

# Oracle8i

for AIX-Based Systems 管理者リファレンス

リリース 8.1.5

おもなトピック：

Oracle8i の管理

AIX での Oracle8i のチューニング

SQL\*Plus の管理

AIX での 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

Net8 の構成

Oracle データ・オプション・デモの実行

最適フレキシブル・アーキテクチャ

ORACLE

---

Oracle8i for AIX-Based Systems 管理者リファレンス リリース 8.1.5

部品番号 : A63021-1

第 1 版 1999 年 7 月 (第 1 刷)

原本名 : Oracle8i Administrator's Reference for AIX-Based Systems, Release 8.1.5

原本部品番号 : A67729-01

Copyright © 1999 Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

プログラムの使用、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。

当ソフトウェア（プログラム）のリバース・エンジニアリング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 Legend が適用されます。

Restricted Rights Legend

Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication and disclosure of the Programs 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 52.227-14, Rights in Data -- General, including Alternate III (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

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

---

# 目次

はじめに .....	ix
用途 .....	ix
対象読者 .....	ix
Oracle8i および Oracle8i Enterprise Edition .....	ix
このマニュアルで使用する表記規則 .....	x
コマンドの構文 .....	x
参照マニュアル .....	xi
Oracle サービスおよびサポート .....	xi
1 Oracle8i の管理 .....	
環境設定 .....	1-2
環境変数の表示 .....	1-2
現行セッションでの変数値の設定およびエクスポート .....	1-2
共通の環境設定 .....	1-2
データベースの例 .....	1-3
Oracle8i の環境変数 .....	1-4
UNIX における Oracle 環境変数 .....	1-5
Oracle8i で使用する UNIX 環境変数 .....	1-7
システム時刻の設定 .....	1-8
初期化パラメータ .....	1-9
initSID.ora ファイルのカスタマイズ .....	1-9
デフォルトの初期化パラメータの値 .....	1-12
データベースの制限 .....	1-13
特殊アカウントおよび特殊グループの管理 .....	1-14
セキュリティの管理 .....	1-15

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 .....	1-16
データベース・ファイルのセキュリティ .....	1-17
セキュリティとリモート・パスワード .....	1-17
ログイン・ホーム・ディレクトリの管理 .....	1-19
Oracle8i のメモリー使用量の見積り .....	1-20
サーバー・リソースの制限 .....	1-22
システム・グローバル領域の制御 .....	1-22
SGA のサイズ制限 .....	1-22
SGA のサイズの計算 .....	1-23
デモンストレーションの作成と実行 .....	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のロード .....	1-26
PL/SQL デモンストレーションの実行 .....	1-26
ネットワーク実行ファイルの再リンク .....	1-28

## 2 AIX での Oracle8i のチューニング

AIX パフォーマンス・ツール .....	2-2
AIX Base Operation System ツール .....	2-2
AIX Performance Toolbox .....	2-2
AIX System Management Interface Tool .....	2-3
Oracle パフォーマンス・ツール .....	2-4
Utlbstat / Utilestat による統計の収集 .....	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
Oracle8i のチューニング .....	2-6
Oracle8i のチューニング順序 .....	2-6
メモリーおよびページング .....	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 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 バッファと SQL*Loader .....	2-11
<b>ディスク I/O の問題</b> .....	2-12
索引と表の分離 .....	2-12
REDO ログの専用ディスク・デバイスへの配置 .....	2-12
ホット・ファイルの他のディスクへの移動 .....	2-12
ホット・ファイルの I/O の削減 .....	2-12
データベースの過剰な断片化のチェック .....	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 パーティションの使用 .....	2-16
非同期 I/O の使用方法 .....	2-18
I/O スレーブ .....	2-19
DB_FILE_MULTIBLOCK_READ_COUNT の使用 .....	2-20
RAID 機能の使用 .....	2-20
後書きの使用 .....	2-20
順次ファイルの先読みのチューニング .....	2-21
ディスク I/O の歩調合せ .....	2-21
ディスク配置の考慮事項 .....	2-22
リモート I/O の最小化 .....	2-22
VSD キャッシュ・バッファを使用しない .....	2-22
<b>CPU のスケジューリングとプロセスの優先順位</b> .....	2-23
プロセスのランタイム・スライスの変更 .....	2-23
SMP マシンでのプロセッサ・バインディング .....	2-23
マルチ・プロセッサ・システムでのスピン・カウン트의チューニング .....	2-26
<b>その他のチューニングのヒント</b> .....	2-26
UDP のチューニング .....	2-26
RAW デバイスのバックアップ .....	2-27
パラレル回復の使用方法 .....	2-27
Pro*C プログラムのコンパイル .....	2-27
<b>関連資料</b> .....	2-29
Oracle マニュアル .....	2-29
IBM マニュアル .....	2-29

### 3 SQL\*Plus の管理

SQL*Plus の管理 .....	3-1
設定ファイル .....	3-1
Site Profile .....	3-1
User Profile .....	3-1
PRODUCT_USER_PROFILE 表 .....	3-2
デモンストレーション表 .....	3-2
ヘルプ機能 .....	3-3
SQL*Plus の使用 .....	3-4
SQL*Plus からのシステム・エディタの使用 .....	3-4
エディタの順序設定 .....	3-5
_editor オプションの設定 .....	3-5
環境変数の設定 .....	3-5
デフォルト設定 .....	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
制限事項 .....	3-6
COPY コマンド .....	3-6
ウィンドウのサイズ変更 .....	3-7
リターン・コード .....	3-7

### 4 AIX での 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

Oracle プリコンパイラの概要 .....	4-2
プリコンパイラ実行ファイルの再リンク .....	4-2
プリコンパイラ構成ファイル .....	4-3
すべてのプリコンパイラに共通の問題 .....	4-3
参照マニュアル .....	4-4
Pro*C/C++ .....	4-4
Pro*C/C++ の管理 .....	4-4
Pro*C/C++ の使用 .....	4-4
Pro*COBOL .....	4-6
Pro*COBOL の管理 .....	4-7
環境変数 .....	4-7

Pro*COBOL の使用 .....	4-7
<b>Pro*FORTRAN</b> .....	4-11
Pro*FORTRAN の管理 .....	4-11
Pro*FORTRAN の使用 .....	4-11
<b>SQL*Module for Ada</b> .....	4-13
SQL*Module for Ada の管理 .....	4-13
SQL*Module for Ada の使用 .....	4-13
<b>Oracle コール・インタフェース</b> .....	4-15
Oracle コール・インタフェースの使用 .....	4-15
<b>Oracle プリコンパイラと Oracle コール・インタフェースのリンクおよび Make ファイル</b> .....	4-17
カスタム Make ファイル .....	4-17
未定義シンボル .....	4-17
スレッドのサポート .....	4-18
<b>Oracle ライブラリを使用した静的リンクおよび動的リンク</b> .....	4-18
シグナル・ハンドラの使用方法 .....	4-19
シグナル .....	4-19
<b>XA 機能</b> .....	4-22

## 5 Net8 の構成

参照ドキュメント .....	5-2
README ファイルへの補足情報 .....	5-2
主な Net8 製品および特徴 .....	5-2
Net8 ファイルおよびユーティリティ .....	5-2
Oracle Connection Manager .....	5-4
マルチスレッド・サーバー .....	5-4
<b>Net8 プロトコルのサポート</b> .....	5-5
ADDRESS 指定 .....	5-5
<b>BEQ プロトコル</b> .....	5-6
BEQ の ADDRESS 指定 .....	5-7
<b>IPC プロトコル</b> .....	5-8
IPC の ADDRESS 指定 .....	5-8
<b>RAW プロトコル</b> .....	5-9
<b>TCP/IP プロトコル</b> .....	5-9
TCP/IP の ADDRESS 指定 .....	5-10
<b>SPX/IPX プロトコル</b> .....	5-11

<b>ntisb8dm</b> 同報通信デーモン .....	5-11
<b>ntspxctl</b> ユーティリティ .....	5-11
SPX/IPX プロトコルのコマンドの要約 .....	5-13
<b>getname</b> コマンド .....	5-13
SPX/IPX の ADDRESS 指定 .....	5-15
<b>APPC/LU6.2</b> プロトコル .....	5-16
APPC/LU6.2 の ADDRESS 指定 .....	5-16
<b>Net8 Naming Supprt</b> .....	5-17
<b>Oracle Enterprise Manager (OEM)</b> .....	5-17
<b>Oracle SNMP</b> での <b>Oracle Intelligent Agent</b> の構成 .....	5-18
<b>Oracle Advanced Security</b> .....	5-19

## 6 Oracle データ・オプション・デモの実行

参照ドキュメント .....	6-2
<b>Oracle8i interMedia</b> .....	6-2
Text .....	6-2
Audio、Video および Image .....	6-3
Locator .....	6-7
<b>Oracle8i Time Series</b> デモ .....	6-8
<b>Oracle8i Visual Information Retrieval</b> .....	6-9
<b>Oracle8i Spatial</b> .....	6-11

## A 最適フレキシブル・アーキテクチャ

<b>最適フレキシブル・アーキテクチャ (OFA)</b> .....	A-2
OFA に準拠したデータベースの特長 .....	A-2
<b>UNIX で適用される OFA</b> .....	A-5
マウント・ポイントの名前 .....	A-5
ディレクトリの名前 .....	A-5
ファイルの名前 .....	A-6
表領域の名前 .....	A-9
OFA 構造に基づいた Oracle ファイル .....	A-9
OFA ファイルのマッピング .....	A-10
RAW デバイスのサイズ .....	A-11
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-11



ディレクトリ構造 .....	A-12
----------------	------

## 索引



---

# はじめに

## 用途

このマニュアルでは、Oracle8i を管理およびチューニングするために必要な AIX 固有の情報を説明します。このマニュアルの内容は、Oracle8i ドキュメンテーション・ライブラリ・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 システムでの Oracle8i の管理者を対象としています。

## Oracle8i 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 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

クーリエ体	クーリエ体は、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] キーを押す）。ただし、コマンド構文の中で使用される場合、大カッコは別の意味になります。
イタリック体	イタリック体は、変数を表します。
大文字	大文字は、SQL の予約語、初期化パラメータまたは環境変数を表します。

UNIX では大文字と小文字が別のものとして認識されるため、このマニュアルで使用する表記規則は、他の Oracle 製品のマニュアルとは多少異なります。

## コマンドの構文

コマンド構文は、クーリエ体で表します。コマンドで使用する構文は、次のとおりです。

バックスラッシュ \	バックスラッシュは、このマニュアルのページの 1 行に入りきらない場合に使用しています。このマニュアルで記載され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。 <pre>dd if=/dev/rdisk/c0t1d0s6 of=/dev/rst0 bs=10b \ count=10000</pre>
中カッコ {}	中カッコは、必ず選択する項目を表します。 <pre>.DEFINE {macro1}</pre>
大カッコ []	大カッコは、任意に選択する項目を表します。 <pre>cvtcrt termname [outfile]</pre> <p>ただし、本文中で使用される場合、大カッコは別の意味になります。</p>
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。 <pre>CHKVAL fieldname value1 value2 ... valueN</pre>
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。 <pre>library_name</pre>
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。 <pre>SIZE filesize [K M]</pre>

## 参照マニュアル

本番データベース・システム用の拡張構成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 管理者ガイド』  
Oracle8i に関するさまざまなタスク（データベースの作成、データベース・オブジェクトの管理、ユーザーの作成など）を行う際に、まずお読みください。
- 『Oracle8i Net8 管理者ガイド』
- 『Oracle8i チューニング』

Oracle リレーショナル・データベース管理システムに慣れていない場合は、インストレーションを開始する前に『Oracle8i 概要』の第 1 章をお読みください。

## Oracle サービスおよびサポート

Oracle 製品およびグローバル・サービスについては、インターネット上の <http://www.oracle.com> で参照できます。

次に、選択したサービスの URL を示します。

### Oracle カスタマ・サポート

技術サポートの連絡先については、<http://www.oracle.com/support> にアクセスしてください。テンプレートが用意されていますので、連絡する前に、現状の問題に関する情報を準備するのに役立ててください。また、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。

カスタマ・サポートの連絡先については、<http://www.oracle.com/support> にアクセスしてください。

### 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ついては、<http://education.oracle.com> にアクセスしてください。



---

## Oracle8i の管理

- 環境設定
- Oracle8i の環境変数
- 初期化パラメータ
- 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の管理
- セキュリティの管理
- ログイン・ホーム・ディレクトリの管理
- Oracle8i のメモリー使用量の見積り
- サーバー・リソースの制限
- システム・グローバル領域の制御
- デモンストレーションの作成と実行
- ネットワーク実行ファイルの再リンク

## 環境設定

この項では、Oracle8i システムの共通の環境を構築する方法について説明します。

### 環境変数の表示

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
```

### 現行セッションでの変数値の設定およびエクスポート

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。

### 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 (C シェルの場合は coraenv) 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、データベースを /usr/oracle から /usr1/oracle へ移動する必要があるとします。共通の環境設定のルーチンがない場合は、ユーザー起動ファイルを個々に更新する必要があります。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。そのため、oraenv の変更のみが必要となります。



## ローカル bin ディレクトリ

oraenv (または coraenv) 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、ローカル bin ディレクトリに入れることによって、パスが別の ORACLE\_HOME を示すように変更した場合でも、oraenv (または coraenv) は有効のままです。ローカル bin ディレクトリは、次のインストレーションで実行される 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、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。必ず oraenv コマンド・ファイルのフルパス名を入力してください。次に例を示します。

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

## データベースの例

次に、ローカル bin ディレクトリが /usr/lbin で、本番データベースが PROD の場合の例を示します。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。

次の例では、oraenv が実行された後、ORAENV\_ASK はデフォルトの Yes にリセットされます。これによって、次に oraenv が実行されたときに、別の ORACLE\_SID を入力するプロンプトが表示されます。

## 単一インスタンス

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
```

前述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
```

前述の行を次の行に置き換えます。

```
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

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。

```
PATH=${PATH}:/usr/lbin
ORACLE_SID=PROD
export PATH ORACLE_SID
SIDLIST= 'awk -F: '/^[^#]/ {printf "%s ", $1}' /etc/oratab
echo "SIDS on this machine are $SIDLIST"
ORAENV_ASK=
. /usr/bin/oraenv
```

C シェルの場合、次のように入力します。

```
setenv PATH ${PATH}:/usr/local/bin
setenv ORACLE_SID PROD
set sidlist = 'awk -F: '/^[^#]/ {printf "%s ", $1}' /etc/oratab
echo "SIDS on this machine are $sidlist"
unset ORAENV_ASK
source /usr/lbin/coraenv
```

## Oracle8i の環境変数

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。これらの変数のいくつかは、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。環境変数の一覧は、『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。

## UNIX における Oracle 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

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	出力に使用する言語とキャラクタ・セットを指定します。設定値の詳細は、『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。
	構文	<i>language_territory.characterset</i>
	例	french_france.we8dec
ORA_NLS33	機能	言語とキャラクタ・セットが保存されているディレクトリを示します。
	設定値	\$ORACLE_HOME/ocommon/nls/admin/data
ORACLE_BASE	機能	OFA に準拠した Oracle データベースのディレクトリ構造の基本となるディレクトリを指定します。
	構文	<i>directory_path</i>
	例	/mount_point/app/oracle
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	<i>directory_path</i>
	例	\$ORACLE_BASE/product/8.1.5
ORACLE_PATH	機能	SQL*Plus などの Oracle アプリケーションが使用するファイルの検索パス名を指定します。指定しないと、アプリケーションは現行のディレクトリに読み込みおよび書き込みを行います。
	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	/u01/oracle/adhoc/8.1.5/bin:.
		<b>注意:</b> 最後にピリオドを付けることによって、検索パスに現行の作業ディレクトリが追加されます。
ORACLE_SID	機能	Oracle のシステム ID を指定します。
	構文	英字で始まる英数文字列を指定します。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 インストレーション・ガイド』を参照してください。
	例	SAL1

表 1-1 UNIX における Oracle8i 環境変数

変数	項目	定義
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 <code>set -x</code> フラグがオンの状態で実行されます。
	許容値	T など
ORAENV_ASK	機能	(c)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 など
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	<i>directory_path</i>
	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（たとえば、arch、pmon、dbwr など）は設定しないでください。

---

## ORACLE\_HOME および ORACLE\_SID の略称

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.ora' size 2M
```

@ 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@.ora'
```

## Oracle8i で使用する UNIX 環境変数

表 1-2 に、Oracle8i で使用する UNIX 環境変数の機能、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

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。
DISPLAY	機能	X ベースのツールで使います。入出力に使用するディスプレイ・デバイスを指定します。詳細は、各ベンダーの X-Windows 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、マシン名です (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 つの場合は、番号は 0 です。
	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。
LD_LIBRARY_PATH	機能	AIX で、Java アプリケーションの実行時に使います。
	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	/usr/lpp/J1.1.6/:\$ORACLE_HOME/lib
LIBPATH	機能	AIX で、共有ライブラリ・ローダーが実行時に共有オブジェクト・ライブラリを検索するために使います。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	/usr/dt/lib:\$ORACLE_HOME/lib
PATH	機能	シェルが実行プログラムを配置するために使います。 \$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	/bin:/usr/bin:/usr/sbin: /usr/bin/X11:\$ORACLE_HOME/bin:\$HOME/bin:.
		<b>注意:</b> 最後にピリオドを付けることによって、検索パスに現行の作業ディレクトリが追加されます。

表 1-2 Oracle8i で使用する UNIX 環境変数

変数	項目	定義
PRINTER	機能	AIX システム用の、デフォルト・プリンタを指定します。
	構文	printer_name
	例	docqms
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	/bin/sh、/bin/csh、/bin/ksh または AIX に付属のその他のコマンド・インタプリタ
	例	/bin/sh
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	vt100
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。
	構文	directory_path
	例	/u02/oracle/tmp
XENVIRONMENT	機能	X-Window システムのリソース定義を含むファイルを指定します。詳細は、X-Window のドキュメントを参照してください。

## システム時刻の設定

TZ 変数には、タイム・ゾーンを設定します。TZ 変数によって、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。

**警告：** できるだけ、個々に TZ 値を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。

## 初期化パラメータ

初期化パラメータによって、システムを構成およびチューニングできます。この項では、次のことを説明します。

- Oracle8i インスタンスの `initSID.ora` ファイルでの初期化パラメータのカスタマイズ
- デフォルト初期化パラメータの事前設定

オプションの初期化パラメータのほとんどは、共通の Oracle8i ドキュメントで参照できます。

**参照：** 詳細は、『Oracle8i 管理者ガイド』および『Oracle8i チューニング』を参照してください。

## initSID.ora ファイルのカスタマイズ

この項では、Oracle8i ソフトウェアとともに提供されるデフォルトの `initSID.ora` ファイルについて説明します。Oracle Universal Installer (OUI) は、`$ORACLE_BASE/admin/db_name/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 インストレーションをカスタマイズできます。

`initSID.ora` パラメータには、どんなサイズのインストレーションにも共通の設定値があります。インストレーションのサイズに応じて値を変更する必要があるパラメータに対しては、小、中、大の3つのパターンを提供しています。サンプル `initSID.ora` ファイルでは、インストレーション・サイズによって設定が異なるパラメータが示されています。インストレーションには適用しない設定があれば、行頭に「#」を付けてコメントにできます。

表 1-3 に、`initSID.ora` ファイルに用意されている3つのパターンに対するおよその SGA サイズを示します。

**表 1-3 サンプル `initSID.ora` ファイルのブロック・サイズおよび SGA サイズ**

インストレーション/データベース・サイズ			
ブロック・サイズ	小	中	大
2KB	4500KB	6800KB	17000KB
4KB	5500KB	8800KB	21000KB

### サンプル `initSID.ora` ファイル

サンプル `initSID.ora` ファイルは、RDBMS インストレーションをカスタマイズする際に使用すると便利です。

## 初期化パラメータ

---

```
# replace DEFAULT with your database name
db_name=DEFAULT

db_files = 80                                # SMALL
# db_files = 400                             # MEDIUM
# db_files = 1500                            # LARGE

db_file_multiblock_read_count = 8            # SMALL
# db_file_multiblock_read_count = 16         # MEDIUM
# db_file_multiblock_read_count = 32         # LARGE

db_block_buffers = 100                      # SMALL
# db_block_buffers = 550                     # MEDIUM
# db_block_buffers = 3200                    # LARGE

shared_pool_size = 3500000                  # SMALL
# shared_pool_size = 5000000                 # MEDIUM
# shared_pool_size = 9000000                 # LARGE

log_checkpoint_interval = 10000

processes = 50                              # SMALL
# processes = 100                           # MEDIUM
# processes = 200                           # LARGE

parallel_max_servers = 5                    # SMALL
# parallel_max_servers = 4 x (number of CPUs) # MEDIUM
# parallel_max_servers = 4 x (number of CPUs) # LARGE

log_buffer = 32768                          # SMALL
# log_buffer = 32768                         # MEDIUM
# log_buffer = 163840                        # LARGE

# audit_trail = true                        # if you want auditing
# timed_statistics = true                  # if you want timed statistics
max_dump_file_size = 10240                  # limit trace file size to 5 Meg each

# Uncommenting the lines below will cause automatic archiving if archiving has
# been enabled using ALTER DATABASE ARCHIVELOG.
# log_archive_start = true
# log_archive_dest = disk$rdlms:[oracle.archive]
# log_archive_format = "T%TS%S.ARC"

# If using private rollback segments, place lines of the following
# form in each of your instance-specific init.ora files:
# rollback_segments = (name1, name2)
```



```
# If using public rollback segments, define how many
# rollback segments each instance will pick up, using the formula
#   # of rollback segments = transactions / transactions_per_rollback_segment
# In this example each instance will grab 40/5 = 4:
# transactions = 40
# transactions_per_rollback_segment = 5

# Global Naming -- enforce that a dblink has same name as the db it
# connects to:
# global_names = TRUE

# Edit and uncomment the following line to provide the suffix that will be
# appended to the db_name parameter (separated with a dot) and stored as the
# global database name when a database is created. If your site uses
# Internet Domain names for e-mail, then the part of your e-mail address after
# the '@' is a good candidate for this parameter value.

# db_domain = us.acme.com
# global database name is db_name.db_domain

#_db_block_cache_protect = true                # memory protect buffers
# event = "10210 trace name context forever, level 2" # data block checking
# event = "10211 trace name context forever, level 2" # index block checking
# event = "10235 trace name context forever, level 1" # memory heap checking
# event = "10049 trace name context forever, level 2" # memory protect cursors

# define parallel server (multi-instance) parameters
# ifile = ora_system:initsps.ora

# define two control files by default
control_files = (ora_control1, ora_control2)

# Uncomment the following line if you wish to enable the Oracle Trace product
# to trace server activity. This enables scheduling of server collections
# from the Oracle Enterprise Manager Console.
# Also, if the oracle_trace_collection_name parameter is non-null,
# every session will write to the named collection, as well as enabling you
# to schedule future collections from the console.

# oracle_trace_enable = TRUE

# Uncomment the following line if you want to use some of the new 8.1
# features. Please remember that using them may require some downgrade
# actions if you later decide to move back to 8.0.

# compatible = 8.1.0
```

