

Oracle8i

for HP 9000 Servers and Workstations 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

Oracle8i のチューニング

SQL*Plus の管理

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 HP 9000 Servers and Workstations 管理者リファレンス リリース 8.1.5

リリース 8.1.5

部品番号: A63016-1

第1版: 1999年7月 (第1刷)

原本名: Oracle8i Administrator's Reference, Release 8.1.5 for HP 9000 Servers and Workstations, Release 8.1.5

原本部品番号: A67163-01

Copyright © 1998, 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。

ドラフトのアルファ版およびベータ版ドキュメント

ドラフトのアルファ版およびベータ版ドキュメントはプレリリース状態のものです。これらのドキュメントは、オラクル社の機密かつ所有のドキュメントであり、デモおよび暫定使用のみを目的としたものです。タイプミスからデータの不正確さに至るまでのいくつかの誤りが存在することが考えられます。このドキュメントは予告なく変更する場合がありますが、当ソフトウェアを使用するハードウェアに限定するものではありません。オラクル社はプレリリースのドキュメントに対して、無謬性を保証しません。またそのドキュメントを使用したことによって損失および損害が発生した場合も一切責任を負いかねますのでご了承ください。

目次

はじめに	vii
用途	vii
対象読者	vii
Oracle8i および Oracle8i Enterprise Edition	vii
このマニュアルで使用する表記規則	viii
コマンドの構文	viii
参照マニュアル	ix
Oracle サービスおよびサポート	ix
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	1-5
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	1-9
初期化パラメータ	1-9
init.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	1-14
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	1-16

グループとセキュリティ	1-16
Oracle Server ユーティリティのセキュリティ	1-17
Server Manager コマンドのセキュリティ	1-17
データベース・ファイルのセキュリティ	1-17
ネットワーク・セキュリティ	1-18
Net8 での自動ログイン	1-19
確認の順番	1-20
セキュリティとリモート・パスワード	1-21
ログイン・ホーム・ディレクトリの管理	1-23
Oracle8i のメモリー使用量の見積り	1-25
サーバー・リソースの制限	1-26
システム・グローバル領域の制御	1-26
SGA のサイズ制限	1-26
SGA のサイズの計算	1-29
デモンストレーションの作成と実行	1-29
SQL*Loader のデモンストレーション	1-29
SQL*Loader の管理	1-30
PL/SQL デモンストレーションのロード	1-32
PL/SQL デモンストレーションの実行	1-33
ネットワーク実行ファイルの再リンク	1-35

2 Oracle8i のチューニング

チューニングの重要性	2-2
パフォーマンス・ボトルネックのタイプ	2-2
HP-UX のツール	2-3
gprof	2-3
monitor	2-3
netfmt	2-3
netstat	2-4
nfsstat	2-4
nettl	2-4
prof	2-4
profil	2-5
sar	2-5
top	2-6

vmstat	2-6
iostat	2-7
swapinfo	2-8
mpstat	2-8
HP パフォーマンス分析ツール	2-9
GlancePlus/UX	2-9
HP PAK	2-9
SQL スクリプト	2-9
メモリー管理のチューニング	2-9
十分なスワップ領域の割当て	2-10
ページングの制御	2-10
単一共有メモリー・セグメントでの SGA の確保	2-10
物理メモリーの SGA ロック	2-11
Oracle ブロック・サイズをオペレーティング・システム・ブロック・サイズの倍数にする	2-11
データベース・バッファ数の最適化	2-11
REDO バッファ数の最適化	2-12
共有プール・サイズの最適化	2-12
データ・ディクショナリ・キャッシュの効果の検証	2-13
適切なライブラリ・キャッシュ・スペースの割当て	2-13
ディスク I/O のチューニング	2-14
データベース・ライターをチューニングして書込みバンド幅を大きくする	2-14
IOSTAT を使用した大きなディスクへの要求キューの検索	2-15
HP-UX での非同期 I/O	2-15
適切なファイル・システム・タイプの選択	2-18
ディスク・パフォーマンスの監視	2-18
ディスクのパフォーマンスに関する問題	2-19
CPU の使用状況のチューニング	2-19
Oracle ユーザー / プロセスの優先順位をすべて同じにする	2-19
マルチ・プロセッサ・システムでのプロセッサ親和性 / バインディングの使用	2-20
大量のエクスポート / インポートおよび SQL*Loader ジョブでのシングルタスク・リンクの使用	2-20
Oracle リソースの競合のチューニング	2-20
UNIX カーネル・パラメータのチューニング	2-20
V\$ 表を使用して競合を特定する方法	2-21
競合の原因となるセグメントの特定	2-22
ブロック・サイズおよびファイル・サイズのチューニング	2-22

Oracle ブロック・サイズの指定	2-22
HP-UX バッファ・キャッシュ・サイズのチューニング	2-23
トレース・ファイルとアラート・ファイルの使用	2-23
トレース・ファイルの名前	2-23
アラート・ファイル	2-24
RAW デバイス / ボリューム	2-25
RAW デバイス / ボリュームの欠点	2-25
RAW デバイス / ボリュームを使用する場合のガイドライン	2-25
RAW デバイスの設定	2-27

3 SQL*Plus の管理

SQL*Plus の管理	3-2
設定ファイル	3-2
Site Profile	3-2
User Profile	3-2
PRODUCT_USER_PROFILE 表	3-3
デモンストレーション表	3-3
ヘルプ機能	3-4
SQL*Plus の使用	3-5
SQL*Plus からのシステム・エディタの使用	3-5
SQL*Plus からのオペレーティング・システム・コマンドの実行	3-5
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
ウィンドウのサイズ変更	3-6
リターン・コード	3-6

4 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
参照ドキュメント	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	4-7
Pro*COBOL の管理	4-7
環境変数	4-8
Pro*COBOL の使用	4-8
Pro*FORTRAN	4-11
Pro*FORTRAN の管理	4-11
Pro*FORTRAN の使用	4-11
Oracle コール・インタフェース	4-13
Oracle コール・インタフェースの使用	4-13
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	4-15
カスタム Make ファイル	4-15
未定義シンボル	4-16
スレッドのサポート	4-17
Oracle ライブラリを使用した静的リンクおよび動的リンク	4-17
シグナル・ハンドラの使用	4-18
シグナル	4-18
XA 機能	4-21

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
Oracle Names	5-4
Net8 Assistant	5-4
Net8 プロトコルのサポート	5-4
ADDRESS 指定	5-5
BEQ プロトコル	5-6
BEQ の ADDRESS 指定	5-6
IPC プロトコル	5-7
IPC の ADDRESS 指定	5-7
RAW プロトコル	5-8
TCP/IP プロトコル	5-9

TCP/IP の ADDRESS 指定	5-9
APPC/LU6.2 プロトコル	5-10
APPC/LU6.2 の ADDRESS 指定	5-10
Oracle Enterprise Manager (OEM)	5-12
Oracle SNMP での Oracle Intelligent Agent の構成	5-12
Oracle Advanced Security	5-13

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
Oracle8i <i>interMedia</i>	6-2
Text	6-2
Audio、Video および Image	6-3
デモの実行	6-5
Locator	6-7
Oracle8i Time Series デモ	6-9
Oracle8i Visual Information Retrieval	6-10
Oracle8i Spatial	6-12

A 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)	A-2
OFA に準拠したデータベースの特長	A-2
UNIX で適用される OFA	A-4
マウント・ポイントの名前	A-4
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-8
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-9
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-13

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 HP-UX 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、HP 9000 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/rdsd/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>cvtrct 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

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。

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

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

```
$ echo $ORACLE_SID
```

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。HP-UX の場合、ローカル bin ディレクトリのデフォルト位置は /usr/local/bin 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/local/bin/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/local/bin 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/local/bin
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/local/bin
setenv ORACLE_SID PROD
setenv ORAENV_ASK NO
source /usr/local/bin/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/local/bin
ORACLE_SID=PROD
export PATH ORACLE_SID
SIDLIST= 'awk -F: '/^[^#]/ {printf "%s ", $1}' /var/opt/oracle/oratab
echo "SIDS on this machine are $SIDLIST"
ORAENV_ASK=
. /usr/local/bin/oraenv
```

C シェルの場合、次のように入力します。

```
setenv PATH ${PATH}:/local/bin
setenv ORACLE_SID PROD
setenv SIDLIST 'awk -F: '/^[^#]/ {printf "%s ", $1}' /var/opt/oracle/oratab
echo "SIDS on this machine are $SIDLIST"
unset ORAENV_ASK
source /usr/local/bin/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 HP 9000 Servers and Workstations インストール・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 Oracle8i 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 HP 9000 Servers and Workstations インストール・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	/u01/app/oracle
ORACLE_HELP	機能	ヘルプ・ファイルがあるディレクトリを指定します。
	構文	<i>directory_path</i>
	例	\$ORACLE_HOME/help/admin/resource
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

表 1-1 Oracle8i/UNIX における Oracle8i 環境変数

変数	項目	定義
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:. 注意: 最後にピリオドを付けることによって、検索パスに現行の作業ディレクトリが追加されます。
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 HP 9000 Servers and Workstations インストレーション・ガイド』を参照してください。
	例	SAL1
ORACLE_TRACE	機能	インストレーション時の Bourne シェル・スクリプトのトレース状態を切り替えます。T を設定すると、多くの Oracle シェル・スクリプトは set -x フラグがオンの状態で実行されます。
	許容値	T 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 HP 9000 Servers and Workstations インストレーション・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
DISPLAY	機能	X ベースのツールで使します。入出力に使用するディスプレイ・デバイスを指定します。詳細は、各ベンダーの X-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 HP 9000 Servers and Workstations インストレーション・ガイド』を参照してください。

表 1-2 Oracle8i で使用する UNIX 環境変数

変数	項目	定義
PATH	機能	シェルが実行プログラムを配置するために使用します。 \$ORACLE_HOME/bin が含まれている必要があります。
	構文	ディレクトリをコロンで区切ったリスト (<i>directory:directory:directory</i>)
	例	/bin:/usr/bin:/usr/local/bin: /usr/bin/X11:\$ORACLE_HOME/bin:\$HOME/bin:.. 注意: 最後にピリオドを付けることによって、検索パスに現行の作業ディレクトリが追加されます。
PRINTER	機能	HP-UX 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 HP-UX に付属のその他のコマンド・インタプリタ
	例	/bin/sh
SHLIB_PATH	機能	HP-UX では、実行時に共有ライブラリ・ローダーが共有オブジェクト・ライブラリを検索するために使用します。
	構文	ディレクトリをコロンで区切ったリスト (<i>directory:directory:directory</i>)
	例	/usr/dt/lib:\$ORACLE_HOME/lib
TERM	機能	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。HP-UX のドキュメントを参照して、オペレーティング・システムでこの環境変数を使用できるかどうかを確認してください。

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/SID/pfile` ディレクトリ中にファイルを作成します。このファイルを変更することによって、Oracle8i 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 つのパターンが用意されています。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、`init.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 `init.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 `init.ora` ファイル

サンプル `init.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。

```
# 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$rdbmns:[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 40/8 = 5:
# transactions = 40
# transactions_per_rollback_segment = 8

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

初期化パラメータ

```
# 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、HP-UX 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_BASE/admin/ <code>sid</code> /bdump
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	200
DB_BLOCK_SIZE	2048
DB_FILES	80 (最大値は 2000000)

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
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。)
DISTRIBUTED_TRANSACTIONS	16
HASH_AREA_SIZE	0
HASH_MULTIBLOCK_IO_COUNT	1
LOCK_SGA	FALSE
LOCK_SGA_AREAS	0
LOG_ARCHIVE_BUFFER_SIZE	64
LOG_ARCHIVE_BUFFERS	4 (最大値は 128)
LOG_ARCHIVE_DEST	\$ORACLE_HOME/dbs/arch/
LOG_ARCHIVE_FORMAT	"%t_%s.dbf"
LOG_BUFFER	8192
LOG_CHECKPOINT_INTERVAL	10000
LOG_SMALL_ENTRY_MAX_SIZE	80
MTS_MAX_DISPATCHERS	5
MTS_MAX_SERVERS	20
MTS_SERVERS	0
MTS_LISTENER_ADDRESS	ADDRESS=address
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	50
SHARED_POOL_SIZE	3500000
SORT_AREA_SIZE	65536
SORT_READ_FAC	5
SORT_SPACEMAP_SIZE	512

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
USER_DUMP_DEST	\$ORACLE_BASE/admin/sid/udump

データベースの制限

表 1-5 に、CREATE CONTROL FILE 文のパラメータのデフォルト値および最大値を示します。

注意： それらの値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	65534
MAXINSTANCES	1	63
MAXLOGFILES	16	255
MAXLOGMEMBERS	2	5
MAXLOGHISTORY	100	65534

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

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

表 1-7 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。
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。

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 のみが実行できるプログラムには、権限 710 が与えられています。
- oinstall という名前のグループを作成します。oinstall グループは、OUI oraInventory を所有します。また、Oracle8i システムをインストールおよびアップグレードする責任があります。すべての oracle アカウントは、このグループに属している必要があります。
- 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 751 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勧めしません）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。

Oracle Server ユーティリティのセキュリティ

権限のないユーザーが Oracle8i 実行プログラムを使用しないように保護する必要があります。Oracle8i 実行プログラムを保護する方法は、環境やシングル・タスク・ユーティリティを使用するかどうかによって異なります。次の方法で Oracle8i 実行プログラムを保護します。

- すべてのプログラムを \$ORACLE_HOME/bin ディレクトリに置き、*oracle* ソフトウェア所有者に所有権を与えます。
- マシン上のすべてのユーザーが Oracle Server にアクセスできるように、すべてのユーザー・ユーティリティ (sqlplus、exp、imp) に 711 の保護を与えます。
- すべての DBA ユーティリティに 700 の保護を与えることによって、通常、それらのユーティリティを使用できるのは *oracle* ソフトウェア所有者である DBA ユーザー名のみになります。

