

# Oracle8 Server

---

## 移行ガイド

リリース 8.0

1998 年 2 月

部品番号 A56821-1

**ORACLE®**

---

Enabling the Information Age™

Oracle8 Server 移行ガイド

部品番号 A56821-1

リリース 8.0

第 1 版 : 1998 年 2 月

原本名 : Oracle8 Migration

原本部品番号 : A58243-01

Copyright© Oracle Corporation, 1997, 1998.

All rights reserved.

Printed in Japan.

原著者 : Randy Urbano

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。

本書の情報は、予告なしに変更されることがあります。本書に問題を見つけたら、当社にコメントをお送りください。オラクル社は、本書の無謬性を保証しません。

危険な用途への使用について

当社製品は、原子力、航空産業、大量輸送、又は医療の分野など、本質的に危険が伴うアプリケーションを用途として特に開発されておりません。当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って行っていただきたく、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、当社および開発元である米国 Oracle Corporation（その関連会社も含みます）は一切責任を負いかねます。

ORACLE は、Oracle Corporation の登録商標です。

本文中の他社の商品名は、それぞれ各社の商標または登録商標です。

---

# 目次

はじめに .....	ix
対象読者と前提条件 .....	x
このマニュアルの構成 .....	x
このマニュアルで使用する表記規則 .....	xii
1 移行の概要 .....	
用語 .....	1-2
移行ステップの概要 .....	1-2
ステップ 1: 移行のための準備 .....	1-3
ステップ 2: 移行処理のテスト .....	1-3
ステップ 3: 移行したテスト・データベースのテスト .....	1-3
ステップ 4: ソース・データベースの準備と保存 .....	1-3
ステップ 5: 本番データベースの移行 .....	1-4
ステップ 6: 新しいバージョン 8 の本番データベースのチューニングと調整 .....	1-4
移行におけるデータベース管理者の役割 .....	1-4
移行におけるアプリケーション開発者の役割 .....	1-5
2 移行の準備 .....	
ステップ 1: 移行のための準備 .....	2-2
新しいバージョン 8 のデータベースの機能を理解する .....	2-2
移行方法の選択 .....	2-2
システム要件と使用可能なリソースの見積り .....	2-8
Oracle バージョン 7、リリース 7.X 以上から開始 .....	2-9
共通移行問題の回避 .....	2-9

バックアップ計画の準備 .....	2-9
テスト計画の作成 .....	2-10
ステップ 2: 移行処理のテスト .....	2-12
ステップ 3: 移行したテスト・データベースのテスト .....	2-13
<b>3 移行ユーティリティによる移行</b>	
バージョン 8 の移行ユーティリティを使った移行の概要 .....	3-2
移行処理の概要 .....	3-2
移行ユーティリティの使用 .....	3-3
システムの考慮事項と要件 .....	3-3
領域要件 .....	3-3
ブロック・サイズの考慮事項 .....	3-4
レプリケーション環境のための考慮事項 .....	3-4
異なるコンピュータ・アーキテクチャへの移行 .....	3-5
文字のコード化に関する考慮事項 .....	3-5
移行のためのバージョン 7 のソース・データベースの準備 .....	3-5
バージョン 8 の移行ユーティリティのインストール .....	3-7
バージョン 8 の移行ユーティリティ・コマンド行オプションの検討 .....	3-8
バージョン 7 のソース・データベースの移行 .....	3-10
バージョン 7 の環境での移行ステップ .....	3-10
バージョン 7 のソース・データベースの保存 .....	3-12
バージョン 8 の環境での移行ステップ .....	3-13
移行時のエラー .....	3-18
移行の中止 .....	3-18
<b>4 エクスポート / インポートによる移行</b>	
エクスポート / インポートの基本 .....	4-2
エクスポート要件 .....	4-2
インポート要件 .....	4-2
その他のエクスポート / インポート情報源 .....	4-2
その他のオプション .....	4-2
エクスポート / インポートによるバージョン 8 より前のソース・データベースの移行 .....	4-3

5	データベースの移行後	
	バージョン 8 のデータベースのバックアップ.....	5-2
	無効な日付制約のチェック.....	5-2
	無効にされたビットマップ索引の再構築.....	5-3
	データベースのテストと結果の比較.....	5-3
	ターゲット・データベースのチューニング.....	5-3
	新機能の適宜追加.....	5-4
	必要な新管理手順の作成.....	5-4
6	バージョン 7 のアプリケーションのアップグレード	
	Oracle アプリケーションのバージョン 8 へのアップグレード.....	6-2
	XA コール: リリース 7.1 XA コールとの非互換性.....	6-2
	ブリコンパイラおよび OCI アプリケーションのアップグレード.....	6-2
	ブリコンパイラ・アプリケーションのアップグレード.....	6-3
	既存アプリケーションの簡易アップグレード.....	6-3
	OCI アプリケーションのアップグレード: 制約の使用可能化.....	6-4
	OCI アプリケーションのリンク・ライン.....	6-5
	バージョン 6 の OCI ライブラリを使用するアプリケーション.....	6-5
	LONG から LOB へのアップグレード.....	6-5
	バージョン 7 の Forms または Developer/2000 アプリケーションのアップグレード.....	6-6
	データ・ディクショナリ・ビューの更新.....	6-6
	SQL*Plus スクリプトのアップグレード.....	6-6
	PL/SQL V2 互換モード.....	6-7
	PLSQL_V2_COMPATIBILITY フラグ.....	6-7
	バージョン 7 とバージョン 8 でのキーワード動作の違い.....	6-8
	バージョン 7 とバージョン 8 での新しいキーワードまたは型の動作の違い.....	6-8
	SQL*Net または Net8.....	6-9
	SQL*Net バージョン 1 から SQL*Net バージョン 2 へのアップグレード.....	6-9
	バージョン 7 の Net2 クライアントおよび接続マネージャ.....	6-9
	再リンクされたバージョン 7 のクライアントで使用可能な Net8 機能.....	6-10
	バージョン 8 の Net8 クライアントの場合.....	6-10
	バックアップ管理: EBU および Recovery Manager.....	6-11

ディクショナリ保護.....	6-11
パスワード管理.....	6-12
バージョン7またはそれ以前のクライアントとバージョン8のサーバー.....	6-13
バージョン8のクライアントとバージョン7またはそれ以前のサーバー.....	6-13
エクスポート/インポートの使用方法、パーティション・オブジェクト.....	6-13
OCIのスレッド・セーフティについての移行および互換性の問題.....	6-13
スタンバイ・データベースのアップグレードおよび互換性の問題.....	6-14
エクスポート/インポートの互換性の問題.....	6-14
下位互換性の技術および制約.....	6-15
NCHAR および NLS の使用.....	6-15
移行および NCHAR と NLS.....	6-15
NCHAR と NLS の互換性および相互運用性.....	6-16
<b>7 バージョン8のROWIDに関する移行の問題</b>	
アプリケーションおよびデータの移行.....	7-2
DBMS_ROWID パッケージ.....	7-3
ROWID 変換型.....	7-3
ROWID 変換関数.....	7-3
変換プロシージャの例.....	7-5
例1.....	7-5
例2.....	7-5
例3.....	7-5
例4.....	7-5
例5.....	7-5
スナップショット・リフレッシュ.....	7-6
バージョン8より前のクライアントとの互換性の問題.....	7-6
ROWID 関連の移行に関する質問・回答.....	7-6
<b>8 アップグレードおよびダウングレード</b>	
バージョン8の新リリースへのアップグレード.....	8-2
製品の構成およびアップグレード.....	8-4
アドバンスト・キューイング・オプションのアップグレード.....	8-5
AQ\$_AGENT データ型で使用可能な新しいフィールド.....	8-6

拡張アドレス・フィールド .....	8-6
新しいディクショナリ表 .....	8-6
ダウングレード .....	8-6
リリース 8.0.4 からリリース 8.0.3 へのダウングレード .....	8-7
バージョン 8 からリリース 7.x へのダウングレード .....	8-9

#### A 移行ユーティリティのメッセージ

#### B 制御ファイル固定ビューの変更

制御ファイル・ビュー内の日付列 .....	B-1
バージョン 8 で保持されている廃止されたビュー .....	B-2
V\$LOG_HISTORY の保持とアップグレード .....	B-2
V\$ARCHIVED_LOG による V\$LOG_HISTORY の置換え .....	B-2
V\$BACKUP_CORRUPTION .....	B-3
V\$BACKUP_DATAFILE .....	B-4
V\$BACKUP_DEVICE .....	B-5
V\$BACKUP_PIECE .....	B-5
V\$BACKUP_REDOLOG .....	B-6
V\$BACKUP_SET .....	B-6
V\$CONTROLFILE_RECORD_SECTION .....	B-7
V\$COPY_CORRUPTION .....	B-7
V\$DATABASE の新規列 .....	B-8
V\$DATAFILE の新規列 .....	B-8
V\$DATAFILE_COPY .....	B-9
V\$DATAFILE_HEADER .....	B-10
V\$DELETED_OBJECT .....	B-11
V\$INSTANCE .....	B-11
V\$OFFLINE_RANGE .....	B-13
V\$RESOURCE_LIMIT .....	B-13
V\$TABLESPACE .....	B-14
V\$THREAD .....	B-14
変更された列型 .....	B-14
データベース・スケジュール機能 .....	B-15

変更された固定ビュー .....	B-15
新しい固定ビュー .....	B-15
表 (ビュー) 名の変更 .....	B-16
<b>C バージョン 8 の INIT.ORA の変更点</b>	
COMPATIBLE パラメータ .....	C-2
リリース 8.0.4 への移行またはアップグレード .....	C-2
データ・ディクショナリの保護 .....	C-3
DML_LOCKS .....	C-3
NCHAR パラメータと NLS パラメータ、および互換性 .....	C-4
バージョン 8 で改名されたバージョン 8 より前のパラメータ .....	C-4
バージョン 8 で廃止されたリリース 7.3 のパラメータ .....	C-5
レプリケーション・サーバー用の REPLICATION_DEPENDENCY_TRACKING .....	C-6
バージョン 8 ではサポートされなくなっている機能 .....	C-6
SERIALIZABLE=TRUE または _SERIALIZABLE .....	C-6
<b>D 新しい SQL キーと予約語</b>	
<b>E 移行に関する一般的なシステム要件</b>	
メモリー要件 .....	E-2
基本メモリー要件 .....	E-2
バージョン 8 の実行可能ファイル .....	E-2
同時アクセス .....	E-2
Oracle Parallel Server の使用 .....	E-4
バージョン 8 の新しいサイズおよび制限 .....	E-4
CHAR および NCHAR の最大サイズのサポート .....	E-4

## 索引

---

# はじめに

このマニュアルは、Oracle データベース・システムの移行、アップグレード、ダウングレードの計画と実行の手引きを示しています。基本的な原則および Oracle 製品の機能について説明し、移行およびアップグレード/ダウングレード操作で使用する詳細な作業手順を示します。

この章では、次のトピックについて一般的な説明を行います。

- 対象読者と前提条件
- このマニュアルの構成
- このマニュアルで使用する表記規則

『Oracle8 Server 移行ガイド』では、Oracle8 および Oracle8 Enterprise Edition 製品のさまざまな機能について説明します。Oracle8 および Oracle8 Enterprise Edition の基本機能は同じです。ただし、いくつかの拡張機能は Oracle8 Enterprise Edition でのみ使用でき、その一部はオプションです。たとえば、オブジェクトを使うには、Oracle8 Enterprise Edition とオブジェクト・オプションが必要です。

関連項目：Oracle8 と Oracle8 Enterprise Edition の違い、役に立つ機能やオプションなどの詳細は、『Oracle8 と Oracle8 Enterprise Edition の解説』を参照してください。

## 対象読者と前提条件

このマニュアルは、Oracle ソフトウェアの移行またはアップグレード、ダウングレード操作の計画と実行を担当する、データベース管理者 ( DBA )、アプリケーション・プログラマ、セキュリティ管理者、システム・オペレータを対象に作成されています。ユーザーが Oracle Server のバージョン 7 ( Oracle7 ) とそのオペレーティング・システム環境に慣れていることを前提としています。また、ユーザーが Oracle データベース管理システム ( DBMS ) の概要を理解していることが前提です。『Oracle8 Server 概要』の最初の章では、この移行マニュアルで使用する概念と用語について広範囲に紹介しています。

## このマニュアルの構成

このマニュアルは、次の章と付録で構成されています。

### 第 1 章: 移行の概要

移行の手順、およびデータベース管理者とアプリケーション・プログラマの役割について概説します。

### 第 2 章: 移行の準備

データベースを移行する前に行う手順について説明します。

### 第 3 章: 移行ユーティリティによる移行

移行ユーティリティを使用して、バージョン 7 のデータベースをバージョン 8 へ移行する方法について説明します。

### 第 4 章: エクスポート/インポートによる移行

エクスポート/インポート・ユーティリティを使用して、バージョン 7 またはバージョン 6 のデータベースをバージョン 8 へ移行する方法について説明します。

### 第 5 章: データベースの移行後

データベースをバージョン 8 へ移行した後で行う手順について説明します。

### 第 6 章: バージョン 7 のアプリケーションのアップグレード

バージョン7のアプリケーションおよびツールをバージョン8で使えるようにアップグレードする方法について説明します。

## 第7章: バージョン8のROWIDに関する移行の問題

ROWIDを含む列のバージョン8への移行に伴う、新しいバージョン8のROWIDに関連する問題について説明します。

## 第8章: アップグレードおよびダウングレード

データベースをリリース8.0.3からリリース8.0.4へアップグレードするときの手順について説明します。また、バージョン8のダウングレード、つまりリリース8.0.4のデータベースからリリース8.0.3またはバージョン7、リリース7.3へのダウングレードについても説明します。

## 付録A: 移行ユーティリティのメッセージ

移行ユーティリティによって表示されるメッセージをリストで示し、各メッセージおよび考えられる原因、エラー訂正のための処置について説明します。

## 付録B: 制御ファイル固定ビューの変更

制御ファイル固定ビューの、バージョン7リリース7.3からバージョン8のサーバーへの変更点について簡単に説明します。

## 付録C: バージョン8のINIT.ORAの変更点

移行の際に重要なOracle INIT.ORAファイル初期化パラメータについて簡単に説明します。特に、リリース7.3以降に追加または変更、削除された初期化パラメータについて説明します。

## 付録D: 新しいSQLキーと予約語

バージョン8で新規に追加されたキーワードと予約語をリストで示します。

## 付録E: 移行に関する一般的なシステム要件

バージョン8への移行を成功させるために重要なシステム要件について説明します。

## このマニュアルで使用する表記規則

このマニュアルでは、以下に示す表記上の規則に従います。

UPPERCASE  
WORDS (大文字) アルファベットの大文字は、コマンドのキーワード、オブジェクト名、パラメータ、ファイル名などを明示するために使用されています。次にその例を示します。

「プライベート・ロールバック・セグメントを作成する場合は、PARAMETERファイルのROLLBACK\_SEGMENTSパラメータにそのロールバック・セグメントの名前を指定する必要があります。」

コーディング例 SQL および Server Manager 行モード、SQL\*Plus のコマンドと文は、次に示す例のように通常のテキストとは別に固定幅で表示されます。

```
INSERT INTO emp (empno, ename) VALUES (1000, 'SMITH');  
ALTER TABLESPACE users ADD DATAFILE 'users2.ora' SIZE 50K;
```

サンプル文には、コンマまたは引用符などの句読点が含まれている場合があります。サンプル文にある句読点は、すべて必須です。セミコロンまたはその他の終了文字が文を終了するために必須であるかどうかは、アプリケーションによって異なります。

サンプル文の大文字は、Oracle SQL 内のキーワードを示しています。ただし、文を発行するとき、キーワードの大文字と小文字は区別されません。

サンプル文の小文字は、その単語がたんなる例として使用されていることを示しています。たとえば、小文字の単語は、場合によって表、列、ファイルなどの名前を示します。

---

## 移行の概要

この章では、バージョン 8 以前のデータベースをバージョン 8 へ移行するために必要な手順の概要を示します。

これらの移行の手順によって、既存のバージョン 8 以前のデータベース・システム（対応付けられたアプリケーションを含む）をバージョン 8 のデータベース・システムへ変換します。バージョン 8 は、Oracle の以前のバージョンおよびリリースのすべてと互換性があります。したがって、このマニュアルで説明する移行手順によって変換されたデータベースは、以前のバージョンの場合と同様に機能し、バージョン 8 の新機能を任意に活用できます。

現行の本番データベースを移行する前に、いくつかの準備処理を実行する必要があります。データベースを移行した後、移行をテストするために、さらにテスト・ステップをいくつか実行します。他の複数の手順により、既存のバージョン 8 以前のアプリケーションにバージョン 8 の新機能を追加できます。

この章では、次のトピックについて説明します。

- 用語
- 移行ステップの概要
- 移行におけるデータベース管理者の役割
- 移行におけるアプリケーション開発者の役割

## 用語

次の用語は、このマニュアル全体で使われます。

移行は、Oracle データベースのインストール済みのバージョンを新しいバージョンに変換する処理です。たとえば、バージョン7のデータベースをバージョン8に変換することが、データベース・システムの移行です。

アップグレードは、Oracle データベースのインストール済みのバージョンを、インストールされているリリースから同じバージョンの新しいリリースに変換する処理です。たとえば、リリース 8.0.3 からリリース 8.0.4 への変換がアップグレードです。