## デフォルトの初期化パラメータの値

表 1-4 に、AIX での初期化パラメータのデフォルト値を示します。init`sid`.ora ファイルのこれらのパラメータに別の値を設定しなかった場合、すべての Oracle8i インスタンスでこれらの値が使用されます。init`sid`.ora ファイルには、デフォルトとは異なる値のパラメータのみを設定してください。

システムでの現在のパラメータ値を表示するには、SQL\*Plus を使用して SQL 文 SHOW PARAMETERS を実行します。

**参照：** 詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。

表 1-4 デフォルトの初期化パラメータ

パラメータ	デフォルト値
BACKGROUND_DUMP_DEST	\$ORACLE_HOME/rdbms/log
BITMAP_MERGE_AREA_SIZE	1048576
COMMIT_POINT_STRENGTH	1
CONTROL_FILES	\$ORACLE_HOME/dbs/ctrl@.dbf (@ は ORACLE_SID)
CREATE_BITMAP_AREA_SIZE	8388608
DB_BLOCK_BUFFERS	32768
DB_BLOCK_SIZE	4096
DB_FILE_DIRECT_IO_COUNT	64 (最大値は 1048576)
DB_FILE_MULTIBLOCK_READ_COUNT	8 (許容値は 1 ~ 128。ただし、 DB_BLOCK_BUFFERS の 1/4 以下で設定 してください。)
DB_FILES	200 (最大値は 2000000)
DISTRIBUTED_TRANSACTIONS	10
HASH_AREA_SIZE	0
HASH_MULTIBLOCK_IO_COUNT	1
LOG_ARCHIVE_BUFFER_SIZE	64
LOG_ARCHIVE_BUFFERS	4 (最大値は 128)
LOG_ARCHIVE_DEST	なし
LOG_ARCHIVE_FORMAT	"%t_%.s.dbf"
LOG_BUFFER	524288

表 1-4 デフォルトの初期化パラメータ

パラメータ	デフォルト値
LOG_CHECKPOINT_INTERVAL	0
LOG_SMALL_ENTRY_MAX_SIZE	80
MTS_LISTENER_ADDRESS	なし
MTS_MAX_DISPATCHERS	5
MTS_MAX_SERVERS	20
MTS_SERVERS	0
NLS_LANGUAGE	AMERICAN
NLS_TERRITORY	AMERICA
OBJECT_CACHE_MAX_SIZE_PERCENT	10
OBJECT_CACHE_OPTIMAL_SIZE	102400
OPEN_CURSORS	50
OS_AUTHENT_PREFIX	ops\$
PROCESSES	30
SHARED_POOL_SIZE	8000K
SORT_AREA_SIZE	65536
SORT_READ_FAC	5
SORT_SPACEMAP_SIZE	512
USER_DUMP_DEST	\$ORACLE_HOME/rdbms/log

## データベースの制限

表 1-5 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。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 1-5 制御ファイルのパラメータ作成

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

## 特殊アカウントおよび特殊グループの管理

データベース管理者 (DBA) には、Oracle Server が必要とする特殊アカウントに関する知識と経験が必要で、それらのアカウントが適切なグループに属するように設定する必要があります。表 1-6 に UNIX アカウント、表 1-7 に Oracle Server アカウントを示します。また、表 1-8 に特殊グループ・アカウントを示します。

表 1-6 UNIX アカウント

<i>oracle</i>	<i>oracle</i> 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 接続するための DBA 権限が必要です。 <i>oracle</i> ソフトウェア所有者は、スーパーユーザーにはなりません。
<i>root</i>	<i>root</i> 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 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・ディクショナリの実表すべてを所有します。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 ユーザーでその他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、root ユーザーなどの 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 接続するための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。デフォルトのグループ ID は dba です。
oinstall グループ	ORACLE_HOME では、Oracle をインストールするすべてのユーザーは 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。現行のプライマリ・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	ユーザーが作成できるオプションの 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。
dba 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、hagsuser グループのメンバーになれるのは oracle ユーザーです。

## セキュリティの管理

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・アーキテクチャによって、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。

## グループとセキュリティ

Oracle8i データベースのセキュリティを高めるため、オペレーティング・システム・レベルでユーザー・グループを作成します。グループは、UNIX ファイルの /etc/group が制御します。Oracle プログラムは、セキュリティを高めるために 2 つのグループに分けられます。すべてのユーザー（UNIX 用語では *other*）がアクセスできる実行プログラムおよび DBA 専用の実行プログラムです。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。

- Oracle8i をインストールする前に、データベース管理者のグループ dba を作成し、このグループに root および *oracle* ソフトウェア所有者 ID を割り当てます。dba が実行できるプログラムには、権限 710 のみが与えられています。
- 別のユーザー・グループを追加して、制限された UNIX ユーザーのサブセットが Oracle8i へアクセスできるようにします。Oracle ユーティリティ、SQL\*Plus など、一般的に実行できるプログラムは、このグループから実行できなければなりません。ユーティリティの権限を 710 に設定して、このグループ（ただし、*other* を除く）に実行権限を与えます。そのようなユーザー・グループを作成した場合、dba グループのメンバーも、その中に含めてください。
- *other* が実行できるプログラムに、権限 711 を与えます。この権限は、データベース・セキュリティに影響しないプログラムにのみ与えてください。

データベース管理者のグループには任意の名前を割り当てられますが、デフォルトのグループ名は dba で、このマニュアルでもその名前を使用しています。同じ ORACLE\_HOME でデータベースが複数ある場合（このような構成はお薦めしません）は、それらのデータベース管理者のグループは同じでなければなりません。

---

**注意：** *oracle* ソフトウェア所有者および root ユーザーは dba グループに属しますが、*oracle* ソフトウェア所有者は root グループのメンバーにはなりません。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。

---

## Server Manager コマンドのセキュリティ

SQL\*Plus がない場合は、Server Manager を使用して SQL 問合せを実行できます。ただし、Server Manager へのアクセスを割り当てるには注意が必要です。*oracle* ソフトウェア所有者および dba グループのユーザーには特別なオペレーティング・システム権限が与えられているため、*oracle* ソフトウェア所有者および dba グループのユーザーのみが、システムによって権限が与えられた次の文にアクセスできます。

- STARTUP
- SHUTDOWN
- CONNECT INTERNAL

---

**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、必要な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なければなりません。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権限 0600 に設定します。これによって、所有者のみに読みおよび書き込み (rw) 権限が与えられ、グループや他のユーザーに書き込み権限は与えられません。

*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の所有者でなければなりません。セキュリティを高めるために、グループおよび他のユーザーの読み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* 実行ファイルのユーザー ID に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
```

## セキュリティとリモート・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・コンピュータなどのリモート・マシンからデータベースにアクセスしたり、データベースを管理したりできます。ユーザー検証は、*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行われます。オペレーティング・システム・アカウントが使用するシステムでは、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。

### *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* を実行するには、次のように入力します。

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

表 1-9 に、*orapwd* の構文を示します。

表 1-9 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 <i>orapwsid</i> 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 ALTER USER 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。

**注意：** ユーザーの最大数を増やさなければならない場合は、新しいパスワード・ファイルを作成する必要があります。そのため、*max\_users* は、必要な数より多く設定してください。

## orapwd の例

```
$ orapwd file=/u01/app/oracle/product/8.1.5/dbs/orapw815 \
password=rsdb3t4 entries=30
```

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

## 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、ネットワーク上の DBA 権限を持つ 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 1-10 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 1-10 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。

**注意：** REMOTE\_OS\_AUTHENT を true に設定すると、リモート・マシンの dba グループのメンバーであるユーザーは、パスワードなしで INTERNAL として接続できます。

## ログイン・ホーム・ディレクトリの管理

ログイン・ホーム・ディレクトリを参照するプログラムを変更せずにそのディレクトリを追加または移動するには、次のようにします。

- 明示的なパス名は、そのパス名を保管するファイル（/etc/passwd、/etc/oratab など）で参照します。
- グループのメンバーは、/etc/group ファイルで参照します。

次のいずれかの方法でユーザーのホーム・ディレクトリを参照できるため、主要な参照ファイル以外にパス名を記録しておく必要はありません。

- C シェルおよび Korn シェルのユーザーは、`~login` を使用してユーザーのホーム・ディレクトリを参照できます。
- Bourne シェルのユーザーは、ユーザーのホーム・ディレクトリを参照するための簡単なプログラムを作成できます。この項で示すサンプル `lhd` スクリプトを参照してください。

同じように、グループのメンバーも /etc/group で参照できます。この項で示すサンプル `grp` スクリプトを参照してください。

### サンプル `lhd` スクリプト

```
#!/bin/sh
#
# lhd - print login home directory name for a given user
#
# SYNTAX
# lhd [login]
#
prog='basename $0'
```

```
if [ $# -eq 0 ] ; then
    login='whoami'

elif [ $# -eq 1 ] ; then
    login=$1
else
    echo "Usage: $prog login" >&2

    exit 2
fi
awk -F: '$1==login {print $6}' login=$login /etc/passwd
```

### サンプル grpx スクリプト

```
#!/bin/sh
# grpx - print the list of users belonging to a given group
#
prog='basename $0'
if [ $# -ne 1 ] ; then
    echo "Usage: $prog group" >&2
    exit 2
fi
g=$1
# calculate group id of g
gid='awk -F: '$1==g {print $3}' g=$g /etc/group'
# list users whose default group id is gid
u1='awk -F: '$4==gid {print $1}' gid=$gid /etc/passwd'
# list users who are recorded members of g
u2='awk -F: '$1==g {gsub(/,/," "); print $4}' g=$g /etc/group'
# remove duplicates from the union of the two lists
echo $u1 $u2 | tr " " "\012" | sort | uniq | tr "\012" " "
```

#### 例 1-1 lhd スクリプトおよび grpx スクリプトの例

次に、管理者が、スケルトン .profile ファイルをグループの各メンバーのホーム・ディレクトリに反映させる例を示します。clerk グループのメンバー・リストを変更しても、コードを修正する必要はありません。

```
$ for u in `grpx clerk` ; do
> cp /etc/skel/.profile `lhd $u`
> done
```

## Oracle8i のメモリー使用量の見積り

Oracle8i を起動する前に、Oracle8i のメモリー使用量を知っておく必要があります。必要なメモリーの使用量を把握しておくと、システム上のユーザー数を決定するのに役立ちます。

また、必要な物理メモリーおよびスワップ領域を決定するのに役立ちます。次の式を使用して、必要なメモリーを見積もります。

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

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。

SGA サイズを見積もるには、1-23 ページの「SGA のサイズの計算」を参照してください。

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 *bss*）を見積もるには、`size` コマンドを使用します。すべての Oracle 実行テキストは常に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ときに、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 Oracle バックグラウンド・プロセスの項目を検索します。各項目について、SZ 列を合計します。

**参照：** `ps` コマンドで使用できるスイッチのリストの詳細は、AIX 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、ページでのプロセス・サイズを返します。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 (/etc/security/limits ファイルおよび getrlimit(2) の man ページを参照)。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、どの Oracle8i 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。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。詳細は、AIX のシステム管理者にお問い合わせください。

ファイル・システムで 1GB 以上の Oracle データ・ファイルを使用する場合は、すべてのユーザーに対するデフォルトのファイル・サイズ制限を高くする必要があります。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 ID 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お問い合わせください。

## システム・グローバル領域の制御

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。

### SGA のサイズ制限

AIX での単一共有メモリー・セグメントの最大サイズは 256MB です。プロセスでは、最大 10 個の連続共有メモリー・セグメントを割り当てられるので、AIX での最大 SGA サイズは 2.5GB です。

---

**注意：** Oracle を実行するときに、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、Oracle を実行できません。

---

SGA のサイズは、initsid.ora ファイルの次のパラメータによって決定します。

- DB\_BLOCK\_BUFFERS
- DB\_BLOCK\_SIZE
- JAVA\_POOL\_SIZE
- LARGE\_POOL\_SIZE
- SORT\_AREA\_SIZE
- SHARED\_POOL\_SIZE

これらのパラメータの値は、十分注意して設定してください。値が大きすぎると、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

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
```

次のいずれかの方法で、SGA サイズを確認できます。

- 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。
- データベース・システムを起動するときに、SGA のサイズも参照できます。ヘッダー「Total System Global Area」の横に SGA サイズが表示されます。

## デモンストレーションの作成と実行

### SQL\*Loader のデモンストレーション

SQL\*Loader のデモンストレーションには、次の要件があります。

- 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

デモンストレーションを作成して実行するには、次の手順に従います。

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。次のように入力して、`scott/tiger` でコマンド行から SQL\*Plus を起動します。

```
$ sqlplus scott/tiger ulcasen.sql
```

2. 次のように入力して、`scott/tiger` でコマンド行からデモンストレーションを起動します。

```
$ sqlldr scott/tiger ulcasen.ctl
```

`scott/tiger` で、次の順に SQL\*Loader のデモンストレーションを実行します。

- ulcase1: 前述の手順 1 ～ 2 を実行します。
- ulcase2: デモンストレーションを起動するために、手順 2 を実行します (ulcase2.sql スクリプトを実行する必要はありません)。
- ulcase3: 前述の手順 1 ～ 2 を実行します。
- ulcase4: 前述の手順 1 ～ 2 を実行します。
- ulcase5: 前述の手順 1 ～ 2 を実行します。
- ulcase6: scott/tiger で ulcase6.sql スクリプトを実行し、コマンド行に次のように入力します。

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

- ulcase7: scott/tiger で ulcase7s.sql スクリプトを実行し、コマンド行に次のように入力します。

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

例を実行した後、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・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。

### ファイル処理オプション

SQL\*Loader の制御ファイルには、次のファイル処理オプション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 1-11 ファイル処理オプション**

str	(デフォルト)。改行文字で終了するレコードを、1 度に 1 レコードずつ読み込むストリームを指定します。
fix	ファイルが <i>n</i> 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。 <i>n</i> は整数です。
var	ファイルが <i>n</i> 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。 <i>n</i> 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走査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。

---

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

---

### 改行文字の削除

改行文字をロードしないで、固定長レコードから改行文字を削除するには、制御ファイルの position(x:y) 関数を使用します。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 )
```

これによって、固定長レコードの 4 桁目にあるために、改行文字は削除されます。

## PL/SQL デモンストレーションのロード

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、plsql ディレクトリにあります。Oracle8i データベースを起動してマウントした状態で、次の操作を行ってください。

1. SQL\*Plus を起動し、ユーザー / パスワードが 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。

---

## PL/SQL デモンストレーションの実行

表 1-12 に、カーネルのデモンストレーションを示します。

**表 1-12 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			



表 1-13 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 1-13 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
-----------	------------	------------	------------

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 @ 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
```

1 つ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
```

現行のシェルから 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、ファイル listener.ora に次の行を追加します。