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込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
```

ネットワーク・セキュリティ

ネットワーク上でのパスワードの使用

ネットワーク上のリモート・ユーザーは、クリア・テキストまたは暗号化テキストを使用してパスワードを入力できます。クリア・テキストを使用すると、許可されていないユーザーにパスワードが知られてしまい、セキュリティが失われます。Net8 では、暗号化パスワードが使用できます。

ネットワークでの DBA 権限

ネットワークで DBA 権限を制御するには、次の手順を行います。

1. \$ORACLE_HOME/network/admin/listener.ora ファイルで、リモート DBA アクセスを denied に設定します。
2. listener.ora ファイルに次の行を追加します。

```
REMOTE_DBA_OPS_DENIED=oracle_sid
```

oracle_sid は、DBA アクセスを拒否する Oracle インスタンスのリストか、またはすべての Oracle インスタンスに対してリモート DBA アクセスを拒否する場合は、ALL_SIDS です。

3. orapwd を使用して、リモート DBA アクセスの管理者のパスワードを設定します。orapwd 実行ファイルの使用方法については、『Oracle8i 管理者ガイド』を参照してください。

自動 (ops\$) ログイン

Oracle8i では、ネットワークでの自動ログイン（オペレーティング・システムで許可されたログイン）をサポートしています。

UNIX では、ドル記号 (\$) が環境変数の始まりを示します。そのため、コマンド行またはスクリプトの中で、オペレーティング・システムによって許可された (ops\$) ログインを指定する場合、まず、バックスラッシュ (\) で \$ をエスケープします。たとえば、リモートにログインする場合、ユーザー ID scott には、ops\$\scott を指定する必要があります。

root ユーザー ID では、自動ログインできません。

注意： PC、Apple Macintosh および OS/2 ユーザーが自動ログインすることはお薦めしません。だれでも Oracle 構成ファイルを編集したり、そのユーザー ID を変更したりできます。それらのシステムのユーザーがネットワークでログインする場合は、セキュリティ上の理由から listener.ora ファイルの ops\$ ログインを使用禁止にしてください。

Net8 での自動ログイン

Net8 は、自動ログインおよびリモート DBA ログインを制御しません。自動ログインおよびリモート DBA ログインは Oracle8i によって制御され、init^{tsid}.ora ファイルのパラメータを使用して構成されます。自動ログインはサポートされていますが、デフォルトでは使用禁止になっています。使用可能にするには、REMOTE_OS_AUTHENT 初期化パラメータを true に設定し、データベースを起動します。

oracle プログラムがそれらのログインを制御するので、root への setuid として Net8 リスナーを実行する必要はありません。

参照： Net8 の構成の詳細は、第 5 章を参照してください。

Net8 での自動ログインを実行するには、/etc/passwd ファイルに daemon というユーザーを作成します。この daemon ユーザーは、どのローカル・データベースにも ops\$ アカウントを持っていてはいけません。また、どの DBA グループにも属してはいけません。つまり、外部ユーザーがローカル・データベースに侵入できる ops\$daemon アカウントを持っていてはいけません。

DBA グループ ID のキーワード

表 1-9 に、リモート・ログインを使用可能にして制御するために、\$ORACLE_HOME/network/admin/listener.ora ファイルで使用するキーワードを示します。

表 1-9 リモート・ログインを制御するキーワード

DBA_GROUP	名前が、リスナーのサービス対象となるすべてのインスタンスの定数の場合に使用します。
DBA_GROUP_sid	2 つ以上の \$ORACLE_HOME がリスナーのサービス対象で、そのグループ ID が異なる場合、ORACLE_SID ごとに使用します。
OPS_DOLLAR_LOGIN_ALLOWED OPS_DOLLAR_LOGIN_DENIED	リモート・ログインを制御します。デフォルトは、OPS_DOLLAR_LOGIN_DENIED です。
REMOTE_DBA_OPS_ALLOWED REMOTE_DBA_OPS_DENIED	リモート DBA アクセスを制御します。デフォルトは、REMOTE_DBA_OPS_DENIED です。

アクセスされるデータベースの DBA グループ ID がデフォルト名 (dba) でない場合、デフォルト以外の名前を指定できます。

リモート・ログインおよびリモート DBA アクセスのパラメータには、ネットワーク上のデータベースの個々の ORACLE_SID に設定するか、またはすべての sid を一度に設定します。たとえば、次の文はいずれも有効です。

```
PARAMETER=ALL_SIDS
PARAMETER=sid1[, sidn...]
```

sid にどちらの権限が割り当てられているかを調べるには、次のように入力します。

```
$ lsnrctl status
```

確認の順番

システムは、リモート・ログインのパラメータを次の順に確認します。

1. アクセスを拒否するパラメータ
2. アクセスを許可するパラメータ
3. デフォルト値 (拒否)

これらの権限は、Net8 リスナーの対象となるシャドウ・プロセスのユーザー ID とグループ ID を操作することによって実行されます。次に例を示します。

- 指定したインスタンスの OPS_DOLLAR_LOGIN_DENIED が true に設定されている場合、または、クライアント側のオペレーティング・システムによって通知されたユーザー ID のアカウントがデータベース・ホスト・マシンにない場合、そのユーザー ID およびグループ ID は daemon 項目の下の /etc/passwd ファイルにあります。

- 指定した ORACLE_SID の OPS_DOLLAR_LOGIN_ALLOWED および REMOTE_DBA_OPS_ALLOWED が true で、クライアントのオペレーティング・システムによって通知されたユーザー ID のアカウントがこのシステムにない場合、そのユーザー ID およびグループ ID はこのユーザー ID の /etc/passwd にあります。
- 指定した ORACLE_SID の OPS_DOLLAR_LOGIN_ALLOWED が true で、REMOTE_DBA_OPS_ALLOWED が false の場合、そのユーザー ID に DBA 権限が与えられていれば、そのプロセスのユーザー ID およびグループ ID は daemon になります。それ以外の場合、プロセスのユーザー ID およびグループ ID は、このユーザーのユーザー ID およびグループ ID になります。

注意： デフォルトでは、REMOTE_DBA_OPS_ALLOWED は false です。この値は変更しないでください。このパラメータを false に設定すると、DBA 権限を持つユーザーは、オペレーティング・システムで認可されたログインをネットワークで実行できなくなります。ただし、通常のネットワーク・ログイン（パスワードの入力が必要なネットワーク・ログイン）は実行できます。

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 1-10 に、orapwd の構文を示します。

表 1-10 orapwd の実行構文

<i>filename</i>	パスワード情報が書き込まれているファイル名です。ファイル名は <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/orapwV815 \
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。

複数のデータベース用の共有パスワード・ファイル

1つのパスワード・ファイルを使用して複数のデータベースの DBA アクセスを制御する場合 (REMOTE_LOGIN_PASSWORDFILE = shared の場合)、パスワード・ファイルを作成してから、異なるデータベースの \$ORACLE_HOME/dbs ディレクトリからそのファイルへのシンボリック・リンクを作成する必要があります。HP-UX の場合、必ず共有ファイルへの各リンクは *orapwsid* という名前にしてください。

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 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。権限がないユーザーの場合は、このデータベースを使用する Oracle アプリケーションを起動することによって、データベースにアクセスできます。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
```

OPERATOR として接続するには、同じコマンドに OPERATOR パスワードを追加して使用します。

リモート認証

表 1-11 に、安全性の低いプロトコルを使用したり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 1-11 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、そのパス名を保存するファイル（`/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。

注意： 共通の目的で使用するローカルなユーティリティは、`/usr/local/bin` ディレクトリに格納してください。

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 grp_x スクリプト

```
#!/bin/sh
# grp_x - 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" " "
echo
```


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、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。

```
<oracle 実行ファイルのテキストのサイズ>
+ <SGA のサイズ>
+ n × ( <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>
+ <oracle 実行ファイルの未初期化データ・セクションのサイズ>
+ <スタック用に 8192 バイト>
+ <プロセス・ユーザー領域用に 2048 バイト>)
```

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、1-29 ページの「SGA のサイズの計算」を参照してください。

各 Oracle バックエンド接続に対して、次の式を使用して仮想メモリー要件を見積もります。

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

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、HP-UX 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。`getconfig` 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

HP-UX では、親プロセスからリソースの制限を継承します（オペレーティング・システムのドキュメントの `getrlimit(2)` を参照）。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。HP-UX 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、HP-UX のシステム管理者にお問い合わせください。

`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 HP-UX 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

1 つの共有メモリー・セグメントの最大サイズは、HP-UX の SHMMAX パラメータで指定します。たとえば、SHMMAX が 512KB で SGA が 2048KB の場合、SGA には 4 つの共有メモリー・セグメントが必要になります。

SGA のサイズが共有メモリー・セグメントの最大サイズ（SHMMAX）を超える場合、Oracle8i は、要求された SGA サイズになるように、連続セグメントをさらに連結しようとします。SHMSEG は、プロセスが接続できるセグメントの最大数です。SHMMAX のサイズが制限されているシステムでは、連続するアドレスでセグメントを連結するために、SHMMAX を最大値に設定する必要があります。

注意： SHMMAX がデータベースの SGA サイズより小さい場合、緊密共有メモリー（ISM）によって問題が発生する場合があります。

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
- 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。

DB_BLOCK_BUFFERS パラメータの設定

このパラメータでは、データ・バッファ・キャッシュのサイズを設定します。SGA のデータ・バッファ・キャッシュには、システムが現在使用している、または最近使用した Oracle データ・ブロックが保持されます。

これは、おそらく Oracle で最も重要なチューニング・パラメータです。Oracle で特定のブロックを検索する場合に、それがバッファ・キャッシュに見つかればディスクから読み込む必要はなく、I/O 操作の回数を削減できるので、システムのパフォーマンスが向上します。

他にボトルネックがなく、システムのページングを回避するのに十分な追加メモリーが使用可能ならば、SGA のサイズを増やすとパフォーマンスが向上します。

バッファ・キャッシュのサイズは次の条件を満たす必要があります。

- I/O を最小化するのに十分な大きさである
- プロセスの追加によってシステムのページングが始まるほど大きくない
- SGA の他の部分用のメモリーが不足するほど大きくない

パフォーマンス標識

現行のバッファ・キャッシュの効率、キャッシュ・ヒット率を見ることによって判断できます。キャッシュ・ヒット率とは、読み込みの合計回数に対する論理読み込みの回数です。

キャッシュ・ヒット率は、次のようにして判断します。

- 論理読み込みの回数を判断する（固定取得＋データベース・ブロック取得）
- キャッシュ・ヒット率を計算する（論理読み込み－物理読み込み / 論理読み込み）

固定取得、データベース・ブロック取得および物理読み込みの値は、`v$sys_stat` 表から検索できます。また、`utlblockstat.sql` (BSTAT) と `utlestat.sql` (ESTAT) を使用して検索できます。

チューニングのガイドライン

- キャッシュ・ヒット率が 0.9 (90%) 未満で、システム上に使用可能なメモリーが十分にある場合は、`db_block_buffers` の値を増やします。
- キャッシュ・ヒット率が 90 ～ 95% の範囲の場合は、再チューニングによってパフォーマンスを改善できます。
- `db_block_buffers` の値を増やしてもパフォーマンスの問題が解決しない場合は、このパラメータをさらに増やすのではなく、別のタイプのボトルネックを探します。

DB_BLOCK_SIZE パラメータの設定

このパラメータを設定する機会は、データベースを作成する前の 1 度のみです。既存のデータベースに対してこのパラメータを変更することはできません。Oracle では、配布するソフトウェアに 2KB のデフォルト・ブロック・サイズを設定しています。ほとんどの場合、2KB のブロック・サイズでは大規模データベースに対して不十分です。4 ～ 8KB の範囲で選択してください。

サイズ設定のガイドライン

- Oracle Application、SAP およびその他の市販の大規模アプリケーション・パッケージには、ブロック・サイズとして 8KB を使用します。
- バッチ処理を行うデータベースや、全表走査によって大量のデータが頻繁に読み込まれる大規模レポートを最適化する場合は、8KB のブロックを使用します。
- UNIX ファイル・システムに保存されるデータベースの場合は、Oracle のブロック・サイズをファイル・システムのブロック・サイズと同じにします。

たとえば、ファイル・システムのデフォルト・ブロック・サイズが 8KB の場合は、データベースのブロック・サイズも 8KB にします。これより小さいサイズをデータベースに対して選択すると、データベースの書込みを行うたびに、より大きなファイル・システム・ブロックを読み込んでから、ディスクに書き込まなければなりません（書込み前の読み込み）。

- 長さが 4KB を超える行を持つ表がデータベースにある場合は、`db_block_size` パラメータに 8KB のブロック・サイズを設定します。

4KB 未満の行に対して直接読み込み / 書込みを行うアプリケーションの場合は、このパラメータに 4KB のブロック・サイズを設定します。

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
```

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
```

デモンストレーションの作成と実行

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 1-12 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 n バイトの固定長レコードで構成されていることを示します。 n は整数です。
var	ファイルが n バイトの可変長レコードで構成されていることを示します。 n は整数です。レコードの最初の 5 文字を指定します。

ファイル処理オプションを選択しないと、情報はデフォルトでレコード・ストリーム (str) として処理されます。fix モードではレコード終了記号を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 1-13 に、カーネルのデモンストレーションを示します。

表 1-13 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 1-14 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 1-14 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
-----------	------------	------------	------------