ダウングレードは、Oracle データベースのインストール済みのバージョンを、新しいリリースから以前のリリースに逆変換する処理です。たとえば、Oracle データベースをリリース 8.0.4 からリリース 8.0.3 に戻すことがダウングレードであり、バージョン 8 からバージョン 7 に戻すこともダウングレードです。

関連項目：アップグレードまたはダウングレードの詳細は、第 8 章「アップグレードおよびダウングレード」も参照してください。

ソース・データベースはバージョン 8 へ移行されるデータベースであり、通常は古いバージョンの Oracle ソフトウェアを使います。ターゲット・データベースはソース・データベースを移行するデータベースであり、通常は新しいバージョン 8 のソフトウェアを使います。

## 移行ステップの概要

データベースの移行を行う前に、移行処理の主なステップについて理解しておく必要があります。これらのステップはすべてのオペレーティング・システムに適用されますが、例外として一部のプラットフォームに固有の詳細項目があり、それについてはインストレーション・ガイドに示されています。

慎重に計画を立ててバージョン 8 のツールを使うことで、データベースをバージョン 8 へ移行する処理が簡単になります。移行ユーティリティはデータベース全体を移行するときに最も簡単な方法ですが、エクスポート / インポート・ユーティリティと SQL コピー・ユーティリティでは、データベース各部の部分的な移行が可能になります。

移行処理には、主に次のステップがあります。

- ステップ 1: 移行のための準備
- ステップ 2: 移行処理のテスト
- ステップ 3: 移行したテスト・データベースのテスト
- ステップ 4: ソース・データベースの準備と保存
- ステップ 5: 本番データベースの移行
- ステップ 6: 新しいバージョン 8 の本番データベースのチューニングと調整

これらのステップの簡単な概要を、以下に示します。これらの説明は、移行プロセスの主なステップを理解することを目的としています。詳細は、このマニュアルの該当する章を参照してください。

### ステップ 1: 移行のための準備

- バージョン 8 のデータベースの機能を理解します。これらの機能の概要は、『Oracle8 と Oracle8 Enterprise Edition の解説』を参照してください。
- 移行に必要なシステム・リソースを見積もり、確保します。
- 現行の本番データベースおよび移行の目的、使用可能な移行方法の動作と機能に関する考慮事項に基づいて、使用する移行方法を決定します。
- バージョン 8 のテスト・データベースでの移行をテストする計画と、移行したバージョン 8 の本番データベースをテストする計画を作成します。
- 予期しない問題や遅れが生じた場合にすぐに回復できるように、バックアップ計画を準備します。

### ステップ 2: 移行処理のテスト

- バージョン 7 のテスト・データベースを使って、移行をテストします。テスト移行は、移行をテストするために作成した環境で行う必要があり、実際のバージョン 7 の本番データベースに干渉してはなりません。

### ステップ 3: 移行したテスト・データベースのテスト

- 移行前のバージョン 7 のテスト・データベースと、バージョン 8 へ移行されたバージョン 7 のテスト・データベースに対して、ステップ 1 で計画したテストを実行します。
- 結果を比較して、移行前のバージョン 7 のテスト・データベースに対するテスト結果と、バージョン 8 へ移行されたバージョン 7 のテスト・データベースに対するテスト結果の間の相違点を記録します。
- 見つけた相違点を訂正する方法を調べて、その訂正をインプリメントします。
- 移行が問題なく完了し必要なアプリケーションで正常に動作するまで、必要に応じてステップ 1 およびステップ 2、ステップ 3 の最初の部分を繰り返します。

第 2 章「移行の準備」では、ステップ 1 からステップ 3 までを詳しく説明します。

### ステップ 4: ソース・データベースの準備と保存

- バージョン 8 への移行が正しく行われるように、現行の本番データベースを準備します。
- バージョン 8 以前の本番データベースのバックアップ処理とバージョン 8 への移行処理に必要なダウン時間をスケジューリングします。
- 現行の本番データベースの全体バックアップを作成します。このステップは、移行方法として移行ユーティリティを使用する場合にだけ必要です。

## ステップ 5: 本番データベースの移行

- バージョン 8 以前の本番データベースをバージョン 8 へ移行します。
- 移行の後、本番データベースの全体バックアップを作成します。

第 3 章では、ステップ 4 およびステップ 5 での移行ユーティリティの使用方法について説明します。第 4 章では、ステップ 4 およびステップ 5 でのエクスポート/インポート・ユーティリティの使用方法について説明します。第 5 章では、移行後のバックアップ手順について説明します。

関連項目：アドバンスド・レプリケーションをインストール済みのバージョン 8 以前のデータベース・システムを移行する場合は、『Oracle8 Server レプリケーション』の付録 B 「移行および互換性」を参照してください。

## ステップ 6: 新しいバージョン 8 の本番データベースのチューニングと調整

- 新しいバージョン 8 の本番データベースをチューニングします。バージョン 8 の本番データベースのパフォーマンスは、移行前の Oracle データベースと同等以上になる必要があります。第 5 章では、これらのチューニング調整について説明します。
- バージョン 8 のデータベースのどの新機能をデータに適用すれば適切かを決定し、それに応じてアプリケーションを更新します。
- 必要に応じ、新しいデータベース管理の手順を作成します。
- すべてのアプリケーションがテストされ、適切に動作するまでは、本番ユーザーをバージョン 8 のデータベースに移行しないでください。第 6 章では、アプリケーションを更新する際の考慮事項について説明します。

## 移行におけるデータベース管理者の役割

通常、移行処理を確実に成功させる責任は、データベース管理者 ( DBA ) が負います。DBA は、通常、移行処理の各ステップに関係しますが、移行したデータベースでのアプリケーションのテストに関連するステップにはかかりません。

データベース管理者に固有の任務は、次のとおりです。

- 移行処理に関連するすべてのスタッフと会合を持ち、移行処理における各スタッフの役割を明確に決定する
- テスト移行の実行
- テストおよび本番移行処理のスケジュールリング
- 移行前のバージョン 7 の本番データベースのバックアップを行う
- 本番データベースの移行を完了する
- 新規に移行したバージョン 8 の本番データベースのバックアップの作成

すべてのアプリケーションがテストされ、適切に動作するまでは、移行されたバージョン 8 のデータベースに対する運用を開始しないでください。

## 移行におけるアプリケーション開発者の役割

アプリケーション開発者は、移行前のバージョン 7 のデータベース用に設計されたアプリケーションが、移行されたバージョン 8 のデータベースでも正しく動作することを保証する責任があります。アプリケーション開発者は、移行されたバージョン 8 のデータベースに対するアプリケーションのテストを頻繁に行い、使用すべきバージョン 8 の新機能を決定します。

バージョン 7 の本番データベースを移行する前に、DBA またはアプリケーション開発者は、バージョン 8 のテスト・データベースをインストールする必要があります。次に、アプリケーションが元の機能（または拡張されたバージョン 8 の機能）で動作するまで、アプリケーション開発者は必要に応じてそのアプリケーションをテストしたり変更したりします。

次の参照先には、特定のアプリケーションに影響するような、移行されたバージョン 8 のデータベースでの相違点を判断するための情報があります。アプリケーション開発者は、これらの相違点を利用して、既存のアプリケーションへの変更を行うことができます。

- 第 6 章「バージョン 7 のアプリケーションのアップグレード」では、（バージョン 7 のデータベースにアクセスする）既存のアプリケーションがバージョン 8 のデータベースにアクセスできるようにするのに必要な変更点について説明します。また、バージョン 8 の機能を利用するためにバージョン 7 のアプリケーションをアップグレードする方法についても説明しています。
- 『Oracle8 と Oracle8 Enterprise Edition の解説』では、バージョン 8 の拡張機能について説明しています。
- 付録 B では、アプリケーションが必要とするデータ・ディクショナリ・ビューの変更と、バージョン 8 の SQL 予約語およびキーワードをリストしています。
- 『Oracle8 Parallel Server 概要および管理』と『Oracle8 Server SQL リファレンス』では、バージョン 8 での変更点と新機能について説明しています。
- バージョン 8 以前のデータベース・システムにアドバンスド・レプリケーションをインストール済みの場合は、『Oracle8 Server レプリケーション』の付録 B「移行および互換性」も参照してください。

バージョン 8 では、既存のアプリケーションをバージョン 8 にアップグレードするのを支援するいくつかの機能を提供しています。

- たとえば、Net8 および SQL\*Net バージョン 2 では、Oracle の各バージョン間の通信をサポートしています。
- プログラミング・インタフェースは、Oracle の各バージョン間でも未変更のままです。
- Oracle の下位互換性では、異なるバージョンとリリース間で非互換な部分はわずかしかなかったりありません。



# 2

---

## 移行の準備

この章では、本番データベースを移行する前に完了しなければならない作業について説明します。また、この章では、移行処理（第1章「移行の概要」ですでに概説）のステップ1からステップ3までを詳しく説明します。

- ステップ1: 移行のための準備
- ステップ2: 移行処理のテスト
- ステップ3: 移行したテスト・データベースのテスト

この章の情報は一般的なものであり、バージョン7およびバージョン6の本番データベース全般に適用されます。

関連項目：アドバンスド・レプリケーションをインストールしているバージョン8以前のデータベース・システムを移行する場合は、『Oracle8 Server レプリケーション』の付録B「移行および互換性」を参照してください。

## ステップ 1: 移行のための準備

このステップには次のような処理が含まれており、次のセクションで詳しく説明します。

- 新しいバージョン 8 のデータベースの機能を理解する
- 移行方法の選択
- システム要件と使用可能なリソースの見積り
- Oracle バージョン 7、リリース 7.X 以上から開始
- 共通移行問題の回避
- バックアップ計画の準備
- テスト計画の作成

### 新しいバージョン 8 のデータベースの機能を理解する

移行処理を計画する前に、バージョン 8 のデータベースの新機能を理解する必要があります。バージョン 8 の RDBMS とリリース 7.3 の RDBMS の違いを理解するには、まず『Oracle8 と Oracle8 Enterprise Edition の解説』を読んでください。

関連項目：Parallel Server オプションを使用する場合は、Parallel Server での変更点について『Oracle8 Parallel Server 概要および管理』を参照してください。

---

---

注意：バージョン 8 で使用可能な機能を活用する方法を学ぶには、バージョン 8 のトレーニング・クラスを受講するのが最適です。

---

---

### 移行方法の選択

データベースをバージョン 8 へ移行するには、次の 3 つの方法のどれかを使います。

- 移行ユーティリティ。バージョン 7 のデータベースをバージョン 8 へ移行します。使用しているプラットフォームで移行ユーティリティによって移行できる最初のリリースについては、プラットフォームに固有の Oracle マニュアルを参照してください。たとえば、一部のプラットフォームでは、移行ユーティリティによって移行できるのはリリース 7.1.4 以降のデータベースだけです。
- バージョン 7（またはバージョン 6）のソース・データベースのエクスポート（全体または部分的）。その後で、バージョン 8 のターゲット・データベースへ全体または部分的なインポートを行います。
- COPY コマンドまたは CREATE TABLE コマンドの AS 句を使用しての、ソース・データベースからバージョン 8 のデータベースへのデータのコピー。

図 2-1 では、これらの移行方法についてまとめ、それぞれの長所と短所を示しています。

表 2-1 移行方法の長所と短所

移行方法	長所	短所
<p>移行ユーティリティ： バージョン 7 からバージョン 8 への完全なデータベースの移行のため データ・ファイル・ヘッダーは変更するが、実際のデータは未変更のまま。 データはコピーしない。</p>	<p>自動的に実行され、DBA の介入が最小限で済む。 変更されるオブジェクトはデータ・ディクショナリ・オブジェクトだけであるため、データベースのサイズに関係なく比較的高速である。 本来、移行できるデータベースのサイズには制限がない。 他の移行方法と比べて、通常は比較的少量のディスク領域しか必要としない。</p>	<p>バージョン 7 からバージョン 8 への移行し か実行できず、バージョン 7 へのダウングレードはできない。 リリース間のアップグレード（たとえば、リリース 8.0.3 からリリース 8.0.4 へ）は実行できない。ただし、Oracle Installer を使えば簡単にアップグレードできる。 データベースの選択した部分だけを移行することはできない。データベース全体を移行する。</p>
<p>エクスポート/インポート： データベース各部の移行のため データ・ファイル・ヘッダーと実際のデータは変更しない。 データの新しいコピーを作成する。</p>	<p>バージョン 6 およびバージョン 7 のデータベースをバージョン 8 へ移行できる。 データベースの特定の部分を移行できる。 Oracle のバージョン間のダウングレード、たとえばバージョン 8 からバージョン 7 へのダウングレードを実行できる。 リリース間でのアップグレード操作またはダウングレード操作、たとえば 8.0.3 から 8.0.4 へのアップグレードを実行できる。 データ・ファイルの断片化を解消し、データを圧縮して移行できるので、パフォーマンスを向上できる。 変更した表領域または新しい表領域を使用するか、表をパーティション化することでデータベースを再構築できる。</p>	<p>小規模なデータベースを除いて非常に遅い。データ型 LONG の使用とデータの量に応じて所要時間が伸びる。数ギガバイトもある非常に大規模なデータベースでは数時間かかることがあり、テラバイト規模のデータベースでは数日かかることもある。 エクスポート・ファイルヘッダーをコピーするために大量のディスク領域が必要である。</p>
<p>データのコピー： データベース各部の移行のため データ・ファイル・ヘッダーと実際のデータは変更しない。 データの新しいコピーを作成する。</p>	<p>データ・ファイルの断片化を解消し、データを圧縮して移行できるので、パフォーマンスを向上できる。 変更した表領域または新しい表領域を使用してデータベースを再構築できる。 バージョン 6 またはバージョン 7 のデータベースをバージョン 8 へ移行できる。 データベースの特定の部分を移行できる。 リリース間でのアップグレード操作またはダウングレード操作、たとえば 8.0.3 から 8.0.4 へのアップグレードを実行できる。</p>	<p>小規模なデータベースを除いて非常に遅い。データ型 LONG の使用とデータの量に応じて所要時間が伸びる。数ギガバイトもある非常に大規模なデータベースでは数時間かかることがあり、テラバイト規模のデータベースでは数日かかることもある。 コピー操作中にソース・データベースとターゲット・データベースの両方が同時に使用可能である必要がある。各移行方法の長所と短所</p>

次のセクションではそれぞれの移行方法について詳しく説明し、それぞれに必要な時間と領域、およびどのような状況にどの方法が適しているかを説明します。

### 移行ユーティリティ

移行ユーティリティは、データ・ファイルのファイル・ヘッダーだけを変更し、また必要に応じてファイル内のデータの定義を変更して、バージョン7のソース・データベースのファイルと構造をバージョン8の形式に変換します。移行ユーティリティがデータ・ファイル内のデータ部分を変更することではなく、その形式や内容を変更することはありません。

移行ユーティリティの第一の利点は、そのスピードと使い易さです。移行ユーティリティは、エクスポート/インポートに比べてかなり短時間で実行することができます。また、このユーティリティを使用する移行処理は、標準化された一連の簡単なステップしか必要としません。さらに、移行ユーティリティを使ってデータベースを移行するのにかかる時間は、データベースのサイズよりはデータ・ディクショナリ内のオブジェクト数によって決まります。

移行ユーティリティは、ソース・データベース全体を迅速に移行するときに特に便利です。エクスポート/インポートとは異なり、移行ユーティリティでは特定のデータ・ファイルのみを選択して移行することはできません。しかし、大量のデータおよび大きなデータ型、その他いくつかバージョン7の機能を使用するデータベースの場合は、移行ユーティリティがバージョン8へ移行する際の唯一の実用的なツールです。

移行ユーティリティが必要とするのは、バージョン7(ソース)とバージョン8(ターゲット)のデータ・ディクショナリを両方とも同時に保持できる、SYSTEM表領域内の十分な一時領域だけです。

移行ユーティリティは、データベース・ファイルおよびロールバック・セグメント、制御ファイルを含むデータベース全体を変換します。実際にバージョン7のデータベースを移行する前ならばいつでも、バージョン7のインスタンスでデータをオープンし、アクセスできます。ただし、移行ユーティリティによってバージョン7のソース・データベースをバージョン8に移行した後でバージョン7に戻る方法は、バージョン7のソース・データベースの全体バックアップを復元する以外にありません。

関連項目：移行ユーティリティの詳細な使用法は、第3章「移行ユーティリティによる移行」を参照してください。

### エクスポート/インポート

移行ユーティリティとは異なり、エクスポート/インポート・ユーティリティはソース・データベース内のデータを新しいデータベースに物理的にコピーします。ソース・データベースのエクスポート・ユーティリティは、ソース・データベースの特定部分をエクスポート・ファイルにコピーします。次に、バージョン8のインポート・ユーティリティが、このエクスポートされたデータを新しいバージョン8のデータベースにロードします。ただし、エクスポート・ファイルを移行する前に、新しいバージョン8のターゲット・データベースを準備しておく必要があります。

次のセクションでは、データベースの移行にエクスポート/インポートを使うかどうか決める際に役に立つ、エクスポート/インポート操作のいくつかの局面について説明します。

関連項目：エクスポート/インポートによる移行方法の詳細は、第4章「エクスポート/インポートによる移行」と、『Oracle8 Serverユーティリティ』を参照してください。

