle システム・グローバル領域などの他の目的にメモリーを使用する方が賢明です。

## 十分なページング領域の割当て

通常、ページング領域が不十分な場合、システムがハングしたり、応答時間が非常に長くなったりします。AIX-Based Systems 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。構成するページング領域の量は、存在する物理メモリーの量およびアプリケーションが必要とするページング領域によって異なります。ページング領域の使用状況の監視には、AIX-Based Systems の `lsps` コマンドを使用し、システムのページング・アクティビティの監視には `vmstat` を使用します。

一般に、ページング領域は、大半のワークロードで必要とされる物理メモリー量の 2～3 倍に設定するのが最適です。CASE、Oracle Application または Oracle Office を使用する場合は、もう少し高い値に設定します。ページング領域の使用状況を見ながら、必要に応じて値を高くしてください。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。大量の物理メモリー（1GB 以上）を保持するシステムの場合、ページング領域は物理メモリーの 1.5 倍で十分です。

## ページングの制御

ページングが定期的および頻繁に発生する場合は、実メモリーの負担が大きすぎることを示します。一般に、次の作業を行う必要があります。

- システムが非常に高速な拡張記憶領域を備えているために、メモリーと拡張記憶領域間のページングが、Oracle による SGA とディスク間のデータの読取り / 書込みよりも高速である場合を除き、継続的なページングを避けます。
- 限られたメモリー・リソースを、システム・パフォーマンス向上に最も役立つところに割り当てます。メモリー・リソース要件とその結果とのバランスをとるには、再帰的な処理が必要な場合があります。
- メモリーが不足している場合、システムのプロセス / 要素を必要とするメモリーの優先順位リストを作成します。最も高いパフォーマンスが得られるところにメモリーを割り当てます。優先順位リストの例を、次に示します。
  1. OS および RDBMS のカーネル
  2. ユーザー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. PCM ロックおよびDLM
6. DB 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 / ビューへの問合せを実行して、共有プールとデータベース・バッファ・キャッシュの両方がより多くのメモリーを必要としていることがわかった場合、限られた余分のメモリーを、データベースのブロック・バッファ・キャッシュよりも共有プールに割り当てる方が、パフォーマンスがより向上します。

次の AIX-Based Systems の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] および  
$ sar -r interval [repeats]
```

## データベース・ブロック・サイズの設定

Oracle データベース・ブロック・サイズを設定することで、I/O スループットを向上させることができます。AIX-Based Systems では、DB\_BLOCK\_SIZE の範囲は 2KB ～ 16KB で、デフォルトは 4KB です。Oracle データベースが JFS ベースの場合、ブロック・サイズを JFS のブロック・サイズ（AIX-Based Systems では 4 KB）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存在する場合、Oracle データベース・ブロック・サイズは、オペレーティング・システムの物理ブロック・サイズ（AIX-Based Systems では 512 バイト）の倍数になります。

Oracle データベース・ブロック・サイズは、OLTP または複合的作業の環境では小さめ（2KB または 4KB）に設定し、DSS タイプの作業負荷環境では大きめに（8KB または 16KB）設定することをお勧めします。

## アーカイバ・バッファのチューニング

より多くの専用バッファを割り当てることにより、データベースのアーカイブ速度を改善できます。ただし、バッファを多く設定し過ぎると、システム全体のパフォーマンスを低下させることがあります。init.ora のパラメータ LOG\_ARCHIVE\_BUFFER\_SIZE を最大値の 128 まで増やすことができます。LOG\_ARCHIVE\_BUFFERS のデフォルト値は 4 です。

通常のデータベース・アクティビティ全体のパフォーマンスが大幅に低下することがないよう、チューニングは注意深く行ってください。アーカイブ処理のパフォーマンスを 0 ～ 20% 改善することが可能です。より高いパフォーマンス向上についても報告されています。

## SGA サイズのチューニング

SGA で複数の共有メモリー・セグメントを使用する際、現行のハードウェア・アーキテクチャがサポート可能な物理メモリーの量が制約となります。メイン・メモリーが足りないために Oracle SGA がページング領域に移されることは、明らかに望ましいことではありません。多数のユーザーが同時にデータベースを利用する場合は、SGA サイズをより大きく取ることをお勧めします。

SGA サイズに最も大きな影響を及ぼす `init.ora` のパラメータは、`DB_BLOCK_BUFFERS` および `SHARED_POOL_SIZE` です。Oracle Parallel Server が実行中の場合、`init.ora` ファイルの `LM_LOCKS`、`LM_RESS` および `LM_PROCS` パラメータが、SGA のサイズに大きな影響を与えます。これらのパラメータ値をどの程度増やすかは、SGA に割り当てられる可能なメモリーの量によって異なります。`DB_BLOCK_BUFFERS` および `SHARED_POOL_SIZE` に大きな値を指定すると、パフォーマンスのオーバーヘッドが発生します。その場合は、これらの値を引き下げると、CPU の使用率が減少します。`initsid.ora` ファイル内の `DB_BLOCK_BUFFERS` パラメータを変更し、インスタンスを再起動することによって、Oracle8i バッファ・キャッシュを大きくまたは小さくすることができます。

## I/O バッファおよび SQL\*Loader

データをパラレルでロードする以外に、SQL\*Loader ダイレクト・パス・オプションを使用するなどしてデータを高速にロードする場合、CPU 使用時間の大半が I/O 操作の完了待機に費やされます。通常、バッファ数を増やすことにより、CPU 使用率を高め、結果として全体のスループットを向上させることができます。

選択するバッファ数 (SQL\*Loader の `BUFFERS` パラメータ) は、使用可能なメモリーの量、および CPU 使用率をどこまで高めるかによって変化します。`BUFFERS` パラメータを使用して、ファイル処理オプション文字列を調整する方法の詳細は、『Oracle8i ユーティリティ・ガイド』を参照してください。

パフォーマンス向上の程度は、CPU 使用率およびロード時の並列度によって異なります。

**参照：** SQL\*Loader ユーティリティに関する一般的な情報は、『Oracle8i ユーティリティ』を参照してください。

### imp の BUFFER パラメータ

高速ネットワークを使用している場合、そのパフォーマンスを活かすには、`imp` の `BUFFER` パラメータに大きな値を設定する必要があります。たとえば、SP スイッチを使用する場合には、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（によるページングおよびスワッピング）が貧弱な場合、または表領域やファイルの複数ディスク間での分散が不十分な場合に発生します。I/O の負荷は、すべてのディスクに均等に分散させる必要があります。

### 索引を表から分離する

索引および索引が参照する表が、同一のディスク・ドライブに存在する場合、索引検索に関連した I/O はすべて、同一ディスク上に集中します。ドライブの I/O 負荷が大きい場合には、索引と表を別々のドライブに格納して、I/O 負荷を分散させる必要があります。

### REDO ログを専用ディスク・デバイスに配置する

Oracle アプリケーションが負荷の大きい INSERT および UPDATE アクティビティに関係している場合、他のディスク・アクティビティをサポートしないディスク上に REDO ログを配置することにより、Oracle のパフォーマンスを最大限引き出すことができます。また、ARCHIVELOG オプションを使用可能に設定している場合、各 REDO ログを別々のディスクに配置することで、LGWR プロセス（現行の REDO ログへの書込み）と ARCH プロセス（クローズされた REDO ログからの読取り）間のディスク競合を最小限に抑えることができます。REDO ログを RAW デバイスに配置すると、パフォーマンスはさらに向上します。次のような理由で、REDO ログは、RAW デバイス上に配置すべき最も優先度の高いファイルといえます。

- REDO ファイルではデータが順次に読み書きされ、RAW デバイスのメリットを最大限に活用できます。
- REDO ファイルのサイズは固定であるため、RAW デバイスの管理コストを最小限に抑えることができます。

### ホット・ファイルを他のディスクに移動する

I/O バランスを維持するため、アクセス頻度の高いホット・ファイルをアクティブに使用されないディスク・デバイスに分散させます。ファイル全体をアクティブなディスクからアクティブに使用されないディスクへ移動させることも、ホット・ファイルをストライプ化して、各ディスクにファイルの一部を配置することもできます。

### ホット・ファイルへの I/O を削減する

ディスク・デバイスに存在するホット・ファイルが 1 つのみ存在し、そのファイルが大きな要求キューを対象としたものである場合、ホット・ファイルを別のディスクに移動しても効果は得られません。問題の Oracle ファイルまたは表領域に複数のセグメント（表や索引など）からのデータが含まれる場合には、アクセス頻度の高いセグメントを別の表領域やファイルに移動します。データベース・セグメント用の物理デバイスは、表領域レベルでのみ指定できます。関係するセグメントが 1 つのみの場合、表をストライプ化して、セグメント・データを単一の表領域内の複数のファイルに配置する方法を考慮してください。

## データベースの過度の断片化チェック

Oracle データ構造が断片化すると、CPU が複数の物理 I/O から単一の論理 I/O 要素を取得して結合する必要が生じます。この余分なオーバーヘッドのために、応答時間が長くなります。

### エクステンツの断片化

データベース・セグメントには、連続していない複数のディスク領域エクステンツを含めることができます。この場合、ディスクを順次に読み込めないため、または複数の I/O に分割されるため、I/O 時間が長くなります。I/O 分割は、データがディスク上の連続していないエクステンツにまたがっているために、単一の I/O 要求を複数の物理 I/O に分割する場合に発生します。

### 表領域の断片化

Oracle 表領域は、いくつかの個別ファイルで構成されます。表領域内の Oracle セグメント（表や索引など）は多数の個別エクステンツで構成されるため、表領域の断片化が発生します。表のストライプ化のように断片化が望ましい場合もありますが、通常、断片化は望ましくありません。データベース・セグメントが削除されるたびに、表領域の断片化が引き起こされます。表領域の断片化は、空き領域を有効に使用できません。

表領域の断片化により、Oracle の複数ブロック読取り機能が利用できなくなります。セグメント・エクステンツが連続した空きエクステンツより大きい場合、断片化した表領域ファイルによりデータベース領域も浪費されます。表領域の断片化は、次の SQL 文を使用して検出できます。

```
SELECT * FROM DBA_EXTENTS;
```

空き領域の問合せは、次の SQL 文を使用して実行できます。

```
SELECT * FROM DBA_FREE_SPACE;
```

空き領域の断片化には、2 つの種類があります。

- 泡立ち状：複数のアクティブなエクステンツにまたがるエクステンツが削除されると、小さな泡のような連続していない空き領域が形成されます。
- 蜂の巣状：隣接したエクステンツが削除される場合、常に空き領域が連続した小片に分割されます。

utlstat および utlestat の再帰的コール値が報告された場合、表領域が断片化していることを意味します（データ・ディクショナリ・キャッシュが適正にチューニングされていることが前提）。

## AIX 論理ボリューム・マネージャ

AIX-Based Systems の論理ボリューム・マネージャ（LVM）が提供する複数のディスク装置のストライプ化機能を利用すると、ディスクの競合を減らすことができます。ストライプ化の主な目的は、大規模な順次ファイルの読取りおよび書き込み時に、パフォーマンスを向上させることです。LVM のストライプ化機能を効果的に活用すると、I/O を複数ディスク間でより均等に分散できるため、全体のパフォーマンスが向上します。

### 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場合、次の内容を指定する必要があります。

ドライブ	2 台以上の物理ドライブ。パフォーマンスに重点が置かれる順次 I/O の実行時に、ドライブのアクティビティを最小限に抑える必要があります。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ことが必要な場合もあります。
ストライプ・ユニット・サイズ	2KB ～ 128KB の間で任意の 2 の累乗値を指定できますが、大半のワークロードで適切なストライプ・サイズは、32KB および 64KB です。Oracle データベース・ファイルの場合、ストライプ・サイズを、データベース・ブロック・サイズの倍数にする必要があります。
サイズ	論理ボリュームに割り当てる物理パーティションの数は、使用するディスク・ドライブの倍数にする必要があります。
属性	ミラー化はできません。つまり、copies=1 になります。

### ストライプ化論理ボリュームの推奨パラメータ

通常、最高の順次 I/O スループットを達成するストライプ化論理ボリューム・パラメータを次に示します。

ストライプ・ユニット・サイズ	64KB
minpgahead	2（ページ先読込みの最小値）
maxpgahead	ディスク・ドライブ数を 16 倍した値（最大先読みページ数）。その後、ページ先読みが、ストライプ・ユニット・サイズ（64KB）にディスク・ドライブ数を掛けた値に等しい単位で実行されます。その結果、先読みごとに、ディスク・ドライブにつき 1 つのストライプ・ユニットが読み込まれます。
maxfree	maxpgahead の変更を保存します。

（**注意**: minpgahead、maxpgahead および maxfree は、vmtune の調整可能なパラメータです。）

AIX-Based Systems のユーティリティ filemon、sar、iostat、またはホット・ディスクを識別可能な他のパフォーマンス・ツールを使用して、I/O アクティビティが複数のディスク装置に均等に分散されたことを確認してください。



## その他の考慮事項

LVM の効果的な使用によるパフォーマンス向上の程度は、使用する LVM およびワークロードの特性によって異なります。意思決定支援システム (DSS) のワークロードの場合、大幅な向上を見込めます。OLTP タイプまたは混在ワークロード環境でも、かなりのパフォーマンス向上を期待できます。

## 異なるデータベース・ファイルの同一ディスクへの配置の回避

同一の物理ディスクに属する複数のパーティションに、アクセスが同時に行われる可能性がある場合、これらのパーティションの使用は避けてください。たとえば、データ・ファイルとログ・ファイルへのアクセスは通常同時に行われるため、データ・ファイルとログ・ファイルは異なるディスク・セット上でストライプ化する必要があります。そうしないと、ディスク・ヘッドの過剰なシーク時間のために、パフォーマンスが低下します。ディスク上の各パーティションが相互に離れている場合、パフォーマンスはさらに低下します。

## ジャーナル・ファイル・システム (JFS) と RAW パーティション

このトピックは、時として大きな論争の種になることがあります。論争になりやすい理由として、次の点があげられます。

- 様々なファイル・システム実装と同様、ファイル・システムは絶えず改善されています。場合によっては、ファイル・システムの提供する I/O パフォーマンスの方が、RAW デバイスよりも優れていることもあります。
- 様々なディスクの性能を最大限引き出すために、ファイル・システム・レイヤーを実装する方法は、ベンダーによって異なります。このためユーザーの認識は、プラットフォームによって複雑に変化します。
- より強力な LVM インタフェースを導入することで、RAW パーティションをベースとする論理ディスクを構成および支援することは、かなり容易になります。

パフォーマンス向上の程度は、ワークロードの I/O 特性に大きく依存します。

JFS を使用する場合、RAW デバイスを使用する場合よりも、データベース・ファイルの管理およびメンテナンスが容易になります。ただし、現在のところ、JFS がサポートするのはバッファ読み取り / 書き込みのみであるため、I/O サブシステム (Oracle バッファ・キャッシュ、SGA 以外) との間でデータの転送が行われるたびに、新たな AIX-Based Systems ファイル・バッファが作成されます。これが、JFS ベースのデータ・ファイルを使用する場合のマイナス面であり、I/O サブシステムの使用負荷が高まるにつれ、このマイナス面は強調されていきます。

また、高い負荷のかかるシステムでは、AIX-Based Systems バッファの競合が主要なボトルネックになります。次のような副作用も含まれます。

- ページデーモンの活動が増大するため、ファイル・システム・バッファ・キャッシュのデータがディスクにフラッシュされます。
- AIX-Based Systems バッファ・キャッシュにより多くのメモリーが奪われます。

Oracle ファイルで RAW デバイス / ボリュームを使用するのは、Oracle データ・ファイルと同数以上の RAW ディスク・パーティションがサイトに存在する場合のみにしてください。RAW ディスク・パーティションがすでにフォーマットされている場合、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合わせることによって、無駄な領域をなくします。

RS/6000 プラットフォームでのオラクル社の経験では、ワークロードによって差はあるものの、RAW デバイスを使用するほうが、ファイル・システムを使用する場合よりも全体的に高いパフォーマンスとスケーラビリティを得ることができます。このため、負荷のかかるデータベース構成の場合、RAW デバイスを使用することをお勧めします。

**注意：** Oracle Parallel Server では、データベース・ファイルとして、JFS ではなく RAW デバイスを使用する必要があります。

### ファイル・システムから RAW デバイスへの移動

1. UNIX の dd コマンドを使用すると、すべてのデータを手動で再ロードする必要がないため、JFS から RAW デバイスへの移動を簡単に実行できます。dd コマンドは、次のように実行します。

```
dd if=/home/old_JFS_file of=/dev/new_raw_device seek=1 bs=4k
```

次の作業を確実に実行してください。

- RAW デバイスの最初の 4KB ブロックは、AIX-Based Systems の論理ボリューム・マネージャが使用するため、この領域をスキップします。

**注意：** RAW デバイスが HSD の場合、最初の 4KB ブロックをスキップしないでください。

- 領域が無駄にならないように、RAW デバイスのサイズを設定します（RAW デバイスは、JFS ファイルよりも数ブロック大きくする必要があります）。
  - データ・ファイルの名前を変更します。
  - RAW パーティションにアクセス権を設定します。
2. 順次読取りのパフォーマンスが、若干低下する場合があります。アプリケーションが頻繁にフル・テーブル・スキャンを実行していた場合は、表のキャッシュに AIX-Based Systems のファイル・バッファ・キャッシュが使用されなくなるため、フル・テーブル・スキャン時にサーバーのパフォーマンスが低下する場合があります。
  3. メモリーが不足している構成では、AIX-Based Systems のバッファ・キャッシュ・サイズの再設定が必要な場合があります。詳細は、「メモリーおよびページング」を参照してください。
  4. AIX-Based Systems 環境では、RAW デバイスの名前は変更される可能性があるため、可能な場合は常にシンボリック・リンクを使用することをお勧めします。たとえば、ディスク周辺装置を再構成する場合、またはデータベースを新たなハードウェア構成に移動する場合、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係した操作上の問題の詳細は、『Oracle8i for AIX-Based Systems インストレーション・ガイド』を参照してください。

## 双方の利点を最大限活用する

幸い、AIX-Based Systems では、あるデータ・ファイル用に JFS を選択し、他のデータ・ファイル用に RAW パーティションを選択することが可能です。非同期 I/O は、RAW パーティションと JFS のどちらでも機能します。データベース・オブジェクト上の I/O アクティビティの特性があらかじめわかっている場合、特定のオブジェクトに対応するデータ・ファイルを JFS または RAW パーティションに配置するためのスキーマを見積もることができます。可能な場合には、LVM を使用することもできます。

ワークロードの特性およびディスクやファイル・システム構成により、パフォーマンスが向上する度合は大きく異なります。

## 非同期 I/O の使用

Oracle8i は、AIX-Based Systems が提供する非同期 I/O (AIO) を十分に利用するため、データベース・アクセスの速度が向上します。AIO は、複数の I/O 操作を組み合わせて、I/O サブシステムのスループットを改善します。AIO の提供するメリットを活用できるのは、データが異なるディスクに適切に分散されている場合のみです (データベース、ログおよび制御ファイルの分散については『Oracle8i パフォーマンスのための設計およびチューニング』を参照してください)。

論理ボリューム・マネージャ (LVM) およびストライプ化を使用すると、AIO の効率を上げることができます。複数のディスク・スピンドルを使用してデータをストライプ化することで、LVM はディスクの競合を減らします。LVM とともに AIO を使用すると、RDBMS のパフォーマンスは大幅に改善されます。

AIX-Based Systems 4.x は、データベース・ファイルがファイル・システム・パーティションと RAW デバイスのどちらで作成された場合でも、AIO をサポートします。AIO 要求を発行後、アプリケーション・プロセスは実行を継続できます。ファイル・システム上の AIO 要求の場合、サーバーと呼ばれるカーネル・プロセス (kproc) が、キューから取得された時から完了するまで各要求を管理します。サーバーの数により、システムで同時実行可能な AIO 要求の数が制限されます。RAW デバイス上の AIO はカーネル化されます。つまり、AIO サーバーは AIO 要求を処理する必要があります。

---

---

**注意：** カーネル化された AIO を VSD および HSD で使用する場合、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。そうしない場合、EMSGSIZE 「Message too long」 エラーが表示されます。

---

---

サーバー数は、smit (「smit」->「Devices」->「Asynchronous I/O」->「Change/Show Characteristics of Asynchronous I/O」->「{MINIMUM|MAXIMUM} number of servers or smit aio」) または chdev を使用して設定できます。最小値は、システム起動時に起動するサーバー数です。最大値により、同時に発生した多数の要求を処理するために起動可能なサーバー数が制限されます。

デフォルト値は、minservers=1 および maxservers=10 です。これらの値は、通常、カーネル化された AIO を使用しない Oracle RDBMS を実行するには低すぎます。次のように値を設定することをお勧めします。

- `maxservers= 10 × (AIO が同時にアクセスするディスク数) 以上`
- `minservers= maxservers/2`

I/O スレーブ

I/O スレーブは、I/O の実行のみを目的とするプロセスです。I/O スレーブは Oracle7 の複数 DBWR にかわる機能です（実際は、複数 DBWR を汎化したもので、他のプロセスでも同様に実行できます）。また、非同期 I/O が使用可能かどうかに関係なく動作します。I/O スレーブは、`LARGE_POOL_SIZE` が設定されている場合はそこから割り当てられます。そうでない場合は、共有メモリー・バッファから割り当てられます。I/O スレーブには、初期化パラメータ・セットが付属しています。このパラメータ・セットによって、スレーブの動作を制御できます。表 2-1 に、I/O スレーブの初期化パラメータを示します。

表 2-1 I/O スレーブの初期化パラメータ

パラメータ	許容値	デフォルト値
<code>DISK_ASYNC_IO</code>	<code>TRUE/FALSE</code>	<code>TRUE</code>
<code>TAPE_ASYNC_IO</code>	<code>TRUE/FALSE</code>	<code>TRUE</code>
<code>BACKUP_TAPE_IO_SLAVES</code>	<code>TRUE/FALSE</code>	<code>FALSE</code>
<code>DBWR_IO_SLAVES</code>	<code>0 - 999</code>	<code>0</code>
<code>DB_WRITER_PROCESSES</code>	<code>1-10</code>	<code>1</code>

非同期 I/O が必要でなかったり、使用できなかったりする場合があります。表 2-1 に示されている最初の 2 つのパラメータ (`DISK_ASYNC_IO` および `TAPE_ASYNC_IO`) によって、ディスクおよびテープ・デバイスに対する非同期 I/O をオフにできます。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、特別に設定しない限り、I/O スレーブは実行されません。

`DISK_ASYNC_IO` または `TAPE_ASYNC_IO` が使用できない場合、`DBWR_IO_SLAVES` を 0 より大きい値に設定する必要があります。そうしないと、DBWR がボトルネックを起こします。AIX-Based Systems では、このような場合の `DBWR_IO_SLAVES` の最適値は 4 です。

`DB_WRITER_PROCESSES` (Oracle7 の `DB_WRITERS` パラメータの後継) 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` を使用する場合、`DB_WRITER_PROCESSES` の設定に関係なく、使用するデータベース・ライター・プロセスの数は 1 つのみです。

## DB\_FILE\_MULTIBLOCK\_READ\_COUNT の使用

init.ora のパラメータ DB\_FILE\_MULTIBLOCK\_READ\_COUNT の値が大きい場合、通常、I/O スループットが向上します。AIX-Based Systems では、このパラメータの範囲は 1~512 ですが、値が 16 を超えると通常パフォーマンスの向上は見られなくなります。

DB\_BLOCK\_SIZE に DB\_FILE\_MULTIBLOCK\_READ\_COUNT を掛けた値が LVM ストライプ・サイズよりも大きくなるように、このパラメータの値を設定する必要があります。このため、I/O 要求を満たすために、使用するディスク装置の数を増やします。

## RAID 機能の使用

RAID 5 により、順次読取りパフォーマンスは向上しますが、全体的な書込みパフォーマンスは低下します。一般的に、RAID 5 は、集中的な書込みの行われないワークロードでの使用が推奨されます。RAID 5 での集中的な書込みは、RAID 以外の環境と比較して、パフォーマンスを低下させます。

一般的に、RAID 0/1 はパフォーマンスを向上させます。これは、RAID 0/1 により、AIX-Based Systems または Oracle レベルでより効率の高い方法である、ハードウェア・レベルのストライプ / ミラー化が導入されるためです。

RAID 7 は、非同期 I/O（デバイス、パス、バスなど）および複数のホスト・インタフェースの実現に埋込み OS を使用します。RAID 7 は、RAID 0-6 よりも、小単位データおよび大量データの読取り / 書込みパフォーマンスが優れています。

## 後書きの使用

ファイル・システムは、各ファイルを 16KB のパーティションに分割することにより、書込みパフォーマンスを向上させ、メモリー内の使用済ページ数を制限し、ディスク断片化を最小限に抑えます。指定されたパーティションのページは、プログラムが次の 16KB パーティションの最初のバイトを書き込むまで、ディスクに書き込まれません。遅延書込みパーティションのサイズは、vmtune コマンドを使用して変更できます。次のコマンドは、遅延書込み用バッファのサイズを、8 つの 16KB パーティションに設定します。