カーネルの 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、カーネルに接続します。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))
```

デモンストレーションの作成と実行

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 1-15 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

ネットワーク実行ファイルの再リンク

Oracle8i のチューニング

- チューニングの重要性
- HP-UX のツール
- HP パフォーマンス分析ツール
- メモリー管理のチューニング
- ディスク I/O のチューニング
- HP-UX での非同期 I/O
- SQL スクリプト
- ディスク・パフォーマンスの監視
- CPU の使用状況のチューニング
- Oracle リソースの競合のチューニング
- ブロック・サイズおよびファイル・サイズのチューニング
- HP-UX バッファ・キャッシュ・サイズのチューニング
- トレース・ファイルとアラート・ファイルの使用
- RAW デバイス / ボリューム

チューニングの重要性

Oracle8i は、高度に最適化できるソフトウェア製品です。チューニングを頻繁に行うことによって、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。この章の説明は、シングル・プロセッサ・システムを対象としていますが、パフォーマンス・チューニングに関するヒントの多くは、Oracle Parallel Server および Oracle8i で使用可能な機能を使用する場合にも適用できます。

システムのチューニングを始める前に、次の項の「HP-UX のツール」にあるツールを使用して、通常の動作を監視します。

参照： 詳細は、『Oracle8i Parallel Server 概要および管理』および『Oracle8i チューニング』を参照してください。

パフォーマンス・ボトルネックのタイプ

メモリー

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。このメモリー不足を解消するために、システムは、メモリーとディスク間でプロセスのページングおよびスワッピングを行います。

ディスク I/O

ディスク I/O の競合は、メモリー管理がうまく行われていない（そのためページングおよびスワッピングが発生する）ことや、表領域およびファイルのディスクへの分散が適切でないことが原因です。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。

CPU

CPU でも、プロセスの競合が発生する場合があります。ほとんどの場合、UNIX カーネルによって CPU が効果的に割り当てられますが、プロセスが多数あると、それらが CPU サイクルを競合します。システムに複数の CPU がある場合（マルチ・プロセッサ環境）は、さまざまな CPU で異なるレベルの競合が発生することがあります。

Oracle リソース

競合は、ロックやラッチなどの Oracle リソースでもよく起こります。

HP-UX のツール

HP-UX には、データベースのパフォーマンスを評価し、データベースの要件を決定するパフォーマンス監視ツールおよび分析ツールが提供されています。

ここでは、HP-UX の標準コマンドであるパフォーマンス分析ユーティリティを示します。それぞれのユーティリティについて簡単に説明します。詳細は、各ツールの man ページを参照してください。

これらのツールは、oracle プロセスの統計だけでなく、システム全体の CPU の使用状況、割込み、スワップ、ページングおよびコンテキスト・スイッチングについての統計情報も提供します。

参照： HP-UX ツールについては、オペレーティング・システムのドキュメントを参照してください。

gprof

C、PASCAL および FORTRAN プログラムの実行プロファイルを作成するプロファイリング・ユーティリティです。このユーティリティは、プログラムが実行時間のほとんどを費やす場所のパフォーマンス統計を表示します。次のようなデータを表示します。

- プログラムの機能ごとの実行時間の合計
- プログラムの機能ごとの呼出し件数
- サイクル情報（メンバー機能、時間および呼出し件数）

gprof は、ユーザーが作成した共有ライブラリ、またはシステムが提供する共有ライブラリのプロファイルを作成できません。

monitor

プロファイルを作成し、バッファ内の記録、定期的にサンプリングされるプログラム・カウンタの値と特定の機能の呼出し件数のヒストグラムを監視するためのインタフェースです。

基本的に、システム・プログラマおよびツール開発者に役立ちます。

netfmt

ネットワーク・トレーシングおよびロギング機能（nettl を参照）から集めたバイナリ・トレースおよびログ・データをフォーマットします。バイナリ・トレースおよびログ情報は、ファイルまたは標準入力から読み込めます。フォーマットされたデータは、標準出力に書き込まれます。

また、netfmt は、ログ・クラス、トレースの種類、接続、プロセス、パスおよびユーザーによってデータをフィルタできます。

netstat

ネットワーク・インタフェースおよびプロトコルの統計、およびネットワークに関連するさまざまなデータ構造の内容を表示します。

nfsstat

カーネルへのインタフェース、NFS (Network File System) および RPC (Remote Procedure Call) の統計情報を表示します。また、この情報を再初期化する場合にも使用します。nfsstat は、再送信、タイムアウト、反復回数、受信した呼出しなどの累積統計を表示します。

nettl

ログおよびトレースによって、ネットワーク・イベントまたはパケットを獲得します。ログによって、状態変更、エラーおよび接続の確立などのネットワーク・アクティビティを獲得できます。トレースによって、ネットワークを通して受信および送信されるパケットのスナップショット、およびループバックまたはヘッダー情報を獲得できます。

prof

C プログラムの実行プロファイルを作成します。このユーティリティは、ご使用のプログラムのパフォーマンス統計を表示し、プログラムが最も実行時間を費やす場所を示します。

prof は、monitor が作成するプロファイルを解釈します。オブジェクト・ファイル内の記号表を読み込み、プロファイル・ファイルに関連付けます。各外部テキスト記号には、その記号のアドレスとその次のアドレスの間での実行にかかった時間が % で表示されます。また、機能が呼び出された回数および各呼出しにかかった時間の平均がミリ秒で表示されます。

prof を使用して、次のことを判断できます。

- プログラムの機能ごとの実行回数の合計
- プログラムの機能ごとの呼出し件数
- 機能ごとの呼出しにかかった時間
- プログラムの記号表の内容
- プログラム全体の実行時間の合計
- 機能ごとの実行時間の割合 (%)

profil

プログラム・カウンタの情報をバッファにダンプします。profil は、システム呼出しによって実行時間のプロファイリングを制御します。これによって、システムは、アドレス領域内のさまざまな場所で呼出しプログラムが実行する時間を推定します。

sar

オペレーティング・システムから一定の間隔で、累積アクティビティ・カウンタをサンプリングします。以前に記録されたデータ・ファイルの内容も表示できます。

sar を使用して、次のものを表示できます。

- CPU のオペレーティング・モード、バッファ・キャッシュおよび RAW I/O のアクティビティ
- システム呼出し、システム・スワップおよびアクティビティの切り換え
- デバイスのアクティビティ（ターミナル、ブロック・デバイスなど）
- システム・ルーチンにアクセスするファイルのアクティビティ
- メッセージおよびセマフォ・アクティビティ
- テキスト、プロセス、i ノードおよびファイル表の状態
- マルチプロセッサ・システムのプロセッサごとのアクティビティ

ページング・アクティビティのサマリーを 10 秒ごとに 10 回表示する場合は、次のように入力します。

```
$ sar -w 10 10
```

図 2-1 に、sar -w コマンドの出力例を示します。

図 2-1 sar -w コマンドの出力例

	swpin/s	bswin/s	swpot/s	bswot/s	pswch/s
14:14:55					
14:15:05	0.00	0.00	0.00	0.60	722
14:15:15	0.00	0.00	0.00	0.10	1010
14:15:25	0.00	0.00	0.00	0.00	783
14:15:35	0.00	0.00	0.00	0.00	695
14:15:45	0.00	0.00	0.00	0.00	610
14:15:55	0.00	0.00	0.00	0.00	576
14:16:05	0.00	0.00	0.00	0.00	458
14:16:15	0.00	0.00	0.00	0.00	518
14:16:25	0.00	0.00	0.00	0.00	521
14:16:35	0.00	0.00	0.00	0.00	588
Average	0.00	0.00	0.00	0.07	648

top

システムの top プロセスを表示し、情報を定期的に更新します。top プロセスの順位は、RAW CPU の割合 (%) で示されます。システム、メモリーおよびプロセス・データも表示されます。

vmstat

vmstat ユーティリティは、HP-UX におけるプロセス、仮想メモリー、ディスク、ページングおよび CPU のアクティビティを表示します。表示内容は、コマンドで切り換えます。システム・アクティビティのサマリーを 5 秒ごとに 8 回表示する場合は、次のように入力します。

```
% vmstat -S 5 8
```

図 2-2 に、vmstat コマンドの出力例を示します。

図 2-2 vmstat コマンドの出力例

procs			memory		page				disk				faults		cpu						
r	b	w	swap	free	si	so	pi	po	fr	de	sr	f0	s0	s1	s3	in	sy	cs	us	sy	id
0	0	0	1892	5864	0	0	0	0	0	0	0	0	0	0	0	90	74	24	0	0	99
0	0	0	85356	8372	0	0	0	0	0	0	0	0	0	0	0	46	25	21	0	0	100
0	0	0	85356	8372	0	0	0	0	0	0	0	0	0	0	0	47	20	18	0	0	100
0	0	0	85356	8372	0	0	0	0	0	0	0	0	0	0	2	53	22	20	0	0	100
0	0	0	85356	8372	0	0	0	0	0	0	0	0	0	0	0	87	23	21	0	0	100
0	0	0	85356	8372	0	0	0	0	0	0	0	0	0	0	0	48	41	23	0	0	100
0	0	0	85356	8372	0	0	0	0	0	0	0	0	0	0	0	44	20	18	0	0	100
0	0	0	85356	8372	0	0	0	0	0	0	0	0	0	0	0	51	71	24	0	0	100

w 列 (procs の下) は、スワップ・アウトされた（ディスクに書き込まれた）プロセスの数を示します。値が 0 以外の場合は、スワッピングが起きていて、システムはメモリー不足になっています。si 列および so 列は、それぞれ 1 秒あたりのスワップ・インおよびスワップ・アウトの回数を示します。スワップ・アウト (so 列) は、常に 0 にしてください。

iostat

iostat ユーティリティは、端末およびディスク・アクティビティを表示します。表示内容は、コマンドで切り換えます。iostat は、ディスクのビジー（使用中）状況を表示します。ただし、ディスク要求キューは表示されません。iostat で表示される情報は、I/O 負荷のバランスを調整する際に役立ちます。

端末およびディスク・アクティビティのサマリーを 5 秒ごとに 5 回表示する場合は、次のように入力します。