### エクスポート/インポートによって移行されたデータベースへの影響

エクスポート/インポートによる移行方法では、ソース・データベースを変更しません。したがって、ソース・データベースは、移行処理を通じて使用可能な状態にあります。ただし、一貫性のあるデータベース・スナップショットが必要な場合（データの整合性保持またはその他の目的のため）、ソース・データベースは、制限モードで実行するか、またはエクスポート実行時に変更禁止にする必要があります。ソース・データベースを使用可能な状態にできるので、たとえば、既存のバージョン7の本番データベースを実行しながら、エクスポート/インポートによるバージョン8の新規データベースの作成を同時に行うことができます。この移行処理中にデータベースの完全な一貫性を維持するには、バージョン7のデータベースのデータを変更する場合に、バージョン8のデータベースのデータにも同じ変更を加える必要があります。

エクスポート/インポートは、アップグレードやダウングレードにも使用できます。たとえば、バージョン8のデータベースからバージョン7、リリース7.3のデータベースへの逆方向の変換を、エクスポート/インポートによる移行方法によって実現できます。

最も重要なことは、エクスポート/インポート操作の結果としてまったく新しいデータベースが作成されることです。ソース・データベースには最終的に指定したデータのコピーが含まれますが、移行したデータベースは、元のソース・データベースとはまったく異なった方法で運用される場合があります。DBAによるデータベースの再構成またはデータの断片化解消、バージョン8へのアップグレードの結果として、パフォーマンスおよびデータの成長パターン、共有リソースの使用法、データ・ディクショナリのサイズ、オブジェクトの編成などが変わります。

エクスポート/インポートによってデータベースにもたらされる効果を最大限に利用するには、綿密な計画および熟練者による実装、厳密なテストが必要です。これを行わなかった場合には、データベースに変更を加えたときに問題が発生することがあります。移行によってデータベースが再構成された後で、移行したデータベースが異なった動作をする場合は、その違いの原因を判断するのが困難になることがあります。

### エクスポート/インポートの利点

エクスポート/インポートによるデータの移行には、次のような利点があります。

- データの断片化を解消できる。インポートすることによってデータが圧縮されるため、パフォーマンスを向上できる。
- データベースを再構成できる。つまり、新しい表領域を作成したり、既存の表や表領域、またはインポートによってデータがロードされる先のパーティションを変更できる。
- 特定のデータベース・オブジェクトまたはユーザーを移行できる。つまり、オブジェクトおよびユーザー、その他希望する項目だけをインポートできる。
- バックアップ・アーカイブとしての役割を果たす。つまり、全データベース・エクスポートをソース・データベースのアーカイブとして使用できる。

### エクスポート/インポートの制約

エクスポート/インポートによるデータの移行には、次のようないくつかの制約があります。

## ステップ 1: 移行のための準備

---

- エクスポート/インポートによるデータベースの移行では、熟練したデータベース管理者が必要になる。最終的に移行を試みる前に、必要とされる計画と複雑な実行作業には、通常、さまざまな段階と多くの熟練が必要になる。
- 大規模データベースの場合、データベース全体のエクスポート/インポートを行うと、エクスポート・ダンプ・ファイル用に大量の一時記憶領域が必要になる場合がある。
- オペレーティング・システムがデータベースと同等のファイル・サイズをサポートしていない場合は、エクスポートを複数のジョブに分割する必要が生じることがある。
- エクスポート/インポートでは、まったく新しいデータベースが作成される。ソース・データベースの状態を保持し、そのエクスポート・ダンプ・ファイル/データをバージョン 8 のターゲット・データベースにインポートするには、インポートを実行する前にシステムにターゲット・データ・ファイルを作成する必要がある。
- バージョン 8 への移行と断片化解消/データベースの再構築を同時に行うなど、データベースに対して同時に複数の変更を行うと、トラブルシューティングを妨げることがある。
- 移行作業中にソース・データベースおよびターゲット・データベース内のデータの同期を保つには、ソース内のデータ変更を禁止するか、ターゲット・データベースへ移行されたデータへも同じ変更を加えてミラー化する。

### エクスポート/インポートの時間要件

エクスポート/インポートによる全データベースの移行方法は、移行ユーティリティを使用した場合と比較して、長い時間がかかる場合があります。したがって、ピーク時を外して移行するようにスケジュールしたり、移行中にソース・データベースに対して行われた変更を新しいターゲット・データベースに反映させる手配をしたりする必要があります。

エクスポート/インポートによる移行方法に必要な時間とシステム・リソース（特にディスク領域）は、DBAの技術およびデータベースのサイズ、移行するデータの型、特に再構築しなければならない索引の数、サイズ、種類によって異なります。

たとえば、6 ギガバイトの比較的単純なバージョン 7 のデータベースをバージョン 8 に移行するのに、移行ユーティリティでは約 1 時間かかりました。その同じバージョン 7 のデータベースをエクスポートする際には、2 ギガバイトの単一のエクスポート・ダンプ・ファイルが作成されます。その 1 つのエクスポート・ダンプ・ファイルをインポートするのに、20 時間かかります。4-3 ページの「エクスポート/インポートによるバージョン 8 より前のソース・データベースの移行」に示されている各ステップを使って移行を完了すると、2 日かかります。

エクスポート/インポートではデータベースの移行に時間がかかるので、次の各要素について考慮する必要があります。

- 移行に必要な時間は、毎日のスケジュールにおいて利用できる非ピーク時または実動時間外の時間よりはるかに長い。移行処理の期間中に、業務にデータベースを使用できないようにすることは非現実的である。

- ソース・データベースを読取り専用にしたり、エクスポート実行後にソース・データベースへの変更を禁止した場合、インポートおよび移行ステップが完了するまでアプリケーションは利用できない。
- 規模の大きなデータベースについては、時間を削減し、移行の処理を最適化するために、パラレル・エクスポート・ストリームの運用を検討する。

#### バージョン 8 のインポートによるデータ定義の変換

以前のバージョンからデータをインポートする場合、バージョン 8 のインポート・ユーティリティでは、以前のバージョンのエクスポート・ダンプ・ファイルからデータを読み込む際にデータ定義に適切な変更を加えます。つまり、Oracle バージョン 6 およびバージョン 7、バージョン 8 のエクスポート・ユーティリティによって作成されたダンプ・ファイルが処理されます。エクスポートするソース・データベースがバージョン 6 よりも前の場合、エクスポートを実行する前に、ソース・データベースを最低でもバージョン 6 までアップグレードする必要があります。

#### データのコピー

データベース・リンクを使用して、ある Oracle データベースから別の Oracle データベースにデータをコピーできます。たとえば、SQL\*Plus の COPY コマンドを使ってソース・データベースの表からターゲット・データベースの表にデータをコピーしたり、INSERT INTO コマンド、CREATE TABLE ... FROM コマンド、CREATE TABLE ... AS コマンドを使ってターゲット・データベースの中に新しい表を作成し、その表をソース・データベースのデータで埋めたりすることができます。

データのコピーを使用する移行方法には、エクスポート/インポートと同様の利点があります。どちらの移行方法を利用しても、新しい表領域を作成するか既存の表または表領域を変更することによって、データ・ファイルの断片化を解消したりデータベースを再構築したりできます。また、指定したデータベース・オブジェクトまたはユーザーだけを移行することもできます。

ただし、エクスポート/インポートとは異なり、データのコピーでは、表の特定の行を選択してターゲット・データベースに挿入できます。このため、データベース表の一部分だけを移行する場合にはデータのコピーを使用します。逆に、エクスポート/インポート・ユーティリティを使用してバージョン 7 からバージョン 8 にデータを移行すると、表全体の移行だけが可能です。

たとえば、既存の表 ( EMP@V7DB ) のデータの一部分 ( 部門 10、20、30 の従業員のみ ) を含む新しい表 ( NEW\_EMP ) を作成するには、次のような SQL 文を使用します。

```
CREATE TABLE NEW_EMP
(EMPNO          NUMBER(4),
 ENAME          VARCHAR2(10),
 JOB            VARCHAR2(9),
 MGR            NUMBER(4),
 HIREDATE       DATE,
 SAL            NUMBER(7,2),
 COMM          NUMBER(7,2),
```

## ステップ 1: 移行のための準備

---

```
DEPTNO          NUMBER(2)) AS SELECT EMPNO, ENAME, JOB, MGR,  
                HIREDATE, SAL, COMM, DEPTNO  
FROM EMP@V7DB WHERE DEPTNO IN (10, 20, 30);
```

データのコピーによる移行方法は、エクスポート/インポートの場合よりも少ないディスク領域とメモリ・バッファ領域で済みます。データのコピーでは、ソース・データベースとターゲット・データベースの両方がオンラインであればよいからです。一時ファイルまたはエクスポート・ダンプ・ファイル用に、大量の余分な領域を割り当てる必要はありません。

大きいクラスタ表を扱っているときは、SQL COPY コマンドが役立ちます。さらに、SQL\*Plus の COPY コマンドを使用すると、Net8(または SQL\*Net) を使用してクラスタのさまざまな部分を並列に移動できます。データベース間でデータをコピーする方法の詳細は、『Oracle8 Server SQLリファレンス』の「CREATE TABLE コマンド」および『SQL\*Plus ユーザーズ・ガイドおよびリファレンス』の「COPY コマンド」を参照してください。

## システム要件と使用可能なリソースの見積り

移行を成功させるために必要となるシステム・リソースを見積もります。移行方法が異なると、リソース要件も異なることがあります。そのため、使うべき方法がよくわからない場合は、既存のデータベースをバージョン8に移行するための各方法ごとに見積もりを行います。

見積もりに際して、次の各要素について考慮する必要があります。

- オペレーティング・システムとハードウェアの両方の構成要件
- 既存の本番データベースのサイズ
- バージョン8をインプリメントするときに関連するデータベースに対して可能なサイズ調整

関連項目：システム要件の詳細は、付録 E 「移行に関する一般的なシステム要件」およびプラットフォームに固有のインストレーション・ガイドを参照してください。

---

注意：バージョン8のバイナリ・ファイルは、バージョン7の場合の3倍のディスク領域を必要とすることがあります。ディスク領域の要件を見積もるときは、移行ユーティリティを CHECK\_ONLY モードで実行してください。

---

移行方法を選択して要件を見積もった後、移行を成功させるのに必要なリソースを確保します。

## Oracle バージョン 7、リリース 7.X 以上から開始

移行ユーティリティを使用する場合、移行ユーティリティがサポートする最初のリリースはプラットフォームに固有です。たとえば、プラットフォームによっては移行ユーティリティを使って、バージョン 7、リリース 7.1.4 より低いリリース（バージョン 6、リリース 7.0、リリース 7.1.3 など）を移行することはできません。使用しているプラットフォームでサポートされるリリースについては、プラットフォームに固有の Oracle マニュアルを参照してください。

データベースのリリース番号が、使用しているプラットフォームで移行ユーティリティがサポートするリリースよりも低い場合は、データベースを必要なリリースまでアップグレードまたは移行してください。システムを必要なリリースまで移行またはアップグレードするときは、『Oracle7 Server 移行ガイド、リリース 7.3』を使います。その後で、この『Oracle8 Server 移行ガイド』を使ってバージョン 8 へ移行します。

---

---

注意：移行ユーティリティを使用せずにエクスポート / インポートまたはデータのコピーを使う場合は、この制限は適用されません。エクスポート / インポートまたはデータのコピーを使えば、バージョン 8 よりも前のデータベース（バージョン 6、リリース 7.0、リリース 7.1.3 など）からバージョン 8 へデータを直接移行できます。

---

---

## 共通移行問題の回避

データベースを移行する前に共通移行問題を排除することで、時間を節約できます。共通問題の領域には、次の各項目があります。

- 列またはアプリケーション・コードに格納された ROWID を使う場合は、第 7 章「バージョン 8 の ROWID に関する移行の問題」を参照してください。バージョン 8 では ROWID の形式が異なるので旧 ROWID は無効であり、変換する必要があります。
- データベース・オブジェクト（表や列など）の名前に、新しい予約語を使っていないことを確認してください。詳細は、付録 D「新しい SQL キーと予約語」を参照してください。
- すべての Oracle 製品のバージョンおよびオペレーティング・システムのバージョン、サード・パーティー製ソフトウェアのバージョンが、使用しているプラットフォームでのバージョン 8 に対して保証されていることを確認してください。詳細は、プラットフォームに固有のインストール・ガイドを参照してください。

## バックアップ計画の準備

移行が最終的に成功するかどうかは、適切なバックアップ計画の立案と実行次第で決まります。バックアップ計画を展開するには、次のような問題について考慮する必要があります。

- 業務上許容できない状況になるまでのどの程度の期間なら本番データベースが作動不能状態でもかまわないか。
- 可用性要件を満たすには、どのバックアップ計画を使用すべきか。

## ステップ 1: 移行のための準備

---

- サイトから離れた安全な場所にバックアップをアーカイブする必要があるか。
- どのくらいの時間でバックアップを復元できるか（オフサイト記憶域でのバックアップを含む）。
- 回復手順は正常にテストされているか。

バックアップ計画は、これらの問題のすべてに答え、データベースを正常にバックアップおよび回復するための手順を備えている必要があります。

関連項目：詳細は、『Oracle8 Server バックアップおよびリカバリ』を参照してください。

## テスト計画の作成

移行処理のすべての段階を検証するために、一連のテストを慎重に設計してください。緻密なテストを実行し、テストが正常に完了すれば、本番データベースの移行処理を十分に理解し、予測できるようになり、移行処理の成功が確実になります。本番データベースを移行する前に、できる限り多くのテストを実行してください。テスト計画は、非常に重要です。

---

---

警告：移行前に緻密なテストを実行しないことは危険であり、予測できない結果を招くことがあります。

---

---

テスト計画の中で、次のようなテストを実行する必要があります。

### 移行テスト

移行テストでは、ソース・データベースから移行されたデータベースへの移行パスを計画およびテストする必要があります。移行ユーティリティまたはエクスポート/インポートを使用するか、データのコピー方法を使用して、本番データベースのデータをターゲット・データベースへ移行します。これらの方法については、第 3 章「移行ユーティリティによる移行」および第 4 章「エクスポート/インポートによる移行」で説明しています。

どの移行方法を選択する場合でも、移行計画を確立し、テストして検証することが必要です。

### 最小テスト

最小テストでは、ソース・データベースのアプリケーションの全部または一部をターゲット・データベースに移動し、ターゲット・データベースの新機能を使用可能にしないでアプリケーションを実行します。最小テストは非常に限定的なテストであり、「現実の」本番環境で現れるような潜在的問題は明らかにされません。しかし、アプリケーションの起動または呼出しに関して問題があれば、ただちに明らかになります。

### 機能テスト

機能テストとは、システムの新機能と既存機能を移行後にテストする一連のテストです。機能テストには、RDBMS システムのすべてのコンポーネントおよびネットワーキング、アプリケーション・コンポーネントのテストが含まれます。機能テストの目的は、システムの各コンポーネントが移行前と同様に機能し、新機能が正しく動作していることを検証することにあります。

### 統合テスト

統合テストでは、システムの各コンポーネントの相互作用を試験します。統合テストを計画するときは、次の各要素について考慮する必要があります。

- ターゲット・データベースのインスタンスで稼働する Pro\*C/C++ アプリケーションは、新しいソフトウェアで問題がないことを確認するためにテストする必要があります。
- GUI インタフェースをその他のコンポーネントでテストする必要があります。
- バージョン 8 のインスタンスにアプリケーションが直接接続されるかどうかに関係なく、データ型やデータ・ディクショナリのデータの変更（データ・ディクショナリへの行の追加、オブジェクト型の変更など）といったターゲット・データベースのわずかな変更でも、フロントエンド・アプリケーションにまで影響することがある。
- 2 つのコンポーネント間の接続に Net8 または SQL\*Net が係わる場合は、ストレステストとともにその接続のテストも実行しなければならない。

### パフォーマンス・テスト

ターゲット・データベースのパフォーマンス・テストでは、ターゲット・データベースでのさまざまな SQL 文のパフォーマンスを、ソース・データベースでの SQL 文のパフォーマンスとそれぞれ比較します。移行する前に、ソース・データベースでのアプリケーションのパフォーマンス・プロファイルを理解する必要があります。特に、アプリケーションがデータベース・カーネルに対して実行するコールを理解していなければなりません。

関連項目：チューニングの詳細は、『Oracle8 Server チューニング』を参照してください。ソース・データベースでのアプリケーションのパフォーマンス・プロファイルを完全に理解するには、SQL\_TRACE を使用可能にし、TKPROF でプロファイリングしてください。

### ボリューム/ロード・ストレステスト

ボリューム/ロード・ストレステストでは、移行したデータベース全体を、大きいボリュームとロードでテストします（ボリュームとは、操作されるデータの量を表します。ロードとは、システム上の同時要求のレベルを表します）。ボリューム/ロード・テストの目的は、さまざまなボリュームとロードでの本番システムの動作をエミュレートすることです。

ボリューム/ロード・ストレステストは非常に重要ですが、一般に見過ごされがちです。ユーザーはどのようなボリューム/ロード・ストレステストも実行しないことが多いようです。その代わりに、ビジネス・アプリケーションを特に考慮しているわけではないベンチマークが広く利用されています。アプリケーションのベンチマークは、機能およびパフォーマンス、統合に関する問題点を解明するために実行してください。ボリューム/ロード・ストレステストの代わりになるものではありません。