```
vmtune -c 8
```

次のコマンドは、遅延書込みを無効にします。

```
vmtune -c 0
```

遅延書込みを無効にすることで、いくつかのパフォーマンス向上が見られる場合があります。

## 順次ファイルの先読みのチューニング

Virtual Memory Manager (VMM) は、順次ファイルのページの必要性を予測します。VMM は、プロセスがファイルにアクセスするパターンを監視します。ファイルの連続した 2 ページにプロセスがアクセスすると、VMM はプログラムがその後もファイルに順次アクセスするものと見なし、ファイルの順次先読みをスケジュールします。読取りはプログラムの処理とオーバーラップして行われるため、プログラムはデータをより早く処理できるようになります。2 つの VMM しきい値により、先読みするページ数が決定されます。

- minpgahead

VMM による最初の順次アクセス・パターン検出時の先読みページ数

- maxpgahead

VMM による順次ファイルの先読みページ数の最大値

minpgahead および maxpgahead を、アプリケーションに適した値に設定します。デフォルト値は、それぞれ 2 および 8 です。これらの値は、vmtune コマンドを使用して変更できます。ストライプ化された論理ボリュームの順次パフォーマンスが最重要課題のシステムでは、maxpgahead により高い値を指定できます。minpgahead および maxpgahead は、vmtune を使用して設定できます。次のコマンドは、minpgahead を 512 ページに、maxpgahead を 1024 ページに設定します。

```
vmtune -r 512 -R 1024
```

minpgahead および maxpgahead には、2 の累乗を設定してください。

## ディスク I/O の歩調合せ

AIX-Based Systems のメカニズムであるディスク I/O の歩調合せは、システム管理者がファイルに対して保留される I/O 要求数を制限することを可能にします。このメカニズムにより、ディスク I/O の集中するプロセスで CPU が飽和状態になるのを防ぎます。このため、対話型および CPU 集中型プロセスの応答時間が低下しません。

ディスク I/O 歩調合せは、システム・パラメータの最高水位標および最低水位標を調整することで設定します。プロセスが、保留中の I/O 要求が最高水位標に達しているファイルに対してプロセスが書き込みを実行すると、そのプロセスはスリープします。待機中の I/O 要求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は、smit を使用して変更できます。（「smit」->「System Environment」->「Change/Show Characteristics of Operating System」）を選択して、最高水位標および最低水位標を新たに入力します。いろいろな設定を試しながら、最適な水位標を判断します。設定は、注意深く行ってください。適切に設定しない場合、パフォーマンスが悪影響を受けます。ディスク I/O が 4KB を超える場合は、最高水位標および最低水位標をチューニングしても、あまり効果はありません。

## ディスク形状の考慮事項

AIX-Based Systems 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できます。複数のホット論理ボリュームを近接して配置すると、ディスクのシーク時間が減少するため、全体のパフォーマンスが向上します。

## リモート I/O の最小化

SP アーキテクチャ上で稼動する Oracle Parallel Server は、異なるノード上のすべての OPS インスタンスからアクセス可能な共通記憶装置として、仮想共有ディスク（VSD）またはハッシュ共有ディスク（HSD）を使用します。I/O 要求の対象が、ノードに対してローカルな論理ボリュームの VSD である場合、ローカル I/O が実行されます。VSD へのローカルではない I/O 通信量は、ネットワーク通信レイヤーを経由します。

より高いパフォーマンスを得るには、リモート I/O を可能な限り最小化することが重要です。各インスタンスの REDO ログは、ローカルの論理ボリューム上の VSD に配置する必要があります。更新および挿入が集中的に行われる場合、各インスタンスは、独自のプライベート・ロールバック・セグメントを、ローカルの論理ボリュームにマップされた VSD 上に保持する必要があります。

各ユーザーに許可される一時表領域は、各セッションに 1 つのみです。各一時表領域には、OPS を実行する各ノードにローカルなデータ・ファイルを少なくとも 1 つ含める必要があります。

アプリケーションおよびデータベースを注意深く設計して（たとえば、アプリケーションとデータベースをパーティション化して）、リモート I/O を少なくしてください。

## VSD キャッシュ・バッファを使用しない

IBM は、次の理由のために、通常の状態では VSD キャッシュ・バッファを使用しないこと（キャッシュなし）を推奨しています。VSD LRU キャッシュ・バッファは、より効果的な使用が可能な確保済のカーネル・メモリーを使用します。また、キャッシュ・バッファが有効な場合、物理読取りごとに、オーバーラップしたページをキャッシュ・ブロックで検索すること、およびキャッシュ・バッファとの間でデータをコピーすることに伴うオーバーヘッドが発生します。

statvdsd コマンドを使用して、VSD のパフォーマンスをチェックできます。statvdsd が、バディ・バッファ待ちの要求キューを示す場合、バディ・バッファをさらに追加しないようにしてください。かわりに、次のようにして切替え送信プールのサイズを増やします。

```
# /usr/lpp/ssp/css/chgcass -l css0 -a spoolsize=new_size_in_byte
```

送信プール・サイズを増やしたら、次のようにして mbuf の上限値も増やす必要があります。

```
# /etc/no -o thewall=new_size_in_kbyte (Maximum 64MB)
```

現行の送信および受信プール・サイズを確認するには、次のように入力します。

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

---

**参照：** SP および VSD のチューニングの詳細は、  
<http://www.rs6000.ibm.com/support/sp/perf/>、  
[http://www.rs6000.ibm.com/resource/technology/sp\\_papers](http://www.rs6000.ibm.com/resource/technology/sp_papers)  
および <http://www.rs6000.ibm.com/resource/technology/#sp> 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

CPU は、プロセスが競合する可能性のある別のシステム・コンポーネントです。AIX-Based Systems のカーネルは、大抵の場合 CPU を効率的に割り当てますが、多くのプロセスが CPU サイクルを獲得するために競合します。CPU を 2 つ以上搭載したシステム（SMP）では、様々な CPU 上でレベルの異なる競合が発生する可能性があります。

## プロセスのランタイム・スライスの変更

AIX-Based Systems 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。タイム・スライスは、`schedtune` コマンドを使用して変更できます。ただし、タイム・スライスを変更する際には注意が必要です。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、タイム・スライスが長くなると、コンテキスト切替え率が低下します。その結果、プロセスのコンテキスト切替えに費やされる CPU サイクルが減少するため、システムのスループットが向上します。

ただし、実行時スライスが長くなることで、特に単一プロセッサ・システムでの応答時間が低下する可能性があります。通常、デフォルトの実行時スライス（10 ミリ秒）は、標準的なアプリケーションに適切な値です。実行キューが高く、多くのアプリケーションおよび Oracle シャドウ・プロセスがより長い継続時間に対応している場合、次のように入力して、タイム・スライスを増やすことができます。

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

この場合、`n` の値は 0=10ms、1=20ms、2=30ms、3=40ms などです。

## SMP マシンでのプロセッサ・バインディングの使用

特定のプロセスをプロセッサにバインディングすることにより、SMP マシンのパフォーマンスを実質的に改善できます。プロセッサ・バインディングは、AIX-Based Systems 4.x で利用可能で、完全に機能します。

プロセッサ・バインディングを使用することの利点を、次に示します。

- 優先度の高いアプリケーションに、比較的多くの CPU 時間が割り当てられます。
- 長期間にわたって、プロセス・コンテキストをメンテナンスできます。



AIX-Based Systems では、プロセッサ・バインディングは自動ではありません。マルチ・プロセッサの場合、`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（`man` ページを参照）。Oracle プロセスをプロセッサにバインドできるのは、スーパーユーザーまたは Oracle 所有者のみです。子プロセスは、プロセッサ・バインディングを継承します。

様々な Oracle バックグラウンド・プロセス（`dbwr` を除く）を異なるプロセッサにバインドし、1つのプロセッサを `dbwr` 専用にするをお勧めします。これにより、`dbwr` を実行するプロセッサが保証されると同時に、CPU の限界に達した場合に `dbwr` が他のプロセッサに自由に移動できるようになります。

**注意：** プロセッサ・バインディングは複雑な操作であるため、注意深く扱う必要があります。プロセッサにバインドされたプロセスを、別のプロセッサに（たとえそのプロセッサが空き状態にあるとしても）移行することはできません。移動すると、アプリケーションのパフォーマンスを低下させる可能性があります。同種のアプリケーションが複数存在する負荷の均衡が取れた環境の方が、プロセッサ・バインディングに適しています。

プロセスのプロセッサへのバインディングは、排他的な操作ではありません。プロセッサは、他のプロセスを自由に実行できます。

## 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、Net8 リスナー経由で Oracle サーバー・プロセスに接続する場合（ネットワーク化されたクライアントおよびサーバー Oracle）、Net8 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。その後リスナーが起動する Oracle サーバー・プロセスはすべて、同じプロセッサにバインドされます。

これを実行する 1 つの方法は、すべてのリスナーにそれぞれのポートを割り当て、複数のリスナーを起動します。そのために、`$ORACLE_HOME/network/admin/listener.ora` ファイルをカスタマイズして、各リスナーが 1 連になるように、設定する必要があります。サーバー側で複数のリスナーを起動します。クライアント側では、`tnsnames.ora` をカスタマイズすることにより、クライアントだけでなくアプリケーションも、別々のリスナーが割り当てた異なるポートに接続します。たとえば、`listener.ora` を変更してリスナー L1 および L2 にそれぞれポート 1521 と 1522 を割り当てるには、次のように入力します。

```
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)
    )
  )
```

これら 2 つのリスナーを起動します。

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

2 つのリスナーのプロセス ID を取得します。

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

リスナーをプロセッサにバインドします。

```
$ bindprocessor process_id_for_L1 processor_id
$ bindprocessor process_id_for_L2 processor_id
```

この場合、

```
processor_id = 0 ... number_of_CPUs - 1
```

特定のクライアント・アプリケーションの CPU 消費を制限する場合、この方法を使用してリスナーをバインドして、そのプロセッサが使用可能な場合に、そのプロセッサ上でのみアプリケーションが稼動するようにできます。

### ローカル環境でのプロセッサ・バインディング

クライアントと Oracle Server が、パイプ・ドライバを 2 タスクで使用して同一のマシン上で稼動する場合、プロセッサ・バインディングはより複雑です。これには、各サーバー・プロセスのプロセス ID を見つけて手動でプロセッサとバインドする必要があります。サーバーのプロセスが長期間継続しない限り、管理用に過剰なオーバーヘッドが発生するために、多くの場合この労力を払う価値はありません。

Oracle プロセスのプロセッサ・バインディングは、同一マシン上で稼動する Oracle 以外のアプリケーションのパフォーマンスにマイナスの影響を与えます。注意深いチューニングと監視を行うよう、強くお勧めします。

## その他のチューニングのヒント

### UDP のチューニング

Oracle Parallel Server は、UDP を使用してプロセス間通信を行います。UDP カーネル設定をチューニングすることにより、Oracle のパフォーマンスを改善できます。AIX のカーネル UDP バッファの変更は、ネットワーク・オプション `udp_sendspace` および `udp_recvspace` を変更することにより実行します。

- `udp_sendspace` を、最大値の 65536 に設定する必要があります。
- `udp_recvspace` は、`sb_max` 未満である必要があります。

通常、`udp_recvspace` の値は、`udp_sendspace` 値を 10 倍した値から開始されます。これは、UDP では、別のパケットが到着するまで、パケットをアプリケーションに送信することはできないためです。

`udp_recvspace` 設定を評価するには、次のコマンドを実行します。

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

オーバフロー数がゼロ以外の場合、`udp_recvspace` の値を増やします。

### RAW デバイスのバックアップ

RAW デバイスのバックアップには、`dd` コマンドを使用します。RAW デバイスの最初の 4KB ブロックは、AIX-Based Systems の論理ボリューム・マネージャが使用するため、この領域を上書きしてはなりません。ブロックサイズを 4KB より大きくすると、バックアップの速度が向上します。テープにバックアップを作成する場合の望ましいブロックサイズは、256KB です。RAW デバイスとテープ間のバックアップ / リストアに使用可能なサンプル・スクリプトを次に示します。

1. RAW デバイスからテープにバックアップする場合：

```
dd if=/dev/<raw_device> of=/dev/rmt0.1 bs=256k
```

2. テープから RAW デバイスにリストアする場合：

```
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。`restore` コマンドは、次のように簡潔にもできます。

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

---

## パラレル・リカバリの使用

`init.ora` のパラメータ `recovery_parallelism` を `N` に設定することで、パフォーマンスを向上させることができます (`N` は 1 より大きい整数値、通常はマシンのプロセッサ数を設定)。次に、複数のスレーブ・リカバリ・プロセスが生成されて、ログ・ファイルの変更をデータベースに適用するタスクが分割されます。

---

## SQL\*Plus の管理

この章では、SQL\*Plus の次の機能について説明します。

- [SQL\\*Plus の管理](#)
- [SQL\\*Plus の使用](#)
- [SQL\\*Plus の制限事項](#)

## SQL\*Plus の管理

この項では、SQL\*Plus の管理方法について説明します。

### 設定ファイルの使用

SQL\*Plus を起動すると、Site Profile の設定ファイル `glogin.sql` および User Profile の設定ファイル `login.sql` が実行されます。

#### Site Profile ファイルの使用

グローバルな Site Profile は、`$ORACLE_HOME/sqlplus/admin/glogin.sql` です。SQL\*Plus をインストールすると、デフォルトの Site Profile は `$ORACLE_HOME/sqlplus/admin` に保存されます。Site Profile がすでに存在する場合は、上書きされます。SQL\*Plus を削除すると、既存の Site Profile も削除されます。

#### User Profile ファイルの使用

User Profile は、`login.sql` です。SQL\*Plus は現行のディレクトリを最初に検索し、次に指定したディレクトリを `login.sql` が検出されるまで検索します。環境変数 `SQLPATH` には、ファイルを検索するディレクトリのリストをコロンで区切って設定します。

たとえば、現在のディレクトリが `/u02/oracle` で、`SQLPATH` が `/home:/home/oracle:/u01/oracle` に設定されている場合、SQL\*Plus は次の順番で `login.sql` を検索します。

1. `/u02/oracle` (現行のディレクトリ)
2. `/home`
3. `/home/oracle`
4. `/u01/oracle`

`login.sql` に設定されるオプションは、`glogin.sql` に設定されているオプションを上書きします。

**参照：** Profile ファイルの詳細は、『Oracle8i 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

Typical インストール時に、`PRODUCT_USER_PROFILE` 表が自動的に作成されます。この表は、指定した SQL および SQL\*Plus コマンドを使用禁止にする場合に使用します。この表を再作成するには、`SYSTEM` スキーマの `$ORACLE_HOME/sqlplus/admin/pupbld.sql` スクリプトを実行します。

たとえば、次のように入力します。

```
$ sqlplus system/manager
SQL> @?/sqlplus/admin/pupbld.sql
```

SQL\*Plus は、通常は疑問符「?」が指定されている場所に \$ORACLE\_HOME の値を使用します。

## デモンストレーション表の使用

SQL\*Plus 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。

### Typical インストールの実行

Typical インストールの場合、ユーザー SCOTT およびデモンストレーション表が自動的に作成されます。

### 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を作成するには、SQL スクリプト

\$ORACLE\_HOME/sqlplus/demo/demobld.sql を実行します。SQL\*Plus では、任意のユーザー名を使用して demobld.sql ファイルを実行し、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、\$ORACLE\_HOME/sqlplus/demo/demobld.sql 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を削除するには、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、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、\$ORACLE\_HOME/sqlplus/demo/demobld.sql を実行することもできます。

```
$ demodrop scott tiger
```

---

**注意：** SQL スクリプト demobld.sql および demodrop.sql はどちらも、EMP、DEPT、BONUS、SALGRADE および DUMMY 表を削除します。これらと同じ名前をもつ表がスキーマに存在しないことを確認してから、スクリプトを実行してください。そうしないと、表のデータが失われます。

---

## ヘルプ機能の使用

この項では、ヘルプ機能の使用方法について説明します。

### Typical インストールの実行

Typical インストール、または Oracle Database Configuration Assistant のオプションで、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場合、SQL\*Plus は自動的にヘルプ機能をインストールします。

### Database Configuration Assistant の使用

データベースを作成する場合、Oracle Database Configuration Assistant を使用してヘルプ表を作成できます。

### ヘルプ機能の手動インストール

シェル・スクリプト `$ORACLE_HOME/bin/helpins` を使用すると、ヘルプ機能を手動でインストールできます。スクリプトを実行する前に、環境変数 `SYSTEM_PASS` に、`SYSTEM` スキーマ名およびパスワードを設定してください。たとえば、次のように入力します。

```
$ setenv SYSTEM_PASS SYSTEM/MANAGER
$ helpins
```

環境変数 `SYSTEM_PASS` が設定されていない場合、`helpins` から、`SYSTEM` パスワードの入力を求めるプロンプトが表示され、`SYSTEM` スキーマにヘルプ・データがロードされます。ヘルプ機能をインストールするには、`$ORACLE_HOME/sqlplus/help/helpbld.sql helpus.sql` を実行することもできます。SQL\*Plus では、システム・ユーザーはファイル `helpbld.sql` およびその引数 `helpus.sql` を実行して、ヘルプ機能表を作成できます。たとえば、次のように入力します。

```
$ sqlplus system/manager
SQL> @?/sqlplus/admin/help/helpbld.sql helpus.sql
```

---

---

**注意：** シェル・スクリプト `helpins` および SQL\*Plus スクリプト `helpbld.sql` は、新しい表を作成する前に既存のヘルプ機能表を削除します。

---

---

SQL\*Plus で `$ORACLE_HOME/sqlplus/help/helpbld.sql` を実行して、スキーマにあるヘルプ機能表を手動で削除することもできます。たとえば、次のように入力します。

```
$ sqlplus system/manager
SQL> @?/sqlplus/admin/help/helpdrop.sql
```



**参照：** ヘルプ機能の詳細は、『Oracle8i/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 の使用

この項では、SQL\*Plus の使用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`、`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。起動するには、環境変数 `PATH` にエディタのディレクトリが含まれている必要があります。

エディタを起動すると、現行の SQL バッファはエディタに入れられます。エディタを終了すると、変更された SQL バッファが SQL\*Plus に戻されます。

SQL\*Plus の `_editor` 変数の定義によって、エディタを起動するように指定できます。この変数は、`glogin.sql` または `login.sql` に設定するか、または SQL\*Plus セッション中に入力します。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` 変数が設定されていない場合は、環境変数 `EDITOR` または `VISUAL` のどちらかの値が使用されます。環境変数 `EDITOR` および `VISUAL` が両方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` および `VISUAL` が設定されていない場合は、デフォルト・エディタは `ed` になります。

エディタを起動すると、SQL\*Plus はエディタにテキストを渡すためにテンポラリ・ファイル `afiedt.buf` を使用します。`SET EDITFILE` を使用すると、このファイルを改名できます。たとえば、次のように入力します。