```
$ iostat 5 5
```

図 2-3 に、iostat コマンドの出力例を示します。

図 2-3 iostat コマンドの出力例

tty		fd0			sd0			sd1			sd3			cpu			
tin	tout	Kps	tps	serv	Kps	tps	serv	Kps	tps	serv	Kps	tps	serv	us	sy	wt	id
0	1	0	0	0	0	0	31	0	0	18	3	0	42	0	0	0	99
0	16	0	0	0	0	0	0	0	0	0	1	0	14	0	0	0	100
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100
0	16	0	0	0	0	0	0	2	0	14	12	2	47	0	0	1	98

swapinfo

swapinfo -m ユーティリティは、スワップ領域の使用状況を表示します。スワップ領域が不足すると、システムがハングアップしたり、応答時間が長くなったりする場合があります。図 2-4 に、swapinfo -m コマンドの出力例を示します。

図 2-4 swapinfo -m コマンドの出力例

	Mb	Mb	Mb	PCT	START/	Mb		
TYPE	AVAIL	USED	FREE	USED	LIMIT	RESERVE	PRI	NAME
dev	120	13	107	11%	1187340	-	0	/dev/dsk/c201d6s0
fs	246	0	246	0%	246	0	1	/usr2
hold	0	16	-16					

mpstat

mpstat ユーティリティは、プロセッサごとの統計を表示します。表の各行は、1 プロセッサのアクティビティを示します。1 つ目の表は、起動してからのすべてのアクティビティをまとめて表示します。2 つ目以降の各表は、アクティビティの間隔を表示します。特に指定しない限り、すべての値は割合（1 秒あたりのイベント数）です。引数は、統計および反復回数の時間間隔を示します。図 2-5 に、mpstat コマンドの出力例を示します。

図 2-5 mpstat コマンドの出力例

CPU	minf	mjf	xcal	intr	ithr	csw	icsw	migr	smtx	srw	syscl	usr	sys	wt	idl
0	0	0	1	71	21	23	0	0	0	0	55	0	0	0	99
2	0	0	1	71	21	22	0	0	0	0	54	0	0	0	99
CPU	minf	mjf	xcal	intr	ithr	csw	icsw	migr	smtx	srw	syscl	usr	sys	wt	idl
0	0	0	0	61	16	25	0	0	0	0	57	0	0	0	100
2	1	0	0	72	16	24	0	0	0	0	59	0	0	0	100

HP パフォーマンス分析ツール

GlancePlus/UX

Hewlett-Packard のこのユーティリティは、システムのアクティビティを測定するオンライン診断ツールです。GlancePlus は、システム・リソースがどのように使用されているかを表示します。システムの I/O、CPU およびメモリーの使用状況に関する動的な情報が、一連の画面に表示されます。このユーティリティを使用して、個々のプロセスがリソースをどのように使用しているかを監視することもできます。

HP PAK

HP PAK (HP Programmer's Analysis Kit) には、現在、Puma および TTV の 2 つのツールがあります。

Puma は、プログラム実行中にパフォーマンス統計を収集します。また、収集した統計を参照および分析するために、グラフィカル表示もできます。

TTV (Thread Trace Visualizer) は、インストルメント・スレッド・ライブラリ (libpthread_tr.sl) がグラフィカル形式で作成したトレース・ファイルを表示します。TTV を使用すると、スレッドがどのように対話するのかを参照でき、リソースのために待機しているスレッドがブロックしている場所を検索できます。

HP PAK は、HP FORTRAN 77、HP Fortran90、HP C、HP C++、HP ANSI C++ および HP Pascal コンパイラにバンドルされています。

SQL スクリプト

SQL スクリプト utlbstat および utlestat は、Oracle データベースのパフォーマンスを監視し、共有グローバル領域 (SGA) データ構造をチューニングします。これらのスクリプトの詳細は、『Oracle8i チューニング』を参照してください。HP-UX では、スクリプトは \$ORACLE_HOME/rdbms/admin/ に保存されています。

メモリー管理のチューニング

メモリー・チューニング・プロセスでは、ページングおよびスワップ領域をチューニングし、使用可能なメモリーを判断します。

Oracle バッファ・マネージャによって、アクセス頻度の高いデータをキャッシュに長く保存できます。バッファ・マネージャを監視してバッファ・キャッシュをチューニングすると、Oracle のパフォーマンスはかなり向上する場合があります。各システムの Oracle バッファ・サイズの最適値は、システム全体の負荷と他のアプリケーションとを比較した場合の Oracle の優先順位によって異なります。

十分なスワップ領域の割当て

スワッピングはかなりの UNIX オーバーヘッドの原因となるため、最小限に抑える必要があります。スワッピングを調べるには、HP-UX で `swapinfo -m` を使用します。

システムでスワッピングが行われている場合、メモリーを節約するために次の処理を行います。

- 必要以上にシステム・デーモン (daemon)・プロセスまたはアプリケーション・プロセスを実行しないようにします。
- データベース・バッファの数を減らし、使用可能なメモリーを増やします。
- UNIX ファイル・バッファの数を減らします (特に RAW デバイスを使用する場合)。

HP-UX では、`swapinfo -m` を使用して、使用中のスワップ領域を判断します。`swapon` を使用して、システムにスワップ領域を追加します。詳細は、HP-UX のドキュメントを参照してください。

スワップ領域は、まず、システムの RAM の 2 ～ 4 倍に設定します。Oracle Developer、Oracle Application または Oracle InterOffice を使用する場合は、もう少し高い値に設定してください。スワップ領域の使用状況を見ながら、必要に応じて値を高くしてください。

ページングの制御

プログラムを実行するためにプログラム全体がメモリーに常駐する必要はないので、ページングはスワッピングほど深刻な問題ではありません。少しくらいのページアウトでは、システムのパフォーマンスにほとんど影響はありません。

大量のページングを検出するには、高速応答時またはアイドル時の測定値と、低速応答時の測定値を比較します。

ページングを監視するには、`vmstat` または `sar` を使用します。システムで大量のページアウトが常に発生している場合は、次の方法で解決してください。

- メモリーを増やします。
- 一部の作業を別のシステムに移します。
- カーネルで使用するメモリーを少なく設定します。

単一共有メモリー・セグメントでの SGA の確保

十分な共有メモリーが構成されていなければ、データベースを起動できません。共有メモリーを増やすために、UNIX カーネルを再構成できます。詳細は、第 1 章の「システム・グローバル領域の制御」を参照してください。

参照： 詳細は、第1章の「SGA のサイズの計算」を参照してください。

参照： 詳細は、『Oracle8i for HP 9000 Servers and Workstations インストール・ガイド』の第2章の「Oracle8i 用に UNIX カーネルを構成する」を参照してください。

物理メモリーの SGA ロック

可能性のあるパフォーマンス向上率: 0 ~ 20%

SGA の主な機能は、データベース情報のキャッシングです。SGA がディスクのページングを開始すると、キャッシングは利点ではなくオーバーヘッドになります。一部のプラットフォーム・ベンダーは、SGA を物理メモリーにロックする技術を提供しています。

SGA を物理メモリーにロックすると、Oracle のパフォーマンスが改善されることがありますが、同じシステム上の他のアプリケーションのパフォーマンスが低下する可能性もあります。

Oracle ブロック・サイズをオペレーティング・システム・ブロック・サイズの倍数にする

可能性のあるパフォーマンス向上率: 0 ~ 5%

UNIX システムは、オペレーティング・システム・ブロック全体をディスクから読み込みます。データベースのブロック・サイズが UNIX のファイル・システムのバッファ・サイズよりも小さいと、I/O バンド幅の効率が悪くなります。

データベースのブロック・サイズは、オペレーティング・システムのブロック・サイズと同じか、または、その倍数にしてください。init.ora パラメータ DB_BLOCK_SIZE で、データベースのブロック・サイズを設定します。ブロック・サイズは、データベースを再作成することによって変更できます。

DB_BLOCK_SIZE パラメータのデフォルト値を確認するには、V\$PARAMETER データ・ディクショナリ表を問い合わせます。

データベース・バッファ数の最適化

可能性のあるパフォーマンス向上率: 0 ~ 200%

DB_BLOCK_SIZE パラメータは、SGA 内のデータベース・バッファのサイズも決定します。

ヒット率

バッファ・キャッシュのヒット率は、次のように定義されています。

$$\text{ヒット率} = \frac{\text{論理読み込み} - \text{物理読み込み}}{\text{論理読み込み}}$$

ここで、論理読込み = データベース・ブロック取得 + 固定取得です。

ヒット率が 60 ～ 70% 未満の場合は、DB_BLOCK_BUFFERS の値を増やして、キャッシュ内のバッファ数を増やしてください。

参照： X\$KCBRBH 表を使用して、実行中のシステムから収集した統計情報に基づいてバッファ数を見積もる方法については、『Oracle8i チューニング』を参照してください。

REDO バッファ数の最適化

REDO ログ領域の統計は、ユーザー・プロセスが REDO バッファ内の空き領域を待つ回数を表示します。

REDO ログ・バッファのサイズは、init_{sid}.ora の LOG_BUFFER パラメータによって決定します。このパラメータの値は、バイト単位で表します。

共有プール・サイズの最適化

init_{sid}.ora の SHARED_POOL_SIZE パラメータには、共有プールのサイズをバイト単位で設定します。変更された最低使用頻度アルゴリズムは、データ・ディクショナリ・キャッシュのエントリを優先します。つまり、ライブラリ・キャッシュをチューニングすると、十分な量のメモリーをデータ・ディクショナリで使えるようになります。

V\$SGASTAT 表を使用して、特に空き領域を確認しながら、共有プールを監視します。次に、V\$SGASTAT 問合せの例を示します。

```
SELECT * FROM v$sgastat ORDER BY bytes desc
NAME                                BYTES
-----
sql area                            1370876
free memory                         867036
library cache                       785224
db_block_buffers                    409600
dictionary cache                    275740
...
```

共有プールに大きすぎる値が設定されることがよくあります。空きメモリー領域が前述の例と同じくらい大きい場合は、共有プールのサイズを小さくしてください。いずれかの値が増えているかどうかを確認するには、問合せを繰り返し実行してください。

データ・ディクショナリ・キャッシュの効果の検証

SQL 文解析時にパフォーマンスを最適化するには、データ・ディクショナリ・キャッシュが、最も頻繁にアクセスされるデータを十分に保存できる大きさでなければなりません。データ・ディクショナリ・キャッシュ・ミスが発生すると再帰コールが生成され、データベースのパフォーマンスが低下します。

注意： ディクショナリ・キャッシュが小さすぎる場合は、V\$ROWCACHE 表を問い合せてキャッシュ・アクティビティを確認してください。

適切なライブラリ・キャッシュ・スペースの割当て

可能性のあるパフォーマンス向上率: 0 ~ 50%

ライブラリ・キャッシュには、共有 SQL および PL/SQL 領域が含まれます。SQL が再使用可能な場合でも、ライブラリ・キャッシュが小さすぎると再使用は行われません。

V\$LIBRARYCACHE 表を問い合せて、ライブラリ・キャッシュ・ミスがパフォーマンスに影響しているかどうかを確認してください。

次の問合せを実行し、V\$LIBRARYCACHE の統計を一定期間監視します。

```
SELECT SUM(pins) "Executions",
       SUM(reloads) "Cache Misses while Executing"
FROM V$LIBRARYCACHE;
```

問合せは、次のような出力を返します。

```
Executions Cache Misses while Executing
-----
320871                               549
```

PINS の合計（最初の列）は、実行のために SQL 文、PL/SQL ブロックおよびオブジェクト定義が合計で 320,871 回アクセスされたことを示しています。RELOADS の合計（最後の列）は、その実行のうちの 549 回はライブラリ・キャッシュ・ミスであったことを示しています。RELOADS の合計が 0 に近く、その割合は 1% 未満であることが必要です。

PINS に対する RELOADS の割合が 1% を超えている場合は、`initsid.ora` パラメータ `SHARED_POOL_SIZE` の値を増やして、ライブラリ・キャッシュに割り当てられるメモリーを追加してください。

ディスク I/O のチューニング

I/O ボトルネックは、最も簡単に識別できるパフォーマンスの問題です。使用可能なディスク全体で I/O を均等に分散させて、ディスクへのアクセス時間が短くなるようにしてください。小規模なデータベース、および Parallel Query Option を使用しないデータベースでは、異なるデータ・ファイルと表領域を使用可能なディスクに分散させてください。

データベース・ライターをチューニングして書き込みバンド幅を大きくする

Oracle では、データベース・ライター（DBWR）のアクティビティがボトルネックになることを避けるために、次の方法が用意されています。

- 非同期 I/O の使用
- I/O スレーブの使用

非同期 I/O

非同期 I/O を使用することによって、プロセスは、書き込み後すぐに次の操作に進めます。アイドル時間が短くなるため、システム・パフォーマンスが向上します。HP-UX では、RAW デバイスへの非同期 I/O のみをサポートしています。

参照： 詳細は、第 1 章の「HP-UX での非同期 I/O」を参照してください。

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

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
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 に記載されているパラメータのうち、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。HP-UX 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。

IOSTAT を使用した大きなディスクへの要求キューの検索

要求キューは、特定のディスク・デバイスに対する I/O 要求が実行されるまでにかかる時間を示します。要求キューの原因には、特定のディスクに対する I/O のボリュームが大きいことや、I/O の平均シーク時間が長いことなどがあります。ディスク要求キューは 0 (または 0 に近い値) であることが理想的です。

HP-UX での非同期 I/O

背景情報

非同期 I/O ドライバは、RAW デバイスを必要とします。非同期デバイス・ドライバは、RAW ディスクや RAW 論理ボリュームなどの RAW デバイスでのみ動作します。これは、HP-UX の制限です。

非同期 I/O は、特殊な HP デバイス・ドライバを使用して、共有メモリー・セグメントに対するバッチ読み書きを行います。1 回のシステム・コールで、いくつかの書込み / 読込みを行うことができます。デバイス・ドライバは、複数の I/O 操作を低レベルで開始します。このレベルでは、ディスク・スケジューリングを最適化し、パラレル化を最大にして HP-UX のオーバーヘッドを最小にすることができます。

このタイプの非同期は安全です。Oracle には、データがディスクに書き込まれるかどうかはすべて通知されます。トランザクションおよびデータベースの書き込みは、I/O の完了が Oracle によって確認されるまでコミットされません。

HP-UX で非同期 I/O 疑似ドライバを使用すると、Oracle Server は非同期メソッドを使用して RAW ディスク・パーティションに対する I/O を非同期で実行できるので、I/O オーバーヘッドが減り、スループットが向上します。非同期 I/O 疑似ドライバは、HP 9000 Server and Workstation で使用できます。

非同期 I/O のインプリメント

データベース・ファイルが RAW ディスク・パーティションに作成されている場合は、HP 9000 Server 上で非同期 I/O を行うことをお勧めします。非同期 I/O ドライバは、SAM (システム管理者ユーティリティ) を使用して HP-UX カーネルに構成できます。

HP 9000 システム上で実行されている HP-UX システムの非同期 I/O をインプリメントするには、次の作業を行います。

1. SAM を root ユーザーとして実行します。
 2. SAM のメイン・メニューで「Kernel Configuration」を選択します。
 3. 「Drivers」を選択します。
 4. 非同期ディスク・ドライバを構成し、SAM から新規の HP-UX カーネルを再生成します。
 - SAM の「Actions」メニューの「Add Driver to Kernel」を選択します。
 5. SAM を使用して新規の HP-UX カーネルを再生成する手順は、次のとおりです。
 - SAM の「Actions」メニューの「Create a New Kernel」を選択します。
 - 次の 2 つのオプションのどちらかを選択します。
 - (a) 「Move kernel into place and continue shutdown (reboot) now」
 - (b) 「Continue without moving the kernel into place」

(このオプションを選択した場合は、/stand/build/vmunix_test にカーネルが新規に作成され、その作成に使用された構成ファイルが /stand/build/SYSTEM.SAM に置かれます。)
- カーネルの変更を有効にするには、/stand/build/vmunix_test という名前のカーネルを /stand ディレクトリに移動して /stand/vmunix という名前にし、リブートする必要があります。同時に、構成ファイルを /stand/system に移動します。

古いカーネルのバックアップを取り、/stand/build ディレクトリのカーネルを /stand に入れます。

```
$ mv /stand/vmunix /stand/vmunix.prev
$ mv /stand/build/vmunix_test /stand/vmunix
```

6. init.ora ファイルを変更します。

非同期 I/O ドライバを Oracle で使用可能にするには、init.ora のパラメータ DISK_ASYNC_IO を TRUE に設定し、システムをリブートします。

```
$ /sbin/shutdown -r 0
```

これで、非同期ディスク I/O ドライバが使用可能になります。

7. root としてログインし、メジャー番号 101 のデバイス・ファイル /dev/async が存在しなければ、そのデバイス・ファイルを作成します。

このデバイス・ファイルが存在する場合は、次のコマンドを発行して、正しく設定されていることを確認する必要があります。

```
$ ls -l /dev/async
```

このコマンドの出力は、次のようになります。

```
crw----- 1 oracle  oracle    101 0x000000  Oct 28 10:32  /dev/async
```

デバイス・ファイル /dev/async が表示されない場合、またはデバイスが正しく表示されない場合は、既存のデバイス・ファイルを削除し、次のように入力します。