```
SC=(SID_NAME=extproc)(ORACLE_HOME=/u01/app/oracle/8.1.5)(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;
Statement processed.
SQL> connect scott/tiger
Connected.
SQL> create library demolib as '$ORACLE_HOME/plsql/demo/extproc.so';
Statement processed.
```

最後に、テストを実行します。

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

## ネットワーク実行ファイルの再リンク

シェル・スクリプトの再リンクによって、ネットワーク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・システムのアップグレード後、または「**relink network executables**」というエラー・メッセージが表示された場合に必要となります。

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 1-14 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、plsql、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

---

## AIX での Oracle8i のチューニング

この章では、AIX システムにおける Oracle8i データベース・サーバーのパフォーマンスおよびスケーラビリティを改善するツールおよびテクニックについて説明します。この章の情報と一緒に、汎用的なチューニング方法が記載された『Oracle8i チューニング』および『AIX Versions 4.x Performance Tuning Guide』を参照してください。また、Oracle チューニングを扱っているその他の一般的な Oracle ドキュメントも参照してください。

この章では、次のトピックについて説明します。

- AIX パフォーマンス・ツール
- Oracle 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とプロセスの優先順位
- その他のチューニングのヒント
- 関連資料

これらのトピックは読みやすいように並べられているだけで、特定の順序を示すものではありません。

使用している環境に伴う前述のチューニングに必要な作業量は、アプリケーションおよび作業負荷に応じて異なります。

## AIX パフォーマンス・ツール

ここでは、AIX オペレーティング・システムの標準またはオプション機能として使用可能なツールについて説明します。

### AIX Base Operation System ツール

AIX Base Operation System (BOS) には、履歴的に UNIX システムの一部であったツールやインプリメンテーション固有の AIX 機能を管理するのに必要なパフォーマンス・ツールがあります。主要な BOS 機能は、次のとおりです。

iostat	CPU および I/O 統計をレポートします。
lsattr	デバイスの属性を表示します。
lslv	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の情報を表示します。
netstat	ネットワーク関連のデータ構造の内容を表示します。
nfsstat	Network File System (NFS) および Remote Procedure Call (RPC) アクティビティの統計を表示します。
nice	プロセスの最初の優先順位を変更します。
no	ネットワーク・オプションを表示または設定します。
ps	プロセスの状態を表示します。
reorgvg	ボリューム・グループ内の物理的なパーティションの割当てを再編成します。
sar	システム・アクティビティ情報を収集し、レポートまたは記録します。
time	実行が終了するまでの経過時間、ユーザーおよびシステムの CPU 処理時間を出力します。
trace	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。
vmstat	仮想記憶アクティビティおよびその他のシステム統計をレポートします。

### AIX Performance Toolbox

AIX Performance Toolbox (PTX) には、システム・アクティビティをローカルおよびリモートで監視およびチューニングするツールがあります。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、構成内のさまざまなシステムからデータを収集し、表示します。PTX Agent は、xmserd というプログラムを使用して、データを収集して PTX Manager

に送信します。PTX Agent は、Performance Aide for AIX という独立した製品としても提供されています。

PTX および Performance Aide には、監視用とチューニング用のツールが個別に含まれています。

fdpr	特定の作業負荷につい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 ファイル・システムのアクティビティを監視し、レポートします。
fileplace	論理ボリュームまたは物理ボリューム内でのファイルのブロックの配置を表示します。
lockstat	カーネル・ロックの競合について統計を表示します。
lvedit	ボリューム・グループ内の論理ボリュームの対話形式の配置を簡易化します。
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に関連した CPU 使用率をレポートします。
rmss	さまざまなサイズのメモリーでシステムをシミュレートして、パフォーマンスをテストします。
svmon	仮想メモリーの使用方法について情報を獲得し、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モジュールおよびソース・コード文レベルでの CPU 使用率をレポートします。
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をサブルーチン・レベルでの入口または出口として使用できます。

PTX 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。一部のツール・コマンドの構文は、『AIX Version 4.x Commands Reference』にも記載されています。

## AIX System Management Interface Tool

AIX System Management Interface Tool (smit) は、さまざまなシステム管理ツールおよびパフォーマンス・ツールに対するメニュー方式のインタフェースを提供します。smit を使用すると、多数のツールを容易に操作でき、実行するジョブのみに集中できます。

## Oracle パフォーマンス・ツール

ここでは、オラクル社のパフォーマンス監視ツールについて説明します。

### Utlbstat / Utlestat による統計の収集

utlbstat.sql および utlestat.sql スクリプトは、`$ORACLE_HOME/rdbms/admin` ディレクトリにあります。通常の作業負荷期間の開始時に SQL\*Plus から utlbstat を実行し、期間の終わりに utlestat を実行します。これによって、utlestat スクリプトを実行したディレクトリに report.txt ファイルが作成されます。このレポートは、測定期間のパフォーマンス統計が含まれているので、データベース競合の発見とチューニングに役立ちます。初期化パラメータ TIMED\_STATISTICS を TRUE に設定しなければ、タイミング測定がすべて 0 で表示されます。詳細は、『Oracle8i 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

Oracle Server には、リソースの競合およびボトルネック、パフォーマンス問題の検出に役立つ多数の動的サーバー・パフォーマンス表またはビューがあります。よく使用される表の一部を次に示します。

- 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** を設定すると、インスタンスのすべてのセッションで SQL トレースがオンに設定されます。または、コマンド **ALTER SESSION SET SQL\_TRACE=TRUE** を発行して、SQL トレースをセッション・レベルでオンに設定することもできます。該当する各セッションのトレース・ファイルは、初期化パラメータ **USER\_DUMP\_DEST** で指定されたディレクトリに作成されます。このパラメータのデフォルト値は、`$ORACLE_HOME/rdbms/log` です。

Oracle の TKPROF コマンドを実行して、生成されたトレース・ファイルを分析できます。SQL トレース機能および TKPROF の詳細は、『Oracle8i 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 2.0.4 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に次のパッケージが提供されています。

### Oracle Diagnostics Pack

Oracle Diagnostics Pack は、次のパッケージから構成されます。

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

#### Oracle Performance Manager

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および動的に表示する、リアルタイム・パフォーマンス・モニターです。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・カートリッジは、ホストにインストールする必要があります。

#### Oracle Capacity Planner

Oracle Capacity Planner を使用すると、システムおよびデータベース管理者は、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値を収集し、そのデータを履歴データベースに記録できます。さらに、履歴データを分析して、将来的に必要な容量を見積もることができます。

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・カートリッジのほかに、エージェント・データの採集サービスを、オペレーティング・システムのデータを収集す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 アプリケーションでは、Oracle Trace API を使用して計測された製品に対する Oracle Trace の収集データを作成、スケジューリングおよび管理できます。Oracle Intelligent Agent をホストにインストールして、Oracle Trace Collection Services を使用可能な状態にしておく必要があります。

#### Oracle TopSessions

上位 n 個の Oracle セッションのパフォーマンス・アクティビティをいつでも集中的に表示できます。

#### **Oracle Lock Manager**

ロックによってブロックされたセッションを表示および解決するために使用します。

#### **Oracle Tuning Pack**

Oracle Tuning Pack は、次のパッケージから構成されます。

- Oracle Expert
- Oracle Tablespace Manager
- Oracle SQL Analyze

#### **Oracle Expert**

データベースの初めての構築や、既存データベースのパフォーマンスに見られる特徴の収集および評価をサポートして、データベース環境のパフォーマンスを最適化します。

#### **Oracle Tablespace Manager**

表領域の詳細情報を提供し、表の断片化を解消したり使用されていない領域の割当てを解除したりできます。

#### **Oracle SQL Analyze**

ツールを使用して、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 パフォーマンスの解析、異なるオプティマイザ手法の識別や比較、および最適なパフォーマンスの文例の編集を自動的行えます。

## **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 アプリケーションの有効範囲が拡大し、複雑化するなかで、チューニングを頻繁に行うことで、システム・パフォーマンスが最適化され、データ障害の発生を防ぎます。

## **Oracle8i のチューニング順序**

Oracle Server のパフォーマンスを最大にするために、多数のタスクを実行します。1 つのチューニング・アクション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、次の順序で作業してください。

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 の使用率のチューニング
6. Oracle リソース競合のチューニング
7. Oracle Parallel Server の Oracle リソースのチューニング

この章では、メモリーおよびディスク I/O、CPU の使用率に関連した作業 3、4 および 5 を説明します。作業 1、2、6 および 7 の詳細は、「関連資料」のドキュメント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、プロセスが使用できるメモリーが不足した場合に発生します。システムはメモリー不足に対処するために、メモリーとディスクとの間でプログラム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

過度なページング・アクティビティは、パフォーマンスを大きく低下させます。特に JFS ファイルで作成されたデータベースでは、パフォーマンスが著しく低下することがあります。この場合は、JFS バッファに入っている最も頻繁に参照されるデータが、多数の SGA データ・バッファにも入ることがあります。ファイル・システムのバッファ・キャッシュ・マネージャの動作は、パフォーマンスに重大な影響を与えます。I/O ボトルネックが発生し、システム全体のスループットを低下させる可能性があります。

AIX では、バッファ・キャッシュ間のページング・アクティビティをチューニングできますが、回数を控えて慎重に行う必要があります。

次の 4 つの AIX システム・パラメータをチューニングできます。

minfree	空きリスト・サイズの最小値。このサイズ以下では、ページ・スティーリングによって空きリストが補充されます。
maxfree	空きリスト・サイズの最大値。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 用のバッファ・ページ数の最小値。
maxperm	ファイル I/O 用のバッファ・ページ数の最大値。

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 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のユーティリティです。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・システムが故障する可能性があります。

### minfree と maxfree のチューニング

迅速な応答が要求されるプログラムのワーキング・セット・サイズに基づいて **minfree** を設定できます。基本的には、ロードしたプログラムが空きリストを補充する必要がないように、空きリストには十分なページを用意しておく必要があります。vmstat コマンドによってプログラムのワーキング・セットのサイズを設定します。**maxfree** は、8 または **maxpgahead** の値だけ **minfree** よりも大きくなければなりません。たとえば、**maxpgahead** が 16 で、**minfree** を 128 に設定する場合は、次のコマンドを使用します。

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

これによって、**minfree** が 128、**maxfree** が 144 に設定されます。

## AIX ファイル・バッファ・キャッシュのチューニング

AIX バッファ・キャッシュは、JFS ファイルを使用するときにディスクへのアクセス回数を減らすために使用します。キャッシュが小さすぎると、ディスク使用率が増加し、1 つ以上のディスクが一杯になる可能性があります（このパラメータの増加による影響の詳細は、「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）。ただし、キャッシュが大きすぎても、貴重なメモリーが無駄になります。

**minperm** および **maxperm** パラメータを調整して、AIX バッファ・キャッシュを構成できます。一般的に、バッファのヒット率が低い場合（90% 未満、sar -b を実行した結果を参照）、**minperm** を増やしてください。高いバッファ・ヒット率を維持する必要がない場合は、**minperm** を減らすと使用可能な物理メモリーが増えます。詳細は、AIX のドキュメントを参照してください。

パフォーマンスの向上は、マルチプログラミングの度合いおよび作業負荷の I/O 特性に左右されるので、簡単に数値で示すことはできません。

### minperm と maxperm のチューニング

AIX では、**minperm** 値と **maxperm** 値を調整することで、ファイルで使用するページ枠数と計算可能（作業用またはプログラム・テキスト）セグメントで使用するページ枠数の比率を柔軟に制御する方法を管理者に提供します。**minperm** 値と **maxperm** 値は、次のガイドラインに従って調整します。

- 実メモリーでファイル・ページの占める割合が **minperm** 値よりも低くなった場合は、リページ率にかかわらず、ページ置換アルゴリズムはファイルおよび計算可能ページの両方を奪います。
- 実メモリーでファイル・ページの占める割合が **maxperm** 値よりも高くなった場合は、ページ置換アルゴリズムはファイル・ページのみを奪います。
- 実メモリーでファイル・ページの占める割合が **minperm** と **maxperm** の中間である場合、VMM は通常、ファイル・ページのみを奪いますが、ファイル・ページのリページ率が計算可能ページのリページ率よりも高いと、計算可能ページも奪います。

デフォルト値の計算方法は、次のとおりです。

$$\text{minperm (ページ数)} = ((\text{ページ枠数}) - 1024) \times 0.2$$

$$\text{maxperm (ページ数)} = ((\text{ページ枠数}) - 1024) \times 0.8$$

デフォルト値を変換するには、次のコマンドを入力します。

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

このコマンドは、**minperm** と **maxperm** のデフォルト値がそれぞれ 20% と 80% であったのに対して、**minperm** を合計ページ枠数の 5% に変更し、**maxperm** を合計ページ枠数の 20% に変更します。

RAW デバイスのデータベース・ファイルの場合は、**minperm** と **maxperm** を低い値（たとえば、それぞれ 5% と 20%）に設定できます。これは、AIX ファイル・バッファ・キャッシュが RAW デバイスには使用されないからです。したがって、メモリーを他の目的（Oracle システム・グローバル領域など）に有効利用できます。

## 十分なページング領域の割当て

ページング領域が不十分であると、システムがハングしたり、応答時間が非常に遅くなったりします。AIX では、ページング領域を RAW ディスク・パーティションに動的に追加できます。必要なページング領域は、現在の物理メモリー容量および各アプリケーションのページング領域要件に応じて異なります。AIX コマンド `lsps` を使用して、ページング領域の使用率を監視し、`vmstat` コマンドを使用してシステムのページング・アクティビティを監視します。

通常は、ほとんどの作業負荷で必要とされる物理メモリー容量の 2 倍または 3 倍になるように、ページング領域を構成するのが最適です。CASE、Oracle Applications、または Oracle Office を使用する場合は、もう少し高い値に設定してください。ページング領域の使用率を監視し、必要に応じて増やします。ページング領域を増やすには、`smit pgsp` コマンドを使用します。かなり大きい物理メモリーを持つシステム（1GB 以上）の場合は、物理メモリーの 1.5 倍のページング領域で十分です。

## 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。通常は、次のことを行います。

- Oracle が SGA とディスク間でデータを読み書きする速度よりも速くメモリーと拡張記憶領域間のページングを実行する、非常に高速な拡張記憶領域を装備したシステム以外では、定期的なページングを防止する必要があります。

- システム・パフォーマンスにとって一番有利な場所に、限定されたメモリー・リソースを割り当てます。メモリー・リソース要件とつり合いがとれるまで何回も均衡化処理を繰り返すことがあります。
- メモリーが不十分な場合は、メモリーが必要なシステムのプロセスまたは要素の優先順位リストを作成します。パフォーマンスの増加が最大になるように、メモリーを割り当てます。次に、各要素の優先順位リストの例を示します。

OS および RDBMS のカーネル

ユーザー・プロセスおよびアプリケーション・プロセス

REDO ログ・バッファ

PGA および共有プール

PCM ロックおよび DLM

DB ブロック・バッファ・キャッシュ

たとえば、Oracle Server の動的パフォーマンス表またはビューに対して問合せを実行し、共有プールおよびデータベース・バッファ・キャッシュの両方でメモリーの増加が必要であることが判明した場合は、限られた予備のメモリーをデータベース・ブロック・バッファ・キャッシュではなく、共有プールに割り当てた方が有利です。

次の AIX コマンドは、ページングの状況および統計を表示します。

```
$ 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・サイズの設定によって I/O スループットを向上できます。AIX では、2KB から 16KB までの範囲で DB\_BLOCK\_SIZE を設定できます。デフォルトは 4KB です。JFS 対応の Oracle データベースの場合は、ブロック・サイズを JFS ブロック・サイズ（AIX では 4KB）の倍数で設定する必要があります。RAW パーティション上のデータベースでは、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 では 512 バイト）の倍数になります。

OLTP 環境または複合的作業負荷の環境では小さい Oracle データベース・ブロック・サイズ（2KB または 4KB）、DSS タイプの作業負荷では大きいブロック・サイズ（8KB または 16KB）を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

専用の大きいバッファを割り当てることによって、データベース・アーカイブ速度を向上させることができます。ただし、過剰な設定はシステム全体のパフォーマンスを低下させる場合があります。init.ora ファイルのパラメータ LOG\_ARCHIVE\_BUFFER\_SIZE を 128 まで増やせます。LOG\_ARCHIVE\_BUFFERS のデフォルトは 4 です。

通常のデータベース・アクティビティが大きく低下することがないように、慎重にチューニングします。アーカイブ・プロセスにおいて 0 ～ 20% のパフォーマンスの向上が見込まれます。これ以上の向上を得たユーザーもいます。

## SGA サイズのチューニング

SGA での複数の共有メモリー・セグメントの使用を制約する要因は、現在のハードウェア・アーキテクチャがサポートできる物理メモリー容量です。ユーザーは、メイン・メモリー不足による Oracle SGA のページ・アウトは避けたいはずですが、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、大きい SGA をお勧めします。SGA 構成の詳細は、1-22 ページの「システム・グローバル領域の制御」を参照してください。

SGA サイズに最も影響する `init.ora` ファイルのパラメータは、**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** パラメータは、SGA サイズに大きく影響します。これらのパラメータをどれだけ増やせるかは、SGA に割当てられるメモリーに左右されます。**DB\_BLOCK\_BUFFERS** および **SHARED\_POOL\_SIZE** に大きい値を設定すると、パフォーマンスのオーバーヘッドが生じるので、これらの値を小さくすれば CPU 使用率が減少します。

## I/O バッファと 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・オプションなどの高速なデータ・ロード方法を使用すると、CPU が I/O 操作の完了を待っている時間が長くなります。バッファ数を増やすと、通常は CPU 使用量が増加し、全体的なスループットが向上します。

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量および CPU 使用率の拡大範囲に応じて決定します。**BUFFERS** パラメータのファイル処理オプション文字列の調整方法の詳細は、『Oracle8i ユーティリティ・ガイド』を参照してください。

パフォーマンスの向上は、CPU 使用率およびロードするときに使用される並行度によって異なります。SQL\*Loader ユーティリティの一般情報は、『Oracle8i ユーティリティ・ガイド』を参照してください。

### imp の BUFFER パラメータ

`imp` の **BUFFER** パラメータは、`imp` 使用時の高速ネットワークのパフォーマンスを実現するために、大きい値に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、1MB 以上の値が必要です。

## ディスク I/O の問題

ディスク I/O の競合は、(後続のページングおよびスワッピングを伴った) 不適切なメモリー管理、ディスク間での不適切な表領域およびファイルの分布によって生じることがあります。I/O 負荷は、すべてのディスクに均等に分散する必要があります。

### 索引と表の分離

索引とその参照先の表が同じディスク・ドライブに存在すると、索引検索に関連するすべての I/O が同じディスク上に集中します。I/O 負荷が高い場合は、索引と表を別々のドライブに保存して負荷を分散させる必要があります。

### REDO ログの専用ディスク・デバイスへの配置

INSERT および UPDATE アクティビティが頻繁に発生する Oracle アプリケーションでは、その他のディスク・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で、Oracle パフォーマンスを最大化できます。また、ARCHIVELOG オプションが使用可能になっている場合は、すべての REDO ログをそれぞれのディスクに配置し、LGWR プロセス（現行 REDO ログへの書込み）と ARCH プロセス（クローズされた REDO ログからの読込み）との間で生じるディスク競合を最小化します。REDO ログを RAW デバイス上に配置すれば、さらにパフォーマンスが向上します。次の理由から、REDO ログは、最初のファイルの 1 つとして RAW デバイスに配置してください。

- REDO ファイルではデータが順次に読み書きされ、RAW デバイスの利点を最大化する。
- REDO ログのファイル・サイズは固定されているので、RAW デバイスの管理コストを最小化する。

### ホット・ファイルの他のディスクへの移動

頻繁にアクセスされるホット・ファイルを、あまり利用されていないディスク・デバイスに分散し、I/O のバランスをとります。ファイル全体を使用頻度の高いディスクから低いディスクに移動するか、または各ディスク上にホット・ファイルの一部分が配置されるように、ホット・ファイルを複数のディスクに分散させます。

### ホット・ファイルの I/O の削減

ディスク・デバイス上にホット・ファイルが 1 つだけ存在し、このファイルが担当する要求キューが大きい場合は、ファイルを別のディスクに移動しても効果はありません。問題の Oracle ファイルまたは表領域に、複数のセグメント（表および索引など）のデータが入っている場合は、頻繁にアクセスされるセグメントを別の表領域に移動し、別々のファイルに移動します。データベース・セグメントの物理デバイスは、表領域レベルで指定する必要があります。1 つのセグメントのみが関与している場合は、1 つの表領域内の複数のファイルにセグメント・データを配置するように、表のストライプ化を検討してください。

## データベースの過剰な断片化のチェック

Oracle データ構造を断片化すると、複数の物理 I/O から 1 つの論理 I/O の要素を結合する必要があります。余分なオーバーヘッドは、応答時間を長くします。

### エクステントの断片化

データベース・セグメントに、複数の連続していないディスク領域エクステントが含まれている場合があります。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。ディスク上で物理的に連続していない複数のエクステントにデータが分散しているために、1 つの I/O 要求を 2 つ以上の物理 I/O に分割しなければならないときに、I/O 分割が発生します。

### 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ています。表領域内で、Oracle セグメント（表、索引など）は多数の個別エクステントで構成されており、表領域の断片化を生じます。表のストライプ化のように、断片化が好ましい場合もありますが、通常は好ましくありません。データベース・セグメントを削除するたびに、表領域の断片化が起こります。表領域が断片化すると、空き領域を有効に使用できません。

表領域を断片化すると、Oracle のマルチブロック読み込み機能が利用できなくなります。また、連続した空きエクステントよりも大きいエクステントが存在すると、データベース領域が無駄になります。次の SQL 文で表領域の断片化を識別できます。

```
SELECT * FROM DBA_EXTENTS;
```

次の SQL 文を使用し、空き領域に問合せを実行できます。

```
SELECT * FROM DBA_FREE_SPACE;
```

空き領域の断片化には、次の 2 つの形式があります。

- 泡立ち状。アクティブなエクステント間に挟まっていたエクステントを削除したときに、泡のように小さく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トが削除されるたびに、空き領域が連続的な断片に区分化されます。

utl1bstat および utlestat で高い再帰コール値が報告された場合は、表領域の断片化が考えられます（ただし、データ・ディクショナリ・キャッシュが適切にチューニングされている場合）。

**参照：** 詳細は、『Avoiding a Database Reorganization』（Craig A.Shallahamer 著）を参照してください。WWW（<http://www.europa.com/~orapub>）から入手できます。

## AIX 論理ボリューム・マネージャ

AIX 論理ボリューム・マネージャ（LVM）には、ディスク競合を削減するために、複数のディスク装置にわたってデータをストライプ化する機能があります。ストライプ化の主な目的は、大きい順次ファイルを読み書きするときに、高パフォーマンスを得ることです。LVM のストライプ化機能を有効に使用することで、I/O を複数のディスクに均等に分散でき、全体的にパフォーマンスが向上します。

### ストライプ化した論理ボリュームの設計

ストライプ化した論理ボリュームを定義するときに、次の項目を指定する必要があります。

ドライブ	2 つ以上の物理ドライブ。パフォーマンス・クリティカルな順次 I/O を実行するときに、これらの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。2 つ以上のアダプタ間での論理ボリュームのストライプ化が必要になる場合があります。
ストライプ・ユニット・サイズ	2KB から 128KB までの範囲内にある、任意の 2 の累乗です。ただし、32KB および 64KB がほとんどの作業負荷に適しています。Oracle データベース・ファイルでは、データベース・ブロック・サイズの倍数でなければなりません。
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数は、使用するディスク・ドライブ数の倍数でなければなりません。
属性	ミラー化できません。つまり、copies=1 です。

### ストライプ化した論理ボリュームの推奨パラメータ

次に、通常では最高の順次 I/O スループットを達成する、ストライプ化した論理ボリュームのパラメータを示します。

ストライプ・ユニット・サイズ	64KB
minpgahead	2（最小先読みページ数）



**maxpgahead** ディスク・ドライブ数の 16 倍（最大先読みページ数）。ストライプ・ユニット・サイズ（64KB）とディスク・ドライブ数の積を単位として先読みします。つまり、1 回の先読みで、1 ディスク・ドライブにつき 1 ストライプ化ユニットを読み込みます。

**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。）

filemon、sar、iostat またはその他のパフォーマンス・ツールなどの AIX ユーティリティを使用して、ホット・ディスクがないかを調べることで、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していることを確認します。

### その他の考慮事項

LVM を有効に利用した場合にどれだけパフォーマンスが向上するかは、使用した LVM および作業負荷の特性によって大きく異なります。意思決定支援システム（DSS）の作業負荷では、パフォーマンスが大きく向上する場合があります。OLTP タイプまたは複合的な作業負荷でも、かなりのパフォーマンスの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

同時にアクセスされる可能性があるパーティションの場合、同じ物理ディスクからパーティションを使用するのは避けてください。たとえば、データ・ファイルとログ・ファイルは同時にアクセスされることがたいへん多いので、これらのファイルは別々のディスク・セットに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・ヘッドのシーク時間がかかり、パフォーマンスが低下します。このとき、ディスク上のパーティションの位置が互いに離れていると、パフォーマンスがさらに低下します。

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

この話題は、時折論議的になります。この話題が論議的になるいくつかの理由を示します。

- さまざまなファイル・システム・インプリメンテーションが改善されるなかで、ファイル・システムは絶えず進歩し続けています。ファイル・システムの方が RAW デバイスよりも優れたパフォーマンスを提供する場合があります。
- さまざまなベンダーが、さまざまなディスクの長所を利用して、独自の方法でファイル・システム層をインプリメントしています。これによって、各プラットフォームに対するユーザーの理解がますます複雑化します。
- パワフルな LVM インタフェースの導入によって、RAW パーティションを基盤とした論理ディスクの構成や元に戻す作業の負担が著しく軽減されます。

パフォーマンスがどの程度向上するかは、作業負荷の特性によって大きく異なります。

JFS を使用した方が、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簡単です。ただし、JFS ではバッファの読み書きのみがサポートされており、データを I/O システム（Oracle バッファ・キャッシュ以外、SGA）に転送したり受け取ったりするたびに、追加の AIX ファイル・バッファ・キャッシュが作成されます。これは、JFS ベースのデータ・ファイルの主な短所であり、I/O サブシステムの負荷が高いときほど重大な欠点となります。

また、負荷が高いシステムでは、AIX バッファの競合が主なボトルネックになります。次のような影響があります。

- データをファイル・システム・バッファ・キャッシュからディスクへフラッシュするページデーモンの負担が増加します。
- さらに多くのメモリーが、AIX バッファ・キャッシュの犠牲になります。

RS/6000 プラットフォームでのオラクル社の経験では、作業負荷に応じて異なりますが、一般的に、ファイル・システムに比べて RAW デバイスを使用したときの方が全体的なパフォーマンスおよびスケーラビリティが向上します。したがって、使用頻度が高いデータベース構成では、RAW デバイスの使用をお勧めします。

**注意：** Oracle Parallel Server の場合は、データベース・ファイルに JFS ではなく、RAW デバイスを使用してください。

### 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、データを簡単に JFS から RAW デバイスに移動できるので、すべてのデータを手動で再ロードしなくて済みます。次のように dd コマンドを入力します。