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

SQL\*Plus は、テンポラリ・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 プロンプトの後に最初の文字として 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。デフォルトのシェルは、/bin/sh (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。

HOST コマンドまたは ! を入力することによって、SQL\*Plus を終了しなくても、オペレーティング・システム・コマンドを実行できます。たとえば、1 つのコマンドを実行するには、次のように入力します。

```
SQL> ! command
```

この場合、`command` は実行するオペレーティング・システム・コマンドを表します。

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 コマンドまたは ! を入力した後に [Enter] キーを押します。コマンドの実行後、制御は SQL\*Plus に戻ります。

## SQL\*Plus への割込み

SQL\*Plus を実行している間、[Ctrl]+[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拡張子は、`.lis` です。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.lis
```

## 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 LINESIZE および PAGESIZE のデフォルト値によって、ウィンドウ・サイズが自動的に調整されるわけではありません。

## 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。

---

## Oracle Message Broker の構成

この章では、Oracle Message Broker のインストール後に必要なタスクについて説明します。インストールでは様々なドライバを使用できるため、これらのいくつかはオプションのタスクです。

- [JDK のパスの更新](#)
- [ディレクトリのインストールおよび構成](#)
- [Oracle Advanced Queuing の構成（オプション）](#)
- [MQSeries のインストール・タスク（オプション）](#)
- [TIBCO のインストール・タスク（オプション）](#)
- [Asynchronous Component Invocation のインストール・タスク（オプション）](#)
- [OMB インスタンスの作成および Oracle Message Broker の使用](#)

## JDK のパスの更新

ご使用のシステムに正しいバージョンの JDK がインストールされていること、および PATH が適切に設定されていることを確認してください。複数のバージョンの JDK がインストールされている場合は、パスの先頭に、Oracle Message Broker でインストールされたバージョンの JDK が指定されていることを確認してください。AIX-Based Systems では、JDK バージョン 1.1.8.3 がサポートされます。

インストーラにより、JDK をインストールするディレクトリを指定すること、および提供される JDK を使用するために、\$OMB\_HOME/bin ディレクトリ内の ombenv 環境スクリプトにパスを設定するよう促されます。

## ディレクトリのインストールおよび構成

LDAP ディレクトリは、Oracle Message Broker が使用できるシステムにインストールし、また、これを使用するための構成を行う必要があります。LDAP ディレクトリは、Oracle Message Broker と同じシステムにも、別のシステムにもインストール可能です。この項では、ディレクトリのインストールおよび構成について説明します。

Oracle Message Broker は、管理機能を処理するために、LDAP ディレクトリが必要です。Oracle Message Broker は、次の 2 つの LDAP ディレクトリ製品をサポートしています。

- Oracle Internet Directory (OID) リリース 2.1.1
- Netscape Directory Server (Windows NT システムで動作します)

Oracle Internet Directory および Oracle Message Broker に対して同じ \$ORACLE\_HOME を使用する場合は、Oracle Message Broker より先に Oracle Internet Directory をインストールします。

---

**注意：** Oracle Internet Directory をインストールする場合、ORACLE\_HOME のパスに大文字を使用しないでください。大文字を使用すると、インストールが失敗する可能性があります。

---

**参照：** Oracle Internet Directory のインストールの詳細は、『Oracle8i AIX-Based Systems インストレーション・ガイド』およびリリース・ノートを参照してください。

Netscape Directory Server のインストールの詳細は、『Oracle Message Broker Installation Guide』を参照してください。

## ディレクトリのバインド DN の指定

インストール中に使用されるコマンドも含めて、LDAP ディレクトリにアクセスするすべての Oracle Message Broker コマンドには、バインド DN およびパスワードを指定する必要があります。

Oracle Internet Directory の場合、デフォルトのバインド DN は、cn=orcladmin で、デフォルトのパスワードは、welcome です。バインド DN の cn=orcladmin は、デフォルトのスーパーユーザー・バインド DN で、スーパーユーザー権限があります。LDAPSchema コマンドのみ、スーパーユーザー権限が必要ですが、ディレクトリの設定によっては、InitDir コマンドでも必要になります。

ディレクトリにユーザーおよびグループ・エントリを作成し、スーパーユーザー以外のエントリに対するバインド DN とパスワードを使用することをお勧めします。ユーザーおよびグループのエントリの設定には、LDAP および LDAP サーバーに固有のツール（oidadmin など）に関する知識が必要です。

**参照：** Oracle Message Broker で必要なユーザーのタイプについては、『Oracle Message Broker Administration Guide』を参照してください。

バインド DN およびパスワードの指定には、次の 3 つの方法があります。

- InitDir などのコマンド実行中に表示されるプロンプトで指定する
- コマンドライン・オプションの -D および -w を使用する
- JNDI プロパティを直接設定する。JNDI プロパティを設定し、設定した値を Oracle Message Broker コマンド実行時に使用するには、コマンドライン・オプション -noauth を使用するか、または認証プロンプトで、値を指定しないようにします。

セキュリティ・オプションを設定するための JNDI Java プロパティは、次のとおりです。

```
java.naming.security.principal
java.naming.security.credentials
java.naming.security.authentication=simple
```

## ディレクトリ構成

LDAP ディレクトリおよび Oracle Message Broker をインストールした後、ディレクトリに対して複数の構成タスクを実行する必要があります。タスクでは、Oracle Message Broker によってインストールされたツールを使用します。これらのツールはディレクトリを変更し、アクティブなディレクトリ・サーバーを必要とします。ディレクトリの変更によって、Oracle Message Broker の管理機能がサポートされます。

1. Oracle8i データベース・サーバーの起動および実行後に、LDAP ディレクトリを起動します。
  - a. OID コマンド oidmon を使用して OID モニターを起動します。

```
% oidmon connect=connect_string start
```

この場合、connect\_string には、データベース・サーバーに接続するための Net8 接続記述子を指定します。

- b.** OID コマンド `oidctl` を使用して OID インスタンスを起動します。

```
% oidctl connect=connect_string server=oidldapd instance=1 start
```

この場合、`connect_string` には、データベース・サーバーに接続するための Net8 接続記述子を指定します。

2. ご使用のシステムに正しいバージョンの JDK がインストールされていることを確認します。  
複数のバージョンの JDK がインストールされている場合は、パスの先頭に、Oracle Message Broker で必要なバージョンが指定されていることを確認してください。
3. ディレクトリ・スキーマを変更して、Oracle Message Broker をサポートします。

- a.** システム環境を設定します。

Bourne シェルまたは Korn シェルの場合

```
$ . $ORACLE_HOME/omb/2.0/bin/ombenv.sh
```

### C シェルの場合

```
% source $ORACLE_HOME/omb/2.0/bin/ombenv.csh
```

Oracle Message Broker のインストールによって、起動スクリプト `ombenv.sh` または `ombenv.csh` が作成されます。

- b. \$ORACLE\_HOME/omb/2.0/bin ディレクトリの LDAPSchema コマンドを使用して、ディレクトリ・スキーマを更新します。

```
% LDAPSchema [options]
```

表 4-1 にオプションを示します。

LDAPSchema コマンドによって、LDAP ディレクトリ・エントリ `cn=OracleSchemaVersion` がルートに作成されます。このエントリには、Oracle Message Broker の LDAP スキーマのバージョン・ナンバーを定義する、`orclProductVersion` 属性が含まれます。エントリ `cn=OracleSchemaVersion` が存在しない場合、または `orclProductVersion` の値が LDAPSchema コマンドの現在のバージョンよりも低い場合にのみ、LDAPSchema は LDAP スキーマを更新します。

表 4-1 LDAPSchema コマンドライン・オプション

オプション	説明
-D <i>auth_dn</i>	<i>auth_dn</i> には、ユーザー認証に使用する DN を指定します。
-errorlevel <i>level</i>	エラー・レポートのレベルを設定します。パラメータの <i>level</i> には、1 ～ 4 の整数を指定します。 <ul style="list-style-type: none"> <li>1 - 最上位の例外に対するエラー・メッセージの出力</li> <li>2 - リンクされているすべての例外に対するエラー・メッセージの出力</li> <li>3 - 最上位の例外に対するスタック・トレースの出力</li> <li>4 - リンクされているすべての例外に対するスタック・トレースの出力</li> </ul> デフォルト値 : 2
-h <i>host-name</i>	<i>host-name</i> は、ディレクトリがインストールされるホストです。 デフォルト値 : localhost
-noauth	LDAP サーバーでの LDAP 認証が必要でないことを指定します。
-p <i>port</i>	<i>port</i> は、指定されたホストのディレクトリにアクセスするために使用するポートです。 デフォルト値 : 389
-P <i>wallet_password</i>	Wallet パスワードを指定します。-U の値が 0 または 1 の場合は、無視されます。
-ldapv2	LDAP ディレクトリが LDAP バージョン 2 のみをサポートする場合、このオプションを使用します。 デフォルト値 : LDAPSchema は LDAP バージョン 3 をサポートします。
-U <i>value</i>	SSL を使用するかどうか、および使用する場合はその認証レベルを指定します。 <i>values</i> の有効値 : 0、1、2、3 <ul style="list-style-type: none"> <li>0 - SSL は使用されません。-U が指定されなかった場合のデフォルト値です。</li> <li>1 - 認証されていない SSL が使用されます。</li> <li>2 - サーバー側で認証されている SSL が使用されます。</li> <li>3 - サーバー側およびクライアント側で認証されている SSL が使用されます。</li> </ul>
-version	バージョン情報を提供します。バージョン情報の詳細を表示する場合は、-fullVersion を使用します。
-w <i>auth_passwd</i>	LDAP サーバーの認証に対して、パスワード <i>auth_passwd</i> を提供します。
-W <i>wallet_path</i>	エクスポートされた Wallet ファイルへのパスを指定します。-U の値が 0 または 1 の場合は、無視されます。

セキュリティ・オプション -D、-noauth または -w が使用されていない場合、LDAPSchema は認証 DN およびパスワードを入力するプロンプトを表示します。

LDAP ディレクトリが認証可能な場合は、ディレクトリ管理者の DN およびパスワードを入力します。

たとえば、ホスト名が `system1`、デフォルトのポート番号が `389` で実行している LDAP ディレクトリの LDAP スキーマを変更するには、次のコマンドを使用します。

```
% LDAPSchema -h system1
```

LDAPSchema によって、次のメッセージが表示された場合、[4-7 ページ](#)の手順 4 で説明する `catalog.sh` を実行する必要があります。

```
### ATTENTION: The attribute "orcloasentrytype" is not searchable. Please run
"catalog.sh" on OiD to make this attribute searchable
```

- c. `$ORACLE_HOME/omb/2.0/bin` ディレクトリの `InitDir` コマンドを使用して、Oracle Message Broker の接尾辞に対するディレクトリを変更します。次のように、`InitDir` を使用します。

```
% InitDir [options]
```

[表 4-2](#) に使用可能なオプションを示します。

表 4-2 InitDir コマンドライン・オプション

オプション	説明
-b <i>base_dn</i>	指定された <i>base_dn</i> にすべてのエントリを作成します。 <i>base_dn</i> に指定されている DN がディレクトリに存在する必要があります。
-c <i>country</i>	ルートのネーミング・コンテキストに使用する <i>country</i> を指定します。
-D <i>auth_dn</i>	<i>auth_dn</i> には、ユーザー認証に使用する DN を指定します。
-errorlevel <i>level</i>	エラー・レポートのレベルを設定します。パラメータの <i>level</i> には、1 ～ 4 の整数を指定します。  1 - 最上位の例外に対するエラー・メッセージの出力 2 - リンクされているすべての例外に対するエラー・メッセージの出力 3 - 最上位の例外に対するスタック・トレースの出力 4 - リンクされているすべての例外に対するスタック・トレースの出力  デフォルト値 : 2
-h <i>host-name</i>	<i>host-name</i> は、ディレクトリがインストールされるホストです。  デフォルト値 : localhost
-noauth	LDAP サーバーでの LDAP 認証が必要でないことを指定します。
-o <i>org</i>	ルートのネーミング・コンテキストに使用する <i>organization</i> を指定します。
-ou <i>org-unit</i>	ルートのネーミング・コンテキストに使用する <i>organizational unit</i> を指定します。



表 4-2 InitDir コマンドライン・オプション (続き)

オプション	説明
-p <i>port</i>	<i>port</i> は、指定されたホストのディレクトリにアクセスするために使用するポートです。 デフォルト値 : 389
-P <i>wallet_password</i>	Wallet パスワードを指定します。-U の値が 0 または 1 の場合は、無視されます。
-U <i>value</i>	SSL を使用するかどうか、および使用する場合はその認証レベルを指定します。 <i>values</i> の有効値 : 0、1、2、3 0 - SSL は使用されません。-U が指定されなかった場合のデフォルト値です。 1 - 認証されていない SSL が使用されます。 2 - サーバー側で認証されている SSL が使用されます。 3 - サーバー側およびクライアント側で認証されている SSL が使用されます。
-ldapv2	LDAP ディレクトリが LDAP バージョン 2 のみをサポートする場合、このオプションを使用します。デフォルトでは、LDAPSchema は LDAP バージョン 3 をサポートします。
-version	バージョン情報を提供します。バージョン情報の詳細を表示する場合は、-fullVersion を使用します。
-w <i>auth_passwd</i>	LDAP サーバーの認証に対して、パスワード <i>auth_passwd</i> を提供します。
-W <i>wallet_path</i>	エクスポートされた Wallet ファイルへのパスを指定します。-U の値が 0 または 1 の場合は、無視されます。

InitDir の -c、-o および -ou オプションに対して選択した値は、Oracle Message Broker Installer の実行中に接尾辞情報スクリーンに指定した、LDAP ベースのネーミング・コンテキスト情報に一致する必要があります。

セキュリティ・オプションの -D、-noauth または -w が使用されていない場合、InitDir は認証 DN およびパスワードを入力するプロンプトを表示します。

LDAP ディレクトリが認証可能な場合は、ディレクトリ管理者の DN およびパスワードを入力して InitDir を使用します。

たとえば、ホスト名が `system1`、デフォルトのポート番号が 389 で実行している LDAP ディレクトリを初期化するには、次のコマンドを使用します。

```
% InitDir -h system1 -c us -o oracle -ou sales
```

この場合、-c、-o および -ou オプションに指定したルートから、Oracle Message Broker 用にディレクトリが初期化されていることを前提としています。

4. Oracle Internet Directory の属性を検索可能にします。手順 3 で説明した LDAPschema コマンドが次のメッセージを表示した場合のみ、この手順を行います。

```
### Attention: The attribute orcloasentrytype is not searchable. Please run  
catalog.sh on OiD to make this attribute searchable.
```

カタログに追加されていない属性 `orcloasentrytype` を使用しているデータがディレクトリ内にある場合、`LDAPSchema` はこのメッセージを表示します。この場合、`LDAPSchema` は、この属性をカタログに追加できません。`catalog.sh` を手動で実行する必要があります。

- a. Oracle Internet Directory がインストールおよび実行されているシステム上で、次のコマンドを発行します。

```
% cd $ORACLE_HOME/ldap/bin
% catalog.sh -connect connect_string -add -attr orcloasentrytype
```

この場合、`connect_string` には、データベース・サーバーに接続するための Net8 接続記述子を指定します。

- b. `catalog.sh` コマンドがプロンプトを表示した場合、パスワードを入力します。デフォルト・パスワードは `ods` です。これは、OID データベース・サーバーのデフォルト・パスワードです。`catalog.sh` を実行する前に、`oidpasswd` を使用してパスワードを変更する場合は、デフォルト・パスワード以外の新規パスワードを使用してください。

---

**参照：** OID ディレクトリでこれらの手順を行うには、アクティブな Net8 リスナーが必要です。Net8 リスナーの詳細は、『Oracle8i Net8 管理者ガイド』を参照してください。

---

## OID の再起動（オプション）

`catalog.sh` の実行後、Oracle Internet Directory の再起動が必要な場合があります。また、この章の「[ディレクトリ構成](#)」の手順 4 の処理が完全に行われなかった場合にも、再起動が必要になる場合があります。手順 4 の処理が完全に行われなかった場合、`AdminUtil`、`ombadmin` などのディレクトリ操作で、次のエラー・メッセージが表示されることがあります。

```
DSA unwilling to perform: operation not supported
```

次の手順に従って、この問題を解決します。

1. 次のコマンドを実行します。

```
catalog.sh -connect connect_string delete -attr orcloasentrytype
catalog.sh -connect connect_string -add -attr orcloasentrytype
```

この場合、`connect_string` には、データベース・サーバーに接続するための Net8 接続記述子を指定します。

2. `oidctl` を使用して `oidldapd` インスタンスを停止します。
3. `oidmon` を停止します。

```
oidmon connect=connect_string stop
```

4. データベース・サーバーを停止します。

5. データベース・サーバーを起動します。
6. 次のコマンドを使用して、oidmon を起動します。  

```
oidmon connect=connect_string start
```
7. oidctl を使用して oidldapd インスタンスを起動します。

## Oracle Advanced Queuing の構成（オプション）

この項では、Oracle Message Broker とともに、Oracle8i Advanced Queuing の機能を構成する手順を説明します。Oracle Message Broker および Oracle AQ ドライバを使用している場合は、この項で説明する手順を実行してください。

## Oracle Internet Directory の停止

Oracle Internet Directory と同じ Oracle8i データベース・サーバーを使用して Oracle AQ を実行している場合、Oracle Internet Directory および Oracle8i データベース・サーバーを停止する必要があります。この項では、Oracle Internet Directory の停止について説明します。

1. OID コマンド oidctl を使用して OID インスタンスを停止します。

```
% oidctl connect=connect_string server=oidldapd instance=1 stop
```

この場合、*connect\_string* には、データベース・サーバーに接続するための Net8 接続記述子を指定します。

2. OID コマンド oidmon を使用して OID モニターを停止します。

```
% oidmon connect=connect_string stop
```

この場合、*connect\_string* には、データベース・サーバーに接続するための Net8 接続記述子を指定します。

**参照：** oidctl および oidmon の詳細は、『Oracle Internet Directory 管理者ガイド』を参照してください。

## データベース・サーバーおよびリスナーの停止

Oracle8i データベース・サーバーを実行しているシステムで、次の手順を実行します。

1. `lsnrctl` コマンドを使用して、Net8 リスナーを停止します。

```
lsnrctl
lsnrctl> stop
lsnrctl> quit
```

2. Oracle8i データベース・サーバーのホーム・ディレクトリに移動します。
3. SQL\*Plus を起動します。

```
% sqlplus
```

4. INTERNAL としてログインします。
5. データベースを停止します。
6. データベース・サーバーを停止します。

```
SQL> SHUTDOWN [MODE]
```

この場合、*MODE* には、次のいずれかを指定します。

Normal	データベース・サーバーは、現在の接続ユーザーがすべて切断するのを待ち、データベース・サーバーが停止されるまでの新しい接続を禁止します。これはデフォルト・モードです。
Immediate	データベース・サーバーが終了します。アクティブ・トランザクションをロールバックしてからクライアントを切断し停止します。

**参照：** リスナーの停止については、『Oracle8i Net8 管理者ガイド』を参照してください。

## データベース・パラメータの構成

Oracle8i データベース・サーバーは、Oracle Message Broker のリソース要件をサポートするように構成する必要があります。アクティブな Oracle Message Broker では、トランザクション、セッション、カーソルおよび処理が消費されます。データベース・サーバーを使用し、独自のリソース要求を持つアプリケーションもあります。

表 4-3 に、Oracle Message Broker のデータベース・サーバー要件を示します。データベース・サーバーを使用する他のアプリケーションの要件に加えて、これらの要件があります。

Oracle8i データベース・サーバーを構成する場合、次の点に注意してください。

- `initSID.ora` を編集した場合、DBMS を再起動する必要があります。
- 制限を設定する際には、慎重になりすぎる必要はありません。

- これらのパラメータの中には、それをサポートするために、OS カーネルを構成する必要があるものもあります。また、システムの再起動が必要な場合もあります。詳細は、プラットフォーム固有の Oracle8i インストレーション・ガイドを参照してください。

表 4-3 Oracle Message Broker に必要な Oracle8i データベース・サーバーのパラメータ

パラメータ	説明
AQ_TM_PROCESSES = <i>num_aq</i>	<i>num_aq</i> の値は、 $\geq 1$ である必要があります。
LICENSE_MAX_SESSIONS = <i>num_license</i>	<p><i>num_license</i> には、次に示すように、<i>num_process</i> より大きい値を設定する必要があります。</p> $num\_license > num\_processes$ <p><i>num_processes</i> の値については、後述の PROCESSES を参照してください。</p>
OPEN_CURSORS = <i>num_open</i>	<p><i>num_open</i> は、次の式で求められる値以上である必要があります。</p> $num\_open = \text{Max} ((2 \times maxConsumers,) numQueues) \times 1.2$ <p>この場合、次のような意味になります。</p> <p><i>maxConsumers</i> は、コンシューマの最大数です。コンシューマは、JMS セッション内の QueueReceiver または TopicSubscriber です。</p> <p><i>numQueues</i> は、Oracle Message Broker で構成されている AQ キューの数です。</p> <p>この式によって、すべての Oracle Message Broker 管理データベース・サーバー接続で使用するカーソルの最大数を決定します。他のユーザーがさらに多くのカーソルを必要とする場合、OPEN_CURSORS には必要に応じた値を設定してください。</p>
PROCESSES= <i>num_processes</i>	<p><i>num_processes</i> は、次の式で求められる値以上である必要があります。</p> $num\_processes = (maxPrivate + maxShared + aq\_tm\_processes) \times 1.2$ <p><i>maxPrivate</i> は、現在データベース・サーバーを使用しているすべての Oracle Message Broker の属性 <i>maxPrivateSessions</i> の合計です。</p> <p><i>maxShared</i> は、現在データベース・サーバーを使用しているすべての Oracle Message Broker の属性 <i>maxSharedSessions</i> の合計です。</p> <p><i>aq_tm_processes</i> は、init.ora ファイルの、AQ_TM_PROCESSES の値です。</p> <p>管理プロセスについては、『Oracle8i 管理者ガイド』の第 4 章「Oracle プロセスの管理」を参照してください。</p> <p>このパラメータは、専用データベース・サーバーの使用を前提としています。他のアプリケーションが同じデータベース・サーバーを使用している場合、それらのリソース要件も考慮する必要があります。</p>

表 4-3（続き）Oracle Message Broker に必要な Oracle8i データベース・サーバーのパラメータ（続き）

パラメータ	説明
SESSIONS = num_sessions	num_sessions ≥ num_processes の場合、num_sessions の値です。
TRANSACTIONS = num_trans	num_trans は、次の式で求められる値以上である必要があります。 num_trans = (2 × maxPrivate + maxShared) × 1.2 maxPrivate は、現在データベース・サーバーを使用しているすべての Oracle Message Broker の属性 maxPrivateSessions の合計です。 maxShared は、現在データベース・サーバーを使用しているすべての Oracle Message Broker の属性 maxSharedSessions の合計です。

データベース・サーバーおよびリスナーの再起動

AQ をサポートしている、Oracle8i データベース・サーバーを実行しているシステムで、データベースを再起動します。Oracle8i データベース・サーバーを起動するには、Oracle8i データベース・サーバーのホーム・ディレクトリに移動します。

- 1. コマンド・プロンプトで SQL\*Plus を起動します。

% sqlplus

- 2. INTERNAL としてログインします。

- 3. データベースを起動します。

SQL> STARTUP

- 4. SQL\*Plus を終了します。

SQL> EXIT

- 5. Net8 リスナーを再起動します。

lsnrctl  
lsnrctl> start  
lsnrctl> quit

**参照：** リスナーの再起動については、『Oracle8i Net8 管理者ガイド』を参照してください。

## Oracle Internet Directory の起動

Oracle Internet Directory と同じ Oracle8i データベース・サーバーを使用して Oracle AQ を実行している場合、OID および Oracle8i データベース・サーバーを起動する必要があります。この項では、データベース・サーバーを起動し、実行した後で、OID を起動する方法について説明します。

1. OID モニターを起動します。