```
# /sbin/mknod /dev/async c 101 0x0
```

注意: この作業は root ユーザーが実行する必要があります。

8. デバイス・ファイルに Oracle インストール時の選択と矛盾しない UNIX 所有者および許可を与えます。

Oracle アカウントの所有者が oracle の場合、次のように入力します。

```
# /sbin/mknod /dev/async c 101 0x0
# /usr/bin/chown oracle:dba /dev/async
# /usr/bin/chmod 660 /dev/async
```

max_async_ports パラメータ

max_async_ports は、構成可能な HP-UX カーネル・パラメータであり、/dev/async を同時にオープンできるプロセスの最大数を制御します。

TPC-C ベンチマークを実行しているときに、一部のシャドウ・プロセスが EBUSY というエラーで非同期ドライバのオープンに失敗することがわかりました。このエラーは、非同期ドライバがこのパラメータで指定されたプロセス数をすでにオープンしていることが原因です。

非同期ドライバのオープンでのエラーは、多数のシャドウ・プロセスや、非同期 I/O を行う多数の平行問合せスレーブが実行されているマシン上で、パフォーマンスに影響を与えることがあります。

このエラーを避けるには、/dev/async が実行するプロセスの最大数を見積もり、その値を max_async_ports に設定する必要があります。

Oracle は、このエラーをアラート・ログ・ファイルまたはトレース・ファイルにフラグしないことに注意してください。

非同期 I/O の検証

非同期 I/O が実行されていることを検証するには、gpm または glance を使用して次の手順を実行します。

1. プロセス・メニューで dbwr プロセスを選択します。
2. プロセスを選択した後で、メニューから「open files」オプションを選択します。
3. /dev/async ファイルを探します。

このファイルが dbwr プロセスによってオープンされている場合、非同期 I/O は「ON」です。

適切なファイル・システム・タイプの選択

HP-UX では、ファイル・システムを選択できます。ファイル・システムの特徴はそれぞれ異なり、それらがデータにアクセスする方法は、データベース・パフォーマンスに大きな影響を与えます。次に、代表的なファイル・システムを示します。

- hfs: BSD UNIX から派生した UNIX ファイル・システム
- vxfs: Veritas ファイル・システム
- RAW デバイス: ファイル・システムなし

ディスク・パフォーマンスの監視

ディスク・パフォーマンスを監視するには、sar -b および sar -u を使用します。

表 2-2 に、ディスク・パフォーマンスに影響する `sar -b` 列を示します。

表 2-2 ディスク・パフォーマンスに影響する `sar -b` 列

<code>bread/s, bwrit/s</code>	読み込んだブロック数および書き込んだブロック数 (ファイル・システム・データベースで重要)
<code>pread/s, pwrnt/s</code>	パーティションの読み込み数およびパーティションの書き込み数 (RAW パーティションのデータベース・システムで重要)

ディスク・パフォーマンスで重要な `sar -u` 列は `%wio` で、これによってブロック I/O での CPU の待ち時間の割合がわかります。

キー・インジケータは次のとおりです。

- `bread`、`bwrit`、`pread` および `pwrnt` の合計は、ディスク I/O サブシステムの状態を示します。この合計が大きいほど、ディスク I/O ボトルネックが起きる可能性が高くなります。物理ドライブの数が多いほど、合計のしきい値が高くなる可能性があります。デフォルトの最適値は、ドライブ 2 個の場合は 40 以下、ドライブ 4～8 個の場合は 60 以下です。
- `%rcache` は 90 を超える値、`%wcache` は 60 を超える値でなければなりません。そうしないと、システムがディスク I/O バウンドになる可能性があります。
- `%wio` が常に 20 を超える値の場合、システムは I/O バウンドになります。

ディスクのパフォーマンスに関する問題

Oracle ブロック・サイズは、ディスクのブロック・サイズと同じか、またはディスクのブロック・サイズの倍数でなければなりません。

できれば、データベース・ファイル用にファイル・システムを使用する前に、パーティションでファイル・システム・チェックを行い、クリーンで断片化されていないファイル・システムを作成します。ディスクの I/O はできるだけ均等に分散し、ログ・ファイルをデータベース・ファイルから分離します。

CPU の使用状況のチューニング

Oracle ユーザー / プロセスの優先順位をすべて同じにする

Oracle は、すべてのユーザーおよびバックグラウンド・プロセスを同じ優先順位で操作するように設計されています。優先順位を変更すると、競合が起きたり応答時間に影響が出たりする場合があります。

たとえば、ログ・ライター・プロセス (LGWR) の優先順位を低くすると、LGWR の実行頻度は低くなり、ボトルネックになります。逆に、LGWR の優先順位を高くすると、ユーザー・プロセスの応答時間は長くなってしまいます。

マルチ・プロセッサ・システムでのプロセッサ親和性 / バインディングの使用

マルチ・プロセッサ環境では、できるだけプロセッサの親和性およびバインディングを使用します。プロセッサ・バインディングを使用すると、ある CPU から別の CPU にプロセスが移されることはないで、CPU キャッシュ内の情報をさらに有効に利用できます。サーバー・シャドウ・プロセスをバインドすると、常にアクティブになるため、キャッシュを使用して、バックグラウンド・プロセスを CPU 間で実行できるようになります。

大量のエクスポート / インポートおよび SQL*Loader ジョブでのシングルタスク・リンクの使用

大量のデータをユーザーと Oracle8i 間で（たとえば、エクスポート / インポートを使用して）転送する必要がある場合、シングルタスク・アーキテクチャを使用すると効果的です。シングルタスク・インポート (impst)、エクスポート (expst) および SQL*Loader (sqlldrst) 実行ファイルを作成する場合、\$ORACLE_HOME/rdbms/lib ディレクトリにある ins_rdbms.mk という Make ファイルを使用します。

次のコマンドを実行すると、impst、expst および sqlldrst の実行ファイルが作成されます。

```
% cd $ORACLE_HOME/rdbms/lib
% make -f ins_utilities.mk singletask
```

注意： Oracle 実行ファイルをシングルタスクとしてリンクすると、ユーザー・プロセスは SGA 全体に直接アクセスできるようになります。さらに、シングルタスクを実行するには、多くのメモリーが必要です。これは、oracle 実行ファイル・テキストがフロントエンド・プロセスとバックグラウンド・プロセス間で共有されなくなるためです。

Oracle リソースの競合のチューニング

UNIX カーネル・パラメータのチューニング

UNIX カーネルをできるだけ小さくすることによって、パフォーマンスが向上します。UNIX カーネルでは、通常、事前に物理 RAM が割り当てられるため、oracle など他のプロセスで利用できるメモリーが少なくなります。

従来は、NBUF、NFILE、NOFILES などのカーネル・パラメータを使用してカーネル・サイズを調整していました。ただし、ほとんどの UNIX インプリメンテーションが UNIX 構成ファイルに含まれていても、それらのカーネル・パラメータは実行時に動的に調整されず。

メモリーが割り当てられているビデオ・ドライバ、ネットワーク・ドライバおよびディスク・ドライバを検索します。それらのドライバは、削除できる場合がほとんどです。それによって、他のプロセスで利用できるメモリーが増えます。

注意： 必ず UNIX カーネルのバックアップを取っておいてください。
バックアップ方法の詳細は、ハードウェア・ベンダーのドキュメントを参照してください。

V\$ 表を使用して競合を特定する方法

V\$SYSTEM_EVENT 表は、データベース・アクティビティのスナップショットに使用します。

V\$SYSTEM_EVENT 内の統計情報は、oracle が時間をどのように使用しているかを示します。これによって、潜在的な問題を識別できます。表は次の SQL 文で問い合わせます。

```
SELECT * FROM V$SYSTEM_EVENT ORDER BY TIME_WAITED;
```

適切にチューニングされたデータベースでも待機することがあるので、この表にイベントが存在する、または存在しないからといって、それが問題を示しているとは限りません。client message、pmon timer、smon timer、rdbms ipc message、rdbms ipc reply などのイベントは正常です。この表の行数は動的に変更されます。イベントについてレポートする情報がない場合は、この表にイベントは表示されません。

V\$SYSTEM_EVENT は、累積される表です。この表は、イベントの発生を測定する表を参照するのにも役立ちます。V\$SESSION_WAIT 表を使用するには、次のように入力します。

```
SELECT sid, event, p1text, p1, p2text, p2
FROM V$SESSION_WAIT;
```

この問合せによって、一連のイベントのスナップショットが提供されます。イベントの頻度がデータベースの負荷によってどのように変わるかを観察すると、実行されている Oracle 操作と SQL 文の性質の両方がよくわかります。

V\$SESSION_WAIT の問合せの出力例を次に示します。

SID	EVENT	P1TEXT	P1	P2TEXT	P2
1	pmon timer		0		0
2	buffer busy waits	file#	7	block#	792
10	latch free	address	8.05E08	number	8
...					

競合の原因となるセグメントの特定

Oracle リソースの競合が問題であると判断した場合は、競合の原因となっているセグメントを特定します。多数のバッファ・ビジー待機が問題であるという判断にいたることがあります。ブロック番号およびファイル番号を使用し、次のように入力して競合のタイプを判別してください。

```
SELECT segment_name, segment_type, block_id, blocks
FROM dba_extents
WHERE file_id=7 AND (792 between block_id and
block_id+blocks);
```

次に出力例を示します。

SEGMENT_NAME	SEGMENT_TYPE	BLOCK_ID	BLOCKS
COT1	TABLE	752	50

1 row selected.

この出力は、索引、クラスタまたはロールバック・セグメントではなく、表セグメントで競合が発生していることを示しています。ファイル番号およびブロック番号がわかっているの
で、次のように入力することで X\$BH 表から追加情報を取得できます。

```
SELECT class
FROM X$BH
WHERE dbafile=7 AND dbablk=792;
```

ブロック・サイズおよびファイル・サイズのチューニング

注意： ブロック・サイズを変更する場合は、新しくデータベースを作成する必要があります。新しいデータベースにデータを転送する前にブロック・サイズを変更してみて、最適な構成を判断してください。

Oracle ブロック・サイズの指定

HP-UX では、デフォルトの Oracle ブロック・サイズは 2KB、最大ブロック・サイズは 16KB です。実際のブロック・サイズは、16KB 以下で 2KB の倍数の値を設定します。

アプリケーションによって異なりますが、通常、最適なブロック・サイズはデフォルトのサイズです。異なる Oracle ブロック・サイズでデータベースを作成する場合は、データベースを作成する前に `init.ora` ファイルに次の行を追加します。