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

必ず、次のことを実行してください。

- RAW デバイスの最初の 4KB ブロックは、AIX 論理ボリューム・マネージャが使用するのをスキップします。

**注意:** RAW デバイスが HSD の場合、最初の 4KB ブロックはスキップする必要はありません。

- 領域が無駄にならないように RAW デバイスのサイズを決定します (RAW デバイスは、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・ファイルを改名します。
  - RAW パーティションに権限を設定します。
2. 順次読み込みが多少遅くなる場合があります。アプリケーションで多数の全表走査を実行していた場合は、これらの表が AIX ファイル・バッファ・キャッシュによってキャッシュされないため、全表走査で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. メモリが不十分な構成では、AIX バッファ・キャッシュのサイズ変更が必要です。詳細は、「メモリーとページング」の項を参照してください。
  4. RAW デバイスの名前が変更される可能性があるため、AIX 環境ではできる限りシンボリック・リンクの使用をお勧めします。たとえば、ディスクの周辺装置を再構成したり、データベースを新しいハードウェア構成に移動したりするときに、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。

RAW デバイスの使用に関連した操作の詳細は、『Oracle8i for AIX-Based Systems インストール・ガイド』を参照してください。

## JFS と RAW デバイスの両方の長所の利用

AIX では一部のデータ・ファイル用に JFS を、その他のデータ・ファイル用に RAW パーティションを自由に選択できます。非同期 I/O は、RAW パーティションおよび JFS の両方で動作します。データベース・オブジェクトの I/O アクティビティの性質が事前にわかっている場合は、特定のオブジェクトに対応するデータ・ファイルを JFS または RAW パーティションのどちらかに配置し、可能であれば LVM を伴った体系を考えることができます。

パフォーマンスの向上は、作業負荷の性質、ディスクまたはファイル・システムの構成に応じて大きく異なります。

## 非同期 I/O の使用方法

Oracle8i では、AIX が提供する非同期 I/O (AIO) を最大限に活用し、高速なデータベース・アクセスを実現します。AIO は複数の I/O 操作を混ぜ合わせて I/O サブシステムのスループットを向上します。データを各ディスク間に適切に分散しなければ、AIO の利点は利用できません (データベース、ログおよび制御ファイルの分散の詳細は、『Oracle8i 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) を使用してストライプ化を行うと、AIO の効果が強化されます。LVM は、データを複数のディスク装置にストライプ化することで、ディスク競合を削減します (詳細は、「AIX 論理ボリューム・マネージャ」の項を参照してください)。AIO を LVM と一緒に使用すると、RDBMS パフォーマンスが著しく向上します。

AIX 4.x では、ファイル・システム・パーティションおよび RAW デバイスの両方で作成されたデータベース・ファイルに対して AIO をサポートします。AIO 要求を発行した後に、アプリケーション・プロセスの実行を継続できます。ファイル・システム (VSD または AIX 4.2.1 の HSD) での AIO 要求の場合、要求をキューから取り出した時点から要求が完了するまで、サーバーと呼ばれるカーネル・プロセス (kproc) がそれぞれの要求を管理します。システムで同時に実行できる AIO 要求数は、サーバーの台数によって制限されます。AIX 4.2.1 では、VSD および HSD 以外の RAW デバイスで AIO をカーネル化します。つまり、AIX サーバーは AIO 要求を処理する必要がありません。AIX 4.3 では、VSD および HSD (APAR IX79690) を含むすべての RAW デバイスで AIO をカーネル化します。

---

**注意：** VSD および HSD でカーネル化された AIO を使用する場合、最大バッファ・サイズは 128KB 以上でなければなりません。そうしないと、EMSGSIZE 「Message too long」 エラーになります。

---

サーバーの台数は、smit (smit->Devices->Asynchronous I/O->Change/Show Characteristics of Asynchronous I/O->{MINIMUM|MAXIMUM} または smit aio) または chdev で設定します。最小値は、システムのブート時に起動されるサーバー数です。最大値は、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を制限します。

デフォルト値は、**minservers=1** および **maxservers=10** です。通常、これらの値はカーネル化された AIO を使用しないで Oracle RDBMS を実行するには小さすぎます。次のように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 を汎用にしたもので、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。設定する場合、I/O スレーブは LARGE\_POOL\_SIZE 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
DISK_ASYNC_IO	TRUE/FALSE	TRUE
TAPE_ASYNC_IO	TRUE/FALSE	TRUE
BACKUP_DISK_IO_SLAVES	TRUE/FALSE	FALSE
BACKUP_TAPE_IO_SLAVES	TRUE/FALSE	FALSE
DBWR_IO_SLAVES	0 ～ 999	0
LGWR_IO_SLAVES	0 ～ 999	0
ARCH_IO_SLAVES	0 ～ 999	0
DB_WRITER_PROCESSES	1 ～ 10	1

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 のパラメータのうち、最初の 2 つ（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が 0 のため、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 では、このような場合の DBWR\_IO\_SLAVES の最適値は 4 です。LGWR\_IO\_SLAVES には 9 以下の値を設定する必要があります。

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 つだけです。

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

一般的に、init.ora ファイルのパラメータ DB\_FILE\_MULTIBLOCK\_READ\_COUNT の値が大きい方が、I/O スループットが優れています。AIX では、このパラメータの範囲は 1 ～ 512 までですが、通常は 16 より大きい値に変更してもパフォーマンスは向上しません。

このパラメータの設定は、DB\_BLOCK\_SIZE × DB\_FILE\_MULTIBLOCK\_READ\_COUNT の方が LVM ストライプ・サイズよりも大きくなるようにする必要があります。つまり、I/O 要求を満たすために、ディスク装置を多く起用します。

## RAID 機能の使用

RAID 5 は、順次読込みのパフォーマンスを向上させますが、全体的な書込みパフォーマンスを低下させます。一般的に、書込みが少ない場合にのみ RAID 5 を使用することをお勧めします。RAID 5 での大量の書込みは、RAID 以外の環境に比べてパフォーマンスが低下する場合があります。

RAID 0/1 では、通常、ストライプ化 / ミラー化をハードウェア・レベルで導入するとパフォーマンスが向上します。これは、AIX または Oracle レベルで行うより効率的です。

RAID 7 は、非同期 I/O（デバイス、バス、バスなど）および複数のホスト・インタフェースを得るために、埋込み OS を採用します。これによって、大小の読み書きにおいて RAID 0 ～ 6 よりも優れたパフォーマンスを提供できます。

## 後書きの使用

ファイル・システムでは、書込みパフォーマンスを増加し、メモリー内の使用済みページ数を制限し、ディスクの断片化を最少化するために、各ファイルを 16KB パーティションに分割します。プログラムが次の 16KB パーティションの先頭バイトに書き込むまで、パーティションのページはディスクに書き込まれません。vmtune コマンドを使用して、後書きパーティションのサイズを変更できます。次のようにコマンドを入力します。

```
vmtune -c 8
```

前述のコマンドは、後書き用のバッファ・サイズを 8 個の 16KB パーティションに設定します。次のようにコマンドを入力します。

```
vmtune -c 0
```

前述のコマンドは、後書きを使用禁止にします。後書きを使用禁止にすると、パフォーマンスがいくらか向上することがあります。

## 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。プロセスがファイルにアクセスし、連続した 2 ページを読み込むと、VMM はプログラムが引続きファイルに順次アクセスすると仮定して、追加の順次読み込みをスケジュールします。これらの読み込みは、プログラムの処理とオーバーラップし、通常よりも速くデータを使用可能にします。先読みするページ数は、次の 2 つの VMM しきい値によって決定します。

### **minpgahead**

VMM が最初に順次アクセス・パターンを検出したときに先読みするページ数。

### **maxpgahead**

順次ファイルで VMM が先読みする最大ページ数。

**minpgahead** および **maxpgahead** は、アプリケーションに応じて適切な値に設定します。デフォルト値は、それぞれ 2 および 8 です。vmtune コマンドを使用して、これらの値を変更できます。ストライプ化した論理ボリュームの順次パフォーマンスが最も重要であるシステムでは、**maxpgahead** の大きい値を使用できます。vmtune を使用して、**minpgahead** および **maxpgahead** を設定できます。次のようにコマンドを入力します。

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

前述のコマンドは、**minpgahead** を 512 ページに、**maxpgahead** を 1024 ページに設定します。**minpgahead** および **maxpgahead** は、2 の累乗に設定します。

## ディスク I/O の歩調合せ

AIX のメカニズムであるディスク I/O の歩調合せによって、システム管理者はファイルに対して保留される I/O 要求の数を制限できます。これによって、ディスク I/O が激しいプロセスによって CPU が占領されるのを防止できます。したがって、対話形式のプロセスおよび CPU 使用量の激しいプロセスの応答が遅くなることはありません。

高水位標および低水位標の 2 つのシステム・パラメータを調整して、ディスク I/O の歩調合せを達成します。保留中の I/O 要求が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状態に設定されます。待機中の I/O 要求が低水位標以下に下がったとき、このプロセスはスリープ状態から解放されます。

smit を使用して、高水位標および低水位標を変更できます。smit -> System Environment -> Change/Show Characteristics of Operating System のように選択し、新しい高水位標および低水位標を入力します。試行錯誤しながら、水位標を判断します。適切に設定しないと、パフォーマンスに悪影響を及ぼすことがあるので、慎重に設定してください。ディスク I/O が 4KB を超える場合は、高水位標および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。

## ディスク配置の考慮事項

AIX では、ディスク上の論理ボリュームの配置をある程度制御できます。ホット論理ボリュームを相互に近接して配置すれば、ディスクのシーク時間が短くなり、全体的なパフォーマンスが向上します。

## リモート I/O の最小化

SP アーキテクチャで実行中の Oracle Parallel Server では、Virtual Shared Disks (VSD) または Hashed Shared Disks (HSD) を、異なるノード上で実行しているすべての 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使用します。論理ボリュームがノードから見てローカルな VSD に対して I/O 要求が出された場合は、ローカル I/O が実行されます。ローカルでない VSD への I/O 通信量は、ネットワーク通信レイヤーを経由します。

優れたパフォーマンスを得るには、リモート I/O をできる限り少なくすることが重要です。各インスタンスの REDO ログは、ローカル論理ボリュームの VSD に配置する必要があります。更新および挿入が激しい場合、ローカル論理ボリュームにマップされる VSD 上のプライベート・ロールバック・セグメントを各インスタンスが持っている必要があります。

各セッションで、各ユーザーが使用できるテンポラリ表領域は 1 つのみです。テンポラリ表領域では、OPS を実行中の各ノードに対してローカルなデータ・ファイルが 1 つ以上必要です。

アプリケーションおよびデータベースを慎重に設計し（たとえば、インスタンスに対してアプリケーションおよびデータベースをパーティション化する）、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 では、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 (nocache) をお勧めします。これは、VSD LRU キャッシュ・バッファが、確保されたカーネル・メモリーをより有効な方法で使用するためです。また、キャッシュ・バッファが使用可能な状態に設定されていると、物理読込みのたびにオーバーラップしたページを検索するためにキャッシュ・ブロックが検索され、キャッシュ・バッファの内外へのデータのコピーによってオーバーヘッドが生じます。

VSD のパフォーマンスをチェックするには、statvsd コマンドを使用します。要求がキューに入ったままバッファを待機している状態が持続している場合でも、バッファを追加しないでください。かわりに、切替え送信プールのサイズを、次のようにして増やします。

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

送信プール・サイズを増やした場合は、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)
```

現行の送受信プールのサイズをチェックするには、次のように入力します。



```
$ /usr/sbin/lssattr -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とプロセスの優先順位

CPU は、プロセス間で競合が生じる可能性のあるもう 1 つのシステム・コンポーネントです。通常は AIX カーネルが効率的に CPU を割り当てますが、多くのプロセスが CPU サイクルのために競合します。CPU を 1 つ以上搭載したシステム (SMP) では、それぞれの CPU で異なるレベルの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 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。schedtune コマンドを使用して、タイム・スライスを変更できます。ただし、十分に注意して変更してください。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・スライスを長くすると、コンテキスト切替え率が低くなります。結果的に、プロセスでコンテキスト切替えに消費される CPU サイクルが減少し、システム・スループットが向上します。

ただし、特に単一プロセッサ・システムでは、ランタイム・スライスが長くなると、応答時間が長くなる場合があります。デフォルトのランタイム・スライス (10 ミリ秒) は通常、標準アプリケーションにとって適切な値です。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・プロセスで長時間の実行が可能となるときは、次のように入力してタイム・スライスを増やします。

```
# /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

SMP マシンで、特定のプロセスをプロセッサにバインドすると、パフォーマンスが大きく向上することがあります。プロセッサのバインディングは、AIX 4.x で使用可能であり、全機能がサポートされています。

プロセッサ・バインディングの利点は、次のとおりです。

- 優先順位の高いアプリケーションに対して CPU 時間が相対的に多く割り当てられます。
- プロセス・コンテキストの維持期間が長くなります。

AIX でのプロセッサ・バインディングは、自動ではありません。bindprocessor コマンド (man ページ参照) を使用して、マルチプロセッサで明示的にプロセスをプロセッサにバインドする必要があります。Oracle プロセスをプロセッサにバインドできるのは、スーパーユーザーか 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。

さまざまな Oracle バックグラウンド・プロセス (dbwr 以外) を別々のプロセッサにバインドし、1 つの空きプロセッサを dbwr 用に残しておくことをお勧めします。これによって、dbwr を実行するプロセッサが保証されるのと同時に、CPU バウンドになった場合、dbwr が自由に他のプロセッサへ移行できます。

**注意：** プロセッサ・バインディングは複雑であるため、慎重に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は、プロセッサが空いている場合でも、別のプロセッサに移行できません。これによって、アプリケーションのパフォーマンスが低下する場合があります。プロセッサ・バインディングに適しているのは、負荷が均等に分散した同一アプリケーション環境です。

プロセスとプロセッサのバインドは、排他的ではありません。つまり、プロセッサは自由に他のプロセスを実行できます。

### ネットワーク化したクライアント・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 リスナーを通じて Oracle サーバー・プロセスに接続されているとき (つまり、ネットワーク化した Oracle クライアント・サーバーを実行しているとき)、Net8 リスナー・プロセスをバインドすることで、サーバー・プロセスを簡単にプロセッサとバインドできます。リスナーから発生する後続のすべての Oracle サーバー・プロセスは、同一プロセッサにバインドされます。

これを達成する 1 つの方法として、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。これには、各リスナーが 1 連になるように、ファイル \$ORACLE\_HOME/network/admin/listener.ora をカスタマイズ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 をカスタマイズし、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、リスナー L1 および L2 にそれぞれポート 1521 および 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)
    )
  )
)

```

次のように入力して 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

```

これらの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 は、次のとおりです。

```

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 使用を制限する場合は、この方法でリスナーをバインドすれば、そのプロセッサが使用可能なときに常に同一プロセッサでアプリケーションを実行することができます。

## ローカル環境でのプロセッサ・バインディング

パイプ・ドライバを 2 タスクで使用し、クライアントおよび Oracle Server が同じマシン上で実行している場合は、プロセッサ・バインディングはさらに複雑です。これには、各サーバーのプロセス 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。管理用に過度なオーバーヘッドが生じ、サーバー・プロセスの寿命が長い場合以外は、一般的に努力する価値はありません。

Oracle プロセスをプロセッサとバインドすると、同じマシンで実行している Oracle 以外のアプリケーションのパフォーマンスが低下する場合があります。慎重なチューニングと監視が必要です。

## マルチ・プロセッサ・システムでのスピン・カウントのチューニング

リソースの取得を 1 回だけ試みた後に、プロセスをスリープさせるのでは、高価な操作になります。かわりに、プロセスが数回スピンしてもリソースを取得できなかった場合に、プロセスをスリープさせることができます。別のプロセスが保持しているラッチをまさに解放しようとしている場合には、この方針によってラッチの取得時のパフォーマンスが向上します。ただし、CPU のオーバーヘッドが大きくなる場合もあります。

Oracle8i for AIX で、`init.ora` ファイルのパラメータ `SPIN_COUNT` のデフォルト値は、シングル・プロセッサでは 0、マルチ・プロセッサでは 2000 です。これは大多数のアプリケーションにとって、適切な値です。`SPIN_COUNT` を増加すると、一般的に CPU 使用量が増加します。シングル・プロセッサ・マシンまたは CPU 使用量の多いマシンでは、`SPIN_COUNT` を 0（スピンなし）に設定するのが適切です。`SPIN_COUNT` の変更によって、パフォーマンスが向上することがありますが、値が大きすぎるとパフォーマンスが低下する可能性があります。アプリケーションをチューニングしてラッチの競合を減少させた方が、一般的によりよい結果が得られます。

リリース 7.3 以降では、`DB_BLOCK_LRU_LATCHES` パラメータによってバッファの LRU リストで使用可能なラッチ数が決定します。このパラメータのデフォルト値はプロセッサ数の半分ですが、1 からシステムのプロセッサ数までの範囲で、値を設定できます。キャッシュ・バッファの LRU（Least Recently Used アルゴリズム）のラッチで競合が発生している場合は、このパラメータ値を大きくしてみてください。

## その他のチューニングのヒント

ここでは、『Oracle for UNIX Performance Tuning Tips』に記載されていない、その他のパフォーマンス・チューニングのヒントを説明します。

### UDP のチューニング

Oracle Parallel Server は、プロセス間通信で UDP を使用します。UDP カーネルの設定によって、Oracle パフォーマンスを改善するようにチューニングできます。

AIX での UDP カーネルのバッファリングは、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って変更できます。

- `udp_sendspace` は、最大値の 65536 に設定する必要があります。
- `udp_recvspace` は、`sb_max` の値より小さくする必要があります。

`udp_recvspace` の値は、通常、`udp_sendspace` の 10 倍の値で開始する必要があります。これは、UDP が、他のパケットに到達する前にアプリケーションにパケットを送信できない場合があるためです。

udp\_recvspace の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"
```

オーバーフロー数が 0 でない場合、udp\_recvspace を増加します。

## 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 論理ボリューム・マネージャが使用するので、上書きしないでください。4KB より大きいブロック・サイズでバックアップした方が早く終了します。テープへのバックアップに適したブロック・サイズは 256KB です。次に、RAW デバイスをテープにバックアップおよび復元するときに使用できるサンプル・スクリプトを示します。

1. RAW デバイスのバックアップをテープに作成します。

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

2. RAW デバイスをテープから復元します。

```
dd if=/dev/rmt0.1 of=/dev/<raw_device> count=63 seek=1 skip=1 bs=4k
```

```
mt -f /dev/rmt0.1 bsf 1
```

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

---

**注意：** HSD の場合、最初の 4KB はスキップしないでください。復元コマンドは次のように入力します。

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

---

## パラレル回復の使用方法

init.ora ファイルのパラメータ RECOVERY\_PARALLELISM=N (N は 1 より大きい整数で、通常はマシンのプロセッサ数) を設定することで、さらにパフォーマンスは向上します。これによって、ログ・ファイルの変更をデータベースに適用するタスクを分担するために、複数の回復スレーブ・プロセスが生成されます。

## Pro\*C プログラムのコンパイル

AIX XLC コンパイラを使用すると、いくつかのコンパイラ・フラグによって、Pro\*C または Oracle OCI インタフェースで書かれた Oracle アプリケーションのパフォーマンスが向上する場合があります。たとえば、次のことを試みてください。

1. 最高の最適化レベル (-O3) を使用します。
2. アーキテクチャ固有のコンパイル・オプションを使用します。AIX XLC コンパイラがサポートするアーキテクチャ固有のコンパイル・オプションは次のとおりです。

### **-qarch=com**

3つの命令セットの共通サブセットのコンパイルです。このオプションでコンパイルしたプログラムは、3つのアーキテクチャのすべてで正しく実行されます。

### **-qarch=pwr**

元の RISC System/6000 の POWER アーキテクチャのコンパイルです。このオプションでコンパイルしたプログラムは、3つのアーキテクチャのすべてで正しく実行されます。ただし、パフォーマンスを犠牲にしていくつかの命令が PowerPC でシミュレートされる場合があります。

### **-qarch=pwrx**

特に、POWER2 アーキテクチャに対するコンパイルです。倍精度浮動小数点または浮動小数点平方根をよく使用するプログラムでは、パフォーマンスが向上する場合があります。この実行プログラムは、Power2 でのみ実行できます。

### **-qarch=ppc**

特に、Power PC アーキテクチャに対するコンパイルです。単精度浮動小数点をよく使用するプログラムでは、パフォーマンスが向上する場合があります。この実行プログラムは、Power PC でのみ実行できます。

アプリケーションの性質に応じて、このように積極的に最適化するとパフォーマンスが向上する場合があります。これらのフラグの詳細は、コンパイラのドキュメントを参照してください。

## 関連資料

### Oracle マニュアル

1. 『Oracle8i for AIX-Based Systems インストレーション・ガイド』
2. 『Oracle8i チューニング』
3. 『Oracle8i Net8 管理者ガイド』
4. 『Oracle8i SQL リファレンス』
5. 『Oracle8i ユーティリティ・ガイド』
6. 『Oracle8i アプリケーション開発者ガイド』

### IBM マニュアル

1. 『AIX Versions 4.x Performance Tuning Guide』
2. 『AIX Performance Toolbox/6000 User's Guide 1.2 and 2.1』
3. 『AIX Version 4.x System Management Guide: Operating System and Devices』
4. 『AIX Version 4.x System Management Guide: Communication and Networks』
5. 『AIX Version 4.x Commands Reference』

## 関連資料

---



---

## SQL\*Plus の管理

- SQL\*Plus の管理
- SQL\*Plus の使用
- 制限事項

## SQL\*Plus の管理

### 設定ファイル

SQL\*Plus の設定ファイルには、サイトごとに定義するグローバルなファイル (glogin.sql)、およびユーザーごとに定義するファイル (login.sql) があります。glogin.sql および login.sql ファイルには、SQL\*Plus セッションの最初に行われる SQL 文または SQL\*Plus コマンドが記述されています。SQL\*Plus を起動すると、最初に glogin.sql が読み込まれ、次に login.sql が読み込まれます。

### Site Profile

Site Profile ファイルは、\$ORACLE\_HOME/sqlplus/admin/glogin.sql です。SQL\*Plus は、ユーザーが SQL\*Plus を起動すると、このコマンド・ファイルを実行します。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

User Profile ファイルは、login.sql です。SQL\*Plus はユーザーが SQL\*Plus を起動すると、このコマンド・ファイルを実行します。SQL\*Plus は、User Profile の現在のディレクトリを検索します。環境変数 SQLPATH には、SQL\*Plus が検索する login.sql ファイルのディレクトリ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、次のような SQLPATH を設定しているとします。