```
% oidmon connect=connect_string start
```

この場合、`connect_string` には、データベース・サーバーに接続するための Net8 接続記述子を指定します。

2. OID インスタンスを起動します。

```
% oidctl connect=connect_string server=oidldapd instance=1 start
```

この場合、`connect_string` には、データベース・サーバーに接続するための Net8 接続記述子を指定します。

**参照:** `oidctl.1` および `oidmon` の詳細は、『Oracle Internet Directory 管理者ガイド』を参照してください。

## AQ の初期化

Oracle Message Broker および Oracle8i データベース・サーバーのインストール後、AQ を初期化します。`$OMB_HOME/admin/plsql` ディレクトリにあるスクリプト `step1.sql`、`step2.sql` および `step3.sql` を使用するには、Oracle8i データベース・サーバーが実行されている必要があります。これらのスクリプトによって、Oracle Message Broker で使用できるように AQ が初期化され、AQ ドライバが AQ キューおよびトピックで動作できるようになります。

1. 次の 3 つの AQ セットアップ・スクリプトを実行します。

```
% sqlplus system/system_password[@service_name] @step1
% sqlplus system/system_password[@service_name] @step2
% sqlplus aq/aq[@service_name] @step3
```

この場合、`system_password` はシステム管理ユーザーのパスワードで、`service_name` はデータベース・サービス名です。

スクリプト `step1.sql` は、Oracle Message Broker PL/SQL サポートが初めてインストールされた場合のみに必要です。これによって、パスワードが `aq` のユーザー `aq` が作成されます。管理サンプル・スクリプト `SetupAQ` では、AQ に対するパスワードの値を入力する必要があります。インストール後、データベース・サーバーの Enterprise Manager またはデータベース・サーバーの `ALTER USER SQL` コマンドを使用して、AQ パスワードを変更できます。

スクリプト `step3.sql` の実行中に、次のエラー・メッセージが 1 つ以上表示される可能性があります。これらのエラー・メッセージを無視できます。

ORA-00942: 表またはビューが存在しません。  
ORA-04043: オブジェクト `xx` は存在しません。  
ORA-04043: オブジェクト `xx` は存在しません。

スクリプト `step2` は、次のことを行います。

- ユーザー `aq` に権限を付与する
- ロール `ombadmin` を作成する
- ロール `ombadmin` に権限を付与する
- ロール `ombuser` を作成する
- ロール `ombuser` に権限を付与する

スクリプト `step3` は、次のことを行います。

- `aq` スキーマにあるパッケージ `ombaq` をインストールする
- `aq` スキーマにあるパッケージ `ombaqadm` をインストールする
- `ombaq` の実行権限を `ombuser` および `ombadmin` に付与する
- `ombaqadm` の実行権限を `ombadmin` に付与する

スクリプト `step3.sql` の実行中に、次のエラー・メッセージが 1 つ以上発生する可能性があります。これらのエラー・メッセージを無視できます。

ORA-00942: 表またはビューが存在しません。  
ORA-04043: オブジェクト `xx` は存在しません。  
ORA-04043: オブジェクト `xx` は存在しません。



## AQ キュー・テーブルの削除

Oracle Message Broker では、AQ に対して作成される表およびユーザーを削除するスクリプトが提供されています。これらの表を削除する必要がある場合、次のコマンドを使用して、データベース・サーバーから、AQ ドライバのサポートを削除します。

```
% sqlplus system/system_password[@service_name] @uninstall
```

この場合、`system_password` はシステム管理ユーザーのパスワードで、`service_name` はデータベース・サービス名です。

スクリプト `uninstall` は、次のことを行います。

- ロール `ombadmin` および `ombuser` を削除します。
- CASCADE オプションを使用して、ユーザー `aq` を削除します。
- AQ スキーマに存在するすべてのキューおよびキュー・テーブルを削除します。

このスクリプトの実行中に、次のタイプのエラーが 1 つ以上発生する可能性があります。

```
ORA-00942: 表またはビューが存在しません。  
ORA-04043: オブジェクト xx は存在しません。
```

---

---

**注意：** AQ スキーマ以外に Oracle Message Broker のオブジェクト・タイプを使用するキュー・テーブルが存在しない場合にのみ、スクリプト `uninstall` は、これらの機能を実行します。AQ スキーマ以外のキュー・テーブルが Oracle Message Broker のオブジェクト・タイプを使用する場合、スクリプト `uninstall` を実行すると、それらのキュー・テーブルが表示され、エラー・メッセージを出力して終了します。

---

---

## MQSeries のインストール・タスク（オプション）

Oracle Message Broker の今回のバージョンでは、IBM MQSeries V5.1 のみをサポートしています。MQSeries のインストールについては、IBM および MQSeries 製品で提供される MQSeries のドキュメントを参照してください。

## TIBCO のインストール・タスク（オプション）

TIBCO ドライバで Oracle Message Broker を使用するには、TIB/Rendezvous リリース 5.x をインストールおよび起動する必要があります。

**参照：** TIB/Rendezvous のインストールおよび管理については、『TIB/Rendezvous Administrator's Guide』を参照してください。  
TIB/Rendezvous については、次の Web サイトを参照してください。

<http://www.rv.tibco.com/>

## Asynchronous Component Invocation のインストール・タスク（オプション）

Oracle Message Broker Asynchronous Component Invocation（ACI）を使用している場合、Oracle Message Broker のクライアント・クラスを Oracle データベース・サーバーにロードし、次の 2 つのコマンドを使用して、EJB を実行するスキーマに権限を付与する必要があります（SCOTT/TIGER をご使用のスキーマおよびパスワードに置き換えてください）。

```
# grant permissions to SCOTT
sqlplus sys/sys_password @$OMB_HOME/admin/plsql/setupaci.sql SCOTT
# Loading OMB client classes
loadjava -r -g SYS -u SCOTT/TIGER ${OMB_HOME}/classes/ombclt.jar
```

Windows NT システムでは、次のようになります。

```
sqlplus sys/sys_password @"%OMB_HOME%\admin\plsql\setupaci.sql SCOTT
loadjava -r -g SYS -u SCOTT/TIGER %OMB_HOME%\classes\ombclt.jar
```

この場合、

`sys_password` は、管理者 SYS のパスワードです。

---

---

**注意：** スキーマで EJB を実行する前に、各スキーマに対してこれらのコマンドを実行してください。

---

---

## OMB インスタンスの作成および Oracle Message Broker の使用

この時点で、Oracle Message Broker、LDAP ディレクトリおよび必要なプロバイダのインストールが完成しています。次に、Oracle Message Broker を構成し起動します。管理ユーティリティで動作する OMB インスタンスの作成、サンプル管理スクリプトおよび Oracle Message Broker の起動については、『Oracle Message Broker Administration Guide』を参照してください。

---

# Oracle プリコンパイラおよび Oracle コール・インタフェースの使用

この章では、次の Oracle プリコンパイラおよび Oracle コール・インタフェースの機能について説明します。

- Oracle プリコンパイラの概要
- Pro\*C/C++ プリコンパイラ
- Pro\*COBOL プリコンパイラ
- Pro\*COBOL の環境変数の設定
- Pro\*FORTRAN プリコンパイラ
- SQL\*Module for Ada の使用
- Oracle コール・インタフェース
- カスタム Make ファイル
- 未定義シンボルの修正
- マルチスレッド・アプリケーション
- XA 機能

# Oracle プリコンパイラの概要

Oracle プリコンパイラとは、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を組み合わせて使用するアプリケーション設計ツールです。Oracle プリコンパイラは、ANSI SQL と互換性があり、Oracle8i 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。Oracle プリコンパイラのデモンストレーションを実行するには、Oracle8i をインストールしておく必要があります。

**参照：** Oracle プリコンパイラおよびインタフェースの概要については、『Oracle8i アプリケーション開発者ガイド』を参照してください。

## プリコンパイラ構成ファイル

プリコンパイラ・システム構成ファイルは、\$ORACLE\_HOME/precomp/admin にあります。表 5-1 に、プリコンパイラ・システム構成ファイルを示します。

表 5-1 システム構成ファイル

製品	構成ファイル
Pro*C/C++ 8.1.7	pccscfg.cfg
Pro*COBOL 8.1.7	pccbcfg.cfg
Pro*COBOL 1.8.52	pcccob.cfg
Pro*FORTRAN 1.8.52	pccfor.cfg
Oracle SQL*Module for Ada 8.1.7	pmscfg.cfg
Object Type Translator 8.1.7	ottcfg.cfg

表 5-2 に、プリコンパイラ README ファイルの位置を示します。README ファイルには、前回のリリースからの変更情報が記載されています。

表 5-2 Oracle 製品の README ファイルの保存場所

製品	README ファイル
Pro*C/C++	\$ORACLE_HOME/precomp/doc/proc2/readme.doc
Pro*COBOL 8.1.7	\$ORACLE_HOME/precomp/doc/procob2/readme.doc
Pro*COBOL 1.8.52	\$ORACLE_HOME/precomp/doc/pro1x/readme.txt
Pro*FORTRAN 1.8.52	\$ORACLE_HOME/precomp/doc/pro1x/readme.txt

## すべてのプリコンパイラに共通の問題

### 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またはサブプログラム名を小文字に変換します。これによって、「No such user exit」エラー・メッセージが表示される場合があります。このエラー・メッセージが表示された場合、オプション・ファイル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グとプリコンパイラに互換性がない場合があります。デバッグで動作するプログラムが、オペレーティング・システムでは動作しない可能性があります。

### ireclen および oreclen の値

ireclen および oreclen パラメータに最大値はありません。

## 静的および動的リンク

プリコンパイラおよび OCI アプリケーションは、静的または動的に Oracle ライブラリにリンクできます。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つの実行ファイル・プログラムにリンクされます。そのため、アプリケーションの実行ファイルのサイズは非常に大きくなります。

動的リンクの場合、実行コードは一部分のみが実行ファイル・プログラムに存在し、他の部分は実行時にアプリケーションと動的にリンクされるライブラリに存在します。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。動的リンクのメリットは、次のとおりです。

- 必要なディスク領域が少なく済む: 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。
- 必要なメイン・メモリーが少なく済む: 同一の動的ライブラリ・イメージをメイン・メモリーに1回のみロードすれば、複数のアプリケーションで共有できます。

## Oracle 共有ライブラリ

Oracle 共有ライブラリは、\$ORACLE\_HOME/lib ディレクトリにある libclntsh.a です。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、デフォルトで Oracle 共有ライブラリが使用されます。

実行ファイルの起動時に次のエラーが発生する場合があります。

```
$ sample1
exec(): 0509-036 Cannot load program ./sample1 because of the following
errors:
0509-022 Cannot load library libclntsh.a [shr.0]
0509-026 System error: A file or directory in the path name does not exist.
```

このエラー・メッセージが表示された場合、次のように入力して、LIBPATH を設定します。

```
$ set LIBPATH $ORACLE_HOME/lib
```

Oracle 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。Oracle 共有ライブラリを再作成する必要がある場合は、Oracle 共有ライブラリを使用している SQL\*Plus、Recovery Manager などのすべてのクライアント・アプリケーションを終了します。Oracle ユーザーとしてログインし、次のように入力します。

```
$ cd $ORACLE_HOME/rdbms/lib
$ make -f ins_rdbms.mk libclntsh.a
```

## Pro\*C/C++ プリコンパイラ

Pro\*C/C++ を使用する前に、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：** オペレーティング・システムの要件については、『Oracle8i for AIX-Based Systems インストレーション・ガイド』を参照してください。

プリコンパイラおよびインタフェース機能の詳細は、『Oracle8i Pro\*C/C++ プリコンパイラ・プログラマーズ・ガイド』を参照してください。

## Pro\*C/C++ デモ・プログラム

デモ・プログラムは、Pro\*C/C++ プリコンパイラの様々な機能を参照するために提供されています。デモ・プログラムには、C、C++ および Object プログラムの 3 種類があります。Object プログラムのデモ・プログラムでは、Oracle8i Object の新機能が表示されます。すべてのデモ・プログラムは、`$ORACLE_HOME/precomp/demo/proc` ディレクトリにあります。プログラムでは、`$ORACLE_HOME/sqlplus/demo/demobld.sql` 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、パスワードは TIGER であると仮定しています。デフォルトでは、すべてのプログラムは、クライアント共有ライブラリ `$ORACLE_HOME/lib/libclntsh.a` に動的にリンクされます。

Make ファイル `demo_proc.mk` は、`$ORACLE_HOME/precomp/demo/proc/` ディレクトリにあり、デモ・プログラムの作成に必要です。

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場合は、次のように入力します。

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

Pro\*C/C++ の C デモ・プログラムを作成するには、次のように入力します。

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

Pro\*C/C++ の C++ デモ・プログラムを作成するには、次のように入力します。

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

Pro\*C/C++ の Object デモ・プログラムを作成するには、次のように入力します。

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

デモ・プログラムによっては、`$ORACLE_HOME/precomp/demo/sql` 内の SQL スクリプトを実行する必要があります。スクリプトを実行しない場合、実行を要求するメッセージが表示されます。このようなデモ・プログラムを作成し、それに対応する SQL スクリプトを実行するには、コマンドラインに `make` マクロ引数 `RUNSQL=run` を追加する必要があります。たとえば、`calldemo` デモ・プログラムを作成し、必要な `$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
```

すべての Object デモ・プログラムを作成し、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
```

**注意：** Make ファイル demo\_proc.mk では、環境変数 ORA\_CLIENT\_LIB が使用されなくなりました。

**参照：** SQL\*Plus を使用したデモ・プログラムの作成については、[3-3 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。

## Pro\*C/C++ ユーザー・プログラム

\$ORACLE\_HOME/precomp/demo/proc/ ディレクトリにある Make ファイル demo\_proc.mk を使用して、プログラムを作成できます。demo\_proc.mk でプログラムをリンク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
```

この場合、次のような意味になります。

- target は、Make ファイルが使用するターゲットです（たとえば、build）。
- objfilen は、プログラムにリンクするオブジェクト・ファイルです。
- EXE は、実行プログラムです。

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

クライアント共有ライブラリに動的にリンクする C ソースの場合、次のように入力します。

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

静的にリンクする C ソースの場合、次のように入力します。

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

クライアント共有ライブラリに動的にリンクする C++ ソースの場合、次のように入力します。

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

静的にリンクする C++ ソースの場合、次のように入力します。

```
$ make -f demo_proc.mk cppbuild_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るオブジェクト・ファイルです。

---

---



# Pro\*COBOL プリコンパイラ

このリリースでは、Pro\*COBOL 8.1.7 または Pro\*COBOL 1.8.52 を使用できます。表 5-3 にこれらのバージョンのネーミング規則を示します。

表 5-3 Pro\*COBOL のネーミング方法の違い

機能	Pro*COBOL 8.1.7	Pro*COBOL 1.8.52
実行ファイル	procob	procob18
デモ・ディレクトリ	procob2	procob
Make ファイル (MicroFocus 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込み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 \$ORACLE\_HOME/lib/libclntsh.a を使用します。動的読込みプログラムは、\$ORACLE\_HOME/bin ディレクトリにある rtsora 実行ファイルを使用します。

# Pro\*COBOL の環境変数の設定

この項では、MicroFocus COBOL コンパイラの環境変数の設定について説明します。

## MicroFocus COBOL コンパイラ

MicroFocus COBOL コンパイラを実行するには、環境変数 COBDIR および LIBPATH を設定する必要があります。

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、次のように入力します。

```
$ set COBDIR /opt/cobol; export COBDIR
```

LIBPATH には、\$COBDIR/coblib ディレクトリを設定します。たとえば、次のように入力して \$COBDIR/coblib を LIBPATH に追加します。

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

LIBPATH に \$COBDIR/coblib が設定されていない場合、プログラムをコンパイルしたときに次のエラー・メッセージが表示されます。

```
System error - errpr data os "rts32
Cannot load library libfhutil.2.0.a[shr.o]
System error: A file or directory in the pathname does not exist
```

**参照：** オペレーティング・システムの要件については、『Oracle8i for AIX-Based Systems インストレーション・ガイド』を参照してください。

プリコンパイラおよびインタフェース機能の詳細は、『Oracle8i Pro\*COBOL プリコンパイラ・プログラマーズ・ガイド』を参照してください。

## Pro\*COBOL デモ・プログラム

### Oracle ランタイム・システム

Oracle では、動的読込み 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・システムが提供されています。動的読込み Pro\*COBOL プログラムを実行するには、MicroFocus の cobrun ランタイム・システムのかわりに rtsora ランタイム・システムを使用する必要があります。cobrun で Pro\*COBOL プログラムを実行すると、次のエラーが発生します。

```
$ 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
```

### デモ・プログラム

デモ・プログラムは、Pro\*COBOL プリコンパイラの様々な機能を参照するために提供されています。すべてのプログラムは、Pro\*COBOL のバージョンによって、

\$ORACLE\_HOME/precomp/demo/procob または \$ORACLE\_HOME/precomp/demo/procob2 にあります。すべてのプログラムでは、\$ORACLE\_HOME/sqlplus/demo/demobld.sql 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、パスワードは TIGER であると仮定しています。デフォルトでは、デモ・プログラムは、クライアント共有ライブラリ \$ORACLE\_HOME/lib/libclntsh.a に動的にリンクされます。

デモンストレーション Make ファイルは、サンプル・プログラムを作成する場合に使用できます。Pro\*COBOL 8.1.7 のデモンストレーション Make ファイルは、\$ORACLE\_HOME/precomp/demo/procob2 ディレクトリにある demo\_procob.mk です。Pro\*COBOL 1.8.52 のデモンストレーション Make ファイルは、\$ORACLE\_HOME/precomp/demo/procob ディレクトリにある demo\_procob18.mk です。

たとえば、Pro\*COBOL 8.1.7 のデモ・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、次のコマンドを使用します。

```
$ cd $ORACLE_HOME/precomp/demo/procob2
$ make -f demo_procob.mk sample1
```

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

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

rtsora で使用する動的読み込みプログラム 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 にある SQL スクリプトを実行する必要があります。スクリプトを実行しない場合、実行を要求するメッセージが表示されます。そのようなデモ・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、コマンドラインに make マクロ引数 RUNSQL=run を追加します。たとえば、sample9 デモ・プログラムを作成し、\$ORACLE\_HOME/precomp/demo/sql ディレクトリにある必要な 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_procob.mk sample9 RUNSQL=run
```

**参照：** SQL\*Plus を使用したデモ・プログラムの作成については、[3-3 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。

デモ・プログラムの詳細は、『Pro\*COBOL プリコンパイラ・プログラマーズ・ガイド』を参照してください。

## Pro\*COBOL ユーザー・プログラム

demo\_procob.mk を使用してユーザー・プログラムを作成できます。ご使用の Pro\*COBOL のバージョンおよび COBOL コンパイラに応じて、適切な Make ファイルを使用してください。デモンストレーション Make ファイルを使用してユーザー・プログラムをリンクするコマンドの構文は、次のとおりです。

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

この場合、次のような意味になります。

- target は、Make ファイルが使用するターゲットです（たとえば、build）。
- objfilen は、プログラムにリンクするオブジェクト・ファイルです。
- EXE は、実行プログラムです。

たとえば、Pro\*COBOL のソース myprog.pco からプログラム myprog を作成する場合、実行ファイルのタイプと共有ライブラリ・リソースの使用の有無に応じて、次のいずれかのコマンドを入力します。