```
db_block_size=new_block_size
```

HP-UX バッファ・キャッシュ・サイズのチューニング

RAW デバイスを最大限活用するには、Oracle8i バッファ・キャッシュのサイズを調整し、メモリーに制限がある場合は、HP-UX バッファ・キャッシュのサイズも調整します。

HP-UX バッファ・キャッシュは、オペレーティング・システムで提供されます。バッファ・キャッシュには、メモリーからディスク、またはディスクからメモリーに転送されるメモリー内のデータ・ブロックを保存します。

Oracle8i バッファ・キャッシュは、Oracle データベース・バッファを保存するためのメモリー領域です。Oracle8i では RAW デバイスを使用できるので、HP-UX バッファ・キャッシュを使用する必要はありません。

RAW デバイスに移動するときは、Oracle8i バッファ・キャッシュのサイズを大きくしてください。システムで使用できるメモリーに制限がある場合は、それに応じて HP-UX バッファ・キャッシュのサイズを小さくしてください。

HP-UX の sar コマンドは、調整が必要なバッファ・キャッシュを判断する際に役立ちます。表 2-3 に、sar コマンドの構文を示します。

表 2-3 sar コマンドの構文

sar -b	HP-UX のバッファ・キャッシュのアクティビティを表示します。
sar -w	HP-UX のスワッピング・アクティビティを表示します。
sar -u	CPU 使用状況を表示します。
sar -r	メモリー使用量を表示します。

キャッシュ・サイズの調整

- キャッシュ・ヒット率が向上する限り、Oracle8i キャッシュ・サイズを大きくします。
- スワッピング / ページング・アクティビティが増える場合は、キャッシュ・サイズを小さくします。

トレース・ファイルとアラート・ファイルの使用

この項では、操作上の問題を診断して解決するために Oracle8i が作成するトレース・ファイル（またはダンプ・ファイル）およびアラート・ファイルについて説明します。

トレース・ファイルの名前

トレース・ファイル名は、*processname_sid_unixpid.trc* という構文です。表 2-4 に、この構文の詳細を示します。

表 2-4 トレース・ファイル名の構文

<i>processname</i>	トレース・ファイルの作成元である Oracle8i を示す 3 ～ 4 文字のプロセス名です（たとえば、pmon、dbwr、ora、reco など）。
<i>sid</i>	インスタンスのシステム識別子です。
<i>unixpid</i>	UNIX プロセス ID 番号です。
<i>.trc</i>	すべてのトレース・ファイル名に付くファイル名の拡張子です。

たとえば、lgwr_TEST_1237.trc などがあります。

アラート・ファイル

alert_sid.log ファイルは、データベースと関連付けられており、initSID.ora のパラメータ BACKGROUND_DUMP_DEST に指定されているディレクトリにあります。デフォルト値は、\$ORACLE_HOME/rdbms/log です。

RAW デバイス / ボリューム

RAW デバイス / ボリュームの欠点

HP-UX で RAW デバイス / ボリュームを使用する場合、次のような欠点があります。

- 1MB より大きな表（もう 1 つのディスクのパーティションなど）をエクスポートするときに発生する可能性がある問題を、ULIMIT では解決できない場合があります。
- RAW デバイスおよびオペレーティング・システムのファイルが 1 つの Oracle8i データベース内に混在している場合、オペレーティング・システムのファイルも ULIMIT パラメータの値の範囲内でなければなりません。
- ディスクに Oracle 配布の内容を読み込むときに発生する可能性がある問題を、ULIMIT では解決できない場合があります。
- 小規模クライアント・システムでは、通常、十分な大きさの RAW デバイス / ボリュームのパーティションを使用できません。ディスク・パーティションが半端なサイズになり、優れたデータベース・アーキテクチャに適さなくなります。
- 特定のディスク・ドライブに I/O 動作が集中していて、Oracle データ・ファイルを別のドライブに移すとパフォーマンスが向上する場合、I/O 動作の少ないドライブには受け入れられるサイズのセクションがないことがよくあります。UNIX の利点であるデータ・ファイルの移動は、RAW デバイス / ボリュームでは欠点になります。
- RAW デバイス / ボリューム環境では、表領域を増やすことは困難です。初期構成時に、すべての RAW パーティションがデータ・ファイルに割り当てられる場合があるため、通常の表領域の拡張に対応する RAW 記憶領域が残らないことがあります。

RAW デバイス / ボリュームを使用する場合のガイドライン

「RAW デバイス / ボリュームの欠点」で示した要因だけでなく、RAW デバイス / ボリュームを使用するかどうかを決定する場合、次の問題を考慮する必要があります。

- Oracle8i Parallel Server のインストレーション
- RAW ディスク・パーティションの可用性
- 論理ボリューム・マネージャ
- 動的パフォーマンスのチューニング
- ミラー化およびオンラインでのディスクの置換え

Oracle8i Parallel Server のインストールレーション

OPS の各インスタンスには、それぞれ個別のログ・ファイルがあります。そのため、表領域および制御ファイルに必要なパーティションの他に、各インスタンスのログ・ファイル用に、少なくとも3つのパーティションが必要になります。すべてのファイルは、HP-UX クラスタのすべてのノードで共有できるディスク上になければなりません。

UNIX クラスタでは、クラスタのすべてのノード間の共有ファイル・システムにアクセスできません。そのため、データベースと関連付けられているすべてのファイルは、RAW デバイス / ボリューム上に作成する必要があります。

RAW ディスク・パーティションの可用性

使用しているサイトに、少なくとも Oracle 表領域と同数の RAW ディスク・パーティションがある場合のみ、Oracle ファイルに RAW デバイス / ボリュームを使用してください。

RAW ディスク・パーティションがすでにフォーマットされている場合、表領域のサイズをパーティションのサイズにできるだけ合わせて、無駄な領域をなくします。

論理ボリューム・マネージャ

論理ボリュームを使用することによって、RAW パーティションの可用性に基づいて論理ディスクを作成できます。論理ディスクは複数のディスク間で移動できるので、論理ディスク・サイズを有効にするためにディスク・ドライブを再フォーマットする必要はありません。

動的パフォーマンスのチューニング

負荷が集中するディスクのデータを負荷の低いドライブに移すことによって、データベースがオンラインにある場合のディスク・パフォーマンスを最適化できます。論理ディスク機能を提供しているほとんどのハードウェア・ベンダーからも、チューニングに使用できるグラフィカル・ユーザー・インタフェースが提供されています。

ミラー化およびオンラインでのディスクの置換え

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

Oracle Parallel Server の場合、論理ボリュームは、UNIX クラスタにある複数のマシンで共有できるドライブと同様に、1 台の UNIX マシンに関連付けられているドライブでも使用できます。複数のマシンに関連付けられているドライブで論理ボリュームを使用する場合、Oracle Parallel Server と関連付けられているすべてのファイルを、これらの共有論理ボリュームに置くようになります。

RAW デバイスの設定

RAW デバイスを作成する場合、次の項目に注意してください。

- ボリュームを作成する場合、それぞれ所有者およびグループが oracle および dba であることを確認します。
- RAW パーティションに作成した Oracle データ・ファイルのサイズは、RAW パーティションのサイズより Oracle ブロック・サイズ×2 以上小さくなければなりません。

3

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。

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、SYSTEM_PASS 環境変数に、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。

```
% setenv SYSTEM_PASS SYSTEM/MANAGER
% 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、PATH 変数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。

SQL*Plus プロンプトで `ed` または `edit` コマンドを入力すると、ed、emacs、ned、vi などの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、PATH 変数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。

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、EDITOR または VISUAL 環境変数のどちらかの値を使用します。EDITOR および VISUAL 環境変数が両方設定されている場合、EDITOR 変数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。

SQL*Plus からのオペレーティング・システム・コマンドの実行

SQL*Plus プロンプトの後に最初の文字として HOST コマンドまたは感嘆符 (!) を入力すると、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、SHELL 環境変数によって設定されます。デフォルトのシェルは `/bin/sh` (sh) です。シェルが実行できない場合は、エラー・メッセージが表示されます。

制限事項

HOST コマンドまたは ! を入力することによって、SQL*Plus を終了しなくても、オペレーティング・システム・コマンドを実行できます。

たとえば、1 つのコマンドを実行するには、次のように入力します。

```
SQL> ! command
```

この場合、command は実行するオペレーティング・システム・コマンドを表します。一度コマンドを実行すると、制御は SQL*Plus に戻ります。

1 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には、[!] または HOST コマンドを入力した後に [Enter] を押します。

SQL*Plus への割込み

SQL*Plus を実行している間、HP-UX マシンの場合は [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。

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

ウィンドウのサイズ変更

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

リターン・コード

UNIX のリターン・コードは 1 バイトですが、Oracle エラー・コードを返すには 1 バイトでは不十分です。リターン・コードの範囲は、0 ～ 255 です。

Oracle 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
- 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・ハンドラの使用方法
- 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て使用するアプリケーション設計ツールです。Oracle プリコンパイラは、ANSI SQL と互換性があり、Oracle8i やその他の ANSI SQL 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
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

表 4-2 に、\$ORACLE_HOME/precomp/admin にある .cfg システム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	pcscfg.cfg
Pro*COBOL 1.8.50	pcccob.cfg
Pro*COBOL 8.1.5	pccbcfg.cfg
Pro*FORTRAN 1.8.50	pccfor.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

注意： Oracle プリコンパイラのデモンストレーションを実行するには、Oracle8i をインストールしておく必要があります。

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。

参照ドキュメント

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』

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、オペレーティング・システムの適切なバージョンのコンパイラが正しくインストールされていることを確認してください。必要なバージョンの詳細は、『Oracle8i for HP 9000 Servers and Workstations インストレーション・ガイド』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 種類があります。Object 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、3-3 ページの「デモンストレーション表」を参照してください。

参照： 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 を使用して、デモンストレーション・プログラムを作成できます。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.sl に動的にリンクされます。

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
```

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
```

HP-UX で共有ライブラリを使用する場合の詳細は、Hewlett-Packard の HP-UX のドキュメントを参照してください。

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
実行ファイル	procob	procob18
デモ・ディレクトリ	procob2	procob
Make ファイル (MicroFocus COBOL)	demo_procob.mk	demo_procob18.mk

注意： Pro*COBOL は、HP-UX 11.0 64-bit システムでは使用できません。

Pro*COBOL では、静的リンク、動的リンク、または動的読み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.sl` 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/pro1x/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 の各バージョンには、次のようなシステム構成ファイルがあります。

- 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 SHLIB_PATH 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（Bourne または Korn シェルの場合）。

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

SHLIB_PATH 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 SHLIB_PATH に追加します（Bourne または Korn シェルの場合）。

```
$ SHLIB_PATH ${SHLIB_PATH}:/opt/cobol/coblib; export SHLIB_PATH
```

SHLIB_PATH 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。

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

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 HP 9000 Servers and Workstations インストレーション・ガイド』を参照してください。

Oracle ランタイム・システム

Oracle では、動的読み込み 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、動的読み込み 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、3-3 ページの「デモンストレーション表」を参照してください。

参照： 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 ファイルを使用して、サンプル・プログラムを作成できます。

- 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.sl 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、コラム 1～6 はオプションの順序番号、コラム 7 はコメントまたは継続行を示す標識です。さらに、コラム 8～11 は段落の名前で、コラム 12～72 が文となります。

FORMAT=TERMINAL を指定した場合、コラム 1～6 は削除され、コラム 7 が左端のコラムになります。

Pro*FORTRAN

Pro*FORTRAN 1.8.50 の追加情報は、README ファイル
\$ORACLE_HOME/precomp/doc/prolx/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 HP 9000 Servers and Workstations 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、3-3 ページの「デモンストレーション表」を参照してください。

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.sl` 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
```

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 Pro*C/C++ が正しくインストールされていることを確認してください。必要なバージョンについては、『Oracle8i for HP 9000 Servers and Workstations 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/rdbsms/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/rdbsms/demo/demo_rdbms.mk を使用して、デモンストレーション・プログラムを作成できます。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.sl 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
```

注意： C++ プログラムのリンク中に、次のエラー・メッセージが表示される場合があります。

```
ld: fatal: library -lsunmath: not found
ld: fatal: library -lC: not found
ld: fatal: library -lC_mtstubs: not found
ld: fatal: library -lcx: not found
```

この場合は、出力されたライブラリが常駐するディレクトリを環境変数 `SHLIB_PATH` の設定に追加してください。

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手動で SQL スクリプトを実行しておく必要があります。すべてのスクリプトは、`$ORACLE_HOME/rdbms/demo` に格納されています。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_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、オラクル社から提供されている 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。リンク中にすべてのシンボルを解決するために、Oracle ライブラリはリンク行に 2 回以上追加されます。これには、次の 2 つの理由があります。
 1. Oracle ライブラリは、相互に参照し合います。つまり、ライブラリ A の関数はライブラリ B の関数を呼び出し、ライブラリ B の関数はライブラリ A の関数を呼び出します。
 2. HP-UX リンカーは、1 パスのリンカーです。つまり、リンカーは、リンク行に出現した時点で 1 回だけライブラリを検索します。
- 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 ライブラリの間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、HP-UX 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、HP-UX の `make` ユーティリティでサポートされています。

- ## 未定義シンボル

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

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 `salcx` が定義されている場所を調べる `symfind` の出力例を示します。

4-16 Oracle8i for HP 9000 Servers and Workstations 管理者リファレンス

スレッドのサポート

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

Oracle ライブラリを使用した静的リンクおよび動的リンク

プリコンパイラおよび OCI アプリケーションは、静的および動的に Oracle ライブラリにリンクできます。静的リンクの場合、アプリケーション全体のライブラリおよびオブジェクトは、1つの実行ファイル・プログラムにリンクされます。そのため、アプリケーションの実行ファイルのサイズは非常に大きくなります。

動的リンクの場合、実行コードは一部分のみが実行ファイル・プログラムに常駐し、実行時にアプリケーションと動的にリンクされるライブラリにも常駐します。実行時にリンクされるライブラリを、動的ライブラリまたは共有ライブラリと呼びます。動的リンクには、主に次の利点があります。

1. **必要なディスク領域が少なくて済む**
異なるアプリケーション、または同一アプリケーションからの異なる呼出しであっても、同一の共有ライブラリまたは動的ライブラリを使用できます。その結果、必要なディスク領域を削減できます。
2. **必要なメイン・メモリーが少なくて済む**
異なるアプリケーションで、同一の共有ライブラリ・イメージまたは動的ライブラリ・イメージ（メモリ内コピー）を共有できます。つまり、1つのライブラリをメイン・メモリーに1回だけ読み込めば、複数のアプリケーションでそのライブラリを使用できます。その結果、必要なメイン・メモリーを削減できます。

Oracle 共有ライブラリ

Oracle 共有ライブラリは、`$ORACLE_HOME/lib/libclntsh.sl` 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 `SHLIB_PATH` 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、`SHLIB_PATH` 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
ld.sl.1: sample1: fatal: libclntsh.sl.1.0: can't open file: errno=2
Killed
```

次のように入力して、SHLIB_PATH を設定します。

```
% setenv SHLIB_PATH $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.sl
```

シグナル・ハンドラの使用方法

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

SIGCONT	アウトオブバンド・ブレイクをユーザー・プロセスから <i>oracle</i> プロセスへ送信する場合に、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、 <i>oracle</i> ではなく、ユーザー・プロセスが受け取ります。
SIGPIPE	通信チャン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、 <i>oracle</i> 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、 <i>oracle</i> プロセスには適用されません。 <i>oracle</i> 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 wait() を使用します）。SIGCLD は、 <i>oracle</i> 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 <i>oracle</i> 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、 <i>oracle</i> プロセスが受け取ります。

表 4-4 2 タスク通信のためのシグナル

SIGIO	ネットワーク・イベントの着信を示すために、Net8 プロトコル・メソッドが使用します。
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、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 のクライアント共有ライブラリより前に設定されていることを確認してください。このリンク制限は、XA の動的登録（Oracle XA スイッチ `xaoswd`）を使用する場合のみ必要です。

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 `$ORACLE_HOME/lib/libclntsh.sl` および静的ライブラリ `$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
- APPC/LU6.2 プロトコル
- 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	\$ORACLE_HOME/network/doc/README.Net8
Advanced Security	\$ORACLE_HOME/network/doc/README.ASO
Oracle Intelligent Agent	\$ORACLE_HOME/oemagent/doc/README.oemagent

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

HP-UX 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 /var/opt/oracle 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. /var/opt/oracle 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。それらは、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 プロトコルを表示するには、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 Net8 プロトコルが表示されます。