```
/home:/home/oracle:/u01/oracle
```

SQL\*Plus は、最初に現在のディレクトリ /u02/oracle にある login.sql を検索します。現在のディレクトリに login.sql がない場合、SQL\*Plus は /home、/home/oracle および /u01/oracle ディレクトリを検索します。SQL\*Plus は、最初に見つかった login.sql ファイルのみ実行します。

login.sql は最後に実行されるので、login.sql のオプション・セットは glogin.sql のオプション・セットを上書きします。

**参照：** 詳細は、『SQL\*Plus ユーザーズ・ガイドおよびリファレンス』の第 3 章を参照してください。

## PRODUCT\_USER\_PROFILE 表

[Typical] インストールの場合、PRODUCT\_USER\_PROFILE (PUP) 表が自動的に作成されます。PUP 表を使用して、特定の SQL および SQL\*Plus コマンドを使用禁止にします。この表を再作成する必要がある場合は、SYSTEM スキーマの \$ORACLE\_HOME/sqlplus/admin/pupbld.sql スクリプトを実行します。

次に例を示します。

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

SQL\*Plus は、"?" 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

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

[Typical] インストールの場合、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 demobld.sql ファイルを実行すると、ユーザーはスキーマにデモンストレーション表を作成できます。次に例を示します。

```
% sqlplus scott/tiger
SQL> @?/sqlplus/demo/demobld.sql
```

次のようにシェル・スクリプト \$ORACLE\_HOME/bin/demobld を使用することによって、\$ORACLE\_HOME/sqlplus/demo/demobld.sql を実行できます。

```
% 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/demodrop.sql を実行します。SQL\*Plus で demodrop.sql ファイルを実行すると、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。次に例を示します。

```
% sqlplus scott/tiger
SQL> @?/sqlplus/demo/demodrop.sql
```

次のようにシェル・スクリプト \$ORACLE\_HOME/bin/demodrop を使用することによって、\$ORACLE\_HOME/sqlplus/demo/demodrop.sql を実行できます。

```
% 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。これらの表と同じ名前の表がスキーマに存在しないことを確認してから、どちらかの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

### Typical インストール

[Typical] インストールで、または Database Configuration Assistant のオプションで、事前作成済みのデータ・ファイルを含む初期データベースをコピーする場合、ヘルプ機能は自動的にインストールされます。

### Database Configuration Assistant (DBCA)

DBCA は、データベースを作成する際にヘルプ表を作成するためのオプションを提供します。

### ヘルプ機能の手動インストール

手動でヘルプ機能をインストールするには、シェル・スクリプト \$ORACLE\_HOME/bin/helpins を使用します。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おく必要があります。次に例を示します。

```
% helpins
```

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

## SQL\*Plus の使用

### SQL\*Plus からのシステム・エディタの使用

SQL\*Plus プロンプトで `edit` コマンドを入力すると、`ed`、`emacs`、`ned`、`vi` などのデフォルト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`、`login.sql` または SQL\*Plus セッション中に入力することによって設定できます。

たとえば、`vi` にデフォルト・エディタを設定するには、次のように入力します。

```
define_editor=vi
```

`_editor` 変数を設定しない場合は、環境変数 `EDITOR` または `VISUAL` のどちらかの値を使用します。環境変数 `EDITOR` および `VISUAL` が両方設定されている場合、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 `afiedt.buf` というテンポラリ・ファイルを使用します。このファイル名を変更するには、`SET_EDITFILE` コマンドを使用します。次に例を示します。

```
SQL> SET_EDITFILE/tmp/myfile.sql
```

SQL\*Plus は、テンポラリ・ファイルを削除できません。

### エディタの順序設定

SQL\*Plus は、次の順序でデフォルトのエディタを検索します。

1. SQL\*Plus セッション中の `_editor` 変数
2. `login.sql` の `_editor` 変数
3. `glogin.sql` の `_editor` 変数
4. 環境変数 `EDITOR`
5. 環境変数 `VISUAL`

これらの値がどれも設定されていない場合、SQL\*Plus はエディタに `ed` を使用します。

## \_editor オプションの設定

SQL\*Plus の `_editor` オプションを設定するには、`login.sql` ファイルに次の行を追加します。

```
define _editor=editor_name
```

この場合、`editor_name` は UNIX エディタを表します。

## 環境変数の設定

Bourne シェルまたは Korn シェルの場合は、次のように入力して環境変数にデフォルト・エディタを設定します。

```
$ UNIX_VAR=editor_name; export UNIX_VAR
```

C シェルの場合は、次のように入力して環境変数にデフォルト・エディタを設定します。

```
% setenv UNIX_VAR editor_name
```

表 3-1 に環境変数の構文を示します。

**表 3-1 UNIX 環境変数の構文**

<code>UNIX_VAR</code>	環境変数 EDITOR または VISUAL
<code>editor_name</code>	UNIX エディタ（たとえば、vi や ed）

## デフォルト設定

システム・エディタを起動すると、現行の SQL バッファは編集バッファに入れられるため、エディタで利用できる文はすべて SQL 文を変更できます。SQL\*Plus はテンポラリ・ファイル `afiedt.buf` を使用します。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 からのオペレーティング・システム・コマンドの実行

SQL\*Plus プロンプトの後の最初の文字を感嘆符 (!) にすると、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` (Bourne シェル) です。シェルが実行できない場合は、エラー・メッセージが表示されます。

次の SQL\*Plus コマンドを使用すると、それぞれに固有のタスクを実行できます。

- `[!]+command` (1 つのオペレーティング・システム・コマンドを実行する場合)。コマンドの実行後、制御は SQL\*Plus に戻ります。
- `[!]+[Return]` (2 つ以上のオペレーティング・システム・コマンドを実行する場合)。

## SQL\*Plus への割込み

SQL\*Plus の実行中に、次のような割込みができます。

- [Ctrl]+[C] 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。
- SQL\*Plus プロンプトで [Ctrl]+[C] を押すと、別の SQL\*Plus プロンプトが表示されます。

## SPOOL コマンドの使用

SPOOL コマンドで生成されるファイルの拡張子のデフォルトは、.lst 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。

次に例を示します。

```
SQL> SPOOL query.lis
```

## 制限事項

### COPY コマンド

SQL\*Plus の COPY コマンドは、同じバージョンのオペレーティング・システムが実行されているマシン上で使用できます。制限事項はありません。

基本的に、COPY コマンドは異なるバージョンのオペレーティング・システムが実行されているマシン間でも動作しますが、COPY コマンドが失敗した場合、rcp または ftp を使用して接続をテストしてください。ベンダーが提供するネットワーク・ソフトウェアによっては、rcp、ftp または COPY コマンドがシステム間で正常に動作しない場合があります。

---

**注意：** rlogin コマンドは、大きなデータ・パケットを送信または受信しないため、接続のテストに適していません。

---

COPY コマンドがシステム間で動作しない場合、システムおよびユーザー ID へのデータベース・リンクを作成して、コピーする表を指定します。そのためには、次のように入力します。

```
SQL> create table newtable as (SELECT * FROM table@database_link_name)
```

これによって、リモート・システムにあるコピー元の表で選択された行および列が、ローカル・システム上のコピー先の表に挿入されます。

## ウィンドウのサイズ変更

SQL\*Plus の LINESIZE のデフォルト値は 80、PAGESIZE のデフォルト値は 25 です。これらの変数によって、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。



---

## AIX での 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

- Oracle プリコンパイラの概要
- Pro\*C/C++
- Pro\*COBOL
- Pro\*FORTRAN
- SQL\*Module for Ada
- Oracle コール・インタフェース
- Oracle プリコンパイラと Oracle コール・インタフェースのリンクおよび Make ファイル
- スレッドのサポート
- Oracle ライブラリを使用した静的リンクおよび動的リンク
- シグナル・ハンドラの使用法
- 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 DBMS で実行するオープンでカスタマイズされたアプリケーションを開発するために使用します。

## プリコンパイラ実行ファイルの再リンク

すべてのプリコンパイラ実行ファイルは、Make ファイル `$ORACLE_HOME/precomp/lib/ins_precomp.mk` を使用して再リンクされます。make コマンドの形式は、次のとおりです。

```
$ make -f ins_precomp.mk relink EXENAME=executable
```

このコマンドを実行すると、新しい実行ファイルが `$ORACLE_HOME/precomp/lib` ディレクトリに作成され、その後、その実行ファイルは `$ORACLE_HOME/bin` ディレクトリに移されます。`$ORACLE_HOME/bin` に移さずに新しい実行ファイルを作成するには、次のコマンドを使用します。

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

表 4-1 に、使用する製品に対する実行ファイル名を示します。

**表 4-1 製品と実行ファイル名**

製品	実行ファイル
Pro*C/C++ 8.1.5	proc
Pro*COBOL 1.8.50	procobl8 または rtsora
Pro*COBOL 8.1.5	procob または rtsora
Pro*FORTRAN 1.8.50	profor
Oracle SQL*Module for Ada 8.1.5	modada
Object Type Translator 8.1.5	ott

たとえば、Pro\*C/C++ の実行ファイルを再リンクするには、次のように入力します。

```
$ cd $ORACLE_HOME/precomp/lib
$ make -f ins_precomp.mk relink EXENAME=proc
```

## プリコンパイラ構成ファイル

\$ORACLE\_HOME/precomp/admin には、6 つの .cfg システム構成ファイルがあります。  
表 4-2 に、製品とその構成ファイルを示します。

表 4-2 システム構成ファイル

製品	構成ファイル
Pro*C/C++ 8.1.5	pccscfg.cfg
Pro*COBOL 8.1.5	pccbcfg.cfg
Pro*COBOL 1.8.50	pcccob.cfg
Pro*FORTRAN 1.8.50	pccfor.cfg
Oracle SQL*Module for Ada 8.1.5	pmscfg.cfg
Object Type Translator 8.1.5	ottcfg.cfg

## 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」エラーが発生する場合があります。この場合、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ガとプリコンパイラに互換性がない場合があります。デバッガ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。

## 参照マニュアル

次のドキュメントでは、プリコンパイラおよびインタフェース機能についての追加情報を説明しています。

- 『Pro\*C/C++ プリコンパイラ・プログラマーズ・ガイド』
- 『Pro\*COBOL プリコンパイラ・プログラマーズ・ガイド』
- 『Oracle8i コール・インタフェース・プログラマーズ・ガイド』
- 『Oracle8i アプリケーション開発者ガイド』
- 『Programmer's Guide to SQL\*Module for Ada』

## Pro\*C/C++

Pro\*C/C++ リリース 8.1.5 の追加情報は、README ファイル  
\$ORACLE\_HOME/precomp/doc/proc2/readme.doc を参照してください。

## Pro\*C/C++ の管理

### システム構成ファイル

Pro\*C/C++ のシステム構成ファイルは、\$ORACLE\_HOME/precomp/admin/pcscfg.cfg です。

## Pro\*C/C++ の使用

Pro\*C/C++ を使用する前に、適切なバージョンの AIX コンパイラがインストールされていることを確認してください。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』の第 1 章を参照してください。

### デモンストレーション・プログラム

Pro\*C/C++ プリコンパイラのさまざまな機能を参照するために、デモンストレーション・プログラムが提供されています。デモンストレーション・プログラムには、C、C++ および Object プログラムの 3 種類のデモンストレーション・プログラムがあります。Object プログラムのデモンストレーション・プログラムでは、Oracle8i Object の新機能が表示されます。すべてのデモンストレーション・プログラムは \$ORACLE\_HOME/precomp/demo/proc にあり、すべてのデモンストレーション・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。

SQL\*Plus を使用してデモンストレーション・プログラムを作成する場合の詳細は、3-2 ページの「デモンストレーション表」を参照してください。また、デモンストレーション・プロ

グラムの詳細は、『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。

Make ファイル `$ORACLE_HOME/precomp/demo/proc/demo_proc.mk` は、デモンストレーション・プログラムの作成に必要です。たとえば、`sample1` というデモンストレーション・プログラムをプリコンパイル、コンパイルまたはリンクする場合は、次のように入力します。

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

また、次のコマンドを実行しても、同様の結果になります。

```
$ make -f demo_proc.mk build OBJS=sample1.o EXE=sample1
```

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 ファイルでは、環境変数 `ORA_CLIENT_LIB` は使用されなくなりました。

すべての Pro\*C/C++ の C デモンストレーション・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
```

すべての Pro\*C/C++ の C++ デモンストレーション・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
```

すべての Pro\*C/C++ の Object デモンストレーション・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
```

デモンストレーション・プログラムによっては、実行する際に `$ORACLE_HOME/precomp/demo/sql` にある SQL スクリプトが必要な場合があります。このようなデモンストレーション・プログラムを作成し、SQL スクリプトを実行するには、コマンド行にマクロ作成引数 `RUNSQL=run` を追加する必要があります。たとえば、`calldemo` デモンストレーション・プログラムを作成し、必要な `$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
```

また、Object デモンストレーション・プログラムを作成し、必要な SQL スクリプトを実行するには、次のように入力します。

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

必要に応じて SQL スクリプトを手動で実行することもできます。

## ユーザー・プログラム

Make ファイル `$ORACLE_HOME/precomp/demo/proc/demo_proc.mk` は、ユーザー・プログラムを作成する場合に使用します。ユーザー・プログラムを `demo_proc.mk` にリンクする場合の構文は、次のとおりです。

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

たとえば、Pro\*C/C++ のソース `myprog.pc` からプログラム `myprog` を作成する場合、次のように入力します。

C ソースの場合で、クライアント共有ライブラリに動的にリンクする場合

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

C ソースの場合で、静的にリンクする場合

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

C++ ソースの場合で、クライアント共有ライブラリに動的にリンクする場合

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

C++ ソースの場合で、静的にリンクする場合

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

## Pro\*COBOL

今回のリリースの Pro\*COBOL には、次の 2 つのバージョンがあります。

- Pro\*COBOL 8.1.5
- Pro\*COBOL 1.8.50

表 4-3 に、この 2 つのバージョンのネーミング方法の違いを示します。

**表 4-3 Pro\*COBOL のネーミング方法の違い**

	Pro*COBOL 8.1.5	Pro*COBOL 1.8.50
<b>実行ファイル</b>	procob	procob18
<b>デモ・ディレクトリ</b>	procob2	procob
<b>Make ファイル (MicroFocus COBOL)</b>	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・プログラムでは、Oracle クライアント共有ライブラリ

`$ORACLE_HOME/lib/libclntsh.a` が使用されます。動的読み込みプログラムでは、`rtsora` 実行ファイルが使用されます。

Pro\*COBOL 8.1.5 の追加情報は、README ファイル  
`$ORACLE_HOME/precomp/doc/procob2/readme.doc` を参照してください。  
Pro\*COBOL 1.8.50 の追加情報は、README ファイル  
`$ORACLE_HOME/precomp/doc/prolx/readme.txt` を参照してください。

## Pro\*COBOL の管理

### システム構成ファイル

Pro\*COBOL 8.1.5 のシステム構成ファイルは、  
`$ORACLE_HOME/precomp/admin/pcbcfg.cfg` です。

Pro\*COBOL 1.8.50 のシステム構成ファイルは、  
`$ORACLE_HOME/precomp/admin/pcccob.cfg` です。

## 環境変数

### MicroFocus COBOL コンパイラ

MicroFocus COBOL コンパイラには、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。次に例を示します。

```
$ setenv COBDIR /usr/lpp/mfcobol
```

`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。

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

`LIBPATH` に `$COBDIR/coblib` が設定されていないと、プログラムをコンパイルしたときに次のエラーが発生します。

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

## Pro\*COBOL の使用

Pro\*COBOL を使用する前に、適切なバージョンの COBOL コンパイラがインストールされていることを確認してください。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』の第1章を参照してください。

## Pro\*COBOL 実行ファイルのリンク

Pro\*COBOL 実行ファイルをリンクする場合は、未定義のシンボルを解決するために、\$COBDIR/coblib/liblist ファイルに指定されたオブジェクト・ファイルの1つを変更する必要があります。\$COBDIR/coblib/liblist ファイルで、次の行を置き換えます。

```
i/lib/crt0.o  
s-lc
```

前述を次のように置き換えます。

```
i/lib/crt0_r.o  
s-lc_r
```

## 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。rtsora ランタイム・システムは、動的読込み Pro\*COBOL プログラムを実行する際に、MicroFocus の cobrun ランタイム・システムのかわりに使用します。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
```

## デモンストレーション・プログラム

デモンストレーション・プログラムは、Pro\*COBOL プリコンパイラのさまざま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 に保存されています。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は 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。

SQL\*Plus を使用してデモンストレーション・プログラムを作成する場合の詳細は、3-2 ページの「デモンストレーション表」を参照してください。また、デモンストレーション・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。

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必要です。Pro\*COBOL 8.1.5 のデモンストレーション Make ファイルは、\$ORACLE\_HOME/precomp/demo/procob2/demo\_procob.mk です。Pro\*COBOL 1.8.50 のデモンストレーション Make ファイルは、\$ORACLE\_HOME/precomp/demo/procob/demo\_procob18.mk です。たとえば、Pro\*COBOL 8.1.5 のデモンストレーション・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
```

また、次のコマンドを実行しても、同様の結果になります。

```
$ make -f demo_procob.mk build COBS=sample1.cob EXE=sample1
```

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 Pro\*COBOL デモンストレーション・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 を作成するには、次のように入力  
します。

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

その後、次のように入力して、rtsora でプログラムを実行します。

```
$ rtsora sample1.gnt
```

デモンストレーション・プログラムによっては、実行する際に  
\$ORACLE\_HOME/precomp/demo/sql にある SQL スクリプトが必要な場合があります。こ  
のようなデモンストレーション・プログラムを作成し、SQL スクリプトを実行するには、コ  
マンド行にマクロ作成引数 RUNSQL=run を追加する必要があります。

たとえば、sample9 デモンストレーション・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_procob.mk sample9 RUNSQL=run
```

必要に応じて SQL スクリプトを手動で実行することもできます。

## ユーザー・プログラム

デモンストレーション Make ファイルは、ユーザー・プログラムを作成する場合に使用しま  
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。ユーザー・プログラムをデモンストレーション Make ファ  
イルにリンクする構文は、次のとおりです。

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

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、次のいずれかのコマンドを実行します。

クライアント共有ライブラリに動的にリンクする実行ファイルの場合

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

クライアント共有ライブラリを使用しないで静的にリンクする実行ファイルの場合

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

rtsora で使用できる動的読み込みモジュールの場合

```
$ 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。さらに、カラム 8～11 は段落の名前で、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

Pro\*FORTRAN 1.8.50 の追加情報は、README ファイル  
\$ORACLE\_HOME/precomp/doc/pro1x/readme.txt を参照してください。

## Pro\*FORTRAN の管理

### システム構成ファイル

Pro\*FORTRAN のシステム構成ファイルは、  
\$ORACLE\_HOME/precomp/admin/pccfor.cfg です。

## Pro\*FORTRAN の使用

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。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』の第 1 章を参照してください。

### デモンストレーション・プログラム

Pro\*FORTRAN プリコンパイラのさまざまな機能を参照するために、デモンストレーション・プログラムが提供されています。すべてのプログラムは、  
\$ORACLE\_HOME/precomp/demo/profor にあり、すべてのデモンストレーション・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。

SQL\*Plus を使用してデモンストレーション・プログラムを作成する場合の詳細は、3-2 ページの「デモンストレーション表」を参照してください。デモンストレーション・プログラムの詳細は、『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。

Make ファイル \$ORACLE\_HOME/precomp/demo/profor/demo\_profor.mk は、デモンストレーション・プログラムの作成に必要です。たとえば、sample1 というデモンストレーション・プログラムをプリコンパイル、コンパイルまたはリンクする場合は、次のように入力します。

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

また、次のコマンドを実行しても、同様の結果になります。

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

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 Pro\*FORTRAN デモンストレーション・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
```

デモンストレーション・プログラムによっては、実行する際に \$ORACLE\_HOME/precomp/demo/sql にある SQL スクリプトが必要な場合があります。このようなデモンストレーション・プログラムを作成し、SQL スクリプトを実行するには、コマンド行にマクロ作成引数 RUNSQL=run を追加する必要があります。たとえば、sample11 デモンストレーション・プログラムを作成し、必要な \$ORACLE\_HOME/precomp/demo/sql/sample11.sql スクリプトを実行する場合は、次のように入力します。

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

必要に応じて SQL スクリプトを手動で実行することもできます。

## ユーザー・プログラム

Make ファイル \$ORACLE\_HOME/precomp/demo/profor/demo\_profor.mk は、ユーザー・プログラムを作成する場合に使用します。ユーザー・プログラムを demo\_profor.mk にリンクする場合の構文は、次のとおりです。

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

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、次のように入力します。

クライアント共有ライブラリに動的にリンクする実行ファイルの場合

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

静的にリンクする実行ファイルの場合

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

## SQL\*Module for Ada

### SQL\*Module for Ada の管理

#### システム構成ファイル

Oracle SQL\*Module のシステム構成ファイルは、  
\$ORACLE\_HOME/precomp/admin/pmscfg.cfg です。

### SQL\*Module for Ada の使用

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。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』の第 1 章を参照してください。