実行ファイルをクライアント共有ライブラリに動的にリンクする場合、次のように入力します。

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

実行ファイルをクライアント共有ライブラリを使用せずに静的にリンクする場合、次のように入力します。

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

---

---

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

---

---

`rtsora` で使用できる動的読み込みモジュールの場合、次のように入力します。

```
$ 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。FORMAT=ANSI (デフォルト) 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 が左端のカラムになります。

## Pro\*FORTRAN プリコンパイラ

Pro\*FORTRAN を使用する前に、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：** ご使用のオペレーティング・システムに必要なバージョンについては、『Oracle8i for AIX-Based Systems インストレーション・ガイド』を参照してください。

## Pro\*FORTRAN デモ・プログラム

デモ・プログラムは、Pro\*FORTRAN プリコンパイラの様々な機能を参照するために提供されています。すべてのプログラムは、`$ORACLE_HOME/precomp/demo/profor` にあります。すべてのデモ・プログラムでは、`$ORACLE_HOME/sqlplus/demo/demobld.sql` 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、パスワードは TIGER であると仮定しています。デフォルトでは、すべてのプログラムは、クライアント共有ライブラリ `$ORACLE_HOME/lib/libclntsh.a` に動的にリンクされます。

Make ファイル `demo_proc.mk` は、`$ORACLE_HOME/precomp/demo/profor` ディレクトリにあり、デモ・プログラムの作成に必要です。

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場合は、次のように入力します。

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

Make ファイル `$ORACLE_HOME/precomp/demo/profor/demo_profor.mk` を使用して、デモ・プログラムを作成できます。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
```

または

```
$ make -f demo_profor.mk build FORS=sample1.pfo EXE=sample1
```

Pro\*FORTRAN のデモ・プログラムを作成するには、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 にある SQL スクリプトを実行する必要があります。スクリプトを実行しない場合、実行を要求するメッセージが表示されます。そのようなデモ・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、コマンドラインに make マクロ引数 RUNSQL=run を追加します。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
```

**参照：** SQL\*Plus を使用したデモ・プログラムの作成については、[3-3 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。

Pro\*FORTRAN プリコンパイラの詳細は、『Oracle プリコンパイラ・ガイド Pro\*FORTRAN サプリメント』を参照してください。

## Pro\*FORTRAN ユーザー・プログラム

demo\_profor.mk を使用してユーザー・プログラムを作成できます。demo\_profor.mk を使用してユーザー・プログラムをリンク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
```

この場合、次のような意味になります。

- target は、Make ファイルが使用するターゲットです（たとえば、build）。
- objfilen は、プログラムにリンクするオブジェクト・ファイルです。
- EXE は、実行プログラムです。

たとえば、Pro\*FORTRAN のソース myprog.pfo からプログラム myprog を作成する場合、実行ファイルのタイプに応じて、次のいずれかのコマンドを入力します。

実行ファイルをクライアント共有ライブラリに動的にリンクする場合、次のように入力します。

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

実行ファイルを静的にリンクする場合、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 の使用

この項では、SQL\*Module for Ada について説明します。

SQL\*Module for Ada を使用する前に、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。ご使用のオペレーティング・システムに必要なバージョンについては、『Oracle8i for AIX-Based 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 デモ・プログラムの使用

SQL\*Module for Ada デモ・プログラムは、SQL\*Module for Ada プリコンパイラの機能を示します。このプログラムは、`$ORACLE_HOME/precomp/demo/modada` ディレクトリにあります。プログラムでは、`$ORACLE_HOME/sqlplus/demo/demobld.sql` 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、パスワードは TIGER であると仮定しています。

プログラムでは、Net8 の接続文字列またはインスタンスの別名 INST1\_ALIAS が定義されていて、適切な表のデータベースに接続できると仮定しています。

**参照：** SQL\*Plus を使用したデモ・プログラムの作成については、[3-3 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。

Make ファイル `$ORACLE_HOME/precomp/demo/modada/demo_modada.mk` を使用して、デモ・プログラムを作成できます。たとえば、`chl_drv` 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、次のいずれかを入力します。

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

または

```
$ make -f demo_modada.mk ada OBJS="chl_mod.ada chl_drv.ada" \  
  EXE=chl_drv MODARAGS=user=modtest/yes
```

デフォルトでは、すべてのプログラムは、クライアント共有ライブラリ `$ORACLE_HOME/lib/libclntsh.a` に動的にリンクされます。

すべての SQL\*Module for Ada デモ・プログラムを作成するには、次のように入力します。

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

`demcalsp` および `demohost` サンプル・プログラムでは、`$ORACLE_HOME/precomp/demo/sql/mktable.sql` で作成した大学のサンプル・データベースが MODTEST スキーマにあることが必要です。MODTEST ユーザーは、SQL スクリプト `$ORACLE_HOME/precomp/demo/sql/grant.sql` を実行して作成できます。MODTEST ユーザーを作成するには、大学のサンプル・データベースを作成し、`make` ターゲットの `makeuser` および `loaddb` を使用してデモ・プログラムを作成します。たとえば、必要な SQL スクリプトを実行して `demohost` プログラムを作成するには、次のように入力します。

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

すべての SQL\*Module for Ada デモ・プログラムおよび大学のサンプル・データベースを作成し、必要な SQL スクリプトを実行して MODTEST ユーザーを作成するには、次のように入力します。

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

SQL スクリプトを手動で実行することもできます。

## SQL\*Module Ada ユーザー・プログラムの作成

Make ファイル `$ORACLE_HOME/precomp/demo/modada/demo_modada.mk` を使用して、プログラムを作成できます。`demo_modada.mk` を使用してユーザー・プログラムをリンク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 MODARAGS=SQL*Module_arguments
```

## Oracle コール・インタフェース

Oracle コール・インタフェース (OCI) を使用する前に、適切なバージョンの Pro\*C/C++ 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。ご使用のオペレーティング・システムに必要なバージョンについては、『Oracle8i for AIX-Based Systems インストレーション・ガイド』を参照してください。

## OCI デモ・プログラム

デモ・プログラムは、OCI の様々な機能を参照するために提供されています。デモ・プログラムには、C および C++ の 2 種類があります。すべてのデモ・プログラムは、`$ORACLE_HOME/rdbms/demo` にあります。デフォルトでは、すべてのプログラムは、クライアント共有ライブラリ `$ORACLE_HOME/lib/libclntsh.a` 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` で作成したデモンストレーション表が SCOTT にあり、パスワードは TIGER であると仮定しています。

`$ORACLE_HOME/rdbms/demo` ディレクトリにある Make ファイル `demo_rdbms.mk` を使用して、デモ・プログラムを作成できます。たとえば、`cdemo1` をコンパイルおよびリンクするには、次のように入力します。

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

OCI の C デモ・プログラムを作成するには、次のように入力します。

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

OCI の C++ デモ・プログラムを作成するには、次のように入力します。

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

デモ・プログラムによっては、`$ORACLE_HOME/precomp/demo/sql` にある SQL スクリプトを実行する必要があります。スクリプトを実行しないと、スクリプトの実行を要求するメッセージが表示されます。ほとんどの場合、SQL スクリプト名はプログラム名と同じで、拡張子は `.sql` です。

**参照：** SQL\*Plus を使用したデモ・プログラムの作成については、[3-3 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。

デモ・プログラムの詳細は『Oracle8i コール・インタフェース・プログラマーズ・ガイド』を、各プログラムの詳細はプログラム・ソースを参照してください。

## OCI ユーザー・プログラム

`$ORACLE_HOME/rdbms/demo` ディレクトリにある Make ファイル `demo_rdbms.mk` を使用して、プログラムを作成できます。`demo_rdbms.mk` を使用してユーザー・プログラムをリンク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
```

この場合、次のような意味になります。

- `target` は、Make ファイルが使用するターゲットです（たとえば、`build`）。
- `objfilen` は、プログラムにリンクするオブジェクト・ファイルです。
- `EXE` は、実行プログラムです。

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、実行ファイルのタイプに応じて、次のいずれかのコマンドを入力します。

C ソースからクライアント共有ライブラリに動的にリンクするプログラム `myprog` を作成する場合、次のように入力します。

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

C ソースから静的にリンクするプログラム `myprog` を作成する場合、次のように入力します。

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

C++ ソースからクライアント共有ライブラリに動的にリンクするプログラム `myprog` を作成する場合、次のように入力します。

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

C++ ソースから静的にリンクするプログラム `myprog` を作成する場合、次のように入力します。

```
$ 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るオブジェクト・ファイルです。

---



## カスタム Make ファイル

オラクル社が提供している Make ファイル `demo_product.mk` を使用してユーザー・プログラムをリンクすることをお勧めします。この Make ファイルを修正する必要がある場合、またはカスタム Make ファイルを使用する場合には、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・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。ユーザー・ライブラリは、Oracle ライブラリの上に置かないでください。
- `nm` 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、AIX 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、すでにテスト済で、AIX の `make` ユーティリティでサポートされています。
- 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

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共通するエラーに、未定義シンボルによって、次のようなエラー・メッセージが表示されます。

```
$ make -f demo_proc.mk sample1
ld: 0711-317 ERROR: Undefined symbol: .sqlcex
ld: 0711-345 Use the -bloadmap or -bnoquiet option to obtain more information.
```

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。

次に、シンボル `sqlcex` の場所を調べる `symfind` ユーティリティの出力例を示します。

[illegible]

## 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。

**参照:** スレッド・サポートの詳細は、『Oracle8i for AIX-Based Systems インストール・ガイド』を参照してください。

## シグナル・ハンドラの方法

シグナルは、Oracle8i の 2 タスク通信で使用できます。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 5-4 に、Oracle8i の 2 タスク通信で使用するシグナルを示します。

**表 5-4 2 タスク通信で使用するシグナル**

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() を使用します。SIGCLD は、Oracle プロセスではなく、ユーザー・プロセスが受け取ります。
SIGCONT	パイプ 2 タスク・ドライバが、アウトオブバンド・ブレイクをユーザー・プロセスから Oracle プロセスへ送信する場合に、SIGCONT を使用します。

表 5-4 2 タスク通信で使用するシグナル (続き)

シグナル	説明
SIGINT	2 タスク・ドライバが、ユーザーの割り込み要求を検出する場合に、SIGINT を使用します。SIGCLD は、Oracle プロセスではなく、ユーザー・プロセスが受け取ります。
SIGIO	Net8 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ために、SIGIO を使用します。
SIGPIPE	パイプ・ドライバが、通信チャネルでファイルの終わりを検出するために、SIGPIPE を使用します。パイプへの書き込み時に、読取りプロセスが存在していなければ、SIGPIPE シグナルが書き込みプロセスに送信されます。SIGPIPE は、Oracle プロセスおよびユーザー・プロセスが受け取ります。SIGCLD は SIGPIPE と似ていますが、ユーザー・プロセスに適用され、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 プロセスに割り込みシグナルを送る場合に SIGTERM を使用します。ユーザーが割り込みキー ([Ctrl]+[c]) を押すと、このシグナルが送られます。SIGTERM は、ユーザー・プロセスではなく、Oracle プロセスが受け取ります。
SIGURG	Net8 TCP/IP ドライバが、アウトオブバンド・ブレイクをユーザー・プロセスから Oracle プロセスへ送信する場合に、SIGURG を使用します。

シグナルは、Pro\*C またはその他のプリコンパイラ・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 1 つのシグナル・ハンドラをインストールできます。osnsui ( ) ルーチン呼び出して設定すると、SIGINT 用のシグナル・ハンドラを複数保持できます。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。その他のシグナルにも、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。

次に、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。

```
/* 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.
*/

```

次に、アプリケーション・プログラムで osnsui() および osncui() を使用する場合のテンプレートを示します。

```

/*
** 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 コール・インタフェースには、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場合、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 の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。

Oracle8i では、Oracle7 リリース 7.1.6 の XA コールをサポートしていません（Oracle7 リリース 7.3 の XA コールはサポートしています）。そのため、Oracle7 リリース 7.1.6 の XA コールを使用する TP モニター XA アプリケーションは、Oracle8i の XA ライブラリにリンクする必要があります。Oracle8i の XA コールは、共有ライブラリ `$ORACLE_HOME/lib/libclntsh.a` および静的ライブラリ `$ORACLE_HOME/lib/libclient8.a` の両方で定義されています。



---

## Net8 の構成

この章では、AIX-Based Systems での、Net8 の次の機能について説明します。

- [主な Net8 製品および特徴](#)
- [Net8 プロトコルのサポート](#)
- [BEQ プロトコル](#)
- [IPC プロトコル](#)
- [RAW プロトコル](#)
- [TCP/IP プロトコル](#)
- [APPC/LU6.2 プロトコル](#)
- [Oracle Enterprise Manager \(OEM\)](#)
- [Oracle SNMP での Oracle Intelligent Agent](#)
- [Oracle Advanced Security](#)
- [参照ドキュメント](#)

## 主な Net8 製品および特徴

サンプル・ファイルは、『Oracle8i Net8 管理者ガイド』を参照してください。

## Net8 ファイルおよびユーティリティ

### Net8 の構成ファイルの保存場所

AIX では、Net8 構成ファイルのデフォルト・ディレクトリは、`/etc` です。

Net8 は、グローバル・ファイルを次の順に検索します。

1. 環境変数 `TNS_ADMIN` に設定されたディレクトリ（設定されている場合）
2. `/etc` ディレクトリ
3. `$ORACLE_HOME/network/admin`

次のように入力して、すべてのネットワーク・ユーザーの起動ファイルに環境変数 `TNS_ADMIN` を設定します。

Bourne または Korn シェルの場合、次のように設定します。

```
%TNS_ADMIN=directory_path
%export TNS_ADMIN
```

C シェルの場合、次のように設定します。

```
% setenv TNS_ADMIN directory_name
```

システム・レベルの構成ファイルごとに、ローカルのプライベート構成ファイル（ユーザーのホーム・ディレクトリに保存されている）を保持できます。プライベート・ファイルの設定は、システム・レベル・ファイルの設定を上書きします。`sqlnet.ora` のプライベート構成ファイルは、`$HOME/.sqlnet.ora` です。`tnsnames.ora` のプライベート構成ファイルは、`$HOME/.tnsnames.ora` です。これらのファイルの構文は、対応するシステム・ファイルの構文と同じです。

### 構成ファイルの例

`cman.ora`、`listener.ora`、`names.ora`、`sqlnet.ora` および `tnsnames.ora` 構成ファイルの例は、`$ORACLE_HOME/network/admin/samples` に保存されています。



## アダプタ・ユーティリティ

Net8 では、様々なネットワーク・プロトコルおよびネーミング方法をサポートしています。アダプタと呼ばれるソフトウェア・モジュールは、個々の実行ファイルにリンクされ、ネットワーク・プロトコルと Net8 間のインタフェースを提供します。インストールされている Net8 アダプタを表示するには、次のように入力します。

```
% adapters
```

指定した実行ファイルにリンクされているアダプタを表示するには、次のように入力します。

```
% adapters executable
```

たとえば、Oracle 実行ファイルにリンクされているアダプタを表示するには、次のように入力します。

```
% adapters oracle
```

Net8 プロトコルが、次のように表示されます。

```
Net8 Protocol Adapters linked with oracle are:
```

```
BEQ Protocol Adapter  
IPC Protocol Adapter  
TCP/IP Protocol Adapter  
RAW Protocol Adapter
```

```
Net8 Naming Adapters linked with oracle are:
```

```
Oracle TNS Naming Adapter  
Oracle Naming Adapter
```

```
Oracle Advanced Security/Network Security products linked with oracle are:
```

## Oracle Connection Manager

Oracle Connection Manager (OCM) は、クライアント接続要求が次のホップに送られるか、または直接データベース・サーバーに送られる際のルーターです。Connection Manager を介して接続要求を送信するクライアントは、Connection Manager で構成された接続集中化、Net8 アクセス制御または複数プロトコル・サポート機能を利用できます。

Oracle Connection Manager はクライアントから受信した要求をリスニングして、宛先サービスへの接続要求を開始します。Oracle Connection Manager は、Oracle Connection ゲートウェイ・プロセスおよび Oracle Connection Manager 管理プロセスを使用して、これらのタスクを実行します。

OCM には、TCP/IP 環境で指定されたサーバーへのクライアント・アクセスを制御するために使用できる機能も含まれています。あるフィルタ・ルールを設定することによって、次の基準に基づいたサーバーへの特定のクライアント・アクセスを可能にしたり、制限することができます。

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

### Net8 のファイアウォール・プロキシ

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

**参照：** Oracle Connection Manager の詳細は、『Oracle8i Net8 管理者ガイド』を参照してください。

## サーバー・モデル

クライアント / サーバーの Net8 接続は、次の 2 つのモデルを使用して構築されます。

- マルチスレッド・サーバー・モデル：リスナーは、ディスパッチャに接続要求を渡すか、またはディスパッチャを介して、接続するクライアントをリダイレクトします。

マルチスレッド・サーバー (MTS) 構成では、クライアントのユーザー・プロセスはディスパッチャに接続します。ディスパッチャは、同時に複数のクライアント接続をサポートできます。各クライアント接続は、仮想サーキットにバインドされます。仮想サーキットは、ディスパッチャがクライアントのデータベース接続の要求および応答に使用する共有メモリーの一部です。この方法によって、サーバー・プロセスの小さなプールで大量のクライアントを処理できるようになります。専用サーバー・モデルに比べて、MTS モデルの大きなメリットは、システム・リソースが少なくすむため、ユーザー数の増加をサポートできることです。

- 専用サーバー・モデル：リスナーは専用サーバーを起動し、専用サーバーに接続要求を渡すか、または専用サーバーにクライアントをリダイレクトします。

専用サーバー・モデルは、クライアントごとに 1 つの単一のサーバー・プロセスがあります。クライアントが専用サーバーに接続するには、リスナーとインスタンスが同じマシンで実行されている必要があります。専用サーバー・プロセスでは、MTS より多くのメモリーが必要です。

**参照：** マルチスレッド・サーバーの詳細は、『Oracle8i Net8 管理者ガイド』を参照してください。

## Oracle Names

Oracle Names は、ネットワーク上のすべてのデータベース・サービスの名前およびアドレスを格納する Oracle 固有のソフトウェアです。サーバーに接続するクライアントは、接続要求を Oracle Names Server に送ります。Oracle Names Server は、ネットワーク・アドレスに名前を変換し、クライアントに情報を戻します。

電話番号を格納するディレクトリ・アシスタンスを使用するコール側と同様、Oracle Names を使用するために構成されたクライアントは、それぞれの接続要求を Oracle Names Server に示します。Oracle Names Server は、クライアントによって提供されるサービス名をネットワーク・アドレスに変換しようとします。Oracle Names Server がネットワーク・アドレスを検索し、クライアントに情報を戻します。クライアントはそのアドレスを使用してサービスに接続します。

**参照：** Oracle Names の詳細は、『Oracle8i Net8 管理者ガイド』を参照してください。

## Net8 Assistant

Oracle Java Runtime Environment が、Net8 Assistant とともにインストールされます（`$ORACLE_HOME/bin/net8asst`）。Net8 Assistant のコマンド・スクリプトの実行時に、システムに他の Java がインストールされているかどうかに関係なく、JRE 1.1.8.5 の提供する Java コマンド・スクリプトが明示的に起動されます。

**参照：** Net8 Assistant の詳細は、『Oracle8i Net8 管理者ガイド』を参照してください。

# Net8 プロトコルのサポート

AIX がサポートする、Net8 リリース 8.1.7 用のプロトコルは、BEQ プロトコル、IPC プロトコル、RAW プロトコル、TCP/IP プロトコルおよび APPC/LU6.2 プロトコルです。

TCP/IP または APPC/LU6.2 プロトコルをインストールする前に、適切なオペレーティング・システム・ソフトウェアがインストールおよび構成されていることを確認してください。要件の詳細は、『Oracle8i for AIX-Based Systems インストレーション・ガイド』を参照してください。BEQ および IPC プロトコルに固有のオペレーティング・システム要件はありません。

## ADDRESS の指定

IPC、TCP/IP および APPC/LU6.2 プロトコルには、それぞれのプロトコル固有の ADDRESS 指定があります。ADDRESS 指定は、Net8 構成ファイルおよび `initSID.ora` ファイル中のデータベースの初期化パラメータ `MTS_LISTENER_ADDRESS` を設定する場合に必要です。詳細は、この章に記載している各プロトコルの項を参照してください。

表 6-1 に、各プロトコルの ADDRESS 指定を示します。

表 6-1 ADDRESS 指定

サポート・プロトコル	ADDRESS 指定
BEQ	(ADDRESS= (PROTOCOL = BEQ) (PROGRAM = ORACLE_HOME/bin/oracle) (ARGV0 = oracleORACLE_SID) (ARGS = ' (DESCRIPTION=(LOCAL=YES) (ADDRESS=(PROTOCOL=BEQ)) ) ') (ENVS = 'ORACLE_HOME=ORACLE_HOME,ORACLE_SID=ORACLE_SID') )
IPC	(ADDRESS= (PROTOCOL=IPC) (KEY=key) )
RAW	なし
TCP/IP	(ADDRESS= (PROTOCOL=TCP) (HOST=hostname) (PORT=port_id) )

表 6-1 ADDRESS 指定 (続き)

サポート・プロトコル	ADDRESS 指定
APPC/LU6.2	(ADDRESS= (PROTOCOL=LU62) (TP_NAME= <i>transaction_program_name</i> ) (SIDE_INFO_PROFILE= <i>side_info_profile</i> ) (MODE= <i>mode_name</i> ) (PLU= <i>partner_lu_name</i> ) )

BEQ プロトコル

BEQ プロトコルは、通信メカニズムであり、また、プロセス起動メカニズムでもあります。クライアントおよびサーバーは、同一のマシン上に存在する必要があります。ネット・サービス名を指定（コマンドラインまたはログイン画面でユーザーが直接指定するか、または TWO\_TASK などの環境変数で間接的に指定する）しなかった場合、BEQ プロトコルが使用されます。その場合、専用サーバーが常に使用され、マルチスレッド・サーバーは使用されません。この専用サーバーは、BEQ プロトコルによって自動的に起動された後、サーバー・プロセスが起動されて既存の SGA に割り当てられるのを待ちます。サーバー・プロセスが正常に起動されると、BEQ プロトコルで、UNIX パイプを介してプロセス間通信ができるようになります。

BEQ プロトコルが動作するためにリスナーは不要です。これは、プロトコルがクライアント・ツールにリンクされていて、外部から操作しなくてもそのサーバー・プロセスを直接起動するためです。ただし、BEQ プロトコルは、クライアント・プログラムおよび Oracle8i が同一のマシンに常駐する場合にのみ使用できます。BEQ プロトコルは常にインストールされ、すべてのクライアント・ツールおよび Oracle8i にリンクされています。