```
% adapters oracle
Net Protocol Adapters linked with oracle are:
    BEQ
    IPC
    TCP/IP
    RAW
    SSL
Net Naming Adapters linked with oracle are:
    Oracle TNS Naming Adapter
    Oracle Naming Adapter
    Host Naming Adapter
    LDAP Naming Adapter
```

Net8 プロトコルのサポート

Oracle Advanced Security/Network Security products linked with oracle are:
SecurID
Identix

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/netasst) 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

HP-UX 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
- APPC/LU6.2 プロトコル

TCP/IP、APPC/LU6.2 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 HP 9000 Servers and Workstations インストール・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 および APPC/LU6.2 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 initsid.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。

サポート・プロトコル	ADDRESS 指定
BEQ	(ADDRESS= (PROTOCOL = BEQ) (PROGRAM = ORACLE_HOME/bin/oracle) (ARGV0 = oracleORACLE_SID) (ARGS = ' (DESCRIPTION= (LOCAL=YES) (ADDRESS= (PROTOCOL=BEQ))) ' (ENVS = 'ORACLE_HOME=ORACLE_HOME,ORACLE_SID=ORACLE_SID'))
IPC	(ADDRESS= (PROTOCOL=IPC) (KEY=key))
RAW	N/A
TCP/IP	(ADDRESS= (PROTOCOL=TCP) (HOST=hostname) (PORT=port_id))
APPC/LU6.2	(ADDRESS= (PROTOCOL=LU62) (TP_NAME=transaction_program_name) (LU_NAME=logical_unit_name) (MODE=mode_name) (PLU=partner_lu_name))

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 1521 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)
)
```

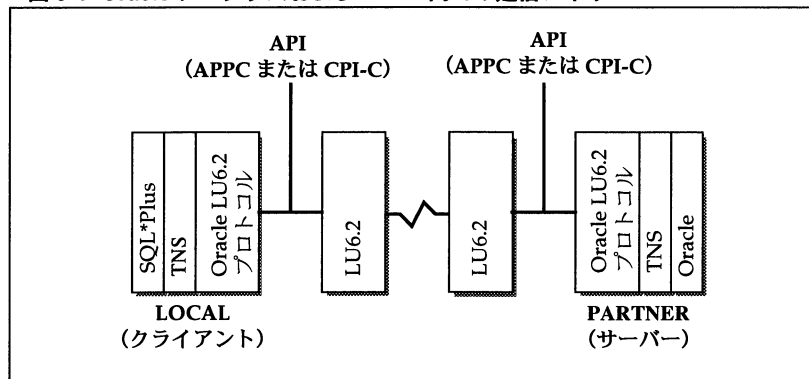
最後のフィールドは、PORT=listener というように名前を指定することもできます。通常、ADDRESS の指定は、接続記述子、構成ファイルなどを構成している内容の一部です。

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=lu_name)
  (MODE=modename)
  (PLU=partner_lu_name)
)
```

表 5-6 に、APPC/LU6.2 プロトコル接続の構文を示します。

表 5-6 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。
LU_NAME	<code>tnsnames.ora</code> を参照する場合、リモート・パートナー LU の名前を指定します。このキーワードが <code>listener.ora</code> に指定されている場合は、ローカル LU の名前を指定します。LU_NAME は、無視されたり、他のパラメータで上書きされたりすることがあります。一部の APPC/LU6.2 実装要件のために、LU_NAME には、常に完全修飾された LU_NAME (<code>netid.lu_name</code>) を指定する必要があります。
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、パートナー LU およびトランザクション・プログラム名とともに、ALLOCATE セグメントで指定します。 <code>modename</code> は、ローカル LU およびパートナー 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	パートナー LU の名前を指定します。HP-UX では、この値は必須で、TP_NAME に設定できます。

Oracle Enterprise Manager (OEM)

エージェント・サービスの展開および自動構成

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）。

参照： 詳細は、『Oracle8 Enterprise Manager 構成ガイド』を参照してください。

Tcl スクリプトのデバッグ

実行ファイル `oratclsh` は、Tcl スクリプトをデバッグする場合に必要です。`oratclsh` を実行する前に、環境変数 `TCL_LIBRARY` に、`$ORACLE_HOME/network/agent/tcl` を設定してください。

参照： 詳細は、『Oracle 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

start_peer スクリプトの確認

`start_peer` スクリプトは、次のような行を含みます。

```
SNMPD = snmpd_executable_path
```

ご使用のシステムの `snmpd` 実行ファイルが、`start_peer` スクリプトによって示された位置にない場合、`snmpd_executable_path` が `snmpd` 実行ファイルの正しい位置を示すように編集します。

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. 次のように入力して、SNMP コンポーネントの `master_peer`、`encap_peer` および `snmpd` が実行されていないことを確認します。

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

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

2. root ユーザーで PEER マスター・エージェント (PEER カプセル化プログラムおよびネイティブな HP-UX SNMP エージェント) を起動するために start_peer スクリプトを実行します。次のように入力します。

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

注意： ご使用のシステムにネイティブな HP-UX SNMP エージェントがない場合、PEER カプセル化プログラムは使用できません。マスター・エージェントのみを起動するには、start_peer -m を実行します。

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 peer
# ps -aef | grep snmp
```

データベース・サブエージェントの構成および起動

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

Oracle Advanced Security

.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。

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 インストレーション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) ・ インストレーションの手順