#### デモンストレーション・プログラム

デモンストレーション・プログラムは、SQL\*Module for Ada のさまざま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/modada に格納されています。

デモンストレーション・プログラム ch1\_drv 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。

デモンストレーション・プログラム demcalsp および demohost では、  
\$ORACLE\_HOME/precomp/demo/sql/mktables.sql で作成した大学のサンプル・データベースが MODTEST スキーマにあると仮定しています。

すべてのプログラムでは、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、適切な表が存在するデータベースに接続できると仮定しています。

SQL\*Plus を使用してデモンストレーション・プログラムを作成する場合の詳細は、3-2 ページの「デモンストレーション表」を参照してください。また、デモンストレーション・プログラムの詳細は、『Programmer's Guide to SQL\*Module for Ada』を参照してください。

Make ファイル \$ORACLE\_HOME/precomp/demo/modada/demo\_modada.mk は、デモンストレーション・プログラムの作成に必要です。たとえば、ch1\_drv というデモンストレーション・プログラム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_modada.mk ch1_drv
```

また、次のコマンドを実行しても、同様の結果になります。

```
$ make -f demo_modada.mk ada OBJS="ch1_mod.ada ch1_drv.ada" \
EXE=ch1_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 デモンストレーション・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
```

サンプル・プログラム demcalasp および demohost では、MODTEST スキーマの  
\$ORACLE\_HOME/precomp/demo/sql/mktables.sql で作成した大学のサンプル・データベースが必要です。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  
ユーザーおよび大学のサンプル・データベースを作成し、デモンストレーション・プログラムを作成する場合、作成ターゲットとして makeuser および loaddb を使用します。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、次の構文のコマンドを実行します。

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

必要に応じて SQL スクリプトを手動で実行することもできます。

すべての SQL\*Module for Ada デモンストレーション・プログラムを作成し、必要な SQL スクリプトを実行して MODTEST ユーザーを作成し、大学のサンプル・データベースを作成する場合、次のように入力します。

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

## ユーザー・プログラム

Make ファイル \$ORACLE\_HOME/precomp/demo/modada/demo\_modada.mk は、ユーザー・プログラムを作成する場合に使用します。ユーザー・プログラムを demo\_modada.mk にリンクする構文は、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・インタフェースの使用

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。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』の第 1 章を参照してください。

#### デモンストレーション・プログラム

デモンストレーション・プログラムは、OCI のさまざまな機能を参照するために提供されています。デモンストレーション・プログラムには、C および C++ の 2 種類があります。すべてのデモンストレーション・プログラムは、`$ORACLE_HOME/rdbms/demo` にあります。ほとんどのデモンストレーション・プログラムでは、`$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。

SQL\*Plus を使用してデモンストレーション・プログラムを作成する場合の詳細は、3-3 ページの「デモンストレーション表」を参照してください。デモンストレーション・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。

Make ファイル `$ORACLE_HOME/rdbms/demo/demo_rdbms.mk` は、デモンストレーション・プログラムの作成に必要です。たとえば、`cdemo1` というデモンストレーション・プログラム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
```

また、次のコマンドを実行しても、同様の結果になります。

```
$ make -f demo_rdbms.mk build OBJS=cdemo1.o EXE=cdemo1
```

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 OCI の C デモンストレーション・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
```

すべての OCI の C++ デモンストレーション・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
```

デモンストレーション・プログラムによっては、プログラムを実行する前に、`$ORACLE_HOME/rdbms/demo`にある SQL スクリプトを実行する必要があります。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。たとえば、プログラム `oci02` の SQL スクリプトは `oci02.sql` です。

プログラムの先頭にコメントがある場合は、そのコメントから必要な SQL スクリプトを判断します。

## ユーザー・プログラム

Make ファイル `$ORACLE_HOME/rdbms/demo/demo_rdbms.mk` は、ユーザー・プログラムを作成する場合に使用します。ユーザー・プログラムを `demo_rdbms.mk` にリンクする構文は、次のとおりです。

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

たとえば、C のソース `myprog.c` からプログラム `myprog` を作成する場合、次のいずれかのコマンドを実行します。

C ソースの場合で、クライアント共有ライブラリに動的にリンクする場合

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

C ソースの場合で、静的にリンクする場合

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

C++ のソース `myprog.cc` からプログラム `myprog` を作成する場合は、次のいずれかのコマンドを実行します。

C++ ソースの場合で、クライアント共有ライブラリに動的にリンクする場合

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

C++ ソースの場合で、静的にリンクする場合

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



## Oracle プリコンパイラと Oracle コール・インタフェースのリンクおよび Make ファイル

### カスタム Make ファイル

この章で説明している各製品には、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は、次のことに注意してください。

- Oracle ライブラリの順番は変更しないでください。
- リンク行に独自のライブラリを追加する場合は、リンク行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。
- `nmake` または `GNU make` などの `make` ユーティリティを使用する場合は、マクロおよび接尾辞の処理について、AIX で提供されている `make` ユーティリティ (`/usr/ccs/bin/make`) との違いに注意してください。Oracle の Make ファイルは、すでにテスト済みで、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

プログラムのリンク時に共通するエラーに、次のような未定義シンボルがあります。

```
$ make -f demo_proc.mk sample1
ld: 0711-317 ERROR: Undefined symbol: .sqlcex
ld: 0711-317 ERROR: Undefined symbol: .sqlglm
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。通常、このような問題を解決するには、シンボルが定義されているライブラリまたはオブジェクト・ファイルがリンク行にあるかどうか、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。

オラクル社が提供している `symfind` という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のに役立ちます。次に、シンボル `sqlcex` が定義されている場所を調べる `symfind` の出力例を示します。



## Oracle 共有ライブラリ

Oracle 共有ライブラリは、`$ORACLE_HOME/lib/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。

環境変数 `LIBPATH` は、プロセスの起動時にランタイム・ローダーが Oracle 共有ライブラリを検索できるように設定する必要があります。実行ファイルの起動時に次のエラーが発生する場合は、`LIBPATH` に Oracle 共有ライブラリがあるディレクトリを設定してください。

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

`LIBPATH` を次のように設定します。

```
% setenv 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、*oracle* ユーザーでログインして、次のように入力します。

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

## シグナル・ハンドラの使用法

ここでは、2 タスク通信で Oracle8i が使用するシグナル、および独自のシグナル・ハンドラを設定する方法について説明します。

### シグナル

シグナルは、データベースに接続するとユーザー・プロセスに作成され、切断すると削除されます。

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

表 4-4 2 タスク通信のためのシグナル

シグナル	用途
SIGCONT	アウトオブバンド・ブレイクをユーザー・プロセスから oracle プロセスに送信する場合に、パイプ 2 タスク・ドライバが使用します。
SIGINT	ユーザーの割り込み要求を検出する場合に、すべての 2 タスク・ドライバが使用します。SIGINT は、oracle ではなく、ユーザー・プロセスが受け取ります。
SIGPIPE	通信チャネルでファイルの終わりを検出するために、パイプ・ドライバが使用します。パイプへの書き込み時に、読み込み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	パイプ・ドライバが使用します。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。oracle プロセスが異常終了すると、UNIX カーネルはユーザー・プロセスに SIGCLD 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 <code>wait()</code> を使用します）。SIGCLD は、oracle ではなく、ユーザー・プロセスが受け取ります。
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。
SIGIO	ネットワーク・イベントの着信を示すために、Oracle Net8 プロトコル・アダプタが使用します。
SIGURG	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、Oracle Net8 TCP/IP ドライバが使用します。

ここに示されているシグナルは、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() ルーチンを呼び出して設定すると、複数のシグナル・ハンドラを SIGINT 用にインストールできます。その他のシグナルには、必要なだけのシグナル・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。

### シグナル・ルーチンの例

次に、独自のシグナル・ルーチンおよび受取りルーチンを設定する方法を示します。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。

```
/* 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。

```
/*
** My own user interrupt handler.
*/
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
Do something appropriate here. */
...
}
...
/* clear my interrupt handler */
if (err = osncui(handle))
{
/* if the return value is non-zero, an error has occurred
Do something appropriate here. */
...
}
...
}
```

## XA 機能

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）がリンク行で、Oracle 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（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、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 ライブラリに再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。

# 5

---

## Net8 の構成

- 参照ドキュメント
- 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
- SPX/IPX プロトコル
- APPC/LU6.2 プロトコル
- Net8 Naming Supprt
- Oracle Enterprise Manager (OEM)
- Oracle SNMP での Oracle Intelligent Agent の構成
- Oracle Advanced Security

## 参照ドキュメント

Net8 の特徴の詳細は、次のドキュメントを参照してください。

- 『Oracle8i Net8 管理者ガイド』
- 『Oracle Networking Quick Reference Card for Net8』
- 『Oracle8i Advanced Security 管理者ガイド』
- 『Oracle Cryptographic Toolkit Programmer's Guide』

## README ファイルへの補足情報

表 5-1 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 5-1 Oracle 製品の README ファイルの保存場所

製品	README ファイル
Net8	<code>\$ORACLE_HOME/network/doc/README.Net8</code>
Oracle Advanced Security	<code>\$ORACLE_HOME/network/doc/README.OAS</code>
Oracle Intelligent Agent	<code>\$ORACLE_HOME/network/doc/README.oemagent</code>

## 主な Net8 製品および特徴

**参照：** サンプル・ファイルの詳細は、『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 および Connection Manager のグローバル・ファイルは、デフォルトで `/etc` ディレクトリに保存されています。

Net8 および Connection Manager 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` に設定したディレクトリ（設定した場合）
2. `/etc` ディレクトリ
3. `$ORACLE_HOME/network/admin`



ファイルがデフォルトのディレクトリにない場合、すべてのネットワーク・ユーザーの起動ファイルの環境変数 TNS\_ADMIN で、異なるディレクトリを設定します。

C シェルの場合、次のように入力します。

```
% 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）があります。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。

## サンプル構成ファイル

cman.ora、listner.ora、names.ora、sqlnet.ora および tnsnames.ora のサンプル構成ファイルは、\$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、さまざまなネットワーク・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。それぞれのソフトウェア・モジュールであるアダプタは、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、次のコマンドを実行すると、oracle 実行ファイルにリンクされているアダプタが表示されます。

```
% adapters oracle
Net Protocol Adapters linked with oracle are:
    BEQ
    IPC
    TCP/IP
    RAW
Net Naming Adapters linked with oracle are:
    Oracle TNS Naming Adapter
    Oracle Naming Adapter
    HOST TNS Naming Adapter
    LDAP Naming Adapter
Oracle Advanced Security/Network Security products linked with oracle are:
```

## Oracle Connection Manager

**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。

## マルチスレッド・サーバー

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

## Oracle Names

**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 は、Net8 Assistant (`$ORACLE_HOME/bin/net8asst`) とともにインストールされます。Net8 Assistant のコマンド・スクリプトを実行すると、その他の Java 製品をシステムにインストールしたかどうかに関係なく、Java のコマンド・スクリプトが実行されます。JAVA のコマンド・スクリプトは、JRE 1.1.7 とともに提供されています。

**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 では、Net8 バージョン 8.1.5 に対して次のプロトコルがサポートされています。

- 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
- SPX/IPX プロトコル
- APPC/LU6.2 プロトコル

TCP/IP、APPC/LU6.2 または SPX/IPX Net8 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 Net8 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 指定

IPC、TCP/IP、APPC/LU6.2 および SPX/IPX Net8 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 `init.ora` ファイル中の `MTS_LISTENER_ADDRESS` データベースの初期化パラメータ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 5-2 に、各プロトコルの ADDRESS 指定を示します。

表 5-2 ADDRESS 指定

サポート・ プロトコル	ADDRESS 指定
BEQ	<pre> (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' ) ) </pre>
IPC	<pre> (ADDRESS=   (PROTOCOL=IPC)   (KEY=key) ) </pre>
RAW	N/A
TCP/IP	<pre> (ADDRESS=   (PROTOCOL=TCP)   (HOST=hostname)   (PORT=port_id) ) </pre>
SPX/IPX	<pre> (ADDRESS=   (PROTOCOL=SPX)   (SERVICE=servicename) ) </pre>
APPC/LU6.2	<pre> (ADDRESS=   (PROTOCOL=LU62)   (TP_NAME=transaction_program_name)   (SIDE_INFO_PROFILE=side_info_profile)   (MODE=mode_name)   (PLU=partner_lu_name) ) </pre>

## 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。サービス名を指定（コマンド行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、専用のサーバーが使用され、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ネットワーク・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 の ADDRESS 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。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 5-3 に、BEQ プロトコルの接続パラメータの構文を示します。

**表 5-3 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。

例 5-1 に、BEQ ADDRESS の例を示します。

### 例 5-1 クライアントを指定するための BEQ ADDRESS

```
(ADDRESS =
  (PROTOCOL = BEQ)
  (PROGRAM = /u01/app/oracle/product/8.1.5/bin/oracle)
  (ARGV0 = oracleV815)
  (ARGS = ' (DESCRIPTION=(LOCAL=YES) (ADDRESS=(PROTOCOL=BEQ))) ' )
  (ENVS = 'ORACLE_HOME=/u01/app/oracle/product/8.1.5,ORACLE_SID=V815' )
)
```

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。

## 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、ネットワーク・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 (IPC) 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 (/var/tmp/o)、および現在保存されているディレクトリ (/var/tmp/.oracle) 間にシンボリック・リンクを作成します。

## IPC の ADDRESS 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。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 5-4 に、IPC プロトコルの接続パラメータの構文を示します。

**表 5-4 IPC 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 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。

例 5-2 に、IPC ADDRESS の例を示します。

**例 5-2 クライアントを指定するための IPC ADDRESS**

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

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。

## RAW プロトコル

データがクライアントとサーバー間で転送されると、Net8 は、そのヘッダー情報を各パケット（ネットワークを介して送信される情報のひとまとまり）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。たとえば、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

Net8 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、Net8 リスナー用のポートを予約してください。ほとんどの場合、ポートは 1521 です。リスナー名とポート番号という形式で入力します。たとえば、次のように入力します。

```
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してください。

## TCP/IP の ADDRESS 指定

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。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 5-5 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 5-5 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 <code>1521</code> です。

例 5-3 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。

**例 5-3 クライアントを指定するための TCP/IP ADDRESS**

```
(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。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。



## SPX/IPX プロトコル

Oracle SPX/IPX プロトコルは、Oracle8i と、DOS、UNIX、OS/2 または Netware OS で実行するクライアント・アプリケーションとの間の透過的な 2 タスク通信インタフェースをサポートします。

### ntisbsdsm 同報通信デーモン

クライアントは名前を使用し、その名前を SPX アドレスに変換してサーバーを識別し、サーバーと通信します。Netware バインダリは、変換メカニズムを提供するディレクトリ・サービスです。サーバーがバインダリに登録されると、そのアドレスのバインダリを定期的に通知します。これは、サーバー通知プロトコル (SAP) を使用して行われます。

サーバーは、IPX データグラムで 60 秒ごとに SAP パケットを同報通信します。この SAP パケットには、関連するすべてのアドレス情報が含まれています。その後、どのクライアントからも、必要なサーバーのアドレスに最も近いサーバーに対して問合せできます。

Oracle SPX/IPX プロトコルは、\$ORACLE\_HOME/bin の ntisbsdsm 同報通信デーモンを使用して同報通信を行います。ntisbsdsm は、ntspcxctl ユーティリティによって起動および終了されます。

Oracle SPX/IPX プロトコル・アダプタを使用する前に、Network Transport が動作していることを確認してください。

1. 次のコマンドを実行して、Network Transport が動作していることを確認します。

```
# ps -aef | grep npsd
```

2. Network Transport が実行されていない場合は、次のようにして実行します。

```
# cd /usr/lpp/netware/bin
# sconsole
```

3. SP デーモンおよび Netware プロトコル・スタックが実行されていることを確認します。

```
# ps -aef | grep npsd
# ps -aef | grep sapd
```

### ntspcxctl ユーティリティ

ntspcxctl ユーティリティには、名前の登録と取消し、およびバインダリの間合せを実行する機能が含まれています。また、同報通信デーモンを起動または停止する場合に使用します。リスナーは、このデーモンを使用して、使用中のサービス名を登録します。

例 5-4 に、ntspcxctl ユーティリティのユーザーが複数の場合の例を示します。

**例 5-4 ntspxctl ユーティリティの使用**

ntspxctl ユーティリティは、コマンド行からコマンドを読み取ります。パラメータを指定していない場合、それらのパラメータを指定するためのプロンプトが表示されます。

ntspxctl を起動するには、次のように入力します。

```
$ ntspxctl
```

次のように表示されます。

```
ntspxctl: Version 8.1.5 - on Fri Jul 3 11:43:50 1998
```

同報通信デーモンを起動するには、次のように入力します。

```
ntspxctl> startup
```

次のように表示されます。

```
ntisbsdsm started at Fri Jul 3 11:43:47 1998
```

デーモンがすでに起動されている場合は、システム・メッセージが表示されます。

同報通信デーモンは、マシンの起動時に常に自動的に起動されるように設定する必要があります。/etc/inittab ファイルにエントリを追加して、デーモンの起動を自動化します。たとえば、システムの起動時に ntisbsdsm を起動するには、/etc/inittab に次の行を追加します。

```
ntspxctl:2:once:/u/oracle/bin/ntisbsdsm &
```

この場合、/u/oracle は \$ORACLE\_HOME へのフルパス名を示します。

テスト用の名前を登録するには、register および名前を入力します。

```
ntspxctl> register
```

これによって ntisbsdsm が所有するソケットが作成され、登録されます。

次のメッセージが表示されます。

```
Name YYY successfully registered
YYY address 00eee045:000000000001:4454
```

ntisbsdsm の状態を表示するには、次のように入力します。

```
ntspxctl> status
```

または

```
ntspxctl> summary
```

次のメッセージが表示されます。

```
ntisbsdsm started at Fri Jul 3 11:43:47 1998
Tracing is off
Pid: 14784 YYY
```

## SPX/IPX プロトコルのコマンドの要約

表 5-6 に、SPX/IPX プロトコルの help コマンドを示します。

**表 5-6 help コマンド**

<i>register name</i>	エントリを登録します。
<i>remove name</i>	エントリを取り消します。
<i>shutdown [force]</i>	ntisbsdsm を停止します。
<i>startup</i>	状態の要約を表示します。
<i>tracemon</i>	トレースをアクティブにします。
<i>traceoff</i>	トレースを非アクティブにします。
<i>status</i>	状態の詳細情報を表示します。
<i>getname name   hex_number</i>	ネーム・サービスの問合せを実行します。
<i>exit</i>	プログラムを終了します。
<i>help [command]</i>	コマンド情報を印刷します。
<i>!</i>	シェルを取り消します。

## getname コマンド

getname コマンドは、Novell システムに名前を要求する場合に使用します。同報通信デーモンは必要ありません。

次のように入力します。

```
getname name servicetype
```

次のメッセージが表示されます。

```
getname name servicetype (address number_of_hops)
```

表 5-7 に、getname コマンドの構文を示します。

**表 5-7 getname コマンドの構文**

<i>name</i>	入力した名前です。
<i>servicetype</i>	Novell で割り当てられた番号です。Oracle の番号は 103 です。
<i>address</i>	入力した名前のアドレスです。
<i>number_of_hops</i>	宛先までのホップ数（16 進数）です。値が 10 の場合は、名前が登録解除されたことを表します。SAP 問合せがサポートされていない場合、値は 0000 です。

付けることができるすべての名前を表示するには、次のように入力します。

```
getname * *
```

例 5-5 に、getname コマンドで名前を付ける例を示します。

**例 5-5 getname コマンドの使用**

```
ntspxctl> getname YYY *
YYY servertype x0103 address 00eee045:000000000001:
4465 hops 0000
ntspxctl> getname * 103
LSNR servertype x0103 address 00eee053:000000000001:
502c hops 0000
IBM6 servertype x0103 address 00eee058:000000000001:
507f hops 0000
DESK servertype x0004 address 00eee055:000000000001:
5451 hops 0000
DESK servertype x0107 address 00eee055:000000000001:
5104 hops 0000
CXY4 servertype x009e address 00eee055:000000000001:
5063 hops 0000
IBM2 servertype x0004 address 00eee057:000000000001:
5451 hops 0000
```

ntisbsdsm を停止するには、次のように入力します。

```
ntspxctl> shutdown
```

daemon は、名前がまだ登録されている場合は停止しません。次のメッセージが表示されます。

```
1 names are registered
ntisbsdsm not stopped
```

エントリを取り消すには、remove および名前を入力します。名前が YYY の場合の例を次に示します。

```
ntspxctl> remove YYY
```

次のメッセージが表示されます。

```
Name xxxremoved.  
ntspxctl> shutdown  
ntisbsdsm stopped
```

強制終了するには、次のように入力します。

```
ntspxctl> shutdown force
```

次のメッセージが表示されます。

```
ntisbsdsm stopped
```

## SPX/IPX の ADDRESS 指定

SPX/IPX プロトコルおよび Oracle SPX/IPX プロトコルがシステムにインストール済みの場合、TNS 接続記述子を指定した SPX/IPX パラメータを使用して、SPX/IPX コミュニティ・ノードを識別できます。

ADDRESS のキーワードとその値の組合せが SPX/IPX プロトコル・パラメータです。

```
(ADDRESS=  
  (PROTOCOL=SPX)  
  (SERVICE=service)  
)
```

表 5-8 に、SPX/IPX プロトコル接続の構文を示します。

**表 5-8 SPX/IPX プロトコル接続の構文**

PROTOCOL	プロトコル名を指定します。SPX/IPX の場合、値は spx です。
SERVICE	ネットワーク上のアプリケーションを示す一意の名前を 30 文字以下で指定します。サービスには起動中に名前が付けられ、ネットワーク全体で使用できます。クライアントは、ネットワーク・ディレクトリのパイナダリの検索を使用してこのサービスを参照します。