ソース・データベースを移行した後、すべてのデータにアクセスできてアプリケーションが正しく機能することを確認するために、データをテストする必要があります。データベースの調整が必要かどうか判断する必要もあります。可能な場合は、これらのテスト手続きを自動化してください。

## ステップ 2: 移行処理のテスト

---

テスト計画は、サイトで実行される作業を反映するものでなければなりません。ソースの本番データベースのすべてのアプリケーションの機能やパフォーマンスをテストする必要があります。通常の使用状況とピーク時の使用状況の両方についてパフォーマンスの統計数値を収集してください。

### 特定の移行前後のテスト

テスト計画には、次のようなテスト項目を組み込みます。

- タイミングのテスト
- データ・ディクショナリ成長の監視
- ロールバックおよび一時セグメントの使用など、データベース・リソース使用率の監視

この情報を収集することにより、ソース・データベースと移行したターゲット・データベースを比較しやすくなります。

Oracle で各 SQL 文を実行するための実行計画を決定するには、ソース・データベースとターゲット・データベースの両方で EXPLAIN PLAN を使用します。この情報を複数の表に保存するには、INTO パラメータを使用します。

移行後に、移行されたデータベースの実行計画とソース・データベースの実行計画を比較できます。相違点がある場合は、移行されたデータベース上でコマンドを実行し、そのパフォーマンスを、ソース・データベース上で実行したコマンドのパフォーマンスと比較します。

## ステップ 2: 移行処理のテスト

現行の本番データベースに干渉しないテスト環境を作成します。テスト環境は、選択した移行方法によって異なります。

- 移行ユーティリティを使用する場合は、ソース・データベースのテスト・バージョン（通常はサブセット）を作成してテスト移行を行います。
- エクスポート/インポートを使用する場合は、実際のバージョン 7 の本番データベースからほんの一部をテストとしてエクスポート/インポートします。

テスト環境を使ってデータベースの移行を行います。最良の移行テストとは、ダウンサイズしたコピーまたはテスト・データに対してではなく、移行するデータベースの正確なコピーに対して実行することです。

---

---

**警告:** このデータベースのテスト・サブセットの移行と、次のステップで説明するアプリケーションでのテストが成功するまでは、実際の本番データベースを移行しないでください。

---

---

バージョン 8 のデータベースで使用する予定の OCI とプリコンパイラ・アプリケーションを、必ずアップグレードしてください。そうすることによって、本番データベースを移行する前にそれらのアプリケーションをサンプルの Oracle データベースでテストできます。詳細は、6-2 ページの「プリコンパイラおよび OCI アプリケーションのアップグレード」を参照してください。

関連項目：本番データベース用に定義されたオペレーティング・システム変数がテスト・データベースに影響されないようにそのテスト・データベースを構成する方法の詳細は、プラットフォーム固有の Oracle マニュアルを参照してください。

### ステップ 3: 移行したテスト・データベースのテスト

バージョン 7 のソース・データベースおよびバージョン 8 へ移行したテスト・データベースに対して、計画したテストを実行します。結果を比較し、相違点を記録します。必要に応じて、この章に記載のステップ 1 およびステップ 2、ステップ 3 を繰り返します。

移行されたバージョン 8 のデータベースで既存のアプリケーションが正常に動作するか確認するために、この新たに移行されたバージョン 8 のテスト・データベースをテストします。また、利用可能なバージョン 8 の機能を使用するための機能を追加して、機能強化についてもテストします。ただし、アプリケーションがソース・データベースでの動作状況と同様に動作するかを、最初に確認してください。

関連項目：バージョン 8 でアプリケーションを使用する際の詳細は、第 6 章「バージョン 7 のアプリケーションのアップグレード」を参照してください。

### ステップ 3: 移行したテスト・データベースのテスト

---

---

## 移行ユーティリティによる移行

この章では、移行ユーティリティを使用してバージョン7のデータベースをバージョン8へ移行する処理について説明します。

関連項目：移行処理の一部は、プラットフォームに固有のものです。使用しているプラットフォームでの移行に関するその他の指示は、プラットフォームに固有の Oracle マニュアルを参照してください。

この章で説明するトピックは、次のとおりです。

- バージョン8の移行ユーティリティを使った移行の概要
- システムの考慮事項と要件
- 移行のためのバージョン7のソース・データベースの準備
- バージョン8の移行ユーティリティのインストール
- バージョン8の移行ユーティリティ・コマンド行オプションの検討
- バージョン7のソース・データベースの移行
- 移行時のエラー
- 移行の中止

---

警告: Oracle8 Enterprise EditionからOracle8（以前はWorkgroup Server）へ移行する場合は、移行を実行する前に、Oracle8 Enterprise Editionの拡張機能を使うアプリケーションを変更し、それらの拡張機能を使わないようにしなければなりません。エディション間の違いに関する詳細は、『Oracle8とOracle8 Enterprise Editionの解説』を参照してください。

---

## バージョン 8 の移行ユーティリティを使った移行の概要

移行により、バージョン 7 のデータベースのデータ・ディクショナリや構造がバージョン 8 形式に変換されます。データベースを移行するには、最初に DBA がバージョン 7 のデータベース上にバージョン 8 の移行ユーティリティをインストールして実行します。そして、その DBA が新しいバージョン 8 のデータベースに対して一連の ALTER DATABASE コマンドを実行します。これらの処理の結果として、次のようなバージョン 7 の構造体が、バージョン 8 で使用できる構造体に変換されます。

- データ・ファイル (ファイル・ヘッダーだけ)
- データ・ディクショナリ
- 制御ファイル
- ロールバック・セグメント

### 移行処理の概要

以下のセクションでは、移行処理の概要について説明します。

#### バージョン 7 の環境

- DBA がバージョン 8 の移行ユーティリティを実行します。この移行ユーティリティは、バージョン 7 のデータベースのデータ・ディクショナリに基づいて新しいデータ・ディクショナリを作成およびデータ移入し、さらにバージョン 7 のデータベースの制御ファイルに基づいて 1 つのバイナリ・ファイルを作成します。このバイナリ・ファイルを変換ファイルと呼びます。

---

注意: バージョン 8 でデータベースをオープンしなくても、DBA はバージョン 8 の移行ユーティリティを複数回実行できます。それでも、バージョン 7 のデータベースへ戻れなくなることはありません。ただし、移行ユーティリティを実行するとバージョン 7 のデータベースのカatalog・ビューが自動的に削除されます ( 3-18 ページの「移行の中止」を参照 )。

---

#### バージョン 8 の環境

- DBA が ALTER DATABASE CONVERT 文を実行します。この文は、移行ユーティリティが生成した変換ファイルに基づいて新しい制御ファイルを作成し、すべてのオンライン・データファイル・ヘッダーをバージョン 8 形式に変換し、バージョン 8 のデータベースをマウントします。

オフライン・データファイルおよび読取り専用表領域のファイル・ヘッダーが、移行処理中に更新されることはありません。オフライン・データファイルのファイル・ヘッダーはファイルがオンラインになったときに変換され、読取り専用表領域のファイル・ヘッダーは、移行後に読取り/書込みになったときに変換されます。ただし、必ず読取り/書込みになるとは限りません。

- DBAがALTER DATABASE OPEN RESETLOGS文を実行します。このコマンドは、新しいディクショナリ内で定義されているすべてのオブジェクトとユーザーをバージョン8の仕様に自動的に変換し、すべてのロールバック・セグメントをバージョン8形式に変換します。

バージョン8のデータベースをオープンするときに、オフライン状態の表領域にソース・データベースのロールバック・セグメントがある場合、そのロールバック・セグメントはバージョン8のデータベース形式にすぐには変換されません。その代わりに、表領域がバージョン8で最初にオンラインになるときに变換されます。

## 移行ユーティリティの使用

このセクションでは、移行ユーティリティを使う際の重要な考慮事項について説明します。

### バージョン7より前のデータベース

バージョン6のデータベースは、バージョン8へ移行する前に、最低でもバージョン7へ移行する必要があります。使用しているプラットフォームの移行ユーティリティがサポートする最初のリリースについては、プラットフォームに固有のOracleマニュアルを参照してください。たとえば一部のプラットフォームでは、移行ユーティリティによって移行できるのはリリース7.1.4以降のデータベースだけです。

関連項目：使用しているプラットフォームの移行ユーティリティによって移行できるリリースへのデータベースの移行またはアップグレードの詳細は、『Oracle7 Server 移行ガイド、リリース7.3』を参照してください。

### ダウングレード

ダウングレードは、既存のOracleデータベースを以前のバージョンまたはリリースに変換する処理です。移行ユーティリティでは、バージョン8のデータベースをバージョン7へ戻すことはできません。そのために、エクスポート/インポートを使ったり、バックアップから復元したり、他のファンクションを使うなどの、他の機能を使ってダウングレードすることができます。

### アドバンスト・レプリケーションをインストール済みのデータベース

Oracleデータベース・システムにアドバンスト・レプリケーションをインストール済みの場合は、移行ユーティリティを実行する前に『Oracle8 Server レプリケーション』の付録B「移行および互換性」を参照してください。

## システムの考慮事項と要件

以下のセクションでは、移行ユーティリティを使う際のシステムの考慮事項と要件について説明します。

### 領域要件

バージョン8のバイナリ・ファイルは、バージョン7の場合の3倍のディスク領域を必要とすることがあります。これにより、移行処理中にディスク領域がなくなることがあります。

しかし、移行ユーティリティでは、一時領域は比較的少量で済みます。十分な領域を必要とするのは、既存のバージョン7のデータ・ディクショナリと同時に新しいバージョン8のデータ・ディクショナリを保持するための SYSTEM 表領域だけです。

Oracle データ・ディクショナリを保持するのに必要なディスク領域は、データベース内のオブジェクトの数によって異なります。通常、新しいバージョン8のデータ・ディクショナリは、バージョン7のソース・データ・ディクショナリよりも50パーセント大きくなります。必要ならば、SYSTEM 表領域にディスク領域を追加してください。

移行を開始する前に十分なディスク領域が SYSTEM 表領域に割り当てられていないと、移行ユーティリティは移行処理を完了しません。十分な領域がない場合、移行ユーティリティは、必要な追加ディスク領域の大きさを示すエラーを戻します。

移行を成功させるのに必要なディスク領域を調べるには、CHECK\_ONLY コマンド行オプションを TRUE に設定してバージョン8の移行ユーティリティを実行します。CHECK\_ONLY コマンド行オプションを指定すると、移行ユーティリティは、移行に必要なディスク領域を算定し、利用可能な領域の大きさをチェックし、ディスク領域の要件に関する情報メッセージを発行します。CHECK\_ONLY コマンド行オプションを TRUE に設定すると、移行ユーティリティは、バージョン8のデータ・ディクショナリを構築したりその他の移行処理を実行したりしません。

### ブロック・サイズの考慮事項

バージョン7のデータベースと移行したバージョン8のデータベース内の DB\_BLOCK\_SIZE の値 (INIT.ORA ファイル内のパラメータ) は同じでなければなりません。バージョン8での最小ブロック・サイズは2048バイト (2KB) です。この値より大きい場合は、プラットフォームの物理ブロック・サイズの整数倍にします。ただし、2KBの倍数、特に2乗 (つまり、2KB、4KB、8KB、16KB など) が最も堅牢な操作を実現します。

バージョン8のブロック・サイズ設定が次の基準を満たすようにしてください。

- バージョン7の設定と一致していること。
- 最低でも2048バイト (2KB) であること。バージョン7のブロック・サイズが2KBより小さい場合は、バージョン8の移行ユーティリティによりエラー・メッセージが表示されます。
- プラットフォームの物理ブロック・サイズの整数倍であること。できれば2KBの倍数が望ましい。

### レプリケーション環境のための考慮事項

バージョン7のレプリケーション環境は、バージョン8へ移行できます。バージョン7のサイトは、レプリケーション環境内でバージョン8のサイトと共存し、正常に実行できます。ただし、各システムにインプリメントされているさまざまなレプリケーション機能に適應するように、特別の注意を払ってください。

関連項目：レプリケーション機能を使ったシステムの移行に関する詳細は、『Oracle8 Server レプリケーション』の付録B「移行および互換性」を参照してください。

## 異なるコンピュータ・アーキテクチャへの移行

バージョン8の移行ユーティリティでは、アーキテクチャが異なるコンピュータ・システムにデータベースを移行することはできません。たとえば、32ビット・プラットフォームから64ビット・プラットフォームへの移行や、Solaris上のバージョン7からWindows NT上のバージョン8へのデータベースの移行は行われません。通常、種類の異なるオペレーティング・システムは異なるアーキテクチャに対応付けられています。したがって、ほとんどの場合、移行ユーティリティはオペレーティング・システム間でのデータベースの移行ができません。ただし、エクスポート/インポートを使えば、新しいアーキテクチャへのデータベースの移行が可能です。

## 文字のコード化に関する考慮事項

移行を成功させるには、バージョン8のデータベースのデータに使用する文字コード体系を正しく指定する必要があります。バージョン8のデータベースのすべての文字データは、データベースを作成した CREATE DATABASE コマンドで指定した文字コード体系になっています。

関連項目: これらの文字コード化を指定するための各国語サポート (NLS) の詳細は、『Oracle8 Server リファレンス・マニュアル』を参照してください。

移行ユーティリティは、INIT.ORA初期化ファイルの言語パラメータで別の指定がされていない限り、バージョン7のソース・データベースのキャラクタ・セットをバージョン8のデータベースのキャラクタ・セットとして使用します (6-15 ページの「NCHAR および NLS の使用」と C-4 ページの「NCHAR パラメータと NLS パラメータ、および互換性」も参照)。このキャラクタ・セットは移行が完了した後では変更できませんので、移行ユーティリティを実行する前に、正しいキャラクタ・セットが指定されていることを確認してください。

## 移行のためのバージョン7のソース・データベースの準備

バージョン7のデータベースをバージョン8へ移行する前に、次のステップを実行してください。

1. データベースのリリース番号が、使用しているプラットフォームで移行ユーティリティがサポートするリリースよりも低い場合は、データベースをサポートされるリリースまでアップグレードまたは移行します。たとえば、プラットフォームによっては、移行ユーティリティを使ってバージョン7リリース 7.1.4 より低いデータベース (バージョン6、リリース 7.0、リリース 7.1.3 など) を移行することはできません。

関連項目: 使用しているプラットフォームの移行ユーティリティでサポートされるリリースについては、プラットフォームに固有の Oracle マニュアルを参照してください。そして、システムを必要なリリースまで移行またはアップグレードする必要がある場合は、『Oracle7 Server 移行ガイド、リリース 7.3』を使ってください。

2. プロシージャ・オプションがインストールされていなければ、バージョン7のインストール・メディアを使ってインストールします。詳細は、プラットフォームに固有のインストール・ガイドを参照してください。
3. すべてのデータファイルおよび表領域が正常にオンラインまたはオフラインであることを確認します。

メディア回復を必要とするデータファイルが存在する場合、バージョン8の移行ユーティリティは、先へ進まずにエラーを表示します。きれいにオフラインにできなかった表領域は、移行前に削除するかオンラインにする必要があります。これは、そうしないと移行後にバージョン8で表領域を使用できないからです。通常、ALTER TABLESPACE OFFLINE IMMEDIATEまたはOFFLINE TEMPORARYコマンドを使ってオフラインにされる表領域には、メディア回復が必要です。

---

---

注意: バージョン8のデータベースをオープンするときにオフラインになっている表領域は、バージョン7のデータベース・ファイル形式のまま残ります。オフラインの表領域は、移行後ならいつでもオンラインにできますし、ファイル・ヘッダーはそのときにバージョン8形式に変換されます。さらに、障害が発生した場合に大規模な復元を避けたいければ、SYSTEMとROLLBACKを除くすべての表領域をオフラインにし、別の移行処理を実行する必要がある場合はその後でSYSTEMとROLLBACKのデータファイルだけを復元することができます。

---

---

4. MIGRATE という名前のユーザーまたはロールがないようにします。これは、バージョン8の移行ユーティリティがこのスキーマを作成し、既存のユーザーまたはロールをこの名前で置き換え、最終的にはそれをシステムから削除するからです。

MIGRATE という名前のユーザーをチェックするには、次のコマンドを入力します。

```
Select * from dba_users where username = 'MIGRATE';
```

MIGRATE という名前のロールをチェックするには、次のコマンドを入力します。

```
Select * from dba_roles where role = 'MIGRATE';
```

5. 次のコマンドを入力して、SYSTEM ロールバック・セグメントが十分に大きいことを確認します。

```
Select count(*) from dba_extents where segment_name = 'SYSTEM' and  
segment_type = 'ROLLBACK';
```

6. 次のコマンドを入力して、SYSTEM ロールバック・セグメントが最大のエクステントに達していないことを確認します。

```
Select max_extents from dba_rollback_segs where segment_name = 'SYSTEM';
```

7. 未処理の REDO 情報およびコミットされていないトランザクションがないことを確認し、未処理のインダウト・トランザクションをすべて解決します。つまり、『Oracle8 Server 分散システム』の「インダウト・トランザクションの手動上書き」の項で説明したとおり、インダウト・トランザクションが DBA\_2PC\_PENDING 状態のままにならないようにします。
8. SYSTEM 表領域に十分な空き領域を確保して、バージョン 8 のデータ・ディクショナリおよび既存のバージョン 7 のデータ・ディクショナリを同時に保持します。