BEQ ADDRESS の指定

ADDRESS のキーワードとその値の組合せが BEQ プロトコルの接続パラメータです。通常、ADDRESS の指定は、接続記述子、構成ファイルなどを構成している内容の一部です。パラメータの入力順に決まりはありません。

```
(ADDRESS =  
    (PROTOCOL = BEQ)  
    (PROGRAM = ORACLE_HOME/bin/oracle)  
    (ARGV0 = oracleORACLE_SID)  
    (ARGS = ' (DESCRIPTION=(LOCAL=YES) (ADDRESS=(PROTOCOL=BEQ))) '  
    (ENVS = 'ORACLE_HOME=ORACLE_HOME,ORACLE_SID=ORACLE_SID')  
)
```

表 6-2 に、BEQ プロトコルの接続パラメータの構文を示します。

表 6-2 BEQ プロトコルの接続パラメータの構文

PROTOCOL	使用するプロトコルを指定します。 値は beq で、大文字または小文字のどちらかで指定します。
PROGRAM	Oracle 実行ファイルのフルパス名を指定します。
ARGV0	ps コマンドで表示されるプロセスの名前を指定します。推奨値は oracleORACLE_SID です。
ARGS	' (DESCRIPTION= (LOCAL=YES) (ADDRESS= (PROTOCOL=BEQ) ) ) '
ENVS	環境を指定します。ORACLE_HOME には、接続先データベースの ORACLE_HOME ディレクトリをフルパスで指定します。ORACLE_SID には、接続先データベースのシステム識別子を指定します。

次に、BEQ ADDRESS の例を示します。

```
(ADDRESS =
  (PROTOCOL = BEQ)
  (PROGRAM = /u01/app/oracle/product/8.1.7/bin/oracle)
  (ARGV0 = oracleV817)
  (ARGS = ' (DESCRIPTION= (LOCAL=YES) (ADDRESS= (PROTOCOL=BEQ) ) ) ' )
  (ENVS = 'ORACLE_HOME=/u01/app/oracle/product/8.1.7,ORACLE_SID=V817' )
)
```

# IPC プロトコル

IPC プロトコルは、クライアント・プログラムおよび Oracle8i が同一マシンに常駐する場合にのみ使用できるという点が、BEQ プロトコルと似ています。IPC プロトコルと BEQ プロトコルの違いは、IPC プロトコルが専用サーバーおよびマルチスレッド・サーバーで使用できることです。IPC プロトコルが動作するには、リスナーが必要です。IPC プロトコルは常にインストールされ、すべてのクライアント・ツールおよび Oracle8i にリンクされています。

IPC プロトコルでは、Oracle7 リリース 7.1 以降、UNIX での UNIX Domain Socket ファイルの保存場所が変わりました。そのため、Oracle7 リリース 7.1 を Oracle8i と同じマシンにインストールし、2つのインスタンス間で IPC 接続しようとするとう失敗します。この問題を解決するには、IPC ファイルが以前に保存されていたディレクトリ (/tmp/o)、および現在保存されているディレクトリ (/tmp/.oracle) 間にシンボリック・リンクを作成します。

## IPC ADDRESS の指定

ADDRESS のキーワードとその値の組合せが IPC プロトコルの接続パラメータです。通常、ADDRESS の指定は、接続記述子、構成ファイルなどを構成している内容の一部です。パラメータの入力順に決まりはありません。

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

表 6-3 に、IPC プロトコルの接続パラメータの構文を示します。

表 6-3 IPC プロトコルの接続パラメータの構文

PROTOCOL	使用する IPC プロトコルを指定します。 値は ipc で、大文字または小文字のどちらかで指定します。
KEY	データベースのサービス名またはデータベースの識別子 (ORACLE_SID) を指定します。

次に、IPC ADDRESS の例を示します。

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

## RAW プロトコル

データがクライアント / サーバー間で転送されると、Net8 は、そのヘッダー情報を各ネットワーク・パケットに追加します。Raw Transport 機能によって、ネットワークを介する各パケットのヘッダー情報を最小限に抑えることができます。

接続後、2 種類の情報（データおよびブレイク・ハンドリング）がネットワーク上に送信されます。接続パケットには、適切に接続するための Net8 のヘッダー情報が必要です。ただし、接続後、すべてのデータ・パケットは Net8 のヘッダー情報から削除され、Net8 のネットワークおよびプロトコル・レイヤーを通過して、直接オペレーティング・システムに渡されます。データが流れるプロトコル・スタック・レイヤーの数が減り、ネットワーク上に送信されるデータのバイト数が減るため、接続のパフォーマンスが向上します。

この機能は、ユーザーが意識しなくても、必要な場合にいつでも有効になります。つまり、ヘッダー情報を送信しなくてもよい場合、ヘッダーは削除されます。暗号化および認証では、各情報のパケットと一緒に特定の情報を送信する必要があります。そのため、Raw Transport は有効になりません。

この機能を使用する場合、特別な構成はありません。Net8 では、ユーザーが意識しなくても、Raw Transport モードに切り替える必要があるかどうか判断され、必要な場合には切り替えられます。

## TCP/IP プロトコル

リスナー・ポートを定義するネットワーク上の各ノードの `/etc/services` ファイルで、リスナー用のポートを予約してください。ポートは、通常 1521 です。次に例を示します。

```
listener      1521/tcp
```

この場合、`listener` は `listener.ora` で定義されているリスナーの名前を示します。複数のリスナーを起動する場合は、複数のポートを予約する必要があります。

ADDRESS のキーワードとその値の組合せが TCP/IP プロトコルの接続パラメータです。通常、ADDRESS の指定は、接続記述子、構成ファイルなどを構成している内容の一部です。パラメータの入力順に決まりはありません。

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

表 6-4 に、TCP/IP プロトコルの接続パラメータの構文を示します。

表 6-4 TCP/IP プロトコルの接続パラメータの構文

PROTOCOL	使用するプロトコルを指定します。 値は大文字でも小文字でもかまいません。デフォルトは <code>tcp</code> です。
HOST	ホスト名またはホスト IP アドレスを指定します。
PORT	TCP/IP のポートを指定します。 <code>/etc/services</code> ファイルで指定されている番号または名前を指定します。推奨値は 1521 です。

次に、MADRID ホストにクライアントを指定する TCP/IP ADDRESS の例を示します。

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

最後のフィールドは、`PORT=listener` というように名前を指定することもできます。

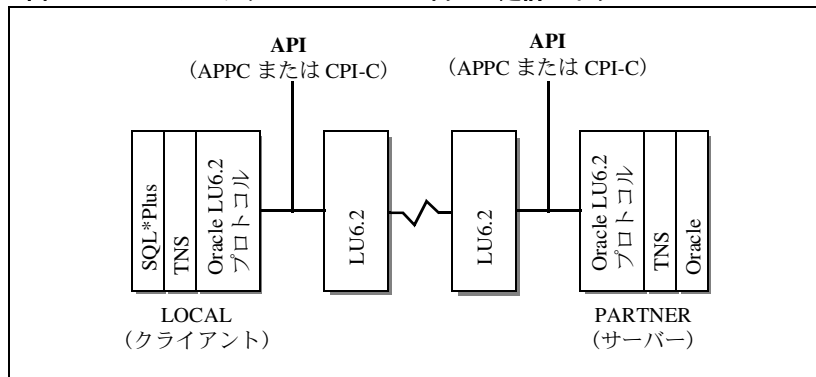


## APPC/LU6.2 プロトコル

Oracle APPC/LU6.2 プロトコルは、Oracle プログラム間の通信に LU6.2 サービスを使用するネットワークで使用できます。たとえば、APPC/LU6.2 を使用することで TNS アプリケーションは、標準インターフェースとして API を使用できるようになります。

図 6-1 に、LU6.2 通信サービスおよび Oracle APPC/LU6.2 プロトコルを使用した Oracle プログラム間での通信レイヤーを示します。

図 6-1 Oracle プログラムおよび LU6.2 間での通信レイヤー



## APPC/LU6.2 の ADDRESS 指定

APPC/LU6.2 プロトコル・パラメータは、各ノードの接続記述子で指定します。各接続記述子には、[ キーワード ]=[ 値 ]という形の組合せがいくつか含まれています。APPC/LU6.2 固有のキーワードは、接続記述子内で任意の順に入力できます。

```
(ADDRESS=
  (PROTOCOL=LU62)
  (TP_NAME=tpname)
  (LU_NAME=luname)
  (MODE=modename)
  (PLU=partner_lu_name)
)
```

表 6-5 に、APPC/LU6.2 プロトコル接続の構文を示します。

表 6-5 APPC/LU6.2 プロトコルの構文

PROTOCOL	使用するプロトコルを指定します。値は大文字でも小文字でもかまいません。APPC/LU6.2 の場合、値は lu62 です。
TP_NAME	ターゲットで実行するトランザクション・プログラム名、または着信接続要求をリスニングするときに使用するトランザクション・プログラム名を指定します。この値は必須です。
SIDE_INFO_PROFILE	このパラメータで、リモート TP との通信を確立する際に使用する特性を指定します。使用する AIX SNA 構成内の対応するプロファイルの名前を指定してください。
MODE	論理デバイス間のセッションの特性を定義します。モードは、パートナー LU およびトランザクション・プログラム名とともに、ALLOCATE セグメントで指定します。modename は、ローカル LU およびパートナー LU の両方に共通である必要があります。この値は必須です。
PLU	パートナ LU の名前を指定します。

## Oracle Enterprise Manager (OEM)

### エージェント・サービスの展開および自動構成

Simple Network Management Protocol (SNMP) システムで統合しない限り、Oracle Intelligent Agent での構成は必要ありません（「[Oracle SNMP での Oracle Intelligent Agent](#)」を参照）。

**参照：** Oracle Names および Net8 Assistant の詳細は、『Oracle Enterprise Manager 構成ガイド』を参照してください。

詳細は、『Oracle8i Net8 管理者ガイド』を参照してください。

### Tcl スクリプトのデバッグ

oratclsh 実行ファイルは、tcl スクリプトをデバックする場合に必要です。oratclsh を実行する前に、環境変数 TCL\_LIBRARY に、\$ORACLE\_HOME/network/agent/tcl を設定してください。

**参照：** 詳細は、『Oracle Enterprise Manager 開発者ガイド』を参照してください。

## Oracle SNMP での Oracle Intelligent Agent

Oracle Intelligent Agent は、Simple Network Management Protocol (SNMP) を必要としませんが、Intelligent Agent を起動する前に Oracle SNMP 用に構成することができます。

### マスター・エージェントの構成

/etc/snmpd.conf ファイルに、次の行を追加します。

```
smux 0.0 "" ip_address
```

この場合、*ip\_address* は、Oracle サブエージェントを実行するホストの *ip\_address* です。

### SNMP コンポーネントの起動

次の手順を実行して、SNMP コンポーネントを起動します。

1. `snmpd` が実行中でないことを確認します。

```
$ ps -aef | grep snmpd
```

実行中の場合、`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. `root` ユーザーで、`startsrc` スクリプトを実行し、ネイティブな AIX SNMP エージェントを起動します。

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

このコマンドにより、`snmpd` デーモンが起動し、`/tmp/snmpd.log` に情報が記録されます。

3. 次のように入力して、SNMP コンポーネントが実行されていることを確認します。

```
# ps -aef | grep snmpd
```

### 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

## .bak ファイル

Oracle Advanced Security のインストール時に、3つの .bak ファイル（naeet.o.bak、naect.o.bak、naedhs.o.bak）が作成されます。これらのファイルは、\$ORACLE\_HOME/lib に保存されます。これらのファイルは、Oracle Advanced Security の削除（デインストール）中の再リンクに必要なため、削除しないでください。

## セキュリティおよびシングル・サインオン

セキュリティおよびシングル・サインオンの構成の詳細は、『Oracle Advanced Security 管理者ガイド』を参照してください。

## DCE 統合

DCE 統合の構成の詳細は、『Oracle Advanced Security 管理者ガイド』を参照してください。

# 参照ドキュメント

[表 6-6](#) に、各バンドル製品の README ファイルの保存場所を示します。README ファイルには、前回のリリースからの変更情報が記載されています。

表 6-6 Oracle 製品の README ファイルの保存場所

製品	README ファイル
Net8	\$ORACLE_HOME/network/doc/README.Net8
Advanced Security	\$ORACLE_HOME/network/doc/README.ASO
Oracle Intelligent Agent	\$ORACLE_HOME/network/doc/README.oemagent

---

## Oracle データ・オプション・デモの実行

この章では、このリリースに含まれるデモに関する、AIX 固有の情報を説明します。このリリースで使用可能な、Oracle オプション・デモの AIX に依存しない情報の詳細は、[「Oracle8i interMedia」](#) に示されているドキュメントを参照してください。

- [Oracle8i interMedia](#)
- [Oracle8i Time Series デモ](#)
- [Oracle8i Visual Information Retrieval](#)
- [Oracle8i Spatial](#)

## Oracle8i *interMedia*

Oracle8i *interMedia* には、次のコンポーネントがあります。

- Text
- Audio、Video および Image
- Locator

### Text

Oracle8i には、Text のデモンストレーション（デモ）はありません。ただし、*interMedia* Text には、コード・サンプルが含まれています。次の URL で、ご使用のブラウザを示してください。

`$ORACLE_HOME/ctx/sample/api/index.html`

**参照：** 詳細は、『Oracle8i *interMedia* Text リファレンス』および『Oracle8i *interMedia* Text 移行ガイド』を参照してください。

### Audio、Video および Image

Oracle8i *interMedia* には、次に示すディレクトリにスクリプトおよびサンプル・プログラムが多数あります。

`$ORACLE_HOME/ord/aud/demo/`  
`$ORACLE_HOME/ord/img/demo/`  
`$ORACLE_HOME/ord/vid/demo/`

#### サンプル Audio スクリプト

Audio スクリプトは、次のファイルで構成されています。

- `auddemo.sql` - 次に示す Audio オブジェクトの機能を含む Audio デモンストレーションです。
  - *interMedia* オブジェクトをチェックします。
  - デモ内の Audio でサンプル表を作成します。
  - Audio 表に NULL 行を挿入します。
  - アウトラインをチェックします。
  - すべての Audio 属性を直接チェックします。
  - すべての Audio 属性をメソッドを呼び出すことによってチェックします。
  - 2つのファイル (`fplugins.sql` および `fpluginb.sql`) を使用してユーザー独自の形式のプラグインをインストールします。

- `fplugins.sql` - サポートする任意の形式のプラグインを書き込むためのガイドラインとして使用するデモ形式のプラグイン仕様です。
- `fpluginb.sql` - サポートする任意の形式のプラグインを書き込むためのガイドラインとして使用するデモ形式のプラグイン本体です。

---

**参照：** この SQL デモを実行する要件および手順の詳細は、  
`$ORACLE_HOME/ord/aud/demo` ディレクトリの `README.txt` ファイル  
 を参照してください。

---

## Image の変更または Image インストールのテスト用のサンプル・プログラム

一度 Oracle8i interMedia Image をインストールすると、Oracle8i interMedia Image デモ・プログラムを実行できます。このプログラムは、インストールが正常に完了したかを確認するためのテストにも使用できます。

この項では、interMedia Image デモを作成および実行するために必要な手順を示します。

interMedia Image デモは、`$ORACLE_HOME/ord/img/demo` に保存されています。この場合、`$ORACLE_HOME` は `ORACLE_HOME` ディレクトリです。

## デモンストレーション・インストールの手順

1. Oracle8i interMedia Image デモは、SCOTT/TIGER データベース・ユーザーを使用します。このユーザーが存在しない場合は、作成する必要があります。次のように入力します。

```
% svrmgrl
SVRMGR> connect internal;
SVRMGR> create user SCOTT identified by tiger;
SVRMGR> grant connect,resource to SCOTT;
```

2. `$ORACLE_HOME` が `ORACLE_HOME` ディレクトリを示すように Image デモ・ディレクトリを作成します。

```
% svrmgrl
SVRMGR> connect internal;
SVRMGR> create or replace directory imgdemodir as '$ORACLE_HOME/ord/img/ demo';
```

3. 次のように入力して、ディレクトリの権限を PUBLIC に与えます。

```
SVRMGR> grant read on directory imgdemodir to public with grant option;
```

4. 必要ならば、imgdemo プログラムを作成します。

```
% cd $ORACLE_HOME/ord/img/demo
% make -f demo_ordimg.mk imgdemo
```

## デモの実行

imgdemo ファイルは、Oracle8i *interMedia Image* をプログラム内からどのようにして使用できるかを示すサンプル・プログラムです。デモは C 言語で書かれており、OCI (Oracle Call Interface) を使用して、データベースにアクセスし、Oracle8i *interMedia Image* を実行します。

プログラムは、デモ・ディレクトリ内にビットマップ (BMP) ・イメージがある `imgdemo.dat` で動作します。オプションとして、ファイルがデモと同じディレクトリに常駐する場合は、コマンドラインでイメージ・ファイル名を指定できます。いずれの場合も、一度イメージが Oracle8i *interMedia Image* で処理されると、結果のイメージは `imgdemo.out` ファイルに書き込まれて、指定した共通のレンダリング・ツールで表示されます。

デモが実行されると、デフォルト・データベースの SCOTT/TIGER スキーマ内の IMGDEMOTAB という名前の表が削除および再作成されます。この表は、デモ・データを保持するために使用されます。一度表が作成されると、イメージ・ファイルへの参照が表に挿入されます。次に、データが表にロードされ、ORDImage の `processCopy()` メソッドを使用して JFIF に変換されます。

イメージ・プロパティは、`setProperties()` メソッドを使用してデータベース内に抽出されます。UPDATE コマンドは、`setProperties()` の起動後に発行されます。これは、`setProperties()` の起動がタイプ属性のローカル・コピーを更新するのみなので必須となります。

次に、Oracle8i *interMedia Image* の `process()` メソッドが、データベース内でイメージをカットおよびスケール変更するために使用されます。この次に、変更をコミットする更新が行われます。プログラムは、ピクセル位置 (100,100) から始めて、100 ピクセルの幅、100 ピクセルの高さでイメージの一部分をカットします。このサブイメージは、元のサイズの 2 倍の大きさになります。結果のイメージは、現行のディレクトリにある `imgdemo.out` という名前のファイルへ書き込まれます。

### 例 7-1 コマンドラインでのデモの実行

コマンドラインで `imgdemo` を入力することによって、デモを実行できます。オプションとして、デモ内で異なるイメージを使用できます。そうするには、最初にデモがあるディレクトリにファイルをコピーしてから、`imgdemo` の引数としてコマンドラインでそのファイル名を指定します。次のように使用します。

```
$ imgdemo optional-image-filename
```

デモは、その進行状況を示す多くのメッセージを表示します。それとともに、何かが正しく設定されていない場合に発生するエラーも表示します。次のようなメッセージが表示されます。

```
Dropping table IMGDEMOTAB...
Creating and populating table IMGDEMOTAB...
Loading data into cartridge...
Modifying image characteristics...
```



```
Writing image to file imgdemo.out...
Disconnecting from database...
Logged off and detached from server.
Demo completed successfully.
```

プログラムでエラーが発生した場合、Oracle8i *interMedia* Image ソフトウェアが正しくインストールされない、またはデータベースが起動されない場合があります。プログラムが正常に完了した場合、前述したカットおよびスケールによって処理された元のイメージおよび結果のイメージは共通のイメージ・レンダリング・ツールで表示できます。

## サンプル Video スクリプト

Video スクリプトは、次のファイルで構成されています。

- `viddemo.sql` - 次に示す Video オブジェクトの機能を含む Video デモです。
  - *interMedia* オブジェクトをチェックします。
  - デモ内の Video でサンプル表を作成します。
  - Video 表に NULL 行を挿入します。
  - アウトラインをチェックします。
  - すべての Video 属性を直接チェックします。
  - すべての Video 属性をメソッドを呼び出すことによってチェックします。
  - 2つのファイル (`fplugins.sql` および `fpluginb.sql`) を使用してユーザー独自形式のプラグインをインストールします。
- `fplugins.sql` - サポートする任意の形式のプラグインを書き込むためのガイドラインとして使用するデモ形式のプラグイン仕様です。
- `fpluginb.sql` - サポートする任意の形式のプラグインを書き込むためのガイドラインとして使用するデモ形式のプラグイン本体です。

**参照：** この SQL デモの実行方法の要件および手順の詳細は、  
`$ORACLE_HOME/ord/vid/demo` ディレクトリの `README.txt` ファイルを参照してください。

## Java デモ

Java デモは、Audio および Video 両方のクライアント側で Java クラスを使用して、ユーザー独自のアプリケーションの作成方法を理解するために役立ちます。これら 2つのデモでは、Audio および Video オブジェクトがクライアント側で即座に、多数のアクセサ方式が呼ばれます。Audio の Java デモ・ファイルは `$ORACLE_HOME/ord/aud/demo` に、Video の Java デモファイルは `$ORACLE_HOME/ord/vid/demo` に保存されています。それぞれの Java デモを実行する要件および手順の詳細は、`README.txt` ファイルを参照してください。

## MediaAnnotator

MediaAnnotator プログラムは、Oracle8i *interMedia* CD に含まれていません。その他の Oracle ソフトウェアとともに、次の URL で検索できます。

[http://technet.oracle.com/software/products/intermedia/software\\_index.htm](http://technet.oracle.com/software/products/intermedia/software_index.htm)

**参照：** 詳細は、『Oracle8i *interMedia* Audio,Image,Video ユーザーズ・ガイドおよびリファレンス』および『Oracle8i *interMedia* Audio,Image,Video Java Classes 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。

## Locator

Oracle8i *interMedia* Locator には、変更および実行できるスクリプトが多数あります。

### サンプル・スクリプト

Oracle8i *interMedia* Locator のサンプル・スクリプトは、この製品をインストールした後に、次のディレクトリで使用可能となります。

`$ORACLE_HOME/md/demo/geocoder`

これらのスクリプトは、次のファイルで構成されています。

- `geohttp.sql`

このファイルは、2 つに分かれています。1 つは、対話モードでジオコード機能を実行します。もう 1 つは、バッチ・モードでジオコード機能を実行します。

- 対話モード

ファイルのこの部分のリストについては、『Oracle8i *interMedia* Locator ユーザーズ・ガイドおよびリファレンス』の「GEOCODE1 ファンクション (lastline フィールドを含む)」の例 1 を参照してください。