例 5-6 に、リモート・サーバー上にサービス MAILDB1 を指定する SPX/IPX ADDRESS を示します。

**例 5-6 SPX/IPX プロトコル接続**

```
(ADDRESS=
  (PROTOCOL=SPX)
  (SERVICE=MAILDB1)
)
```

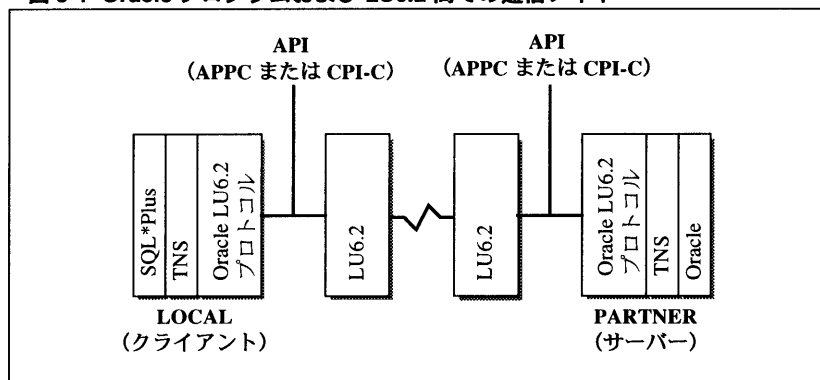
通常、ADDRESS は、接続記述子、構成ファイルなどを構成している内容の一部です。

**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、標準インタフェースとして API を使用できるようになります。

図 5-1 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 5-1 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 5-9 に、APPC/LU6.2 プロトコル接続の構文を示します。

**表 5-9 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 <code>lu62</code> 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	このパラメータには、リモート TP と対話する場合に使用する特性を指定します。AIX SNA 構成に対応するプロファイルの名前を指定します。
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 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。 <code>modename</code> は、ローカル LU およびパートナ LU の両方に共通でなければなりません。この値は必須です。
PLU	パートナ LU の名前を指定します。

## Net8 Naming Supprt

NIS Naming Support の構成の詳細は、『Oracle8i Net8 管理者ガイド』を参照してください。

## Oracle Enterprise Manager (OEM)

### エージェント・サービスの展開および自動構成

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

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 の構成』を参照)。

### Tcl スクリプトのデバッグ

実行ファイル `oracclsh` は、Tcl スクリプトをデバッグする場合に必要です。`oracclsh` 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 Intelligent Agent ユーザーズ・ガイド』を参照してください。

## Oracle SNMP での Oracle Intelligent Agent の構成

Oracle Intelligent Agent が動作するために、Simple Network Management Protocol (SNMP) は必要ありませんが、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` ファイルは、次のように変更されます。

1. 次の行を追加します。

```
snmp 0.0 "" <ip_address>
```

この場合、`<ip_address>` は oracle サブエージェントを実行しているホストの IP アドレスです。

### 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` が実行されていないことを確認します。

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

実行されている場合、次に進む前に `root` ユーザーでログインして、`kill` コマンドを使用してプロセスを終了させます。

2. `root` ユーザーでネイティブな AIX SNMP エージェントを起動するために `startsrc` スクリプトを実行します。次のように入力します。

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

このコマンドは、`snmpd` デーモンを起動して `/tmp/snmpd.log` に情報を記録します。

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 のインストール中に、3 つの .bak ファイル（naeet.o.bak、naect.o.bak および naedhs.o.bak）が作成されます。これらのファイルは、\$ORACLE\_HOME/lib に保存されています。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ので、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、『Oracle8i Advanced Security 管理者ガイド』を参照してください。

### DCE 統合

**参照：**DCE 統合の構成の詳細は、『Oracle8i Advanced Security 管理者ガイド』を参照してください。



---

## Oracle データ・オプション・デモの実行

- 参照ドキュメント
- Oracle8i interMedia
- Oracle8i Time Series デモ
- Oracle8i Visual Information Retrieval
- Oracle8i Spatial

## 参照ドキュメント

リリース 8.1.5 で使用できる Oracle オプションの詳細は、次のドキュメントを参照してください。

- 『Oracle8i *interMedia* Audio, Image, Video ユーザーズ・ガイドおよびリファレンス』
- 『Oracle8i *interMedia* Audio, Image, Video Java Client ユーザーズ・ガイドおよびリファレンス』
- 『Oracle8i *interMedia* Web』
- 『Oracle8i *interMedia* Locator ユーザーズ・ガイドおよびリファレンス』
- 『Oracle8i *interMedia* Text リファレンス』
- 『Oracle8i *interMedia* Text 移行ガイド』
- 『Oracle8i Visual Information Retrieval User's Guide』
- 『Oracle8i Visual Information Retrieval Java Client User's Guide』
- 『Oracle8i Time Series ユーザーズ・ガイド』
- 『Oracle8i Spatial ユーザーズ・ガイドおよびリファレンス』

## 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

**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。

Oracle8i には、Text のデモはありません。

## Audio、Video および Image

**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 Client 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 *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。
  - 行外をチェックします。
  - 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`) を使用したユーザー独自形式のプラグインのインストールです。これらのファイルについては、次の2つの項目で説明します。また、新しい Audio データ形式をサポートするための *interMedia* Audio の拡張方法については、『Oracle8i *interMedia* Audio, Image, Video ユーザーズ・ガイドおよびリファレンス』で説明しています。
- `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。

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* デモンストレーション・プログラムの実行を選択できます。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・ファイルは、\$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 (デモ)・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。

```
% sqlplus
SQL> connect internal;
SQL> create user SCOTT identified by tiger;
SQL> 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。

```
% sqlplus
SQL> connect internal;
SQL> 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。

```
SQL> 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_ording.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、コマンド行でイメージ・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・コピーを更新するだけ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 6-1 コマンド行でのデモの実行

コマンド行で `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行でそのファイル名を指定します。

次のコマンドを入力します。

```
$ 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。たとえば、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。
  - 行外をチェックします。
  - 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`) を使用したユーザー独自形式のプラグインのインストールです。これらのファイルについては、次の2つの項目で説明します。また、新しい Video データ形式をサポートするための *interMedia Video* の拡張方法については、『Oracle8i *interMedia Audio, Image, Video ユーザーズ・ガイド* および *リファレンス*』で説明しています。
- `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。

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のに役立ちます。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` ディレクトリに保存されています。また、Video の Java デモ・ファイルは、`$ORACLE_HOME/ord/vid/demo` ディレクトリ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。



## Locator

**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 interMedia Locator には、変更および実行できる多数のスクリプトがあります。

### 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（最終行フィールド）」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（最終行フィールド）」の例 2 または「GEOCODE1 ファンクション（最終行フィールド）」の例 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 マイル以上の距離の半径を使用する問合せを、距離内の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 2-17 ページの「`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、関心事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「LOCATOR\_WITHIN\_DISTANCE」の例 2 を参照してください。

## Oracle8i Time Series デモ

**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。

表 6-1 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。

**表 6-1 Oracle8i Time Series デモ**

説明	ディレクトリ
クイック・スタート・デモは、素早く簡単に 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 1.6.1 項を参照）。	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 1.6.2 項を参照）。	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 1.7 項を参照）。	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 1.7 項を参照）。	demo/java
管理ツールを使用して既存の Time Series 元表をレトロフィットする手順を示すデモです。また、Time Series 元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

表 6-1 Oracle8i Time Series デモ

説明	ディレクトリ
上級開発者デモは、Oracle8i Time Series 機能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 機能を呼び出す方法を示します。	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 機能を呼び出す方法を示します。	demo/proc
Oracle 開発者デモは、Oracle Forms アプリケーションでの Oracle8i Time Series 機能を呼び出す方法を示します。	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。

## Oracle8i Visual Information Retrieval

**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。

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ていて、PUBLIC READ にアクセス権限を与えられていて、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・プログラムを実行するには、2つの方法があります。1つは、含まれているサンプル・イメージを使用する方法、もう1つはユーザー独自のイメージを使用する方法です。