関連項目：詳細は、3-3 ページの「領域要件」を参照してください。

## バージョン 8 の移行ユーティリティのインストール

次のステップを実行し、バージョン 8 のインストール・メディアからバージョン 7 の \$ORACLE\_HOME ディレクトリへ、バージョン 8 の移行ユーティリティの実行可能ファイルをインストールします。

関連項目：インストール・ユーティリティの詳細は、プラットフォームに固有のインストール・ガイドを参照してください。

---

注意：次の手順の一部のステップにある特定のコマンドは、プラットフォームによって異なります。詳細は、プラットフォームに固有の Oracle マニュアルと README ファイルを参照してください。

---

1. バージョン 8 のインストール・ユーティリティを実行します。ユーティリティの名前はプラットフォームによって異なります。たとえば、UNIX システムでは orainst、VMS では oracleins となります。コマンドについては、プラットフォームに固有の Oracle インストール・マニュアルを参照してください。

---

注意：移行ユーティリティをインストールするのにインストール・ユーティリティを使わない場合は、必要なファイルをインストール・メディアからシステムにコピーするだけでもかまいません。必要な移行ファイルのコピーに関する詳細は、プラットフォームに固有の Oracle マニュアルを参照してください。

---

2. 「Install Type」画面で、「Default or Customer Install」を選択します。
3. 「Select the Installer Activity」画面で、「Install, Upgrade, or De-Install Software」を選択します。
4. 「Select the Installer Option」画面で、「Migrate from ORACLE7 to ORACLE8」を選択します。

5. 「Installation Options: Home Locator」画面で、バージョン7の\$ORACLE\_HOMEの位置を示す絶対パスを入力します。
6. 「Logging and Status」画面で、ログ・ファイルの位置を確認または変更します。
7. 「Select an ORACLE7 to ORACLE8 Migration Action」画面で、「Install Migration Utility」を選択します。
8. 「Install Source」画面で、「Install from CD-ROM」または「Install from Staging Area」のどちらかを選択します。「Install from Staging Area」を選択すると、「Source Staging Area」画面が表示され、ソース・ステージング領域のパス名を入力するように求められます。
9. 「NLS」画面で、インストールに使用する母国語を選択します。
10. 「Software Asset Manager」画面で、「Migration Utility: Oracle7 to Oracle8 8.0.X」を選択します。ここで、X は最新の Oracle リリース番号です。

インストール・ユーティリティにより、移行ユーティリティがインストールされます。

11. インストール・ユーティリティをインストールして、バージョン7のデータベース上で移行ユーティリティを実行するための環境を、インストール・ユーティリティが作成していることを確認します。

たとえば、UNIX プラットフォームでは、インストール・ユーティリティによって次のインストール処理が実行されたことを確認します。

- バージョン 8 の移行ユーティリティの実行可能ファイルである mig が、バージョン 7 環境の \$ORACLE\_HOME/bin ディレクトリにインストールされます。
- バージョン 8 のメッセージ・ファイルである migus.msb が、バージョン 7 環境の \$ORACLE\_HOME/rdbms/msg ディレクトリにインストールされます。
- バージョン 8 の migrate.bsq が、バージョン 7 環境の \$ORACLE\_HOME/dbs ディレクトリにインストールされます。
- 必須の NLS ファイルが、バージョン 7 環境の \$ORACLE\_HOME/migrate/nls/admin/data ディレクトリにインストールされます。

## バージョン 8 の移行ユーティリティ・コマンド行オプションの検討

移行処理の次の作業は、バージョン 8 の移行ユーティリティを実行することです。作業を始める前に、次の移行ユーティリティのコマンド行オプションを検討してください。移行処理の中でその一部を使いたくなる場合があるからです。また、使用しているプラットフォームでの移行ユーティリティのコマンド行オプションに関する詳細が、プラットフォームに固有の Oracle マニュアルに記載されていることがあります。

CHECK_ONLY	TRUE にすると、移行ユーティリティは、移行を実行せずに領域使用率を計算します。FALSE にすると、移行ユーティリティは領域使用率の計算と移行の両方を実行します。このコマンド行オプションは、NO_SPACE_CHECK の反対です。CHECK_ONLY コマンド行オプションを指定しなければ、移行ユーティリティはこれをデフォルトで FALSE に設定します。
DBNAME	移行するデータベース名 ( INIT.ORA ファイル内の DB_NAME ) を指定します。
MULTIPLIER	バージョン 8 の i_file#_block# 索引の初期サイズを、バージョン 7 の i_file#_block# 索引との比例で指定します。たとえば、MULTIPLIER=30 とすると、索引が作成されたときの初期サイズが 3 倍になります。MULTIPLIER コマンド行オプションを指定しない場合、移行ユーティリティは i_file#_block# 値 15 を使用し、バージョン 8 用に作成される i_file#_block# 索引は、バージョン 7 の i_file#_block# 索引よりも 1.5 倍大きくなります。
NEW_DBNAME	移行したデータベースに新しい名前を指定します。デフォルト名に「 DEFAULT 」は使わないでください。より意味のある名前を選んでください。
NLS_NCHAR	バージョン 8 のデータベースの props\$ に各国語サポート ( NLS ) NCHAR キャラクタ・セットを指定します (たとえば、W52DEC または US7ASCII)。NLS_NCHAR コマンド行オプションを指定しなければ、移行ユーティリティはバージョン 7 のデータベースのキャラクタ・セットを使います。
NO_SPACE_CHECK	TRUE にすると、移行ユーティリティは移行前に領域使用率チェックを行いません。FALSE にすると、移行ユーティリティは移行前に領域使用率チェックを行います。このコマンド行オプションは、CHECK_ONLY の反対です。NO_SPACE_CHECK コマンド行オプションを指定しなければ、移行ユーティリティはこれをデフォルトで FALSE に設定します。
PFILE	パラメータ・ファイルの名前を指定します。PFILE コマンド行オプションを指定しなければ、移行ユーティリティはデフォルトの INIT.ORA ファイルを使います。 注意: UNIX 上では、次の例のようにパス名を "¥" 記号でマスクされた二重引用符で囲む必要があります。 mig PFILE=¥"/tmp/mig/pfile¥"
SPOOL	出力をスプールするファイルのファイル名を指定します。 注意: UNIX 上では、次の例のようにパス名を "¥" 記号でマスクされた二重引用符で囲む必要があります。 mig SPOOL=¥"/tmp/mig/spool¥"

## バージョン7のソース・データベースの移行

移行ユーティリティを使ってバージョン7のソース・データベースをバージョン8へ移行するには、以下のセクションに示すステップを実行してください。

### バージョン7の環境での移行ステップ

バージョン7の環境では、次の移行ステップを実行してください。

1. README.doc ファイルをよく読んでおきます。
2. バージョン8の移行ユーティリティを実行するのに必要なDBA権限を持っていることを確認します。
3. プラットフォームに固有のマニュアルを参照し、作成しようとしているバージョン8の新しいデータベース・オブジェクトの配置を制御するシステム設定を確認します。

次の例は、プラットフォームに固有の変数の設定を示します。

- UNIXシステムでは、TWO\_TASK環境変数が設定解除されている必要があります。
- NTシステムでは、TWO\_TASKが設定状態のままではなりません。
- VMSでは、ORA\_DFLT\_HOSTSTR論理変数が設定解除されている必要があります。

4. RESTRICTED SESSION権限を持つ他のDBA（CONNECT INTERNAL）が、移行ユーティリティの実行中にデータベースへ接続しないようにします。「通常」のユーザーは移行処理中にデータベースへ接続することはできません。
5. ORA\_NLS33変数が正しく設定されていることを確認します。

たとえば、NLSファイルをデフォルト位置（\$ORACLE\_HOME/migrate/nls/admin/data）にインストールしているUNIXでは、ORA\_NLS33を\$ORACLE\_HOME/migrate/nls/admin/dataに設定しなければなりません。

6. バージョン7のsql.bsqファイルがカスタマイズされている場合は、\$ORACLE\_HOME/dbsディレクトリ内のmigrate.bsqファイルに対して同じ変更を加えます。カスタマイズされていない場合は、ステップ7へ進みます。

移行処理中、migrate.bsqは、通常はsql.bsqによって作成される多数のオブジェクトを作成しますが、オブジェクトをMigrateスキーマのもとで作成します。後に移行処理の中で、これらのオブジェクトの所有権はSYSに変更されます。

7. SHUTDOWN NORMALまたはSHUTDOWN IMMEDIATEコマンドを使って、バージョン7のデータベースを完全に停止します。SHUTDOWN ABORTは使わないでください。バージョン7のソース・データベースは、完全に停止する必要があります。これにより、REDO情報やコミットされていないトランザクションは残りません。

---

注意：移行を開始する前にバージョン7のデータベースが停止していない場合は、移行ユーティリティが停止してエラー・メッセージが表示されません。

---

8. バージョン8の移行ユーティリティを実行します。移行ユーティリティは、システム・プロンプトで移行ユーティリティのコマンドを入力するか、インストール・ユーティリティを使うことによって実行できます。

システム・プロンプトで移行ユーティリティを開始するには、プラットフォームに固有のマニュアルに示されているコマンドを使用します。

たとえば、UNIX システムでは、mig コマンドを入力します。mig とだけ入力して移行ユーティリティをデフォルト設定のオプションで実行するか、mig を入力した後に1つまたは複数の選択オプションを続けます。

関連項目：コマンド行オプションの詳細は、3-8 ページの「バージョン8の移行ユーティリティ・コマンド行オプションの検討」を参照してください。

移行ユーティリティをインストール・ユーティリティから実行するには、次のステップを実行してください。

- a. バージョン8のインストール・ユーティリティを実行します。プラットフォームに応じた正しいコマンドを使ってください。たとえば、UNIX システムでは orainst を使い、VMS では oracleins を使います。

関連項目：詳細は、プラットフォームに固有のインストレーション・ガイドを参照してください。

- b. 「Install Type」画面で、「Default or Customer Install」を選択します。
- c. 「Select the Installer Activity」画面で、「Install, Upgrade, or De-Install Software」を選択します。
- d. 「Select the Installer Option」画面で、「Migrate from ORACLE7 to ORACLE8」を選択します。
- e. 「Installation Options: Home Locator」画面で、バージョン7の\$ORACLE\_HOMEの位置を示す絶対パスを入力します。
- f. 「Logging and Status」画面で、ログ・ファイルの位置を確認または変更します。
- g. 「Select an ORACLE7 to ORACLE8 Migration Action」画面で、「Run Migration Utility」を選択します。
- h. 「ORACLE\_SID」画面で、移行するデータベースのORACLE\_SIDを入力します。
- i. 「Software Asset Manager」画面で、「Migration Utility: Oracle7 to Oracle8 8.0.X」を選択し（Xは現行のリリース番号）、「Install」を選択します。
- j. 「Migrate SID」画面で、移行するデータベースのORACLE\_SIDを入力します。

インストール・ユーティリティが、移行ユーティリティを実行します。

9. 移行ユーティリティの実行後に結果をチェックします。移行ユーティリティは情報メッセージを生成し、migrate.bsq スクリプトの実行時にその進行状況をエコーします（詳細は、付録 A 「移行ユーティリティのメッセージ」を参照）。

移行ユーティリティは、バージョン7の制御ファイルの情報を含む変換ファイルを作成します。後に移行処理の中で、この変換ファイルはバージョン8の新しい制御ファイルを作成するときに ALTER DATABASE CONVERT によって使われます。

変換ファイルの名前と位置は、プラットフォームによって異なります。たとえば、UNIX プラットフォームでは、バージョン7の環境の場合のデフォルト位置は \$ORACLE\_HOME/dbs であり、このディレクトリ内のデフォルト・ファイル名は convSID.dbf です。この SID は、バージョン7のインスタンス ID です。

---

---

警告：バージョン8の移行ユーティリティによって停止されたバージョン7のデータベースはオープンしないでください。データファイルのバージョンを確実に統合するには、データベースをバージョン8へ変換するときに、ディクショナリ内の SCN および変換ファイル、ファイル・ヘッダーのすべてに一貫性がなければなりません。移行ユーティリティの実行後にバージョン7のデータベースをオープンすると、SCN のチェックが、データベースをバージョン8へ変換するときに失敗し、「Oracle7 データファイルは Oracle8 へ移行されたものではありません。」という ORA-1211 エラーが表示されます。したがって、バージョン7のデータベースをオープンする場合は、移行ユーティリティを再実行し、ステップ7から開始します。

---

---

## バージョン7のソース・データベースの保存

移行ユーティリティが正常に終了したら、バージョン7のデータベースのコールド・バックアップを取ってください。このバックアップは、次の目的で使います。

- バージョン8で ALTER DATABASE CONVERT コマンドを実行後にバージョン7のデータベースに戻りたくなった場合は、バックアップを復元してバージョン7のデータベースを開始できます。
- バージョン8を回復するための最初のバックアップとして使用できます。
- バージョン8のデータベースの変換時（ALTER DATABASE CONVERT または ALTER DATABASE OPEN RESETLOGS）にエラーが発生した場合は、このバックアップを復元し、問題を修正して変換処理を継続することができます。ただし、移行ユーティリティの実行前に取ったバックアップを復元する場合は、移行ユーティリティを再実行する必要があります。

関連項目：バックアップ作成の詳細は、『Oracle8 Server バックアップおよびリカバリ』を参照してください。

さらに、バージョン 7 のホーム・ディレクトリ ( UNIX の \$ORACLE\_HOME または VMS の ORA\_ROOT ) を含む、バージョン 7 で配布されるソフトウェア全体のバックアップを作成します。そのバックアップには、次のファイル群が必ず含まれるようにしてください。

- すべてのサブディレクトリ
- 制御ファイル
- データファイルとオンライン REDO ログ・ファイル ( バージョン 7 のデータベースのデータファイルがなくなるか読みとれない場合に備えて )。ただし、これらのファイルには未処理の REDO 情報は含まれていないはずで
- パラメータ・ファイル
- 変換ファイル
- バージョン 7 のデータベースにオブジェクトを作成するスクリプト
- 必要に応じて元のデータベースを復元できるスクリプト

## バージョン 8 の環境での移行ステップ

バージョン 8 の環境では、次の移行ステップを実行してください。

1. バージョン 8 のインストール・ユーティリティを使ってバージョン 8 のソフトウェアをインストールします。Install/Upgrade オプションを選択して、新しいデータベースを作成しないようにしてください。バージョン 8 のデータベースをインストールする手順は、プラットフォームによって異なります。

関連項目：プラットフォームに固有のバージョン 8 のインストレーション・マニュアルを参照し、バージョン 8 に付属する README.DOC ファイル内の追加事項または変更事項を読んでください。

2. 次の環境変数が、バージョン 8 の実行可能ファイルを指定することを確認します。
  - ORACLE\_HOME
  - PATH
  - ORACLE\_PATH
  - LD\_LIBRARY\_PATH

---

注意：ORACLE\_HOME がバージョン 7 の実行可能ファイルを指している場合は、ALTER DATABASE CONVERT コマンドの実行時に ORA-223 エラー「変換データ・ファイルが無効かまたはバージョンが誤っています。」が表示されます。

---

- バージョン8のDBMSを実行するオペレーティング・システムが、バージョン7のDBMSおよびバージョン8の移行ユーティリティの実行に使用するオペレーティング・システムと同じであることを確認します。

- INIT.ORA ファイルをバージョン8用に調整します。

バージョン7の特定の初期化パラメータは、バージョン8では廃止されています。廃止されたパラメータは、バージョン8のインスタンスを開始する初期化パラメータ・ファイルからすべて除去してください。廃止されたパラメータをバージョン8のデータベースで使用すると、エラーになることがあります。また、バージョン8で構文が変更されているパラメータを変更することも必要です。新規のパラメータおよび変更されたパラメータ、廃止されたパラメータのリストは、付録C「バージョン8のINIT.ORAの変更点」を参照してください。

- INIT.ORA ファイル内の COMPATIBLE パラメータを設定します。

関連項目：詳細は、C-2 ページの「COMPATIBLE パラメータ」を参照してください。

- データベースの制御ファイルを、削除または改名します。あるいは、INIT.ORA ファイル内の CONTROL\_FILES パラメータを使って制御ファイルの新しい名前を指定します。

ALTER DATABASE CONVERT コマンドにより、新しい制御ファイルが自動的に作成されます。CONTROL\_FILES パラメータを使わない場合は、このコマンドが移行前のデータベースの制御ファイルの名前を使い、その制御ファイルがすでに存在していたらエラーを戻します。したがって、この場合は、制御ファイルを削除するか改名しなければなりません。

しかし、INIT.ORA ファイル内で CONTROL\_FILES パラメータを使った場合は、指定した名前を持つ新しい制御ファイルが ALTER DATABASE CONVERT コマンドによって作成され、古い制御ファイルを削除する必要はありません。

---

注意：CONTROL\_FILES は、カンマで区切られた1つまたは複数の制御ファイルの名前を指定します。オラクル社では、異なる複数のデバイスで複数のファイルを使うか、オペレーティング・システム・レベルでファイルをミラー化することをお勧めします。詳細は、『Oracle8 Server 管理者ガイド』を参照してください。