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

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

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

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

3. 次のように入力して、ディレクトリの権限を PUBLIC に与えます。

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

4. 必要ならば、imgdemo プログラムを作成します。

```
% cd $ORACLE_HOME/ord/img/demo
% make -f demo_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。

- geohhttp.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。

参照： このファイルのリストについては、『Oracle8i *interMedia* Locator ユーザーズ・ガイドおよびリファレンス』の第 2 章「`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』の第 1 章を参照）。	demo/tsquick
使用方法デモは、エンド・ユーザー向けおよび既存の Oracle8i Time Series 機能を使用する製品開発者向けです（詳細は、『Oracle8i Time Series ユーザーズ・ガイド』の第 1 章を参照）。	demo/usage
電子ユーティリティ・アプリケーションは、15 分間のデータのサマリーをピークおよびオフ・ピークで計算する方法を示します。	demo/usageutl
Time Series データの Java ベース検索は、Oracle8i Time Series Java API のプロトタイプを使用して、Web ブラウザで実行するために設計されています（詳細は、『Oracle8i Time Series ユーザーズ・ガイド』の第 1 章を参照）。	demo/applet
Java コードのサンプル・セグメントは、Time Series を操作して、その結果を印刷します（詳細は、『Oracle8i Time Series ユーザーズ・ガイド』の第 1 章を参照）。	demo/java
管理ツールを使用して既存の Time Series 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
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) 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。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/applmgr* 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

<code>h</code>	標準のディレクトリ名
<code>v</code>	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。ORACLE_HOME 環境変数は、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

<code>adhoc</code>	指定したデータベースの非定型 SQL スクリプト
<code>arch</code>	アーカイブ REDO ログ・ファイル
<code>adump</code>	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。)
<code>bdump</code>	バックグラウンド・プロセスのトレース・ファイル
<code>cdump</code>	コア・ダンプ・ファイル
<code>create</code>	データベースを作成するプログラム
<code>exp</code>	データベース・エクスポート・ファイル
<code>logbook</code>	データベースの状態および履歴を記録するファイル
<code>pfile</code>	インスタンス・パラメータ・ファイル
<code>udump</code>	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 oradata)
<i>d</i>	データベースの DB_NAME
<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
8.0.4/		8.0.4 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
	inst2/	inst2 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
	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
	inst2/	inst2 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
	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、/oa/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	interMedia Text オプション
db	init _{sid} .ora、lk _{sid}
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

表 A-11 ORACLE_HOME ディレクトリ構造

md	Spatial オプション
mlx	Xerox Stemmer (<i>interMedia</i> Text 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 Manager Components
oracore	コア・ライブラリ
ord	データ・オプション
otrace	Oracle TRACE
plsql	PL/SQL
precomp	プリコンパイラ
rdbms	データベースに必要なサーバー・ファイルおよびライブラリ
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)

表 A-12 製品のサブディレクトリの内容

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、DBMS_OUTPUT など）を作成するためにスクリプトを順に実行します。
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

表 A-15 ファイル名の拡張子

.c	C ソース・ファイル
.ctl	SQL*Loader 制御ファイル、Oracle Server 制御ファイル
.dat	SQL*Loader データ・ファイル
.dbf	Oracle Server 表領域ファイル
.dmp	エクスポート・ファイル
.doc	ASCII テキスト・ファイル
.env	環境設定用のシェル・スクリプト・ファイル
.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

記号

- \$
使用, 1-19
- ?
使用例, 1-7

数字

- 2 タスク
アーキテクチャ
セキュリティ, 1-16
シグナル, 4-18

A

- ADDRESS 指定
プロトコル, 5-5
- APPC/LU6.2 プロトコル, 5-10
ADDRESS, 5-10

B

- BEQ プロトコル, 5-6
ADDRESS, 5-6
接続パラメータの構文, 5-6
- Bourne シェル・スクリプトのトレース, 1-6
- BSD-UNIX ファイル・システム・タイプ, 2-18

C

- C
Pro*C/C++, 4-4
- CATPROC.SQL, A-15

- COBOL
Pro*COBOL, 4-7
- CONNECT INTERNAL
セキュリティ, 1-17
- coraenv, 1-6
- CPU の使用状況
シングルタスク・アーキテクチャ, 2-20
チューニング, 2-19
プロセスの優先順位, 2-19
プロセッサのバインディング, 2-20

D

- daemon ユーザー
セキュリティ, 1-19
- DB_BLOCK_SIZE パラメータ, 1-28
- DBA_GROUP
/etc/listener.ora ファイル, 1-20
- DBA_GROUP_sid
/etc/listener.ora ファイル, 1-20
- dba グループ
再リンク, 1-16
メンバー, 1-15
- DBA グループ ID
キーワード, 1-19
デフォルト以外の名前, 1-20
- DBWR
チューニング, 2-14
- Developer/2000 (Oracle Developer) デモ, 6-9

E

- echo コマンド, 1-2
- expst コマンド, 2-20

F

FORMAT プリコンパイラ
Pro*COBOL, 4-10, 4-11

G

Glance/UX, 2-9
glogin.sql, 3-2

H

hfs ファイル・システム・タイプ, 2-18
HP-UX での非同期 I/O, 2-15
HP-UX のツール, 2-3, 2-9
iostat, 2-7
mpstat, 2-8
swap, 2-8

I

impst コマンド, 2-20
initsid.ora ファイルのカスタマイズ
初期化パラメータ, 1-9
Intelligent Agent
SNMP での構成, 5-12
I/O
DBWR チューニング, 2-14
チューニング, 2-14
ディスク要求キュー, 2-15
iostat, 2-7
I/O スレーブ, 2-14
ipcs コマンド, 2-10
IPC プロトコル, 5-7
ADDRESS, 5-7
ireclen, 4-3

J

Java
デモ, 6-9

L

login.sql, 3-2

M

MAXDATAFILES パラメータ, 1-14
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
MicroFocus COBOL コンパイラ, 4-8
mpstat, 2-8

N

Net8
ADDRESS 指定, 5-5
APPC/LU6.2 プロトコル, 5-10
BEQ プロトコル, 5-6
IPC プロトコル, 5-7
Oracle Advanced Security, 5-13
Oracle Connection Manager, 5-4
Oracle Enterprise Manager Intelligent Agent, 5-12
Oracle Names, 5-4
RAW プロトコル, 5-8
TCP/IP プロトコル, 5-9
アダプタ・ユーティリティ, 5-3
自動ログイン, 1-19
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, 5-4
マルチスレッド・サーバー, 5-4
Net8 Assistant
Java Runtime Environment (JRE), 5-4
Net8 の構成ファイル
保存場所, 5-2
NLS_LANG
環境変数, 1-5

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, 1-26
Oracle Advanced Security, 5-13
.bak ファイル, 5-13

- セキュリティおよびシングル・サインオン, 5-13
- Oracle Connection Manager, 5-4
- Oracle Enterprise Manager Intelligent Agent, 5-12
 - tcl スクリプトのデバッグ, 5-12
 - エージェント・サービスの展開および自動構成, 5-12
- Oracle Forms デモ, 6-9
- Oracle Names, 5-4
- Oracle Server
 - アカウント, 1-15
- ORACLE_HELP
 - 環境変数, 1-5
- ORACLE_HOME
 - ? の使用, 1-7
 - 環境変数, 1-5
- ORACLE_SID
 - 環境変数, 1-6
 - 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, 1-5
 - EPC_DISABLED, 1-5
 - NLS_LANG, 1-5
 - ORA_NLS33, 1-5
 - ORACLE_BASE, 1-5
 - ORACLE_HELP, 1-5
 - ORACLE_HOME, 1-5
 - ORACLE_PATH, 1-6
 - ORACLE_SID, 1-6
 - ORACLE_TRACE, 1-6
 - ORAENV_ASK, 1-6
 - TNS_ADMIN, 1-6
 - TWO_TASK, 1-6
 - UNIX 変数, 1-7
- 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, 4-13
 - 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
- 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, 1-15
 - 特殊アカウント, 1-15
- Oracle のシステム ID, 1-6
- Oracle プリコンパイラ, OCI のリンクおよび Make ファイル, 4-15
 - カスタム Make ファイル, 4-15

- 未定義シンボル, 4-16
- Oracle ライブラリ
 - Oracle 共有ライブラリ, 4-17
 - 静的リンクと動的リンク, 4-17
- 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, 1-21
- oreclen, 4-3

P

- PL/SQL
 - デモンストレーション
 - 実行, 1-33
 - ロード, 1-32
- Pro*C/C++
 - Make ファイル, 4-5, 4-6
 - 管理, 4-4
 - シグナル, 4-19
 - 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, 6-9
- Pro*COBOL, 4-7
 - FORMAT プリコンパイラ, 4-10, 4-11
 - Oracle ランタイム・システム, 4-8
 - 環境変数, 4-7, 4-8
 - 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, 4-9
 - ネーミング方法の違い, 4-7
 - ユーザー・プログラム, 4-10
- PRODUCT_USER_PROFILE 表
 - SQL*Plus, 3-3
- Programmer's Analysis Kit (HP PAK), 2-9
- pupbld.sql, 3-3

R

- RAW デバイス, 2-25
 - Oracle8i Parallel Server インストレーション, 2-26

- RAW ディスク・パーティションの可用性, 2-26
 - 欠点, 2-25
 - 使用基準, 2-25
 - バッファ・キャッシュ・サイズ, 2-23
- RAW デバイスの設定, 2-27
- RAW プロトコル, 5-8
- README.txt ファイル, 6-10
- REDO バッファ
 - チューニング, 2-12
- REDO ログ
 - バッファ・サイズのチューニング, 2-12
- root
 - ユーザー, 1-15

S

- sar コマンド, 2-10, 2-23
- Server Manager
 - SHOW PARAMETERS, 1-12
 - コマンド, 1-17
 - セキュリティ, 1-17
- SGA
 - 計算, 1-29
 - チューニング, 2-10
 - 物理メモリーのロック, 2-11
- SHELL, 1-8
- SHUTDOWN コマンド, 1-17
 - セキュリティ, 1-17
- SIGCLD 2 タスク・シグナル, 4-18
- SIGIO 2 タスク・シグナル, 4-19
- SIGINT 2 タスク・シグナル, 4-18
- SIGPIPE 2 タスク・シグナル, 4-18
- SIGTERM 2 タスク・シグナル, 4-18
- SIGURG 2 タスク・シグナル, 4-19
- Site Profile
 - SQL*Plus, 3-2
- SNMP
 - Intelligent Agent, 5-12
- snmpd 実行ファイル, 5-12
- Spatial
 - デモンストレーション
 - Oracle データ・オプション, 6-12
- SPOOL コマンド
 - SQL*Plus, 3-6
 - 使用, 3-6
- SQL
 - 管理, 3-2

- SQL*DBA
 - SHOW PARAMETERS, 1-12
- SQL*Loader
 - デモンストレーション, 1-29
- SQL*Plus
 - PRODUCT_USER_PROFILE 表, 3-3
 - Site Profile, 3-2
 - SPOOL コマンド, 3-6
 - User Profile, 3-2
 - エディタ, 3-5
 - オペレーティング・システム・コマンドの実行,
 - 3-5
 - システム・エディタ, 3-5
 - 使用, 3-5
 - 制限事項, 3-6
 - 設定ファイル, 3-2
 - デフォルト・エディタ, 3-5
 - デモンストレーション表, 3-3
 - ヘルプ機能, 3-4
 - 割込み, 3-6
- SQL*Plus の使用, 3-5
- SQL*Plus への割込み, 3-6
- SQL スクリプト, 2-9
- start_peer スクリプト, 5-12
- STARTUP コマンド
 - セキュリティ, 1-17
- swap, 2-8
- SYS
 - アカウントおよびグループ, 1-15
- SYSDATE
 - TZ, 1-9
- SYSTEM
 - アカウントおよびグループ, 1-15
- SYSTEM アカウント
 - 権限, 1-15
- SYS アカウント
 - 権限, 1-15

T

- TCP/IP プロトコル, 5-9
 - ADDRESS, 5-9
- 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, 1-9
環境変数, 1-9

U

UNIX
セキュリティ, 1-16
UNIX カーネル
チューニング, 2-20
User Profile
SQL*Plus, 3-2

V

V\$SESSION_WAIT 表, 2-21
V\$SGASTAT 表, 2-12
V\$SYSTEM_EVENT 表, 2-21
Veritas ファイル・システム・タイプ, 2-18
vmstat コマンド, 2-10
vxfs ファイル・システム・タイプ, 2-18

X

XA 機能, 4-21

あ

アダプタ・ユーティリティ, 5-3

い

インポート
チューニング, 2-20

う

受取りルーチン
例, 4-19

え

エクスポート
チューニング, 2-20

エディタ
SQL*Plus, 3-5

お

オペレーティング・システム・コマンドの実行
SQL*Plus, 3-5

か

カーネル
UNIX パラメータのチューニング, 2-20
各国語サポート (NLS)
変数, 1-5
カプセル化プログラムの構成
カプセル化プログラム, 5-12
環境変数, 4-8
MicroFocus COBOL コンパイラ, 4-8
NLS_LANG, 1-5
ORA_NLS, 1-5
ORACLE_HELP, 1-5
ORACLE_HOME, 1-5
ORACLE_SID, 1-6
ORACLE_TRACE, 1-6
ORAENV_ASK, 1-6
Pro*COBOL, 4-7, 4-8
SHELL, 1-8
TNS_ADMIN, 1-6, 5-3
TWO_TASK, 1-6
TZ, 1-9
管理
SQL, 3-2

き

キーワード
DBA グループ ID, 1-19
疑問符
使用例, 1-7
キャッシュ
サイズ
チューニング, 2-23
キャッシング
SGA のチューニング, 2-11
共通の環境
oraenv ファイル, 1-2
設定, 1-2

共有プール・サイズ
チューニング, 2-12
共有メモリー
SGA, 1-26
SGA のチューニング, 2-10

く

グループ
サンプル・スクリプト, 1-24

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, 5-12
構成ファイル
Net8, 5-2
プリコンパイラ, 4-3
コマンド
expst, 2-20
impst, 2-20
ipcs, 2-10
orapwd, 1-21
sar, 2-10
vmstat, 2-10
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
Oracle8 for UNIX で適用されるインプリメンテーション, A-2
ディレクトリ構造, A-13
ファイル・マッピング, A-2, A-9
再リンク, 1-35
パラメータ, 1-35
参照マニュアル, ix

し

シグナル
2 タスク, 4-18
ハンドラの作成, 4-18
シグナル・ハンドラ
シグナル, 4-18
使用, 4-18
シグナル・ルーチン
例, 4-19
システム・エディタ
SQL*Plus, 3-5
システム・グローバル領域 (SGA)
要件, 1-26
システム構成ファイル
Pro*C/C++, 4-4
Pro*COBOL, 4-7
システム時刻
設定, 1-9
自動ログイン
listener.ora ファイル, 1-19
remote_os_roles, 1-23
UNIX 以外のシステム, 1-19
自動 (ops\$) ログイン, 1-19
シャドウ・プロセス
セキュリティ, 1-16
上級開発者デモ, 6-9
初期化パラメータ, 1-9
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, 1-13
DB_FILE_MULTIBLOCK_READ_COUNT, 1-13
DB_FILES, 1-12
DISTRIBUTED_TRANSACTIONS, 1-13
HASH_AREA_SIZE, 1-13
HASH_MULTIBLOCK_IO_COUNT, 1-13
LOCK_SGA, 1-13
LOCK_SGA_AREAS, 1-13
LOG_ARCHIVE_BUFFER_SIZE, 1-13
LOG_ARCHIVE_BUFFERS, 1-13
LOG_ARCHIVE_DEST, 1-13
LOG_ARCHIVE_FORMAT, 1-13

LOG_BUFFER, 1-13
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
REMOTE_OS_AUTHENT, 1-19
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, 1-14
デフォルト, 1-12

す

スーパーユーザー, 1-15
スクリプト
 start_peer, 5-12
スレッドのサポート, 4-17
スワップ領域
 チューニング, 2-9

せ

制限
 リソース, 1-26
制限事項 (SQL*Plus), 3-6
 ウィンドウのサイズ変更, 3-6
 リターン・コード, 3-6
静的リンクと動的リンク
 Oracle ライブラリ, 4-17
セキュリティ
 2 タスク・アーキテクチャ, 1-16
 CONNECT INTERNAL, 1-17
 Server Manager アクセス, 1-17
 SHUTDOWN コマンド, 1-17
 STARTUP コマンド, 1-17

グループ・アカウント, 1-16
権限の割当て, 1-16
デフォルト・グループ名, 1-16
ネットワーク, 1-18
ファイル所有権, 1-16
設定ファイル
 SQL*Plus, 3-2

た

タイム・ゾーン
 TZ での設定, 1-9
単一共有メモリー・セグメント, 2-10

ち

チューニング
 CPU の使用状況, 2-19
 HP-UX バッファ・キャッシュ・サイズ, 2-23
 I/O ボトルネック, 2-14
 競合の特定, 2-21
 ディスク I/O, 2-14
 トレース・ファイルおよびアラート・ファイル,
 2-23
 ブロック・サイズおよびファイル・サイズ, 2-22
 メモリー管理, 2-9
 リソースの競合, 2-20

つ

ツール, 2-3, 2-9

て

ディスク
 パフォーマンスの監視, 2-18
 割当て制限, 1-26
ディスク I/O
 I/O スレーブ, 2-14
 チューニング, 2-14
 データベース・ライターのチューニング, 2-14
 非同期 I/O, 2-14
 ファイル・システム・タイプ, 2-18
 要求キュー, 2-15
ディスク・パフォーマンス
 問題, 2-19

- データ・オプション
 - デモの実行, 6-1
- データ・ディクショナリ・キャッシュ
 - チューニング, 2-13
- データベース
 - 管理者
 - 実行プログラムへの権限, 1-16
 - バッファのチューニング, 2-11
 - ファイル
 - セキュリティ, 1-17
 - 認可, 1-17
- データベース I/O
 - DBWR チューニング, 2-14
- データベースの変更, 1-3
- データベース・ライター
 - チューニング, 2-14
- デバック・プログラム, 4-3
- デモの使用, 6-9
- デモンストレーション
 - プリコンパイラ, 1-33
 - プロシージャ・オプション、PL/SQL, 1-32
- デモンストレーション表
 - SQL*Plus, 3-3
 - 削除, 3-3
 - 手動で作成, 3-3
- 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, 4-13
 - Pro*C/C++, 4-4
 - Pro*COBOL, 4-8
- デモ (デモンストレーション・ファイル)
 - Java, 6-9
 - Oracle 開発者, 6-9
 - Oracle コール・インタフェース (OCI), 6-9
 - Pro*C/C++, 6-9
 - 既存表のレトロフィット, 6-9
 - 上級開発者, 6-9
 - 使用方法, 6-9
 - 電子ユーティリティ, 6-9
- 電子ユーティリティ・デモ, 6-9

と

- 動的リンクと静的リンク
 - Oracle ライブラリ, 4-17
- 特殊アカウント, 1-15
- 特殊グループ
 - root, 1-15

- トレース・ファイルおよびアラート・ファイル
 - アラート・ファイル, 2-24
 - 使用, 2-23
 - トレース・ファイル名, 2-23

ね

- ネットワーク
 - DBA 権限, 1-18
 - セキュリティ, 1-18
 - パスワード, 1-18

は

- 配布ソフトウェア, 1-5
- バインディング、プロセス, 2-20
- パスワード
 - リモート, 1-21
- パスワード・ファイル
 - 共有, 1-22
- バッファ
 - チューニング, 2-11
- バッファ・キャッシュ・サイズ
 - チューニング, 2-23
- バッファ・マネージャ, 2-9
- パラメータ
 - CREATE CONTROL FILE, 1-14

ひ

- ヒット率
 - バッファ・キャッシュ用, 2-11
- 非同期 I/O, 2-14
 - 使用, 2-14

ふ

- ファイル
 - トレース・ファイル, 2-23
- ファイル名
 - SQL*Plus でのデフォルト拡張子, 3-6
- プール・サイズ
 - チューニング, 2-12
- 複数のシグナル・ハンドラ, 4-19
- 複数のデータベース
 - 共有パスワード・ファイル, 1-22

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, 4-19
- 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, 1-33
- ベンダー提供のデバッグ・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
 - チューニング, 2-11
- ブロック・サイズおよびファイル・サイズ指定, 2-22
- プロトコル, 5-4
 - ADDRESS 指定, 5-5

へ

- ページング領域
 - チューニング, 2-9, 2-10
- ヘルプ機能
 - SQL*Plus, 3-4
- ヘルプ・ファイル, 1-5

ま

- マスター・エージェントの構成, 5-12
- マルチスレッド・サーバー, 5-4

め

- メモリー
 - SGA のチューニング, 2-10
 - 仮想, 1-25
 - 共有, 1-26
 - 使用量の見積り, 1-25
 - チューニング, 2-9
- メモリー管理, 2-9
 - UNIX カーネル, 2-10
 - スワップ領域, 2-10
 - 単一共有メモリー・セグメント, 2-10
 - ページングの制御, 2-10
- メモリー使用量の見積り, 1-25

ゆ

- ユーザー
 - サンプル・スクリプト, 1-24
- ユーザー・プログラム
 - Oracle コール・インタフェース, 4-14
 - Pro*C/C++, 4-6
 - Pro*COBOL, 4-10
- ユーザー割込みハンドラ, 4-19
- ユーティリティ (電子)・デモ, 6-9

ら

- ライターのアクティビティ
 - チューニング, 2-14
- ライブラリ・キャッシュ
 - チューニング, 2-13

り

- リソースの競合
 - カーネル・パラメータ, 2-20
 - チューニング, 2-20
- リソースの制限, 1-26
- リモート
 - 接続
 - INTERNAL として, 1-23
 - OPERATOR として, 1-23
 - ログイン
 - セキュリティ, 1-19
- リモート DBA アクセスの制御
 - REMOTE_DBA_OPS_ALLOWED, 1-20
 - REMOTE_DBA_OPS_DENIED, 1-20
- リモート接続パラメータ
 - OS_AUTHENT_PREFIX, 1-23
 - REMOTE_OS_AUTHENT, 1-23
 - REMOTE_OS_ROLES, 1-23
- リモート・ログインの制御, 1-20
 - OPS_DOLLAR_LOGIN_DENIED, 1-20
- リンク
 - シングルタスク, 2-20

れ

- レトロフィット・デモ, 6-9

ろ

ログイン・ホーム・ディレクトリ
管理, 1-23
サンプル・スクリプト, 1-24