- バッチ・モード

バッチ・モードで `geohttp.sql` を実行する前に、`nh_cs.sql` ファイルで設定表を更新しておく必要があります。ファイルのこの部分のリストについては、『Oracle8i *interMedia* Locator ユーザーズ・ガイドおよびリファレンス』の「GEOCODE1 ファンクション (lastline フィールドを含む)」の例 2 または例 3 を参照してください。

- `geoindex.sql`

このファイルには、次のものが含まれています。

- `ESTIMATE_LEVEL` という名前のファンクションです。このファンクションは、100 マイルを超える半径距離を使用しての問合せを、`WITHIN_DISTANCE` 空間ロケータ索引で使用する索引レベルでの見積りを改善します。このファイルのリストについては、『Oracle8i interMedia Locator ユーザーズ・ガイドおよびリファレンス』の「`ESTIMATE_LEVEL`」の例を参照してください。
- `SETUP_LOCATOR_INDEX` という名前のプロシージャ文です。このプロシージャ文は、空間情報が格納されている `cust_table` 表内の空間情報を含む位置列において、セットアップ空間ロケータ索引を作成します。このファイルのリストについては、『Oracle8i interMedia Locator ユーザーズ・ガイドおよびリファレンス』の第 2 章「Locator ファンクション」、「`SETUP_LOCATOR_INDEX`」の例を参照してください。

- `geolocate.sql`

このファイルには、ルーチンが含まれています。このルーチンは、関心対象の形状を動的に作成して、オフィスの半径 10 マイルの範囲内でどのくらい格納されているかを検索するために `NH_COMPUTER_STORES` 表に対して問い合わせます。このファイルのリストについては、『Oracle8i interMedia Locator ユーザーズ・ガイドおよびリファレンス』の第 2 章「Locator ファンクション」、「`LOCATOR_WITHIN_DISTANCE`」の例 2 を参照してください。

**参照：** 詳細は、『Oracle8i interMedia Locator 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。

# Oracle8i Time Series デモ

表 7-1 に、Oracle8i Time Series に含まれるデモを示します。この表では、各デモの説明およびそれらのファイルがインストールされるデフォルト・ディレクトリを示します。

デモ・ディレクトリは、\$ORACLE\_HOME/ord/ts で検索できます。

表 7-1 Oracle8i Time Series デモ

説明	ディレクトリ
クイック・スタート・デモは、すばやく簡単に Oracle8i Time Series の使用を開始します（詳細は、『Oracle8i Time Series ユーザーズ・ガイド』の第 1 章を参照）。	demo/tsquick
使用方法デモは、エンド・ユーザー向けおよび既存の Oracle8i Time Series 機能を使用する製品開発者向けです（詳細は、『Oracle8i Time Series ユーザーズ・ガイド』の第 1 章を参照）。	demo/usage
電子ユーティリティ・アプリケーションは、15 分間のデータのサマリーをピークおよびオフ・ピークで計算する方法を示します。	demo/usageutl
Time Series データの Java ベース検索は、Oracle8i Time Series Java API のプロトタイプを使用して、Web ブラウザで実行するために設計されています（詳細は、『Oracle8i Time Series ユーザーズ・ガイド』の第 1 章を参照）。	demo/applet
Java コードのサンプル・セグメントは、Time Series を操作して、その結果を印刷します（詳細は、『Oracle8i Time Series ユーザーズ・ガイド』の第 1 章を参照）。	demo/java
管理ツールを使用して既存の Time Series ディテール表をレトロフィットする手順を示すデモです。また、Time Series ディテール表の複数の修飾子列に対する、Time Series 問合せのサポート方法も示します。	demo/retrofit
上級開発者デモは、Oracle8i Time Series ファンクションを拡張するためのものです。	demo/extend
OCI デモは、Oracle コール・インタフェースを使用して Oracle8i Time Series ファンクションをコールする方法を示します。	demo/oci
PRO*C/C++ デモは、Oracle Pro*C/C++ プリコンパイラを使用して作成されるアプリケーションでの Oracle8i Time Series ファンクションをコールする方法を示します。	demo/proc
Oracle デモは、Oracle Forms アプリケーションで Oracle8i Time Series ファンクションをコールする方法を示します。	demo/dev2k

デモ・ディレクトリにある README.txt ファイルは、デモを説明しています。また、各デモ用のディレクトリには、そのデモの詳細を説明している README.txt ファイルが含まれています。

**参照：** 詳細は、『Oracle8i Time Series ユーザーズ・ガイド』を参照してください。

## Oracle8i Visual Information Retrieval

Visual Information Retrieval には、サンプル・プログラムが含まれています。このサンプル・プログラムは、2つのイメージをデータベースへロードする方法、それらのシグネチャを生成する方法、および類似の重み付け機能を使用してシグネチャを比較する方法を示します。

このプログラムは、2つのデータ・ファイル (virdemo1.dat および virdemo2.dat) をそれらの入力のために使用します。その他の入力またはパラメータは必要ありません。

### 環境

次のように仮定されています。

- Visual Information Retrieval がインストールされていて、PUBLIC に EXECUTE 権限があります。
- インストール・スクリプトが実行されています。VIRDEMODIR ディレクトリが作成され、PUBLIC に READ 権限が与えられていて、イメージ・データ・ファイルをデータベースに読み込むことができます。
- virdemo1.dat および virdemo2.dat が、VIRDEMODIR ディレクトリに常駐する有効なイメージ・ファイルです。また、ユーザーはそのディレクトリに対して読み込み / 書き込みができます。
- ユーザー SCOTT には、デフォルトのパスワード TIGER が設定されています。このサンプル・プログラムを正常に実行するためには、SCOTT に割り当てる表領域を増やす必要があります。

### サンプル・プログラムの実行

サンプル・プログラムには、2つの実行方法があります。1つは含まれるサンプル・イメージを使用する方法で、もう1つは独自のイメージを使用する方法です。

例 7-2 に、提供されているイメージ・ファイルを使用してサンプル・プログラムを実行する例を示します。イメージは、同じ属性の重みを使用して比較されます。

- Globalcolor = 1.0
- Localcolor = 1.0
- Texture = 1.0

- Structure = 1.0

### 例 7-2 提供されているイメージでのサンプル・プログラムの実行

```
% virdemo
Image 1 and 2 have a similarity score of 0.0
```

例 7-3 に、コマンドラインでユーザー独自のイメージを指定する方法を示します。イメージは、\$ORACLE\_HOME/ord/vir/demo ディレクトリに保存しておく必要があります。

### 例 7-3 ユーザー独自のイメージでのサンプル・プログラムの実行

```
% virdemo image1 image2 global_color local_color texture structure
```

このサンプル・プログラムでは、2つのファイル名および4つの属性の重み（0.0～1.0）の6つのパラメータをすべて指定する必要があります。ユーザー独自のアプリケーションで VIRScore() 演算子を使用する場合は、1つ以上の属性の重みのみを指定する必要があります。

いくつかのサンプル・イメージ・ファイルが、異なるビジュアル属性を強調する効果を示すために VIRDEMODIR ディレクトリに提供されています。イメージを表示して、サンプル・プログラムを使用して異なる重みを比較してみるには、イメージ・ビューア（xv など）を使用できます。

**参照：** 詳細は、『Oracle8i Visual Information Retrieval User's Guide and Reference』および『Oracle8i Visual Information Retrieval Java Client User's Guide and Reference』を参照してください。

## Oracle8i Spatial

より詳細な情報は、\$ORACLE\_HOME/md/demo/readme.txt を参照してください。

**参照：** 詳細は、『Oracle8i Spatial 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。

---

## Optimal Flexible Architecture

ここでは、Optimal Flexible Architecture (OFA) について説明します。次のトピックが含まれます。

- [OFA の概要](#)
- [UNIX での OFA の実装](#)

## OFA の概要

オラクル社は、Oracle8i のインストールおよび構成時に、OFA 標準を実装することをお勧めします。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。OFA により、次の操作が行われます。

- 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。
- データの損傷を起こす可能性のある管理タスク（ソフトウェア、データのバックアップ機能など）を簡略化します。
- Oracle データベース間の移動を簡単にします。
- 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 に準拠したデータベースの利点について説明します。

### ファイル・システムの編成

OFA は、ファイル・システムを、管理が容易で、次のスケーラビリティに対応できるように編成します。

- 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

OFA は、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・サポートの回避

OFA 標準を利用するには、大抵の場合は、新規ハードウェアに投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って、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

OFA では、次のアイテムを複数のディスクに分散させることができます。

- 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維持

ログイン・ホーム・ディレクトリを、それを参照するプログラムを変更することなく、追加、移動および削除できます。

## 独立した UNIX ディレクトリ・サブツリーの提供

OFA は、ファイルのカテゴリを、独立した UNIX ディレクトリ・サブツリーに分類します。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。このため、以前のバージョンを削除する前に、新しいバージョンをテストできます。管理者はアップグレード後に新しいバージョンへ容易に移行でき、ユーザーが移行作業によって影響を受けることもありません。

## 各データベースの管理情報の区別

OFA は、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

OFA がデータベース・ファイルに付ける名前により、次のことが容易になります。

- データベース・ファイル、制御ファイル、REDO ログ・ファイルおよびデータ・ファイルを明確に識別できます。
- あるデータベースのファイルと、別のデータベースのファイルとを容易に区別できます。
- 各データ・ファイルを表領域に関連付けます。

## 表領域の内容の分割

OFA は表領域の内容を分割するため、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。

## すべてのドライブでの I/O 負荷のチューニング

OFA は、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。

## Oracle Parallel Server での OFA の使用

Oracle Parallel Server (OPS) のインストールでは、次のメリットがあります。

- OFA は、すべての管理データを一元管理します。管理データには、すべてのデータベース管理者がアクセスできます。
- OFA は、ファイル名を使用して、インスタンスの管理データをインスタンスに関連付けます。

# UNIX での OFA の実装

データベース・ファイルのネーミング規則に従うことによって、データ管理の様々な問題を回避できます。この索引で提供する OFA 規則は、「The OFA Standard: Oracle8 for Open Systems」で公開されている OFA 推奨規則に対応しています。

## マウント・ポイントの使用

### マウント・ポイントの作成

Oracle8i のインストールには、少なくとも 2 つのマウントポイントが必要です。1 つはソフトウェア用のマウント・ポイントで、もう 1 つはデータベース・ファイル用のマウント・ポイントです。OFA では、ソフトウェア用に 1 つのマウント・ポイント、データベース・ファイル用に最低 3 つのマウント・ポイントが必要です。標準的なマウント・ポイントと、超大規模データベース (VLDB) のマウント・ポイントでは、構文が異なります。

標準マウント・ポイントの名前は、次のように指定します。

/pm

それぞれ次の意味があります。

*p* 文字列定数です。

*m* 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。

標準マウント・ポイントの例としては、/u01、/u02、/disk01 および /disk02 をあげることができます。

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、各データベースに十分なドライブがあり I/O のボトルネックが発生しない場合、超大規模データベース（VLDB）のマウント・ポイントに名前を付けるには、次のように入力します。

/q/dm

それぞれ次の意味があります。

*q* Oracle データの保存先を示す文字列です。

*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）です。

たとえば、/u01/oradata/test および /u02/oradata/test という名前の VLDB マウント・ポイントを作成すると、Oracle テスト・データベースに2つのドライブが割り当てられます。

## ディレクトリの名前

### ホーム・ディレクトリの名前

ホーム・ディレクトリに名前を付けるには、次のように入力します。

/mp/h/u

それぞれ次の意味があります。

*mp* マウント・ポイントの名前です。

*h* 標準のディレクトリ名です。

*u* ディレクトリの所有者の名前です。

たとえば、Oracle Server ソフトウェア所有者のホーム・ディレクトリは /u01/app/oracle (ORACLE\_BASE としても知られる) で、Oracle アプリケーション・ソフトウェア所有者のホーム・ディレクトリは /u01/app/applmgr 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ことをお勧めします。こうすることで、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、そのパス名を保存するファイル（/etc/passwd、Oracle oratab ファイルなど）のみで参照します。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・ディレクトリの使用

アプリケーション・ソフトウェアの複数のバージョンを同時に実行するには、次のように入力して、Oracle8i Server ソフトウェアの各バージョンを異なるディレクトリに保存します。

```
/mp/h/product/v
```

それぞれ次の意味があります。

<i>mp</i>	マウント・ポイントの名前です。
<i>h</i>	標準のディレクトリ名です。
<i>v</i>	ソフトウェアのバージョンです。

たとえば、/u01/app/oracle/product/8.1.7 は、Oracle8i Server ファイルが保存されているディレクトリ構造の先頭を示します。環境変数 ORACLE\_HOME をこのディレクトリに設定します。

# ファイルの名前

## 管理ファイルの名前

管理データを編成する場合、次の構文を使用して、各データベース用の管理ファイルを保存することをお勧めします。

h/admin/d/a/

それぞれ次の意味があります。

- h                      Oracle ソフトウェア所有者のホーム・ディレクトリです。
- s                      データベース名 (DB\_NAME) です。
- a                      各データベース管理ファイル用のサブディレクトリです。

表 A-1 に、データベース管理ファイル用のサブディレクトリを示します。

表 A-1 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	指定した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 (configdb_name.ora の AUDIT_FILE_DEST を、この位置に設定します。使用しないファイルを、このディレクトリから定期的に削除してください)。
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
udump	ユーザー SQL トレース・ファイル

たとえば、sab という名前のデータベースの一部である場合、adhoc サブディレクトリのパス名は /u01/app/oracle/admin/sab/adhoc/ になります。

### データベース・ファイルの名前

データベース・ファイルの場合、次のネーミング規則を使用することをお勧めします。

- 制御ファイルの場合、`/mp/q/d/control.ct1`
- REDO ログ・ファイルの場合、`/mp/q/d/redonn.log`
- データ・ファイルの場合、`/mp/q/d/tnn.dbf`

それぞれ次の意味があります。

<code>mp</code>	マウント・ポイントの名前です。
<code>q</code>	Oracle データを他のすべてのファイルと区別するための文字列（通常、ORACLE または <code>oradata</code> ）です。
<code>d</code>	データベース名（ <code>DB_NAME</code> ）です。
<code>t</code>	Oracle 表領域の名前です。
<code>nn</code>	2 桁の数字です。

**注意：** `/pm/q/d` には、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は保存しないでください。

たとえば、データ・ファイルに `/u03/oradata/sab/system01.dbf` という名前を付けることができます。

### セグメントの分類

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。

Oracle データベースごとに、表 A-2 に示す特殊な表領域を作成します。アプリケーション・セグメントに必要な表領域以外に、これらの表領域が必要です。

表 A-2 特殊な表領域

表領域	説明
DRSYS	Oracle <i>interMedia</i> セグメント
INDX	USERS 表領域のデータに対応付けられた索引
OEM_REPOSITORY	Oracle Enterprise Manager のリポジトリ
RBS	ロールバック・セグメント
SYSTEM	データ・ディクショナリ・セグメント

**表 A-2 特殊な表領域（続き）**

表領域	説明
TEMP	テンポラリ・セグメント
USERS	その他のユーザー・セグメント

この方法でセグメントを分割すると、ディクショナリ・セグメントが削除されることはなく、また、削除可能なその他のセグメントが SYSTEM 表領域に保存されることはありません。これによって、表領域の空き領域が断片化したために SYSTEM 表領域を再作成する必要はなくなります。

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。Oracle8i 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・ファイルのベース名を付ける際、次の構文を使用することをお勧めします。

 $t_{nn}.dbf$ 

それぞれ次の意味があります。

t 記述的なテーブル名です。

nn 2桁の数字です。

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・ファイルの名前とそのデータ・ファイルを使用する表領域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、GLD および GLX という名前を指定します。

OFA 構造に基づいた Oracle ファイル

表 A-3 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 A-3 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

構文	ファイルのクラス
/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 apps ソフトウェアのサブツリー
/*/app/oracle/product	Oracle Server ソフトウェアのサブツリー
/*/app/oracle/product/8.1.7	Oracle8i リリース 8.1.7 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 A-4 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 A-4 OFA インストールの階層型ファイル・マッピング

マッピング	説明
/	ルート・マウント・ポイント
u01/	Oracle ソフトウェアのマウント・ポイント #1
app/	app ソフトウェアのサブツリー
oracle/	Oracle ソフトウェア所有者のホーム・ディレクトリ
admin/	データベース管理ファイルのサブツリー
TAR/	サポート・ログのサブツリー



表 A-4 OFA インストールの階層型ファイル・マッピング (続き)

マッピング		説明
	db_name1/	db_name1 データベースの管理サブツリー
	bdump/	background_dump_dest
	cdump/	core_dump_dest
	udump/	user_dump_dest
	create/	データベース作成 SQL スクリプト
	pfile/	データベース init パラメータ・ファイル
	db_name2/	db_name2 データベースの管理サブツリー
	doc/	オンライン・ドキュメント
	local/	ローカルの Oracle ソフトウェアのサブツリー
	aps6/	Oracle6 管理パッケージ
	aps7/	Oracle7 管理パッケージ
	product/	配布ファイル
	7.3.4/	ORACLE_HOME 7.3.4 のインスタンス
	8.0.6/	ORACLE_HOME 8.0.6 のインスタンス
	8.1.7/	ORACLE_HOME 8.1.7 のインスタンス
	oraInventory	Oracle8i Inventory のサブツリー
	log	インストール・ログ・ファイル
home		ログイン・ホーム・ディレクトリのサブツリー
	ltb/	ユーザーのホーム・ディレクトリ
	sbm/	ユーザーのホーム・ディレクトリ
u02/		ユーザー・データのマウント・ポイント #2
	home/	ログイン・ホーム・ディレクトリのサブツリー
	cvm/	ユーザーのホーム・ディレクトリ
	vrm	ユーザーのホーム・ディレクトリ
	oradata/	Oracle データのサブツリー
	db_name1/	db_name1 データベースのサブツリー
	db_name2/	db_name2 データベースのサブツリー

表 A-4 OFA インストールの階層型ファイル・マッピング（続き）

マッピング	説明
u03/  oradata/ db_name1/ db_name2/	ユーザー・データのマウント・ポイント #3  Oracle データのサブツリー db_name1 データベースのサブツリー db_name2 データベースのサブツリー
u04/  oradata/ db_name1/ db_name2/	ユーザー・データのマウント・ポイント #4  Oracle データのサブツリー db_name1 データベースのサブツリー db_name2 データベースのサブツリー
/var opt/ oracle/	oratab および oraInst.loc の保存場所
/usr local/ bin/	oraenv, coraenv および dbhome スクリプトの保存場所

RAW デバイスのサイズ

Oracle データベース・ファイルを保存するための RAW デバイスは、すべて標準的なサイズに設定します。1 つのサイズに標準化することをお薦めします。1 つのサイズに標準化すると、あるパーティションから別のパーティションに RAW ファイルを安全に移動できます。必要な数のデータベース・ファイルを保存でき、かつ、操作性に問題のない程度の大きさに設定してください。

たとえば、2GB のドライブの場合、サイズと数のバランスを取るために、200MB の大きさの 10 個のパーティションに分割します。RAW デバイスを使用する表領域は、複数のドライブにストライプ化してください。可能であれば、ストライプ化には論理ボリューム・マネージャを使用します。

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

OPS 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/ 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 A-5 に、管理ディレクトリ構造の例を示します。

表 A-5 二重インスタンス OPS の管理ディレクトリ構造

構造	説明
u01/ app/oracle/admin/sab/	<b>sab</b> データベース用の管理ディレクトリ
adhoc/	スクリプト用のディレクトリ
arch/	すべてのインスタンス用のログ・アーカイブ先
redo001.arc	アーカイブ REDO ログ・ファイル
bdump/	バックグラウンド・ダンプ・ファイルのディレクトリ
inst1/	<i>inst1</i> インスタンスのバックグラウンド・ダンプ先
inst2/	<i>inst2</i> インスタンスのバックグラウンド・ダンプ先
cdump/	コア・ダンプ・ファイル用のディレクトリ
inst1/	<i>inst1</i> インスタンスのコア・ダンプ先
inst2/	<i>inst2</i> インスタンスのコア・ダンプ先
create/	作成スクリプト用のディレクトリ
1-rdbms.sql	<i>inst</i> データベース作成用の SQL スクリプト
exp/	エクスポート用のディレクトリ
20000120full.dmp	2000 年 1 月 20 日のフル・エクスポート・ダンプ・ファイル
export/	エクスポート・ファイル用のディレクトリ
import/	インポート・ファイル用のディレクトリ
logbook/	<i>inst</i> ログブック・エントリ用のディレクトリ
inst1/	<i>inst1</i> インスタンス・レポート用のディレクトリ
params.lst	<i>inst1</i> インスタンスの v\$parameter レポート
inst2/	<i>inst2</i> インスタンス・レポート用のディレクトリ
params.lst	<i>inst2</i> インスタンスの v\$parameter レポート

表 A-5 二重インスタンス OPS の管理ディレクトリ構造（続き）

構造	説明	
pfile/	user.lst	dba_users レポート
		インスタンス・パラメータ・ファイル用のディレクトリ
	inst1/	inst1 インスタンス・パラメータ用のディレクトリ
	init	inst1 インスタンスのインスタンス・パラメータ
	inst2/	inst2 インスタンス・パラメータ用のディレクトリ
	init	inst2 インスタンスのインスタンス・パラメータ
udump/		ユーザー・ダンプ・ファイル用のディレクトリ
	inst1/	inst1 インスタンスのユーザー・ダンプ先
	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

ORACLE\_BASE ディレクトリ

ORACLE\_BASE は、Oracle ディレクトリ構造のルートです。表 A-6 に、ORACLE\_BASE 内のサブディレクトリを示します。OFA に準拠したデータベースのインストールに Oracle Universal Installer を使用する場合、ORACLE\_BASE は、`/mp/app/oracle` に設定されます。