---

- バージョン8の実行可能ファイルを別の \$ORACLE\_HOME ディレクトリにインストールした場合は、変換ファイルをバージョン7の環境からバージョン8の環境へ移動またはコピーします。同じ \$ORACLE\_HOME を使う場合は、ステップ9へ進みます。

UNIX では、変換ファイル convSID.dbf (SID はバージョン8のインスタンス ID) は、\$ORACLE\_HOME/dbs の中に存在する必要があります。バージョン8のインスタンス ID がバージョン7のインスタンス ID と異なっている場合は、convSID.dbf ファイルを改名して、バージョン8のインスタンス ID と一致させます。

8. バージョン8の実行可能ファイルを別の \$ORACLE\_HOME ディレクトリにインストールし、パスワード・ファイルがある場合は、そのパスワード・ファイルをバージョン7の環境からバージョン8の環境へ移動またはコピーします。

パスワード・ファイルの名前と位置は、プラットフォームによって異なります。たとえば、UNIX プラットフォームでは、デフォルトのパスワード・ファイルは \$ORACLE\_HOME/dbs/orapwSID ですが、Windows NT では、デフォルトのパスワード・ファイルは \$ORACLE\_HOME/database/pwdSID.ora です。どちらの場合も、SID が Oracle のインスタンス ID です。

9. すべてのオンライン・データ・ファイルがアクセス可能で、正しいディレクトリに入っていることを確認します。RAW ディスクを使用する場合は、ログ・ファイルもアクセス可能でなければなりません。
10. 次のように入力し、Server Manager を起動します。

```
SVRMGRL
```

11. バージョン8のデータベースのインスタンスに接続します。

```
SVRMGR> CONNECT INTERNAL
```

12. バージョン8の新しいデータベースをマウントせずに、バージョン8のデータベースのインスタンスを開始します。

```
SVRMGR> STARTUP NOMOUNT
```

---

---

注意：データベースのインスタンスを他のモードで開始すると、そのデータベースを破損する可能性があります。

---

---

13. バージョン8のデータベースの新しい制御ファイルを作成し、すべてのオンライン表領域のファイル・ヘッダーをバージョン8形式に変換します。

```
SVRMGR> ALTER DATABASE CONVERT;
```

このコマンドの実行が成功すると、このデータベースについてはバージョン7に戻れなくなります。ただし、必要に応じてバックアップからバージョン7のデータベースを復元することはできます。

---

---

注意：制御ファイルは、バージョン7よりもバージョン8の方がかなり大きくなっています。バージョン7でのサイズが数十キロバイトの制御ファイルは、バージョン8への移行中に自動的に数十メガバイトのサイズに拡大される場合があります。制御ファイルがRAWデバイス上にあるか使用可能なディスク領域が制限されている場合は、このサイズの増大が重要になります。

---

---

このステップでエラーが発生した場合は、エラーの原因となっている条件を訂正し、移行ユーティリティを再実行してください。3-10 ページのステップ 1 からやり直します。もしくは、移行ユーティリティの実行後に取ったバックアップを復元してください。

14. 次のコマンドを使ってバージョン8のデータベースをオープンします。

```
SVRMGR> ALTER DATABASE OPEN RESETLOGS;
```

バージョン8のデータベースがオープンされているときは、オンライン状態のロールバック・セグメントもすべて新しいバージョン8形式に変換されます。

15. 成功したかどうかを後で確認するために、結果をログ・ファイルにスプールするようにシステムを設定します。

```
SVRMGR> SPOOL CATOUT.LOG
```

16. バージョン8のデータベースの変換スクリプト CAT8000.SQL を実行します。

```
SVRMGR> @CAT8000.SQL
```

CAT8000.SQL スクリプトは、特定のシステム表を作成および変更し、"MIGRATE" ユーザーを削除します。また、システム・カタログ・ビューおよび PL/SQL の使用に必要なすべてのパッケージを作成する、CATALOG.SQL スクリプトと CATPROC.SQL スクリプトを実行します。

関連項目：その他のデータ・ディクショナリ構造体を作成する場合、使用可能なスクリプトの完全な一覧と説明については、『Oracle8 Server リファレンス・マニュアル』を参照してください。

17. Oracle システムにアドバンスド・レプリケーションをインストールしている場合は、CATREP8M.SQL カタログ・スクリプトを実行します。

```
SVRMGR> @CATREP8M.SQL
```

CATREP8M.SQL スクリプトは、自動的に CATREP.SQL スクリプトを実行します。

18. Oracle システムに Parallel Server をインストールしている場合は、新しいリリースに付属する次のカタログ・スクリプトを実行します。

```
SVRMGR> @CATPARR.SQL
```

19. スクリプト結果のログ・ファイルへのスプーリングをオフにします。

```
SVRMGR> SPOOL OFF
```

次にスプール・ファイルをチェックして、すべてのパッケージとプロシージャのコンパイルが成功したかどうかを検証してください。ステップ 15でスプール・ファイルに名前を付けましたが、その名前はCATOUT.LOGでした。このファイルに問題を見つけたら、それを訂正してください。

20. バージョン8のデータベース上で、SHUTDOWNを実行します。

```
SVRMGR> SHUTDOWN NORMAL
```

---

---

警告: SHUTDOWN NORMALまたはSHUTDOWN IMMEDIATEを使ってください。SHUTDOWNABORTは使わないでください。

---

---

この正しい停止を実行すると、すべてのキャッシュをフラッシュし、バッファを消去し、その他の内部的な不整合の解消を行います。これらの手段は、新規に移行したバージョン8のデータベースの整合性と一貫性を保証するための重要な最終ステップです。

21. データベース全体の全表走査を行います。

---

---

注意: このステップは推奨であり、必須ではありません。

---

---

移行処理により表定義が新しい形式に変更されましたが、既存のROWIDは、アクセスがあるまでは変更されません。ユーザーが表の中のROWIDにアクセスするのを待つのではなく、この処理を一度に完了したい場合があります。データベース全体の全表走査を簡単に行うには、データベースの全体エクスポートをFILE=/dev/nullで行います。このエクスポートでは、エクスポート・ファイルは作成されません。

---

---

注意: アプリケーションで使うために表の中にROWIDを格納してある場合は、それらをバージョン8形式に変換しなければなりません。詳細は、第7章「バージョン8のROWIDに関する移行の問題」を参照してください。

---

---

22. 第5章「データベースの移行後」に記載の手順を実行します。

## 移行時のエラー

エラーが発生する原因としては、次のような操作が行われたか、あるいは省略されたことが考えられます。

- 誤った順序で移行ステップを実行する
- 移行の前提条件を満たしていない
- 変換の例外を検出する

関連項目：移行中に発生する可能性のあるエラーとそのエラーの対処措置については、付録A「移行ユーティリティのメッセージ」および『Oracle8 Server エラー・メッセージ』を参照してください。

## 移行の中止

バージョン 8 の移行ユーティリティは複数回実行できますが、それでもバージョン 7 のデータベースへ戻れなくなることはありません。ただし、移行ユーティリティを実行すると、バージョン 7 のデータベースのカタログ・ビューが自動的に削除されます。したがって、移行ユーティリティの実行後にバージョン 7 のデータベースに戻るには、バージョン 7 の CATALOG.SQL スクリプトを実行して、バージョン 7 のデータベースのカタログ・ビューを復元する必要があります。

移行を中止するには、一般的に次のステップを実行してバージョン 7 のデータベースを復元しなければなりません。

1. Server Manager を使ってバージョン 7 のデータベースを起動します。

```
SVRMGRL
SVRMGR> CONNECT INTERNAL
SVRMGR> STARTUP
```

2. ユーザー "MIGRATE" を削除します。

```
SVRMGR> DROP USER MIGRATE CASCADE
```

3. CATALOG.SQL と CATPROC.SQL を再実行します。

```
SVRMGR> @CATALOG.SQL
SVRMGR> @CATPROC.SQL
```

4. Server Manager がインストールされている場合は、CATSVRMG.SQL を実行します。

```
SVRMGR> @CATSVRMG.SQL
```

5. Parallel Server がインストールされている場合は、CATPARR.SQL を実行します。

```
SVRMGR> @CATPARR.SQL
```

6. アドバンスド・レプリケーションがインストールされている場合は、CATREP.SQL を実行します。

```
SVRMGR> @CATREP.SQL
```

---

---

注意：バージョン8の移行ユーティリティは、リリース7.1およびリリース7.2のデータベースをリリース7.3にアップグレードします。元のバージョン7の本番データベースがリリース7.1または7.2であり、移行が実行されてもバージョン8への変換前に中止された場合、バージョン7のデータベースはディクショナリがリリース7.3のままになります。

---

---

移行の中止

---

---

## エクスポート / インポートによる移行

この章では、エクスポート/インポートを使用してデータベースをバージョン7からバージョン8へ移行する方法について説明します。

この章で説明するトピックは、次のとおりです。

- エクスポート/インポートの基本
- その他のエクスポート/インポート情報源
- その他のオプション
- エクスポート/インポートによるバージョン8より前のソース・データベースの移行

## エクスポート/インポートの基本

エクスポート/インポートを使ってデータベースを移行するには、次の3つの基本ステップを行います。

1. 移行元のデータベース（ソース・データベース）からデータをエクスポートする。このエクスポートでは、データをエクスポート・ダンプ・ファイルへコピーする。
2. エクスポートしたデータをインポートするバージョン8のデータベース（ターゲット・データベース）を作成する。
3. エクスポートしたデータを新しいバージョン8のターゲット・データベースにインポートする。

関連項目：エクスポート/インポートによる移行方法の評価については、2-2ページの「移行方法の選択」および『Oracle8 Server ユーティリティ』を参照してください。

### エクスポート要件

データベースを移行またはアップグレードするには、ソース・データベースのバージョンまたはリリースに付属のエクスポート・ユーティリティを使用します。エクスポートの後、インポート・ユーティリティを使用して、エクスポート・ダンプ・ファイルのデータをターゲット・データベースにコピーします。エクスポートしたデータをインポート・ユーティリティでターゲット・データベースに移行する前に、ターゲット・データベースが作成済みですらに操作可能な状態にある必要があります。

### インポート要件

データベースを移行またはアップグレード、ダウングレードするには、ターゲット・データベースのバージョンまたはリリースに付属のインポート・ユーティリティを使用します。

### その他のエクスポート/インポート情報源

エクスポート/インポートを使用してデータベースを移行する場合は、エクスポート/インポート・ユーティリティの使用方法について、『Oracle8 Server ユーティリティ』をよく読んでおいてください。

### その他のオプション

その他のオプションをインストールしている場合は、次の関連情報を参照してください。

関連項目：Trusted Oracle データベース（バージョン7）からデータをエクスポートする場合や、Trusted Oracle データベース（バージョン7）へデータをインポートする場合は、Trusted Oracle データベースの機能の移行について、『Trusted Oracle7 Administrator's Guide』を参照してください。

関連項目：アドバンスド・レプリケーションをインストール済みのバージョン8以前のデータベース・システムを移行する場合は、『Oracle8 Server レプリケーション』の付録B「移行および互換性」を参照してください。

## エクスポート/インポートによるバージョン 8 より前のソース・データベースの移行

---

注意: ソースとなる Oracle データベースがバージョン 6 よりも前の場合は、エクスポートを行う前に、ソース・データベースを最低でもバージョン 6 までアップグレードしてください。

---

エクスポート/インポート・ユーティリティを使用して Oracle データベースを移行するには、次のステップを実行します。

1. ソース・データベースに付属のエクスポート・ユーティリティを使用してソース・データベースをエクスポートします。ソース・データベースでのエクスポート・ユーティリティの使用方法については、ソース・データベースの Server ユーティリティのマニュアルを参照してください。バージョン 7 のデータベースおよびバージョン 6 のデータベースのエクスポートは両方とも、バージョン 8 にインポートできます。

---

注意: 一貫性のあるエクスポート結果を得るには、エクスポートの実行中と実行後はソース・データベースが更新不可でなければなりません。エクスポート実行後にユーザーがソース・データベースを更新できるようにする場合は、インポート完了後にソース・データベースに対する変更をバージョン 8 のターゲット・データベースにコピーする手順を実行し、その後でソース・データベースを使用可能にします。

---

2. バージョン 8 のソフトウェアをインストールします。バージョン 8 のインストール・ステップは、プラットフォームに固有のものです。プラットフォームに固有のバージョン 8 のマニュアルおよびバージョン 8 に含まれる README ファイルに詳細が記述されています。
3. 新しいバージョン 8 のデータベースに既存のソース・データベースと同じ名前を付ける場合は、新しいバージョン 8 のデータベースを作成する前に既存のデータベースを停止する必要があります。
4. バージョン 8 のターゲット・データベースを作成します。

関連項目: バージョン 8 データベースの作成方法については、『Oracle8 Server 管理者ガイド』を参照してください。

5. Server Manager から次のコマンドを入力して、新しいバージョン 8 のターゲット・データベースをオープンします。

```
SVRMGR> CONNECT INTERNAL
SVRMGR> STARTUP
```

6. 記憶領域パラメータを変更してディスク領域の使用状況を改善するなどの目的のために、ターゲット・データベースに表領域およびユーザー、表を事前に作成しておきます。SQL\*Plus や Server Manager を使用して表を事前に作成する場合は、元のデータベースとの互換モードで実行するか、またはインポート中に発生する特定のデータ定義の変換について特に配慮しなければなりません。

---

注意：新しいバージョン 8 のデータベースをソース・データベースと同じマシン上に作成し、ソース・データベースのデータファイルを上書きしたくない場合は、事前に表領域を作成しておき、インポート時に IGNORE=Y と DESTROY=N を指定する必要があります。

---

7. バージョン 8 のインポート・ユーティリティを使用して、ソース・データベースからエクスポートしたオブジェクトをインポートします。インポート・セッションからファイルに情報メッセージやエラー・メッセージを保存するには、LOGパラメータを組み込んでください。

関連項目：インポート・ユーティリティの詳細は、『Oracle8 Server ユーティリティ』を参照してください。

8. 移行作業が完了したら、どのオブジェクトのどのインポートが正常に終了してどれが失敗したかを、インポート・ログ・ファイルを見てチェックします。

関連項目：エラー処理情報については、『Oracle8 Server ユーティリティ』およびバージョン 8 サーバーの README ファイルを参照してください。

9. その他のインポート方法(『Oracle8 Serverユーティリティ』を参照)を使用するか、ソース・オブジェクトの作成に使用する SQL スクリプトを使用して、インポートに失敗したオブジェクトを作成し直します(あるいはまったく新規のインポートを開始します)。
10. エクスポートの実行後にソース・データベースに変更を加えた場合(4-3 ページのステップ 1)は、そのデータベースを使用可能にする前にそれらの変更がバージョン 8 のデータベースに反映されていることを確認してください。
11. 第 5 章「データベースの移行後」に記載の手順を実行します。

# 5

---

## データベースの移行後

この章では、移行ユーティリティまたはエクスポート/インポートを使った移行を完了した後で実行する手順について説明します。また、1-4ページの「ステップ6: 新しいバージョン8の本番データベースのチューニングと調整」についてさらに詳しく説明します。

この章で説明するトピックは、次のとおりです。

- バージョン8のデータベースのバックアップ
- 無効な日付制約のチェック
- 無効にされたビットマップ索引の再構築
- データベースのテストと結果の比較
- ターゲット・データベースのチューニング
- 新機能の適宜追加
- 必要な新管理手順の作成

## バージョン 8 のデータベースのバックアップ

新たに移行した本番データベースの完全なバックアップを、必ず作成してください。このバックアップは、データ・ファイル、制御ファイル、オンライン REDO ログ・ファイル、パラメータ・ファイル、新しいデータベース内のオブジェクトを作成するための SQL スクリプトなどのすべてを含む完全なバックアップでなければなりません。ホット・バックアップは完全な回復可能性を提供できないので、完全なバックアップを作成するには全データベース・エクスポートまたはコールド・バックアップが必要です。後続のステップによって、新しく移行されたデータベースに悪い影響が出るような場合は、必要に応じてここで作業を中止することもできます。

関連項目：データベースのバックアップの詳細は、『Oracle8 Server バックアップおよびリカバリ』および『Oracle8 Server 管理者ガイド』を参照してください。

---

注意：移行ユーティリティを使用すると、ソース・データベースが変換されます。したがって、移行後は3-12ページの「バージョン7のソース・データベースの保存」で作成したバックアップ以外は、ソース・データベースがなくなります。このバックアップは、新規に移行したバージョン8のデータベースを回復するための最初のバージョン8のバックアップとしても使用できます。

---

## 無効な日付制約のチェック

無効な日付の制約は、無効日付操作を必要とします。無効日付操作は、日付の中の世紀を暗黙的に想定するものであり、西暦 2000 年に問題が発生します。UTLCONST.SQL スクリプトは、データベース内のすべてのチェック制約で実行され、無効日付操作を組み込む場合は制約を無効に設定します。このスクリプトは、最後にすべての無効な制約を選択します。

UTLCONST.SQL スクリプトを実行するには、次のように入力します。

```
SVRMGR> SPOOL UTLRESULT.LOG
SVRMGR> @UTLCONST.SQL
SVRMGR> SPOOL OFF
```