例 6-2 に、含まれているイメージ・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 6-2 含ま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 6-3 に、コマンド行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 6-3 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
```

6つすべてのパラメータである2つのファイル名および4つの属性の重み（0.0 から 1.0 まで）が、このサンプル・プログラムでは指定されている必要があります。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、少なくとも1つの属性の重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』の Appendix B を参照してください。

## Oracle8i Spatial

**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。

Oracle8i Spatial には、デモが含まれていません。



---

## 最適フレキシブル・アーキテクチャ

- 最適フレキシブル・アーキテクチャ (OFA)
- UNIX で適用される OFA

## 最適フレキシブル・アーキテクチャ（OFA）

オラクル社では、Oracle8i に対して OFA 標準を適用することをお薦めしています。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 には、次の目的があります。

- デバイスのボトルネックやパフォーマンスの低下が起きないように、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（ソフトウェアやデータのバックアップ機能など）を簡略化します。
- 複数の Oracle データベースで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、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

OFA に準拠したデータベースには、次の利点があります。

### ファイル・システムの編成

ファイル・システムは、管理が容易で、かつ次のシステム拡張に対応できるように編成されています。

- 既存のデータベースへのデータの追加
- ユーザーの追加
- データベースの作成
- ハードウェアの追加

### I/O 負荷の分散

I/O 負荷を十分な数のディスク・ドライブに分散させるので、パフォーマンスのボトルネックが起きません。

### ハードウェア・サポート

操作上の考慮事項に矛盾しない範囲で、ハードウェア・コストを最小限に抑えられます。



## 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、ドライブに障害が起きた場合でも、アプリケーションへの影響を最小限に抑えられます。

## ホーム・ディレクトリの分散

次のものを、複数のディスク・ドライブに分散できます。

- 複数のホーム・ディレクトリ
- 各ホーム・ディレクトリの内容

## ログイン・ホーム・ディレクトリの整合性

ログイン・ホーム・ディレクトリを参照するプログラムを修正しなくても、ログイン・ホーム・ディレクトリを追加、移動または削除できます。

## UNIX ディレクトリ・サブツリーの独立性

ファイルのカテゴリは、独立した UNIX ディレクトリ・サブツリーに分類されています。そのため、あるカテゴリのファイルでの操作によって、別のカテゴリのファイルが受ける影響を最小限に抑えることができます。

## アプリケーション・ソフトウェアの同時実行サポート

複数バージョンのアプリケーション・ソフトウェアを同時に実行できます。これによって、ユーザーはアプリケーションの古いバージョンを削除しなくても、新しいリリースのアプリケーションを使用できます。アップグレード後に新しいバージョンへ移行する作業は、管理者にとっては簡単な作業で、ユーザーがその作業を意識する必要はありません。

## 各データベースの管理情報の区別

データベースごとに管理情報を区別できるため、管理データをわかりやすい構造で編成および保存できます。

## データベース・ファイルのネーミング規則

データベース・ファイルには、次のネーミング規則があります。

- 他のファイルと区別しやすい名前にします。
- 2つのデータベースに属するファイルが、どちらのデータベースに属するかを区別できる名前にします。
- 制御ファイル、REDO ログ・ファイルおよびデータ・ファイルであることが識別できる名前にします。
- データ・ファイルと表領域の関係が明確に識別できる名前にします。

### 表領域の内容の分割

表領域の内容が分割されているため、次の利点があります。

- 表領域内の空き領域の断片化を最小限に抑えられます。
- I/O 要求の競合を最小限に抑えられます。
- 管理面の柔軟性が高くなります。

### すべてのドライブに分散されている I/O 負荷のチューニング

Oracle データを RAW デバイスに保存しているドライブを含め、すべてのドライブへの I/O 負荷をチューニングします。

### Parallel Server における OFA のその他の利点

Oracle Parallel Server のインストレーションでは、他にも次の利点があります。

- すべての管理データを一元管理できます。
- あるインスタンスの管理データは、ファイル名によってそのインスタンスと対応付けることができます。

## UNIX で適用される OFA

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、データ管理のさまざまな問題を回避できます。ここで説明している OFA 規則は、『The OFA Standard: Oracle8i for Open Systems』（Cary V. Millsap 著）に記載されている OFA 推奨事項に対応しています。

## マウント・ポイントの名前

### マウント・ポイントの構文

すべてのマウント・ポイントは、*/pm* という構文で名前を付けます。この場合、*p* は文字列定数、*m* は各マウント・ポイントを区別するための一意の固定長キー（通常、2 桁の数字）です。たとえば、*/u01* および */u02*、*/disk01* および */disk02* などです。

### 超大規模データベース（VLDB）のマウント・ポイントの名前

1 つのアプリケーションが使用するデータベース・ファイルが各ディスク・ドライブにあり、各データベースで I/O のボトルネックが発生しないほど十分なドライブがある場合、*/q/dm* という構文で、マウント・ポイントに名前を付けます（表 A-1 を参照）。

表 A-1 マウント・ポイントの名前の構文

<i>q</i>	Oracle データが保存されていることを示す文字列
<i>dm</i>	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 <i>sid</i> と同じ）

たとえば、*/u01/oradata/test* および */u02/oradata/test* という名前のマウント・ポイントは、Oracle テスト・データベースに 2 つのドライブを割り当てます。

## ディレクトリの名前

### ホーム・ディレクトリの構文

*/pm/h/u* という構文で、ホーム・ディレクトリに名前を付けます（表 A-2 を参照）。

表 A-2 ホーム・ディレクトリの名前の構文

<i>pm</i>	マウント・ポイントの名前
<i>h</i>	標準のディレクトリ名
<i>u</i>	ディレクトリの所有者の名前

たとえば、`/u01/app/oracle` は、Oracle Server ソフトウェアの所有者のホーム・ディレクトリ（`ORACLE_BASE` の値で、OUI を使用してインストールする場合のデフォルト値）で、`/u01/app/appmgr` は、Oracle アプリケーション・ソフトウェアの所有者のホーム・ディレクトリです。

ホーム・ディレクトリを UNIX ファイル・システム内の同一レベルに置くと、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/*`）で参照できます。

### パス名の参照

パス名は、そのパス名を保存するためのファイル（`/etc/passwd`、Oracle `oratab` ファイルなど）で参照します。グループのメンバーは、`/etc/group` ファイルで参照します。

### ソフトウェア・ディレクトリ

複数バージョンのアプリケーション・ソフトウェアを同時に実行するための OFA 要件を満たすために、各バージョンの Oracle8i ソフトウェアを `/pm/h/product/v` という構文の 1 つのディレクトリに保存します（表 A-3 を参照）。

**表 A-3 Oracle8i ソフトウェアを保存するディレクトリ名の構文**

<i>h</i>	標準のディレクトリ名
<i>v</i>	ソフトウェアのバージョン

たとえば、`/u01/app/oracle/product/8.1.5` は、Oracle8i 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` は、この値を設定します。

## ファイルの名前

### 管理ファイル

管理データの編成を簡略化するために、データベース固有の管理ファイルを `h/admin/d/a/` というサブディレクトリに保存しておくことをお勧めします。この場合、*h* は Oracle ソフトウェア所有者のホーム・ディレクトリ、*d* はデータベース名（`DB_NAME`）、*a* は次に示すデータベース管理ファイルのそれぞれのサブディレクトリを表しています（表 A-4 を参照）。

表 A-4 データベース管理ファイルのサブディレクトリ

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 ( <code>configdb_name.ora</code> の <code>AUDIT_FILE_DEST</code> を、この位置に設定します。このサブディレクトリは、定期的に整理してください。)
bdump	バックグラウンド・プロセスのトレース・ファイル
cdump	コア・ダンプ・ファイル
create	データベースを作成するプログラム
exp	データベース・エクスポート・ファイル
logbook	データベースの状態および履歴を記録するファイル
pfile	インスタンス・パラメータ・ファイル
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、サブディレクトリ `adhoc` のパス名は、`/u01/app/oracle/admin/sab/adhoc/` です。

## データベース・ファイル

次のネーミング規則に従ってデータベース・ファイルに名前を付けることによって、それらの識別が簡単になります。

- 制御ファイルの場合、`/pm/q/d/control.ctl`
- REDO ログ・ファイルの場合、`/pm/q/d/redon.log`
- データ・ファイルの場合、`/pm/q/d/tn.dbf`

表 A-5 に、この構文の詳細を示します。

表 A-5 データベース・ファイルの名前の構文

<i>pm</i>	マウント・ポイント名 (前述)
<i>q</i>	Oracle データを他の Oracle ファイルと区別するための文字列 (通常、ORACLE または <code>oradata</code> )
<i>d</i>	データベースの <code>DB_NAME</code>
<i>t</i>	Oracle 表領域名
<i>n</i>	2 桁の数字

---

**注意：** `/pm/q/d` に、データベース `d` に関連する制御ファイル、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、この規則に従って `/u03/oradata/sab/system01.dbf` という名前のデータ・ファイルを作成すると、そのファイルが属しているデータベースを簡単に判断できます。

### 要件に応じた個々のセグメント

異なる表領域間で、存在期間、I/O 要求の必要性およびバックアップの頻度別に、セグメントをいくつかのグループに分類します。

Oracle データベースごとに、表 A-6 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-6 特殊な表領域**

SYSTEM	データ・ディクショナリ・セグメント
TEMP	テンポラリ・セグメント
RBS	ロールバック・セグメント
USERS	その他のユーザー・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
DRSYS	Oracle <i>interMedia</i> 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、削除できる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・ファイル名は、*tn.dbf* という構文で指定することをお勧めします。この場合、*t* は表領域名、*n* は 2 桁の数字からなる文字列です。拡張子と 2 桁の数字からなる文字列の合計が 6 文字になるため、表領域名に使用できるのは、8 文字までになります。

データ・ファイル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 (data) および索引 (indexes) を保存するための表領域には、それぞれ GLD および GLX という名前を指定します。

---

**注意：** 表領域の名前に「tablespace」という語を連想させる文字列を付ける必要はありません。表領域は、前後関係から識別できるので、名前によってタイプに関する情報を示す必要はありません。

---

## OFA 構造に基づいた Oracle ファイル

表 A-7 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-7 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**

/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/8.1.5	Oracle8i リリース 8.1.5 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-8 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-8 OFA インストールの階層型ファイル・マッピング**

/	ルート・マウント・ポイント
u01/	ユーザー・データのマウント・ポイント #1
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
db_name1/	db_name1 データベースの管理サブツリー
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
7.3.3/	7.3.3 インスタンスの ORACLE_HOME
8.1.5/	8.1.5 インスタンスの ORACLE_HOME
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
oradata/	Oracle データのサブツリー
db_name1/	db_name1 データベース・ファイルのサブツリー
db_name2/	db_name2 データベース・ファイルのサブツリー
u02/	ユーザー・データのマウント・ポイント #2
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・ファイルのサブツリー
db_name2/	db_name2 データベース・ファイルのサブツリー
u03/	ユーザー・データのマウント・ポイント #3
home/	ログイン・ホーム・ディレクトリのサブツリー
oradata/	Oracle データのサブツリー
db_name1/	db_name1 データベース・ファイルのサブツリー
db_name2/	db_name2 データベース・ファイルのサブツリー



## 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。できるだけ 1 つのサイズに標準化してください。これによって RAW ファイルをあるパーティションから別のパーティションに確実に移動できます。必要な数のデータベース・ファイルを保存できる程度の大きさで、かつ、操作しにくくならない程度の大きさに設定してください。

たとえば、2GB のドライブのパーティションを切る場合、サイズと数のバランスを取るために、大きさが 200MB のパーティションが 10 個できるように区切るとよいでしょう。RAW デバイスを使用するすべての表領域は、複数のドライブにストライプ化してください。できれば、論理ボリューム・マネージャを使用してストライプ化してください。

## 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

Oracle Parallel Server を使用する場合、そのクラスタに対して Oracle 管理用ホーム・ディレクトリとして機能するノードを 1 つ選択します。管理用ホーム・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-9 に例を示します。

表 A-9 二重インスタンスを持つ Oracle Parallel Server の管理ディレクトリ構造

u01/	app/oracle/admin/sab/	sab 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/	inst1 インスタンスのバックグラウンド・ダンプ先
	inst2/	inst2 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/	inst1 インスタンスのコア・ダンプ先
	inst2/	inst2 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	inst 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
	19990120full.dmp	1999 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/	inst ログブック項目用のディレクトリ
	inst1/	inst1 インスタンス・レポート用のディレクトリ
	params.lst	inst1 インスタンスの v\$parameter レポート
	inst2/	inst2 インスタンス・レポート用のディレクトリ
	params.lst	inst2 インスタンスの v\$parameter レポート
	user.lst	dba_users レポート
	pfile/	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-10 に、ORACLE\_BASE ディレクトリ構造の詳細を示します。Oracle Universal Installer を使用して OFA に準拠したデータベースをインストールすると、ORACLE\_BASE はデフォルトで /pm/app/oracle に設定されています。

表 A-10 ORACLE\_BASE ディレクトリ構造

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 に準拠した Oracle Server をインストールした場合、ORACLE\_HOME ディレクトリは、`/pm/app/oracle/product/release_number` です。表 A-11 に、ORACLE\_HOME ディレクトリ構造の詳細を示します。UNIX の場合、ORACLE\_HOME ディレクトリには、次のサブディレクトリ、および選択した各 Oracle 製品用のサブディレクトリが含まれます。インストールした製品用のみディレクトリがあります。

**注意：** ディレクトリの表示は、対応する製品がインストールされているかどうかによって異なります。

表 A-11 ORACLE\_HOME ディレクトリ構造

assistants	構成アシスタント
bin	すべての製品のバイナリ・ファイル
ctx	<i>interMedia Text</i> オプション
db	<i>init</i> <i>sid</i> .ora、 <i>lksid</i>
install	インストールした関連ファイル
lib	Oracle 製品ライブラリ
jlib	Java クラス
md	<i>Spatial</i> オプション
mlx	Xerox Stemmer ( <i>interMedia Text</i> オプション用)
network	Net8
nlsrtl	NLS ランタイム読み込み可能データ
ocommon	すべての製品に共通のファイル
odg	データ収集機能
opsm	Parallel Server Management Components

表 A-11 ORACLE\_HOME ディレクトリ構造

oracore	コア・ライブラリ
ord	データ・オプション
otrace	Oracle TRACE
plssql	PL/SQL
precomp	プリコンパイラ
rdbsms	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
slax	SLAX 解析機能
sqlplus	SQL*Plus

## 製品のサブディレクトリの内容

各製品のサブディレクトリには、表 A-12 に示すサブディレクトリが含まれます。

表 A-12 製品のサブディレクトリの内容

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-13 に、製品のサブディレクトリの例を示します。

**表 A-13 製品のサブディレクトリの例**

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

rdbms/admin ディレクトリには、表 A-14 に示す SQL スクリプトが含まれます。

**表 A-14 admin ディレクトリのファイルのネーミング規則**

cat*.sql	カタログおよびデータ・ディクショナリ表 / 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 (すべてのインストレーション) catproc.sql (すべてのインストレーション) catparr.sql (Parallel Server のインストレーション) catrep.sql (すべてのインストレーション)  catproc.sql は、標準の PL/SQL パッケージ (DBMS_SQL、 DMS_OUTPUT など) を作成するためにスクリプトを順に実行します。
dbms*.sql	その他のデータベース・パッケージ
utl*.sql	データベース・ユーティリティ用の表およびビューを作成するスクリプト

## ファイル名の拡張子

表 A-15 に、ファイル名の拡張子を示します。

**表 A-15 ファイル名の拡張子**

.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

表 A-15 ファイル名の拡張子

.dmp	エクスポート・ファイル
.doc	ASCII テキスト・ファイル
.env	環境設定用のシェル・スクリプト・ファイル
.h	C ヘッダー・ファイル。 また、sr.h は SQL*Report Writer のヘルプ・ファイルです。
.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

?  
    使用例, 1-6  
@  
    \$ORACLE\_SID, 1-6  
\_editor  
    設定, 3-5

---

## 数字

2 タスク  
    アーキテクチャ  
        セキュリティ, 1-15

---

## A

ADDRESS 指定  
    プロトコル, 5-6  
afiedt.buf, 3-5  
AIX  
    システム・パラメータ, 2-7  
APPC/LU6.2 プロトコル, 5-16  
    ADDRESS, 5-16

---

## B

BEQ プロトコル, 5-6  
    ADDRESS, 5-7  
    接続パラメータの構文, 5-7  
Bourne シェル・スクリプトのトレース, 1-6  
BUFFER  
    imp のパラメータ, 2-11

---

## C

C  
    Pro\*C/C++, 4-4  
    SQL\*Module 用, 4-13  
CATPROC.SQL, A-15  
COBOL  
    Pro\*COBOL, 4-6  
CONNECT INTERNAL  
    セキュリティ, 1-16  
COPY コマンド  
    SQL\*Plus, 3-6  
coraenv, 1-6

---

## D

DB\_BLOCK\_SIZE, 2-10  
dba グループ  
    再リンク, 1-16  
    メンバー, 1-15  
Developer/2000 (Oracle Developer) デモ, 6-9

---

## E

echo コマンド, 1-2

---

## F

FORMAT プリコンパイラ  
    Pro\*COBOL, 4-10

---

## G

getname コマンド, 5-13  
glogin.sql, 3-1

## I

---

imp  
    BUFFER パラメータ, 2-11  
Intelligent Agent  
    SNMP での構成, 5-18  
interMedia, 6-2  
I/O スレーブ, 2-19  
IPC プロトコル, 5-8  
    ADDRESS, 5-8  
ireclen, 4-3

## J

---

Java  
    デモ, 6-8  
JAVA\_POOL\_SIZE, 1-22

## L

---

LARGE\_POOL\_SIZE, 1-22  
login.sql, 3-1

## M

---

MAXDATAFILES パラメータ, 1-14  
maxfree, 2-7  
MAXINSTANCES パラメータ, 1-14  
MAXLOGFILES パラメータ, 1-14  
MAXLOGHISTORY パラメータ, 1-14  
MAXLOGMEMBERS パラメータ, 1-14  
maxper, 2-8  
maxperm, 2-7  
MicroFocus COBOL コンパイラ, 4-7  
minfree, 2-7  
minperm, 2-7, 2-8

## N

---

Net8  
    ADDRESS 指定, 5-5  
    APPC/LU6.2 プロトコル, 5-16  
    BEQ プロトコル, 5-6  
    IPC プロトコル, 5-8  
    Oracle Advanced Security, 5-19  
    Oracle Connection Manager, 5-4  
    Oracle Enterprise Manager Intelligent Agent, 5-17

Oracle Names, 5-4  
RAW プロトコル, 5-9  
README ファイル, 5-2  
SPX/IPX プロトコル, 5-11  
TCP/IP プロトコル, 5-9  
アダプタ・ユーティリティ, 5-3  
製品および特徴, 5-2  
ファイルおよびユーティリティ, 5-2  
プロトコル, 5-5  
プロトコルのサポート, 5-5  
マルチスレッド・サーバー, 5-4

Net8 の構成ファイル  
    保存場所, 5-2

NLS\_LANG  
    環境変数, 1-5  
ntisbsdsm 同報通信デーモン, 5-11  
ntspxtl ユーティリティ, 5-11

## O

---

OCI デモ, 6-9  
OFA  
    「最適フレキシブル・アーキテクチャ」を参照  
ORA\_NLS  
    環境変数, 1-5  
Oracle  
    メモリーの使用量, 1-21  
Oracle Advanced Security, 5-19  
    .bak ファイル, 5-19  
    DCE 統合, 5-19  
    セキュリティおよびシングル・サインオン, 5-19  
Oracle Connection Manager, 5-4  
Oracle Diagnostics Pack, 2-5  
Oracle Enterprise Manager Intelligent Agent, 5-17  
    tcl スクリプトのデバッグ, 5-18  
    エージェント・サービスの展開および自動構成,  
        5-17  
Oracle Forms デモ, 6-9  
Oracle Names, 5-4  
Oracle Server  
    アカウント, 1-14  
Oracle Tuning Pack, 2-6  
ORACLE\_HOME  
    ? の使用, 1-6  
    環境変数, 1-5  
ORACLE\_SID  
    環境変数, 1-5



- プロンプトの抑制, 1-3
- Oracle8i で使用する UNIX 環境変数, 1-7
- Oracle 開発者デモ, 6-9
- Oracle 環境変数
  - EPC\_DISABLED, 1-5
  - NLS\_LANG, 1-5
  - ORA\_NLS33, 1-5
  - ORACLE\_BASE, 1-5
  - ORACLE\_HOME, 1-5
  - ORACLE\_PATH, 1-5
  - ORACLE\_SID, 1-5
  - ORACLE\_TRACE, 1-6
  - ORAENV\_ASK, 1-6
  - TNS\_ADMIN, 1-6
  - TWO\_TASK, 1-6
- oracle グループ
  - 権限と実行プログラム, 1-16
- Oracle コール・インタフェース, 4-15
  - 使用, 4-15
  - デモンストレーション・プログラム, 4-15
  - ユーザー・プログラム, 4-16
- Oracle コール・インタフェース (OCI) ・デモ, 6-9
- oracle ソフトウェア所有者, 1-14
  - 特殊アカウント, 1-14
- Oracle のシステム ID, 1-5
- Oracle プリコンパイラ, OCI のリンクおよび Make ファイル, 4-17
  - カスタム Make ファイル, 4-17
  - 未定義シンボル, 4-17
- Oracle ライブラリ
  - Oracle 共有ライブラリ, 4-19
  - 静的リンクと動的リンク, 4-18
- Oracle ランタイム・システム
  - Pro\*COBOL, 4-8
- ORAENV\_ASK, 1-6
  - 設定, 1-3
- oraenv ファイル
  - 説明, 1-2
  - データベース間の移動, 1-3
- orapwd コマンド, 1-17
- oreclen, 4-3

## P

---

- PL/SQL
  - デモンストレーションロード, 1-26

- Pro\*C/C++
  - Make ファイル, 4-5, 4-6
  - 管理, 4-4
  - シグナル, 4-20
  - システム構成ファイル, 4-4
  - 使用, 4-4
  - デモンストレーション・プログラム, 4-4
  - ユーザー・プログラム, 4-6
- Pro\*C/C++ デモ, 6-8
- Pro\*COBOL, 4-6
  - FORMAT プリコンパイラ, 4-10
  - Oracle ランタイム・システム, 4-8
  - 環境変数, 4-7
  - 管理, 4-7
  - システム構成ファイル, 4-7
  - デモンストレーション・プログラム, 4-8
  - ネーミングの違い, 4-6
  - ユーザー・プログラム, 4-9
- PRODUCT\_USER\_PROFILE 表
  - SQL\*Plus, 3-2
- PTX Agent, 2-3
- PTX Manager, 2-3
- pupbld.sql, 3-2

## R

---

- RAW プロトコル, 5-9
- README.txt ファイル, 6-9
- README ファイル
  - Net8, 5-2
- root
  - ユーザー, 1-14

## S

---

- Server Manager
  - SHOW PARAMETERS, 1-12
  - コマンド, 1-16
  - セキュリティ, 1-16
- SGA, 2-11
- SHELL, 1-8
- SHUTDOWN コマンド, 1-16
  - セキュリティ, 1-16
- SIGCLD 2 タスク・シグナル, 4-20
- SIGIO 2 タスク・シグナル, 4-20
- SIGINT 2 タスク・シグナル, 4-20
- SIGPIPE 2 タスク・シグナル, 4-20

- SIGTERM 2 タスク・シグナル, 4-20
- SIGURG 2 タスク・シグナル, 4-20
- Site Profile
  - SQL\*Plus, 3-1
- smit
  - 「System Management Interface Tool」を参照
- SNMP
  - Intelligent Agent, 5-18
- SPOOL コマンド
  - SQL\*Plus, 3-6
  - 使用, 3-6
- SPX/IPX プロトコル, 5-11
  - ADDRESS, 5-15
  - getname コマンド, 5-13
  - ntisbsdmd 同報通信デーモン, 5-11
  - ntspcxctl ユーティリティ, 5-11
  - コマンドの要約, 5-13
- SQL
  - 管理, 3-1
- SQL\*DBA
  - SHOW PARAMETERS, 1-12
- SQL\*Loader, 2-11
- SQL\*Module
  - Ada 用, 4-13
  - C 用, 4-13
- SQL\*Module for Ada, 4-13
  - 管理, 4-13
  - システム構成ファイル, 4-13
  - 使用, 4-13
  - デモンストレーション・プログラム, 4-13
  - ユーザー・プログラム, 4-14
- SQL\*Plus
  - COPY コマンド, 3-6
  - PRODUCT\_USER\_PROFILE 表, 3-2
  - Site Profile, 3-1
  - SPOOL コマンド, 3-6
  - UNIX コマンド, 3-6
  - User Profile, 3-1
    - エディタ, 3-4
  - 環境変数, 3-5
  - システム・エディタ, 3-4
  - 使用, 3-4
  - 制限事項, 3-6
  - 設定ファイル, 3-1
  - デフォルト・エディタ, 3-4
  - デモンストレーション表, 3-2
  - ヘルプ機能, 3-3

- 割込み, 3-6
- SQL\*Plus の使用, 3-4
- SQL\*Plus への割込み, 3-6
- SQL トレース, 2-4
- STARTUP コマンド
  - セキュリティ, 1-16
- SYSDATE
  - TZ, 1-8
- System Management Interface Tool, 2-3
- SYSTEM アカウント
  - 権限, 1-14
- SYS アカウント
  - 権限, 1-14

## T

---

- TCP/IP プロトコル, 5-9
  - ADDRESS, 5-10
- TIMED\_STATISTICS, 2-4
- TKPROF, 2-5
- TNS\_ADMIN
  - 環境変数, 1-6
- TNS リスナー
  - Oracle TCP/IP プロトコル用の構成, 5-9
- TWO\_TASK
  - 環境変数, 1-6
- TZ
  - SYSDATE, 1-8
  - 環境変数, 1-8

## U

---

- UDP のチューニング, 2-26
- UNIX
  - セキュリティ, 1-15
- UNIX コマンド
  - SQL\*Plus からの実行, 3-6
- User Profile
  - SQL\*Plus, 3-1
- utlbstat.sql, 2-4
- utlestat.sql, 2-4

## V

---

- vmtune, 2-7

## X

---

XA 機能, 4-22

## あ

---

アダプタ・ユーティリティ, 5-3

## う

---

受取りルーチン  
例, 4-20

## え

---

エディタ  
SQL\*Plus, 3-4  
順序の設定, 3-5

## お

---

オペレーティング・システム・コマンド  
SQL\*Plus からの実行, 3-6

## か

---

各国語サポート (NLS)  
変数, 1-5  
環境変数, 4-7  
MicroFocus COBOL コンパイラ, 4-7  
NLS\_LANG, 1-5  
ORA\_NLS, 1-5  
ORACLE\_HOME, 1-5  
ORACLE\_SID, 1-5  
ORACLE\_TRACE, 1-6  
ORAENV\_ASK, 1-6  
Pro\*COBOL, 4-7  
SHELL, 1-8  
SQL\*Plus の設定, 3-5  
TNS\_ADMIN, 1-6, 5-3  
TWO\_TASK, 1-6  
TZ, 1-8  
感嘆符  
SQL\*Plus プロンプト, 3-6  
管理  
SQL, 3-1

## き

---

疑問符  
使用例, 1-6  
共通の環境  
oraenv ファイル, 1-2  
設定, 1-2  
共有メモリー  
SGA, 1-22

## く

---

グループ  
サンプル・スクリプト, 1-20

## け

---

権限  
dba グループ, 1-16  
付与, 1-16  
言語, 1-5

## こ

---

構成  
マスター・エージェント, 5-18  
構成ファイル  
Net8, 5-2  
プリコンパイラ, 4-3  
コマンド  
orapwd, 1-17  
コマンド・インタプリタ, 1-8

## さ

---

最適フレキシブル・アーキテクチャ, A-1  
OFA の特徴, A-2  
ディレクトリ構造, A-12  
ファイル・マッピング, A-2, A-10  
再リンク  
パラメータ, 1-28  
サブシェル  
SQL\*Plus での作成, 3-6  
参照マニュアル, xi

## し

---

### シグナル

- 2 タスク, 4-19
- ハンドラの作成, 4-19

### シグナル・ハンドラ

- シグナル, 4-19
- 使用, 4-19

### シグナル・ルーチン

- 例, 4-20

### システム・エディタ

- SQL\*Plus, 3-4
- 順序の設定, 3-5
- デフォルト設定, 3-5

### システム・グローバル領域 (SGA)

- 要件, 1-22

### システム構成ファイル

- Pro\*C/C++, 4-4
- Pro\*COBOL, 4-7
- SQL\*Module for Ada, 4-13

### システム時刻

- 設定, 1-8

### システム・パラメータ

- AIX, 2-7

### 自動ログイン

- remote\_os\_roles, 1-19

### シャドウ・プロセス

- セキュリティ, 1-15

### 上級開発者デモ, 6-8

### 初期化パラメータ

- BACKGROUND\_DUMP\_DEST, 1-12
- BITMAP\_MERGE\_AREA\_SIZE, 1-12
- COMMIT\_POINT\_STRENGTH, 1-12
- CONTROL\_FILES, 1-12
- CREATE\_BITMAP\_AREA\_SIZE, 1-12
- DB\_BLOCK\_BUFFERS, 1-12
- DB\_BLOCK\_SIZE, 1-12
- DB\_FILE\_DIRECT\_IO\_COUNT, 1-12
- DB\_FILE\_MULTIBLOCK\_READ\_COUNT, 1-12
- DB\_FILES, 1-12
- DISTRIBUTED\_TRANSACTIONS, 1-12
- HASH\_AREA\_SIZE, 1-12
- HASH\_MULTIBLOCK\_IO\_COUNT, 1-12
- LOG\_ARCHIVE\_BUFFER\_SIZE, 1-12
- LOG\_ARCHIVE\_BUFFERS, 1-12
- LOG\_ARCHIVE\_DEST, 1-12
- LOG\_ARCHIVE\_FORMAT, 1-12

### LOG\_BUFFER, 1-12

### LOG\_CHECKPOINT\_INTERVAL, 1-13

### LOG\_SMALL\_ENTRY\_MAX\_SIZE, 1-13

### MTS\_LISTENER\_ADDRESS, 1-13

### MTS\_MAX\_DISPATCHERS, 1-13

### MTS\_MAX\_SERVERS, 1-13

### MTS\_SERVERS, 1-13

### NLS\_LANGUAGE, 1-13

### NLS\_TERRITORY, 1-13

### OBJECT\_CACHE\_MAX\_SIZE\_PERCENT, 1-13

### OBJECT\_CACHE\_OPTIMAL\_SIZE, 1-13

### OPEN\_CURSORS, 1-13

### OS\_AUTHENT\_PREFIX, 1-13

### PROCESSES, 1-13

### SHARED\_POOL\_SIZE, 1-13

### SHOW PARAMETERS コマンド, 1-12

### SORT\_AREA\_SIZE, 1-13

### SORT\_READ\_FAC, 1-13

### SORT\_SPACEMAP\_SIZE, 1-13

### USER\_DUMP\_DEST, 1-13

### デフォルト, 1-12

## す

---

### スーパーユーザー, 1-14

### スレッドのサポート, 4-18

## せ

---

### 制限

- リソース, 1-22

### 制限事項 (SQL\*Plus), 3-6

- COPY コマンド, 3-6

- ウィンドウのサイズ変更, 3-7

- リターン・コード, 3-7

### 静的リンクと動的リンク

- Oracle ライブラリ, 4-18

### セキュリティ

- 2 タスク・アーキテクチャ, 1-15

- CONNECT INTERNAL, 1-16

- Server Manager アクセス, 1-16

- SHUTDOWN コマンド, 1-16

- STARTUP コマンド, 1-16

- グループ・アカウント, 1-15

- 権限の割当て, 1-16

- デフォルト・グループ名, 1-16

- ファイル所有権, 1-15

設定ファイル  
SQL\*Plus, 3-1

## た

---

タイム・ゾーン  
TZ での設定, 1-8

## ち

---

チューニング  
SGA, 2-11  
UDP, 2-26  
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, 1-22  
ディスク I/O  
I/O スレーブ, 2-19  
データ・カートリッジ  
デモの実行, 6-1  
データベース  
管理者  
実行プログラムへの権限, 1-16  
ファイル  
セキュリティ, 1-17  
認可, 1-17  
ブロック・サイズ, 2-10  
データベースの変更, 1-3  
デバック・プログラム, 4-3  
デフォルト設定  
システム・エディタ, 3-5  
デモの使用, 6-8  
デモンストレーション  
プリコンパイラ, 1-27  
プロシージャ・オブション、PL/SQL, 1-26  
デモンストレーション表  
SQL\*Plus, 3-2  
削除, 3-3  
手動で作成, 3-2  
デモンストレーション・プログラム  
Oracle コール・インタフェース, 4-15  
Pro\*C/C++, 4-4

Pro\*COBOL, 4-8  
SQL\*Module for Ada, 4-13  
デモ (デモンストレーション・ファイル)  
Java, 6-8  
Oracle 開発者, 6-9  
Oracle コール・インタフェース (OCI), 6-9  
Pro\*C/C++, 6-8  
既存表のレトロフィット, 6-8  
上級開発者, 6-8  
使用方法, 6-8  
電子ユーティリティ, 6-8  
電子ユーティリティ・デモ, 6-8

## と

---

動的リンクと静的リンク  
Oracle ライブラリ, 4-18  
特殊アカウント, 1-14  
特殊グループ  
root, 1-15

## は

---

配布ソフトウェア, 1-5  
パスワード  
リモート, 1-17  
バッファ・キャッシュ間  
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, 2-3  
Oracle, 2-4

## ふ

---

ファイル名  
SQL\*Plus でのデフォルト拡張子, 3-6  
複数のシグナル・ハンドラ, 4-20  
プリコンパイラ  
ireclen および oreclen の値, 4-3  
値, 4-3  
大文字から小文字への変換, 4-3  
概要, 4-2  
シグナル, 4-20  
実行ファイルの再リンク, 4-2  
デモンストレーションの実行, 1-27  
ベンダー提供のデバック・プログラム, 4-3

プリコンパイラ構成ファイル, 4-3  
プリコンパイラ実行ファイルの再リンク, 4-2  
ブロック・サイズ, 2-10  
プロトコル, 5-5  
    ADDRESS 指定, 5-6

## へ

---

ページング  
    過度な, 2-7, 2-9  
    不十分な, 2-9  
ヘルプ機能  
    SQL\*Plus, 3-3

## ま

---

マスター・エージェントの構成, 5-18  
マルチスレッド・サーバー, 5-4

## め

---

メモリー  
    仮想, 1-21  
    競合, 2-7  
    共有, 1-22  
    使用量の見積り, 1-21

## ゆ

---

ユーザー  
    サンプル・スクリプト, 1-20  
ユーザー・プログラム  
    Oracle コール・インタフェース, 4-16  
    Pro\*C/C++, 4-6  
    Pro\*COBOL, 4-9  
    SQL\*Module for Ada, 4-14  
ユーザー割込みハンドラ, 4-20  
ユーティリティ (電子)・デモ, 6-8

## り

---

リソースの制限, 1-22  
リモート接続パラメータ  
    OS\_AUTHENT\_PREFIX, 1-19  
    REMOTE\_OS\_AUTHENT, 1-19  
    REMOTE\_OS\_ROLES, 1-19

## れ

---

レトロフィット・デモ, 6-8

## ろ

---

ログイン・ホーム・ディレクトリ  
    管理, 1-19  
    サンプル・スクリプト, 1-19