表 A-6 ORACLE\_BASE ディレクトリのサブディレクトリ

サブディレクトリ	説明
admin	管理ファイル
doc	オンライン・ドキュメント
local	ローカルの Oracle ソフトウェアのサブツリー
product	Oracle ソフトウェア

## ORACLE\_HOME ディレクトリ

OFA 準拠の Oracle Server をインストールする場合、ORACLE\_HOME ディレクトリは `/mp/app/oracle/product/release_number` になります。表 A-7 に、ORACLE\_HOME ディレクトリ内のサブディレクトリを示します。ディレクトリには、インストールした各 Oracle 製品のサブディレクトリも含まれます。

**表 A-7 ORACLE\_HOME ディレクトリのサブディレクトリ**

サブディレクトリ	説明
<code>assistants</code>	構成アシスタント
<code>bin</code>	すべての製品のバイナリ・ファイル
<code>ctx</code>	<i>interMedia Text</i> オプション
<code>db</code>	<code>init<sub>sid</sub>.ora</code> 、 <code>lk<sub>sid</sub></code>
<code>install</code>	インストールした関連ファイル
<code>lib</code>	Oracle 製品ライブラリ
<code>javavm</code>	Java Virtual Machine
<code>jdbc</code>	JDBC ドライバ
<code>jlib</code>	Java クラス
<code>md</code>	Spatial オプション
<code>mlx</code>	Xerox Stemmer ( <i>interMedia Text</i> オプション用)
<code>network</code>	Net8
<code>nlsrtl</code>	NLS ランタイム読み込み可能データ
<code>ocommon</code>	すべての製品に共通のファイル
<code>odg</code>	データ収集機能
<code>opsm</code>	Parallel Server Manager Components
<code>oracore</code>	コア・ライブラリ
<code>orb</code>	Object Request Broker
<code>ord</code>	データ・オプション
<code>otrace</code>	Oracle TRACE
<code>plsql</code>	PL/SQL
<code>precomp</code>	プリコンパイラ
<code>rdbms</code>	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
<code>slax</code>	SLAX パーサー

表 A-7 ORACLE\_HOME ディレクトリのサブディレクトリ（続き）

サブディレクトリ	説明
sqlplus	SQL*Plus
svrmgr	Server Manager
sysman	System Management

製品のサブディレクトリの内容

表 A-8 に、各製品のサブディレクトリの内容を示します。

表 A-8 製品のサブディレクトリの内容

サブディレクトリ	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	デモンストレーション・スクリプトおよびデータ・ファイル
doc	README ファイル（readmeunix.doc など）
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
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	アメリカ英語メッセージ・ファイル、多言語オプション（旧各国語サポート）のメッセージ・テキストおよびバイナリ・ファイル（oraus.msg、oraus.msb など）

製品のサブディレクトリの例

表 A-9 に、サブディレクトリの内容の例を示します。

表 A-9 製品のサブディレクトリの例

サブディレクトリ	内容
rdbms	install、lib、admin、doc、mesg、log
sqlplus	install、demo、lib、admin、doc、mesg

admin ディレクトリのファイルのネーミング規則

表 A-10 に、rdbms/admin ディレクトリ内の SQL スクリプトを示します。

表 A-10 admin ディレクトリのファイルのネーミング規則

スクリプト	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) catparr.sql (Parallel Server のインストール用) catrep.sql (すべてのインストール用)  catproc.sql は、標準の PL/SQL パッケージ (DBMS_SQL、DBMS_OUTPUT など) を作成するためのスクリプトを実行します。
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

ファイル名拡張子の規則

表 A-11 に、ファイル名拡張子を示します。

表 A-11 ファイル名拡張子

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 Server 制御ファイル
.dat	SQL*Loader データ・ファイル
.dbf	Oracle Server 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

表 A-11 ファイル名拡張子（続き）

拡張子	説明
.h	C ヘッダー・ファイル（ <code>sr.h</code> は SQL*Report Writer のヘルプ・ファイル）
.jar	Java クラス・アーカイブ
.l	UNIX マニュアル・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 Server REDO ログ・ファイル）
.mk	Make ファイル
.msb	NLS メッセージ・ファイル（バイナリ）
.msg	NLS 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
.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
.tut	Bourne シェル・スクリプト・ファイル
.utd	ユニフォーム端末定義
.zip	Zip ファイル



# 索引

## 記号

? 略称, 1-2  
@ 略称, 1-2

## 数字

10 から 20 へのコマンドの移行, 4-9  
2 タスク, 1-14  
シグナル, 5-16

## A

ADDRESS 指定  
プロトコル, 6-6  
AIX  
システム・パラメータ, 2-7  
APPC/LU6.2 プロトコル, 6-11  
ADDRESS, 6-11

## B

BEQ プロトコル, 6-7  
ADDRESS, 6-7  
接続パラメータの構文, 6-8  
Bourne シェル・スクリプトのトレース, 1-4  
BUFFER  
imp のパラメータ, 2-11

## C

C  
Pro\*C/C++, 5-4  
SQL\*Module, 5-12  
CATPROC.SQL, A-17

CLASSPATH  
環境変数, 1-5  
CONNECT INTERNAL, 1-15  
coraenv, 1-4  
coraenv ファイル, 1-7  
CREATE CONTROL FILE パラメータ, 1-12

## D

dba グループ  
メンバー, 1-14  
db\_block\_size, 2-10  
dbhome ファイル, 1-7  
Developer/2000 (Oracle Developer) デモ, 7-8

## F

FORMAT プリコンパイラ, 5-9  
Pro\*COBOL, 5-10

## G

glogin.sql ファイル, 3-2

## I

imp  
BUFFER パラメータ, 2-11  
InitDir, 4-6  
initsid.ora ファイル, 1-18  
Intelligent Agent  
SNMP での構成, 6-13

## *interMedia*

- Java デモ, 7-5
- MediaAnnotator, 7-6
- サンプル Audio スクリプト, 7-2
- サンプル Video スクリプト, 7-5
- サンプル・プログラム, 7-3

*interMedia* Audio、Video、Image, 7-2

*interMedia* Locator, 7-6

*interMedia* Text, 7-2

I/O スレーブ, 2-18

IPC プロトコル, 6-8

- ADDRESS, 6-9

ireclen, 5-3

## **J**

---

JAVA\_POOL\_SIZE, 1-10

Java デモ, 7-8

## **L**

---

LARGE\_POOL\_SIZE, 1-10

LDAP スキーマ, 4-4

login.sql ファイル, 3-2

## **M**

---

MAXDATAFILES パラメータ, 1-12

maxfree, 2-7

MAXINSTANCES パラメータ, 1-12

MAXLOGFILES パラメータ, 1-12

MAXLOGHISTORY パラメータ, 1-12

MAXLOGMEMBERS パラメータ, 1-12

maxper, 2-8

maxperm, 2-7

MicroFocus COBOL コンパイラ, 5-7

minfree, 2-7

minperm, 2-7, 2-8

## **N**

---

Net8

- ADDRESS 指定, 6-6
- APPC/LU6.2 プロトコル, 6-11
- BEQ プロトコル, 6-7
- IPC プロトコル, 6-8
- Oracle Advanced Security, 6-14

- Oracle Connection Manager, 6-4
- Oracle Enterprise Manager Intelligent Agent, 6-12
- RAW プロトコル, 6-9
- TCP/IP プロトコル, 6-10
- アダプタ・ユーティリティ, 6-3
- ファイルおよびユーティリティ, 6-2
- プロトコル, 6-6
- プロトコルのサポート, 6-6
- マルチスレッド・サーバー, 6-4

Net8 の構成ファイル

- 保存場所, 6-2

NLS\_LANG 環境変数, 1-3

## **O**

---

OCI デモ, 7-8

OFA

- 「Optimal Flexible Architecture」を参照

Optimal Flexible Architecture, A-1

- OFA の特徴, A-2
- Oracle8 for UNIX での実装, A-2
- ディレクトリ構造, A-14
- ファイル・マッピング, A-2, A-10

Oracle8i Spatial, 7-10

Oracle8i で使用する UNIX 環境変数, 1-5

Oracle Advanced Security, 6-14

Oracle AQ

- Oracle Message Broker の初期化, 4-13

Oracle Connection Manager, 6-4

Oracle Developer デモ, 7-8

Oracle Diagnostics Pack, 2-5

ORACLE\_DOC 環境変数, ix

Oracle Enterprise Manager Intelligent Agent, 6-12

- tcl スクリプトのデバッグ, 6-12
- エージェント・サービスの展開および自動構成, 6-12

Oracle Forms デモ, 7-8

ORACLE\_HOME 環境変数, 1-2, 1-3

Oracle HTTP Server, 1-21

Oracle Information Navigator, ix

Oracle Internet Directory

- 起動, 4-13
- 構成, 4-3
- 停止, 4-9

ORACLE\_SID

- 環境変数, 1-4

ORACLE\_SID 環境変数, 1-2

Oracle Tuning Pack, 2-6  
Oracle 環境変数  
    EPC\_DISABLED, 1-3  
    NLS\_LANG, 1-3  
    ORACLE\_BASE, 1-3  
    ORACLE\_HOME, 1-3  
    ORACLE\_PATH, 1-3  
    ORACLE\_SID, 1-4  
    ORACLE\_TRACE, 1-4  
    ORAENV\_ASK, 1-4  
    ORA\_NLS33, 1-3  
    TNS\_ADMIN, 1-4  
    TWO\_TASK, 1-4  
    UNIX 環境変数, 1-5  
Oracle コール・インタフェース, 5-13  
    デモ・プログラム, 5-13  
Oracle コール・インタフェース (OCI)・デモ, 7-8  
Oracle システム識別子, 1-4  
Oracle ソフトウェア所有者, 1-13  
    特殊アカウント, 1-13  
Oracle データ・オプション・デモの実行, 7-1  
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, 5-15  
        未定義シンボル, 5-15  
Oracle ランタイム・システム  
    Pro\*COBOL, 5-8  
ORAENV\_ASK, 1-4  
oraenv ファイル, 1-7  
    説明, 1-7  
        データベース間の移動, 1-8  
ORA\_NLS 環境変数, 1-3  
orapwd コマンド, 1-16  
oratab ファイル, 1-27  
oreclen, 5-3

## P

---

PL/SQL デモンストレーション  
    ロード, 1-25  
Pro\*C/C++  
    Make ファイル, 5-5  
        シグナル, 5-17  
        使用, 5-4  
        デモ・プログラム, 5-5  
Pro\*C/C++ デモ, 7-8  
Pro\*COBOL, 5-7

    FORMAT プリコンパイラ, 5-9, 5-10  
    Oracle ランタイム・システム, 5-8  
    環境変数, 5-7  
    デモ・プログラム, 5-8  
    ネーミング方法の違い, 5-7  
    ユーザー・プログラム, 5-9  
PRODUCT\_USER\_PROFILE 表, 3-2  
PTX Agent, 2-2  
PTX Manager, 2-2

## R

---

RAW プロトコル, 6-9  
README.txt ファイル, 7-9  
root.sh スクリプト, 1-7  
root ユーザー, 1-13

## S

---

Server Manager  
    コマンド, 1-15  
    セキュリティ, 1-15  
SGA, 2-11  
    計算, 1-10  
SHELL, 1-6  
SHUTDOWN コマンド, 1-15  
    セキュリティ, 1-15  
SIGCLD 2 タスク・シグナル, 5-17  
SIGIO 2 タスク・シグナル, 5-17  
SIGINT 2 タスク・シグナル, 5-17  
SIGPIPE 2 タスク・シグナル, 5-17  
SIGTERM 2 タスク・シグナル, 5-17  
SIGURG 2 タスク・シグナル, 5-17  
Site Profile SQL\*Plus, 3-2  
smit  
    「System Management Interface Tool」を参照  
SNMP  
    Intelligent Agent, 6-13  
Spatial デモンストレーション, 7-10  
SPOOL コマンド  
    SQL\*Plus, 3-6  
        使用, 3-6  
SQL\*Loader, 2-11  
SQL\*Module for Ada, 5-12  
    使用, 5-12  
    ユーザー・プログラム, 5-13  
SQL\*Module for C, 5-12

SQL\*Plus, 3-5  
  PRODUCT\_USER\_PROFILE 表, 3-2  
  Site Profile, 3-2  
  SPOOL コマンド, 3-6  
  User Profile, 3-2  
  オペレーティング・システム・コマンドの実行,  
    3-6  
  システム・エディタ, 3-5  
  使用, 3-5  
  制限事項, 3-6  
  設定ファイル, 3-2  
  デフォルト・エディタ, 3-5  
  デモンストレーション表, 3-3  
  ヘルプ機能, 3-4  
  割込み, 3-6  
SQL\*Plus の使用, 3-5  
SQL\*Plus への割込み, 3-6  
SQL トレース, 2-4  
SQL の管理, 3-2  
STARTUP コマンド, 1-15  
step1.sql セットアップ・スクリプト, 4-13  
step2.sql セットアップ・スクリプト, 4-13  
step3.sql セットアップ・スクリプト, 4-13  
SYSDATE, 1-8  
System Management Interface Tool, 2-3  
SYSTEM アカウント, 1-13  
  権限, 1-13  
SYS アカウント, 1-13  
  権限, 1-13

## T

---

TCP/IP プロトコル, 6-10  
timed\_statistics, 2-4  
TKPROF, 2-4  
TNS\_ADMIN 環境変数, 1-4  
TNS リスナー  
  Oracle TCP/IP プロトコルの構成, 6-10  
TWO\_TASK 環境変数, 1-4

## U

---

UDP のチューニング, 2-25  
User Profile  
  SQL\*Plus, 3-2  
utlbstat.sql, 2-4  
utlestat.sql, 2-4

## V

---

vmtune, 2-7

## X

---

XA 機能, 5-19

## あ

---

アカウント, 1-13  
アダプタ・ユーティリティ, 6-3

## い

---

インストール  
  MQSeries, 4-15

## う

---

受取りルーチン, 5-17

## お

---

オペレーティング・システム・コマンドの実行, 3-6

## か

---

各国語サポート (NLS) 環境変数, 1-3  
環境変数, 1-2, 5-7  
  MicroFocus COBOL コンパイラ, 5-7  
  NLS\_LANG, 1-3  
  ORACLE\_DOC, ix  
  ORACLE\_HOME, 1-3  
  ORACLE\_SID, 1-4  
  ORACLE\_TRACE, 1-4  
  ORAENV\_ASK, 1-4  
  ORA\_NLS, 1-3  
  Pro\*COBOL, 5-7  
  SHELL, 1-6  
  TNS\_ADMIN, 1-4, 6-2  
  TWO\_TASK, 1-4

## き

---

共通の環境

- oraenv ファイル, 1-7
- 設定, 1-7

## け

---

言語, 1-3

## こ

---

構成

- Netscape ディレクトリ・サーバー, 4-3
- Oracle Internet Directory, 4-3
- ディレクトリ, 4-3
- マスター・エージェント, 6-13

構成ファイル

- Net8, 6-2
- プリコンパイラ, 5-2

高度なネットワーク・オプション

- .bak ファイル, 6-14
- DCE 統合, 6-14
- セキュリティおよびシングル・サインオン, 6-14

コマンド

- orapwd, 1-16

コマンド・インタプリタ, 1-6

## さ

---

再リンク, 1-9

- パラメータ, 1-9

参照マニュアル, x

## し

---

シグナル

- 2 タスク, 5-16

シグナル・ハンドラ

- 使用, 5-16

シグナル・ルーチン, 5-17

- 例, 5-17

システム・エディタ

- SQL\*Plus, 3-5

システム時刻, 1-8

システム・パラメータ

- AIX, 2-7

自動ログイン

- remote\_os\_roles, 1-17

シャドウ・プロセス, 1-14

上級開発者デモ, 7-8

初期化パラメータ

- BACKGROUND\_DUMP\_DEST, 1-18
- BITMAP\_MERGE\_AREA\_SIZE, 1-18
- COMMIT\_POINT\_STRENGTH, 1-18
- CONTROL\_FILES, 1-18
- CREATE\_BITMAP\_AREA\_SIZE, 1-18
- DB\_BLOCK\_BUFFERS, 1-18
- DB\_BLOCK\_SIZE, 1-18
- DB\_FILE\_DIRECT\_IO\_COUNT, 1-18
- DB\_FILE\_MULTIBLOCK\_READ\_COUNT, 1-18
- DB\_FILES, 1-18
- DISTRIBUTED\_TRANSACTIONS, 1-18
- HASH\_AREA\_SIZE, 1-18
- HASH\_MULTIBLOCK\_IO\_COUNT, 1-19
- LOG\_ARCHIVE\_BUFFERS, 1-19
- LOG\_ARCHIVE\_BUFFER\_SIZE, 1-19
- LOG\_ARCHIVE\_DEST, 1-19
- LOG\_ARCHIVE\_FORMAT, 1-19
- LOG\_BUFFER, 1-19
- LOG\_CHECKPOINT\_INTERVAL, 1-19
- LOG\_SMALL\_ENTRY\_MAX\_SIZE, 1-19
- MTS\_LISTENER\_ADDRESS, 1-19
- MTS\_MAX\_DISPATCHERS, 1-19
- MTS\_MAX\_SERVERS, 1-19
- MTS\_SERVERS, 1-19
- NLS\_LANGUAGE, 1-19
- NLS\_TERRITORY, 1-19
- OBJECT\_CACHE\_MAX\_SIZE\_PERCENT, 1-19
- OBJECT\_CACHE\_OPTIMAL\_SIZE, 1-19
- OPEN\_CURSORS, 1-19
- OS\_AUTHENT\_PREFIX, 1-19
- PROCESSES, 1-19
- SHARED\_POOL\_SIZE, 1-20
- SORT\_AREA\_SIZE, 1-20
- USER\_DUMP\_DEST, 1-20

## す

---

スーパーユーザー, 1-13

スレッドのサポート, 5-16

## せ

---

制限, 1-12

制限事項 (SQL\*Plus), 3-6

ウィンドウのサイズ変更, 3-6

リターン・コード, 3-6

セキュリティ, 1-14, 1-15, 1-16

2 タスク・アーキテクチャ, 1-14

CONNECT INTERNAL, 1-15

Server Manager アクセス, 1-15

SHUTDOWN コマンド, 1-15

STARTUP コマンド, 1-15

UNIX の機能, 1-14

グループ・アカウント, 1-14

ファイル所有権, 1-14

設定ファイル

SQL\*Plus, 3-2

## ち

---

チューニング

SGA, 2-11

UDP, 2-25

アーカイバ・バッファ, 2-10

メモリーおよびページング, 2-7

チューニング、パフォーマンス, 2-5

## て

---

ディスク

割当て制限, 1-12

ディスク I/O

I/O スレーブ, 2-18

ディレクトリ・データの移行, 4-9

データ・オプション, 7-1

データベース

起動, 4-12

停止, 4-10

ブロック・サイズ, 2-10

データベースの変更, 1-8

データベース・ファイル, 1-16

デバッグ・プログラム, 5-3

デモ (デモンストレーション・ファイル)

Java, 7-8

Oracle Developer, 7-8

Oracle コール・インタフェース (OCI), 7-8

Pro\*C/C++, 7-8

既存表のレトロフィット, 7-8

上級開発者, 7-8

使用方法, 7-8

電子ユーティリティ, 7-8

デモの使用, 7-8

デモ・プログラム

Oracle コール・インタフェース, 5-13

Pro\*C/C++, 5-5

Pro\*COBOL, 5-8

デモンストレーション

プリコンパイラ, 1-26

プロシージャ・オプション, PL/SQL, 1-25

デモンストレーション表

SQL\*Plus, 3-3

削除, 3-3

手動で作成, 3-3

電子ユーティリティ・デモ, 7-8

## と

---

ドキュメント

アクセス, ix

インストール先, ix

ナビゲーション・ファイル, ix

特殊 Oracle Server アカウント, 1-13

特殊アカウント, 1-13

特殊グループ

root, 1-14

## に

---

認可, 1-16

## は

---

配布ソフトウェア, 1-3

パスワード, 1-16

バッファ・キャッシュ

チューニング, 2-7

パフォーマンス監視ツール

AIX, 2-2

AIX Performance Toolbox, 2-2

Oracle, 2-4

## ふ

---

### ファイル

- coraenv, 1-7
- dbhome, 1-7
- oraenv, 1-7
- oratab, 1-27
- root.sh, 1-7

### 複数のシグナル・ハンドラ, 5-17

### プリコンパイラ

- ireclen および oreclen の値, 5-3
- 値, 5-3
- 大文字から小文字への変換, 5-3
- 概要, 5-2
- シグナル, 5-17
- デモンストレーションの実行, 1-26
- ベンダー提供のデバッグ・プログラム, 5-3

### プリコンパイラ構成ファイル, 5-2

### ブロック・サイズ, 2-10

### プロトコル, 6-6

- ADDRESS 指定, 6-6

## へ

---

### ページング

- 過度, 2-7, 2-9
- 不十分, 2-9

### ヘルプ機能, 3-4

## ま

---

### マスター・エージェントの構成, 6-13

### マニュアル

- 参照, x

### マルチスレッド・サーバー, 6-4

## め

---

### メモリー

- 仮想, 1-11
- 競合, 2-7
- 使用量の見積り, 1-11

### メモリーの使用量, 1-11

## ゆ

---

### ユーザー・プログラム

- Pro\*COBOL, 5-9

- SQL\*Module for Ada, 5-13

### ユーザー割込みハンドラ, 5-17

### ユーティリティ（電子）デモ, 7-8

## り

---

### リスナー

- 起動, 4-12

- 停止, 4-10

### リソースの制限, 1-12

### リモート接続パラメータ

- OS\_AUTHENT\_PREFIX, 1-17

- REMOTE\_OS\_AUTHENT, 1-17

- REMOTE\_OS\_ROLES, 1-17

### リモート・パスワード, 1-16

## れ

---

### レトロフィット・デモ, 7-8