スクリプトを実行した後、UTLRESULT.LOG ログ・ファイルには、無効な日付制約を持つすべての制約が入っています。

---

注意：UTLCONST.SQL は無効な制約を訂正せずに、使用禁止にします。必要な変更を加えた後に、無効な制約を削除するか、再作成する必要があります。

---

## 無効にされたビットマップ索引の再構築

移行中に、一部のビットマップ索引が無効になることがあります。これらの索引を見つけるには、ALL\_INDEXES や DBA\_INDEXES などの索引ビューを使います。無効にされたビットマップ索引を見つけたら、それを再構築してください。

関連項目：ビットマップ索引の使用の詳細は、『Oracle8 Server チューニング』および『Oracle8 Server 概要』を参照してください。

## データベースのテストと結果の比較

2-10 ページの「テスト計画の作成」で作成したテスト計画を使って、バージョン 8 のデータベースをテストします。テスト結果を元のデータベースで得られたテスト結果と比較し、同じ結果が得られているかまたは改善しているかを確認します。

一般的に、移行したバージョン 8 のデータベースでのパフォーマンスは、ソース・データベースでのパフォーマンス以上になります。バージョン 8 でデータベースのパフォーマンス低下が見られる場合は、初期化パラメータが正しく設定されているかを確認してください。初期化パラメータが不正に設定されていると、パフォーマンスに害を与えることがあるからです。

関連項目：データベースのチューニング情報については、『Oracle8 Server チューニング』を参照してください。

移行したテスト・データベースのテストに加えて、元のソース・データベースと新たに移行したバージョン 8 のデータベースをしばらくの間同時に実行し、新しいデータベースが正しく設定されていて適切に稼動することを確認することもできます。両方のデータベースを同時に実行するには、Oracle の構成を変更する必要があります。もちろん、両方のデータベースに同期した更新が必要です。理想的には、自動化プロセスの一部として更新を行います。

## ターゲット・データベースのチューニング

移行したデータベースのパフォーマンスを改善したい場合は、データベースを調整してください。

関連項目：チューニングの詳細は、『Oracle8 Server チューニング』を参照してください。

バージョン 7 のデータベースおよび関連アプリケーションのチューニングに使用した処理のほとんどは、バージョン 8 のデータベースにも同様に有効な場合と、不適切な場合があります。したがって、ソース・データベースおよびアプリケーションをチューニングするために行った処置が原因で、移行したバージョン 8 のデータベースのパフォーマンスが低下することのないようにしてください。

## 新機能の適宜追加

『Oracle8 と Oracle8 Enterprise Edition の解説』では、バージョン 8 で使用可能な多くの新機能について説明しています。これらの新機能のうちで移行したデータベースとアプリケーションにとって有益であるものを判断し、その機能を利用するための計画を立ててください。

しかし、移行したバージョン 8 のデータベースを使用するためにすぐに変更する必要はありません。データベースおよびそれに対応するアプリケーションに、これらの拡張機能を徐々に取り入れてもかまいません。

第 6 章「バージョン 7 のアプリケーションのアップグレード」では、バージョン 8 の新機能を利用するためにアプリケーションを拡張する方法について説明します。ただし、バージョン 8 の新機能をインプリメントする前にアプリケーションをテストし、それらが移行したテスト・データベース上でバージョン 7 の互換モードで正常に稼動していなければなりません。

## 必要な新管理手順の作成

バージョン 8 の機能に慣れてきたら、データベース管理用のスクリプトや手順を見直し、変更が必要かどうかを判断します。

それぞれのアプリケーションに必要な変更を、データベースにも行う必要があります。たとえば、データベースで整合性制約を使用可能にした場合、アプリケーションでのデータ整合機能のいくつかを取り除けます。

---

## バージョン7のアプリケーションのアップグレード

この章では、バージョン8でOracle Toolsとアプリケーションを使用する方法について説明します。バージョン8の新機能を使用しない既存の(バージョン7の)アプリケーションを変更する必要はありません。既存のアプリケーションは、バージョン8のデータベース上でも同様の機能、または拡張された機能を実行できます。

この章で説明するトピックは、次のとおりです。

- Oracleアプリケーションのバージョン8へのアップグレード
- プリコンパイラおよびOCIアプリケーションのアップグレード
- LONGからLOBへのアップグレード
- バージョン7のFormsまたはDeveloper/2000アプリケーションのアップグレード
- データ・ディクショナリ・ビューの更新
- SQL\*Plusスクリプトのアップグレード
- PL/SQL V2 互換モード
- SQL\*Net または Net8
- バックアップ管理 : EBU および Recovery Manager
- ディクショナリ保護
- パスワード管理
- エクスポート/インポートの使用方法、パーティション・オブジェクト
- OCIのスレッド・セーフティについての移行および互換性の問題
- スタンバイ・データベースのアップグレードおよび互換性の問題
- エクスポート/インポートの互換性の問題
- NCHAR および NLS の使用

## Oracle アプリケーションのバージョン 8 へのアップグレード

次のバージョン8の機能は、アプリケーションのアップグレード・プロセスで役に立ちます。

- SQL\*Net バージョン 2 および Net8 は、両方とも Oracle のさまざまなバージョンおよびリリースで動作する。その結果、バージョン7のデータベースとバージョン8のデータベースは、SQL\*Net バージョン 2 と Net8 を使用して通信できる。

---

注意：SQL\*Net バージョン 1 は、さまざまなネットワーク・アドレス指定スキームを使用していましたが、バージョン 8 では使用できません。

---

- アプリケーションの作成と管理については、プログラミング・インタフェース（たとえば、Oracle プリコンパイラおよび Oracle コール・インタフェース、または OCI）が Oracle の異なるバージョン間で変更されていない。たとえば、SQL\*Net または Net8 を使用できる。あるいは、以前のリリースのデータベース用に設計されたアプリケーションを再リンクして、バージョン 8 の任意のリリースまたはバージョン 7 でそのアプリケーションを実行できる。
- Oracle の異なるリリース間での多少の非互換性に対応するため、下位互換性が組み込まれている。
- 移行の後、多くの新機能および拡張機能を自動的に利用できる。こういった新機能の中には、パフォーマンスを向上させるものもある。

### XA コール：リリース 7.1 XA コールとの非互換性

バージョン 8 のデータベースでは、バージョン 7 リリース 7.1.6 の XA コールはサポートしていませんが、リリース 7.2 および 7.3 のコールはサポートしています。したがって、バージョン 7 リリース 7.1 のデータベースをバージョン 8 へ移行した後に、対応する Tuxedo アプリケーション（および XA コールを使用するその他の対応アプリケーション）をバージョン 8 の XA ライブラリに再リンクしてください。

## プリコンパイラおよび OCI アプリケーションのアップグレード

Oracle データベースをバージョン 7 からバージョン 8 へ移行する前に、バージョン 8 のデータベースで使用する予定の OCI とプリコンパイラ・アプリケーションをアップグレードしてください。そうすることによって、本番データベースを移行する前にそれらのアプリケーションをバージョン 8 のサンプル・データベースでテストできます。

これらのアプリケーションをアップグレードするのに必要な作業量は、プログラム・インタフェースとバージョン 8 を利用する程度によって異なります。簡単なものから困難なもの順に、次の選択が可能です。

- 既存のパフォーマンスと機能を維持する
- アプリケーションのパフォーマンスを向上させる
- バージョン 8 のデータベースの新機能を利用する

次の「プリコンパイラ・アプリケーションのアップグレード」および「OCI アプリケーションのアップグレード：制約の使用可能化」の項では、プリコンパイラおよび OCI アプリケーションの移行に必要な特定のステップについて説明しています。

## プリコンパイラ・アプリケーションのアップグレード

既存のプリコンパイラ・アプリケーションをバージョン 8 のデータベースで使用するには、次のステップを実行してください。

1. バージョン 8 で提供される新機能を活用したい場合は、バージョン 7 とバージョン 8 の違いを反映するように、アプリケーションの再コーディングや新規アプリケーションの作成を行います。既存のアプリケーションでバージョン 8 の機能を活用しない場合は、ステップ 1 を省略してステップ 2 へ進んでください。
2. プリコンパイラに付属するバージョン 8 のランタイム・ライブラリ ( `SQLLIB` ) に、アプリケーションを再リンクします。バージョン 8 でアプリケーションを使うときは、少なくともこのステップを実行する必要があります。

---

---

注意：バージョン 8 のデータベースに対して既存のプリコンパイラ・アプリケーションを実行する場合、アプリケーションの再プリコンパイルや再コンパイルは必要ありません。

---

---

## 既存アプリケーションの簡易アップグレード

Pro\*C 2.2 などのバージョン 7 のプリコンパイラで動作するように作成されたアプリケーションは、相互運用性に優れているため、Pro\*C 3.0 などのバージョン 8 のプリコンパイラへ非常に簡単にアップグレードできます。バージョン 7 のプリコンパイラ・クライアントはバージョン 8 のサーバーとともに運用でき、バージョン 8 のプリコンパイラ・クライアントはバージョン 7 のサーバーとともに運用できます。以下に、プリコンパイラとサーバーの互換性について概要を示します。

- バージョン 7 の Pro\*C 2.2 を使用するアプリケーションは、バージョン 8 のサーバーに対しても同様に動作する ( `SQLLIB` 関数のコールはすべて、バージョン 8 のサーバーに対して動作する )。
- バージョン 8 の Pro\*C 3.0 を使用するアプリケーションは、プリコンパイラまたはサーバーのオブジェクト機能を使用しない場合に限り、バージョン 7 のサーバーで動作できる。
- バージョン 7 およびバージョン 8 の `SQLLIB` 関数は、同一アプリケーションおよび同一トランザクション内に混在できる。

既存のバージョン 7 のプリコンパイラ・アプリケーションで使用可能なアップグレード・パスには、次の 3 つの選択肢があります。再コンパイルや再プリコンパイルは必要ありません。

- バージョン 7 のクライアント・アプリケーションに変更を加えずに保存する。アプリケーションは、新しいバージョン 8 のサーバーに対して同様に動作する。

- バージョン8のクライアントへアップグレードするが、アプリケーションは変更しない。バージョン7のクライアントをバージョン8のクライアントにアップグレードするには、新しいバージョンのOCILIBを再リンクするだけでよい。再リンクされたバージョン7のOCIアプリケーションは、バージョン8のサーバーに対して同様に動作する。
- バージョン8へアップグレードしてアプリケーションに変更を加えることにより、バージョン8の新しいプリコンパイラ機能を利用してパフォーマンスや拡張性を向上したり、バージョン8のサーバーのオブジェクト機能を使用したりする。既存の(バージョン7の)アプリケーションをバージョン8のサーバー上で実行するには、バージョン8の新しいSQLLIB コールを使用して再コーディングし、新しいSQLLIB と再リンクする必要がある。バージョン8のサーバーのオブジェクト機能を使用するには、どのアプリケーションであってもこのパスが必要となります。

### バージョン8での Ada サポートの移行

バージョン7 リリース 7.3 での Pro\*ADA 製品は、サポートされません。

Pro\*ADA を、多数の新機能を備えた Ada 8.0.3 用の SQL\*Module にアップグレードしてください。ただし、Ada 8.0.3 用の SQL\*Module は、オブジェクト・サポートを提供していません。

### PL/SQL の下位互換性およびプリコンパイラ

PLSQL\_V2\_COMPATIBILITY 下位互換性動作 ( 6-7 ページの「PLSQL\_V2\_COMPATIBILITY フラグ」を参照) は、次のようにプリコンパイラ・コマンド行オプション DBMS を設定することによってプリコンパイラ環境で使用できます。

```
... DBMS=V7
```

### プリコンパイラの NCHAR/CHAR の移行または互換性に関する問題の可能性

バージョン7での NCHAR 列は、CHAR として格納されています。バージョン8への移行には特別な注意が必要です。

## OCI アプリケーションのアップグレード：制約の使用可能化

OCI ライブラリは、バージョン8のすべてのリリースに付属しています。既存のバージョン7のOCIアプリケーションは、バージョン8のサーバーでも使用できます。また、バージョン7のアプリケーションにある制約は、バージョン8のデータベースで実行するときにも確実に使用可能になります。

それらの制約を適正に使用可能にするには、次のバージョン8遅延モード設定のどれかを使用して、アプリケーションをバージョン8のランタイム OCI ライブラリである OCILIB に再リンクする必要があります。

- 既存の(バージョン7の)パフォーマンス水準を維持するために非遅延モードを使用する。
- アプリケーションのパフォーマンスを向上させるために遅延モードのリンク方式を使用する。

---

---

注意：遅延リンク方式では、バインド操作および定義操作のすぐ後ではバインドおよび定義のエラーがアプリケーションに報告されないことがあります。その代わりに、その後の DESCRIBE または EXECUTE、FETCH コールの際に報告されることもあります。

---

---

関連項目：詳細は、『Oracle コール・インタフェース・プログラマーズ・ガイド』を参照してください。

## OCI アプリケーションのリンク・ライン

OCI アプリケーションの場合、バージョン 8 のリンク・ラインはバージョン 7 のリンク・ラインとは異なっています。バージョン 8 の OCI アプリケーションをコンパイルするときのバージョン 8 のリンク・ラインの使用例を、`$ORACLE_HOME/rdbms/demo/demo_rdbms.mk` ファイルに示しますので参照してください。

## バージョン 6 の OCI ライブラリを使用するアプリケーション

バージョン 6 の OCI ライブラリは、バージョン 8 のデータベースではサポートされていません。したがって、Oracle6 の OCI ライブラリを使用するアプリケーションは、バージョン 8 のデータベースに対しては実行できません。

## LONG から LOB へのアップグレード

バージョン 8 では、LONG から LOB へ直接移行するための機能は提供していません。ただし、LONG から LOB へアップグレードするための次のような間接パスを使うことができます。

1. LONG RAW 内のデータをサーバー側のファイルに書き込む。
2. バージョン 8 の CREATE DIRECTORY コマンドを使って、ファイルが入っているディレクトリを指定する。
3. バージョン 8 のコマンド、OCILobLoadFromFile または DBMS\_LOB.LOADFROMFILE を使って、ファイル内のデータに BLOB を移入する。

LONG RAW が大きすぎなければ、バッファに LONG RAW を読み込んで OCILobWrite または DBMS\_LOB.WRITE をコールし、LONG RAW データを BLOB に書き込むという別の方法もあります。

## バージョン 7 の Forms または Developer/2000 アプリケーションのアップグレード

Forms アプリケーションは、バージョン 7 でもバージョン 8 でも同じように稼働します。ただし、『Oracle8 と Oracle8 Enterprise Edition の解説』で説明する新機能を確認して、バージョン 8 の新機能がアプリケーションに役立つか、またはアプリケーションに逆効果になるかを判断してください。バージョン 8 の機能が Forms アプリケーションとどのように相互作用するかについては、『Oracle Forms リファレンス・マニュアル Vol.1 および Vol.2』および『Oracle Forms 開発者ガイド』、『Oracle Forms 上級テクニク 4.5』で説明しています。

### データ・ディクショナリ・ビューの更新

CATALOG5.SQL ファイルおよび CATALOG6.SQL ファイルで作成され、Oracle バージョン 5 およびバージョン 6 に対する下位互換性のためにバージョン 7 で保持されていた特定のデータ・ディクショナリ・ビューは、バージョン 8 では廃止されています。これらのデータ・ディクショナリ・ビューに対する参照元は、データベース・ツールとアプリケーションから削除してください。

### SQL\*Plus スクリプトのアップグレード

SQL\*Plus リリース 4.0 およびバージョン 8、PL/SQL バージョン 3 の機能を使うときは、次のステップを実行してください。

- SQL\*Plus リリース 3.x のスクリプトに次のような変更を加えて、SQL\*Plus リリース 4.0 のスクリプトに変換する。
  - スクリプトに SET COMPATIBILITY V7 という行がある場合は、それを SET COMPATIBILITY V8 に変更する。またはこの行を削除して、バージョン 8 のデフォルト設定を使用する。
  - LOGIN.SQL ファイルをチェックし、SET COMPATIBILITY V7 という行があれば SET COMPATIBILITY V8 に変更する。
  - SQL\*Plus リリース 4.0 の新機能を理解するためには、『SQL\*Plus ユーザーズ・ガイド および リファレンス』を参照すること。
- バージョン 8 の新機能を使うには、既存の SQL スクリプトを変更してバージョン 8 の構文に適合させる。既存の SQL スクリプトは、バージョン 8 の新機能を使わなければ、変更を加えなくてもバージョン 8 でも実行できる。

関連項目: SQL スクリプトのアップグレード方法の詳細は、『Oracle8 Server SQL リファレンス』を参照してください。

---

注意: PL/SQL プロシージャについては、変更する必要はありません。

---

## PL/SQL V2 互換モード

PL/SQL V2 の互換モードは、PL/SQL V8.0.3 で使用できます。このモードは、PLSQL\_V2\_COMPATIBILITY フラグによって使用可能になります。

このフラグを設定するには、次のような 3 つの方法があります。

- INIT.ORA ファイルに次の行を追加する。

```
PLSQL_V2_COMPATIBILITY=TRUE
```

- 次のコマンドを発行する。

```
ALTER SYSTEM SET PLSQL_V2_COMPATIBILITY = TRUE;
```

- 次のコマンドを発行する。

```
ALTER SESSION SET PLSQL_V2_COMPATIBILITY = TRUE;
```

## PLSQL\_V2\_COMPATIBILITY フラグ

PLSQL\_V2\_COMPATIBILITY フラグにより、次のような状況で PL/SQL V8.0.3 と PL/SQL V2 の間に互換性が得られる。

- PL/SQL V2 コンパイラを使用すると、ソース内での定義を行う前にソース内でレコード型や索引表の型を参照できます。PL/SQL の V8.0.3 では、ソース内で型を参照する前に必ず型の定義を行っておく必要があります。ただし、PL/SQL V2 の互換モードを使用可能にすると、PL/SQL V8.0.3 は型の定義に関しては PL/SQL V2 のように動作します。
- PL/SQL V2 コンパイラは、次のような不正な構文を許可してしまいます。

```
return variable-expression
```

この構文は誤っており、次の形式に変更する必要があります。

```
return variable-type
```

こうした不正な構文が見つかったら、PL/SQL V8.0.3 コンパイラはエラーを表示します。ただし、PL/SQL V2 の互換モードを使用可能にすると、PL/SQL V8.0.3 は PL/SQL V2 と同様に動作し、エラーは表示しません。

- PL/SQL V2 では、IN パラメータとして渡された索引表の要素を変更 / 削除できます。次にその例を示します。

```
function foo (x IN table_t) is
begin
x.delete(2);
end;
```

このINパラメータの使用方法は誤っています。PL/SQL V8.0.3では、INパラメータの読み取り専用方法を正しく施行し、INパラメータとして渡された索引表を索引表方式から変更することはありません。ただし、PL/SQL V2の互換モードを使用可能にすると、PL/SQL V8.0.3はPL/SQL V2と同様に動作し、そのパラメータを認めます。

- PL/SQL V2では、レコードであるINパラメータのフィールドを(OUTパラメータとして)引き渡すことができますが、PL/SQL V8.0.3では、そのような引渡しは許可しません。ただし、PL/SQL V2の互換モードを使用可能にすると、PL/SQL V8.0.3はPL/SQL V2と同様に動作し、この種の引渡しを許可します。

- PL/SQL V2コンパイラでは、レコード変数であるOUTパラメータのフィールドが式のコンテキスト中で使用されるのを許可してしまいます(たとえば、割当て文の右側にあるドット修飾名など)。

このようなOUTパラメータの使用は許可すべきではありません。PL/SQL V8.0.3では、OUTパラメータを式のコンテキスト中で使用することは許可していません。ただし、PL/SQL V2の互換モードを使用可能にすると、PL/SQL V8.0.3はPL/SQL V2と同様に動作します。

- PL/SQL V2では、SELECTリストのFROM句の中でOUTパラメータを使用できます。PL/SQL V8.0.3では、OUTパラメータのそのような使用法は許可されません。ただし、PL/SQL V2の互換モードを使用可能にすると、PL/SQL V8.0.3はPL/SQL V2と同様に動作します。

## バージョン7とバージョン8でのキーワード動作の違い

バージョン7とバージョン8の両方に含まれるさまざまなキーワードや型は、SELECTリスト内のファンクション名として使用されたとき、バージョン7とバージョン8では次のようにわずかに異なるエラー・メッセージ識別子を出力します。

キーワード	バージョン8の動作	バージョン7の動作
CHARACTER, COMMIT, DEC, FALSE, INT, NUMERIC, REAL, SAVEPOINT, TRUE	エラーを生成 : ORA-06550 および PLS-00222	エラーを生成 : ORA-06552 および PLS-222

## バージョン7とバージョン8での新しいキーワードまたは型の動作の違い

次に示す複数のワードは、キーワードまたは型としてバージョン8で新規に定義されました。

BFILE	CLOB	NCLOB	SYS_OP_NTCIMG
BLOB	DEREF	NVARCHAR2	VALUE
CAST	NCHAR	REF	

バージョン 8 では、それぞれのワードは SELECT リスト内のファンクション名として使用できません。ただし、`schema.function` のようなスキーマで修飾されている場合に限りです。次にその例を示します。

```
select scott.true() ...
```

バージョン 8 では、スキーマ修飾がない場合はエラーを生成しますが、バージョン 7 では、未修飾で使うとエラーを生成せずに機能します。

## SQL\*Net または Net8

バージョン 7 およびバージョン 8 のリリースでは、SQL\*Net のバージョン 2 または Net8 を使用できます。SQL\*Net バージョン 1 は、さまざまなネットワーク・アドレス指定スキームを使用していましたが、バージョン 8 では使用できません。したがって、移行したアプリケーションには、次の要件が適用されます。

- クライアントおよびサーバーの両方で、SQL\*Net バージョン 2 または Net8 を使用していなければならない。
- マルチスレッド・サーバーの場合、サーバー側で SQL\*Net バージョン 2 または Net8 が必要である。したがって、マルチスレッド・サーバーを使用してクライアントと接続するには、クライアント側でも SQL\*Net バージョン 2 または Net8 を使用しなければならない。

## SQL\*Net バージョン 1 から SQL\*Net バージョン 2 へのアップグレード

SQL\*Net バージョン 1 からバージョン 2 にアップグレードするには、次の変更を行います。

- SQL\*Net バージョン 2 または Net8 をインストールする。
- 各接続文字列を次のバージョンの接続記述子として再作成する。SQL\*Net バージョン 2 は、『SQL\*Net 管理者ガイド、バージョン 2.0』または『Oracle Net8 管理者ガイド』で概説している構文を使用する。
- SQL\*Plus や SQL\*Forms などを含む、SQL\*Net バージョン 2 で使用するプリコンパイラ・プログラムおよび Oracle 実行可能ファイルを再リンクする。

バージョン 1 からバージョン 2 への SQL\*Net のアップグレードの詳細は、『SQL\*Net 管理者ガイド、バージョン 2.0』および『SQL\*Net 移行ガイド、バージョン 2』を参照してください。

## バージョン 7 の Net2 クライアントおよび接続マネージャ

既存のバージョン 7 の OCI-Net2 クライアントは、再リンクせずに接続マネージャを利用できます。唯一の条件は、クライアントの接続文字列が CMAN を介して進行しなければならないことです。

シン・クライアント JDBC はバージョン 7 の Net2 クライアントと似ており、再リンクをしないすべての現行クライアントです。

バージョン 8 のその他の利点についても、既存のクライアントは再リンクせずに使用できませんが、そうした機能に含まれる例には、データベース・リンクの集中化、SQL 文メモリの縮小、共有プール消費量全体の縮小、SQL および PL/SQL 用逐次再使用可能 SQL、スケラブル・カーソルの認可、PL/SQL 定数プール・ページング、より効率的な SQL\*Plus-PL/SQL 対話、解析並行性の改善、高速配列挿入、高速表走査、PL/SQL のバルク SQL、高速キャラクタ・セット変換、高速マルチバイト処理などがあります。

## 再リンクされたバージョン 7 のクライアントで使用可能な Net8 機能

Net8 で再リンクするバージョン 7 の OCI/UIPI クライアントは、接続プーリングおよび OSS 認証を使用できます。この機能を得るのに、コード変更は不要です。同じことが、再リンクするプリコンパイラおよびアプリケーション・ツールにも適用されます。

## バージョン 8 の Net8 クライアントの場合

バージョン 8 の OCI を使用するように再コーディングされたアプリケーションは、次に示すようなバージョン 8 の全機能をインプリメントできます。

- アプリケーション・フェイルオーバー
- 透過データのプリフェッチ
- ネットワークの往復とサーバーのフットプリントを減らす、ピギーバック方式のコミットおよび取消し。
- 1 往復での完全動的文の実行 ( SQL\*Plus で select \* from emp など )
- NLS\_LANG の設定時に 3 往復分を排除する高速ログイン
- 新しいオブジェクト表 API、オブジェクト・キャッシュ
- セキュリティ・ツールキット API ( 暗号化、署名 )
- 警告 ( パスワードの期限切れ、フェイルオーバー )
- クライアント側の型変換によるサーバーの拡張性
- NCHAR
- LOB
- より長い CHAR/VARCHAR
- DML 戻り値
- 中間層インプリメンテーションを単純化するための、サーバー、ユーザー、トランザクションのハンドルの分離

---

---

注意：バージョン 8 の OCI に完全に変換されたクライアント・プログラムは、SQL\*Plus だけです。

---

---

バージョン7のクライアントは、バージョン7とバージョン8のコールを混在させながらバージョン8のOCIを選択して使用することもできます。追加機能は、使用するコールによって異なります。SQL\*Loader やエクスポート/インポートなどのユーティリティ・プログラムは、主にオブジェクト表にアクセスするためにコールのサブセットを 変換します。暗号化 API およびパスワード・リセット・コールは、単独で使用することも できます。フェイルオーバーまたはプリフェッチ、ピギーバック方式のコミット/取消し、クライアント側での変換を使用可能にするには、処理する文のすべてのフェーズにバージョン8のOCIを使用して ください。

## バックアップ管理 : EBU および Recovery Manager

EBU および Recovery Manager は、Oracle データベースのバックアップを管理するためのクライアント側のユーティリティです。ただし、バージョン8のデータベースのバックアップを管理するには、Recovery Manager を使用する必要があります。EBU は、バージョン8では使用できません。

EBU と Recovery Manager は両方とも、Legato や EMC などのサード・パーティの記憶サブシステムと対話するためにメディア管理言語 ( MML ) を使います。

EBU とバージョン7用のテープ・サブシステム管理モジュールに対して行った投資は、Recovery Manager とバージョン8でも再利用できます。ただし、バックアップ・ボリューム形式は再利用できません。Recovery Manager は異なる形式を作成し、バージョン7のバックアップはバージョン8での復元には一般的に役に立たないので、バージョン8の下で記憶サブシステムに新しいバックアップを書き込む必要があります。

---

---

注意 : Recovery Manager のスクリプト記述言語は、EBU のスクリプト記述言語とはまったく異なっています。

---

---

## ディクショナリ保護

相対表領域データ・ブロック・アドレス ( DBA ) を次のようなアプリケーションで使うと、問題が発生する可能性があります。

- SYS スキーマでユーザー表を作成する。
- 'ANY' 権限を使用してユーザー表にアクセスする。

SYS スキーマでユーザー表を作成し、それらにアクセスすることは安全性の面で保証できません。したがって、アプリケーション側ではオブジェクトを別のスキーマに移動することが要求されます。一時的な互換性を確保するために O7\_DICTIONARY\_ACCESSIBILITY というスイッチを使いますが、このスイッチは暫定的に使うだけです。

アプリケーションから SYSDBA オプションなしでユーザー SYS に接続することは避けるべきです。ユーザー SYS に接続してパスワードを共有するのではなく、一般ユーザーに SYSDBA 権限を付与し、ユーザーが SYSDBA としてデータベース接続して SYS スキーマに接続するようにしたほうがよいでしょう。SYSDBA 権限の詳細は、『Oracle7 Server 管理者ガイド』を参照してください。

場合によっては、'ANY' 権限を持つユーザーがディクショナリに対して DML を使用します。たとえば、DELETE ANY TABLE 権限を持つユーザーは、delete 文を使用して aud\$ 表内の監査レコードを除去します。ユーザーは、そのような場合に、特定の作業を行えるようにオブジェクトに対する権限をユーザーに付与できます。

理想は、ディクショナリ内にあるユーザー SYS のパスワード列を NULL に更新するための移行スクリプトが利用できるようになることです。

## パスワード管理

バージョン7以前のアプリケーションをバージョン8のパスワード管理に対応させるには、アプリケーションで次の変更を行う必要があります。

- バージョン8のOCIコールである OCI\_SessionBegin() を使用してサーバーに接続する。サーバーから SUCCESS\_WITH\_INFO が戻された場合、パスワード期限切れエラーかどうかチェックする。パスワードの期限が切れていても、まだ猶予期間内であれば、ユーザーに警告を出し、OCI\_Change\_Password() コールを使ってユーザーのパスワードを変更する（パスワード変更コールはオプションだが使用することを薦める）。
- パスワードの期限が切れていてエラーが戻された場合は、バージョン8のOCIコールである OCI\_Change\_Password() を使ってユーザーのパスワードを変更しなければならない。

バージョン7のアプリケーションで上記の変更を行わなかった場合、ユーザー・アカウントの期限が切れた後にパスワードを変更するには、SQL\*Plus や SQL Server Manager といった Oracle ツールが必要になります。

ただし、このバージョン8のパスワード管理機能は、デフォルトでオフになっています。バージョン8のサーバー・システムでパスワード期限切れ機能をインプリメントしていない場合は、バージョン7のクライアント側でパスワード管理について変更を行う必要はありません。DEFAULT プロファイルでは、すべてのパラメータが UNLIMITED に設定されており、パスワード複雑度チェック・ルーチンは NULL に設定されています。

パスワード検証ルーチンは、そのプロファイル定義とともにエクスポートまたはインポートされます。バージョン8リリース 8.0.2 以降では、ユーザーの履歴表もエクスポートまたはインポートできます。

## バージョン 7 またはそれ以前のクライアントとバージョン 8 のサーバー

バージョン 7 のクライアントは、バージョン 7 の OCI コールを使用してサーバーに接続するため、バージョン 8 でのパスワードの期限切れを検出できません。ただし、バージョン 8 のその他のパスワード管理機能は、バージョン 7 のクライアントでも動作します。パスワード満了の処理も含めたバージョン 8 のパスワード管理の全機能は、バージョン 7 のログオン・コールをバージョン 8 のログオン・コールに置き換えるだけで、バージョン 7 のクライアントでも動作します。

## バージョン 8 のクライアントとバージョン 7 またはそれ以前のサーバー

バージョン 8 のクライアントは、バージョン 7 以前のサーバーと動作するようにコーディングできます。バージョン 8 のクライアント用のコーディング例を、以下に示します。

```
OCISessionBegin(...) /* call v8 logon OCI call */
if (SUCCESS_WITH_INFO) then
{ /* Check for password expiration and take appropriate action*/
...
OCIChangePassword(...);
...
}
```

## エクスポート/インポートの使用法、パーティション・オブジェクト

バージョン 8 のエクスポート・ダンプ・ファイル形式は、バージョン 7 の形式とは異なります。したがって、バージョン 8 のエクスポートによって生成されたダンプ・ファイルは、バージョン 7 のインポートを使用してバージョン 7 のデータベースにインポートすることはできません。ただし、CATEXP7.SQL スクリプトの実行後にバージョン 7 のエクスポートを使用して、バージョン 8 のデータベースをバージョン 7 にエクスポートすることはできません。バージョン 7 のエクスポートによってパーティション表をエクスポートすることはできません。パーティション表をバージョン 7 のデータベースに移動する必要がある場合は、最初に非パーティション表に再編成しなければなりません。

## OCI のスレッド・セーフティについての移行および互換性の問題

ORLON コールおよび OLON コールは、バージョン 8 ではサポートされてはいません。しかし、単一スレッド・アプリケーションの場合でも、OLOG を使用してください。

---



---

注意：マルチスレッド・アプリケーションの場合は OLOG コールが必要になります。

---



---

関連項目：スレッド・セーフティおよび OCI の詳細は、『Oracle コール・インタフェース・プログラマーズ・ガイド』を参照してください。

## スタンバイ・データベースのアップグレードおよび互換性の問題

スタンバイ・データベースは、バージョン7 リリース 7.3 またはバージョン8 だけで稼働します。プライマリ・データベースとスタンバイ・データベースは、同じバージョンおよび同じリリース番号のOracle Serverで稼働しなければなりません。また、いずれも同じリリースのオペレーティング・システム・プラットフォームで稼働していなければなりません。

関連項目：プラットフォーム固有の Oracle マニュアルを参照してください。

スタンバイ・データベースを正しく使うには、次のことを考慮してください。

- プライマリ・データベースとスタンバイ・データベースのデータ・ファイル、ログ・ファイル、制御ファイルは、個別の物理媒体に別々に存在していなければならない。1 つの制御ファイルをプライマリ・データベースとスタンバイ・データベースの両方で利用することはできない。
- 必須作業ではないが、プライマリ・データベースとスタンバイ・データベースの両方において、次の事項を同じにすると、管理作業が容易になる。
  - データベース識別文字列
  - データ・ファイル名
  - ログ・ファイル名
  - 制御ファイル名

スタンバイ・データベースをバージョン8に移行するには、次の各ステップを実行してください。

1. バージョン7で作成した REDO ログをすべて適用する。
2. プライマリ・データベースがバージョン8の下で正常にオープンすることを確認する。
3. スタンバイ・サイトにバージョン8のソフトウェアをインストールする。
4. バージョン8の制御ファイルとデータファイル1をスタンバイ・サイトにコピーする。
5. プライマリ・データベースからスタンバイ・データベース用の新しい制御ファイルを作成する。

関連項目：スタンバイ・データベースの詳細は、『Oracle8 Server SQL リファレンス』および『Oracle8 Server管理者ガイド』、『Oracle8 Parallel Server 概要および管理』を参照してください。

## エクスポート/インポートの互換性の問題

ダイレクト・パス・エクスポートは、従来型パス・エクスポートとは異なるコード化形式を使用します。したがって、生成されるエクスポート・ダンプ・ファイルも、ダイレクト・パス・エクスポートと従来型パス・エクスポートでは異なります。

ダイレクト・パス・エクスポートまたは従来型パス・エクスポートで生成されたダンプ・ファイルは、両方ともバージョン 8 のインポート・ユーティリティで使用できます。

バージョン 8 より前のダンプ・ファイルは、バージョン 8 のインポート・ユーティリティと上位互換性を持ちます。ダイレクト・パス・エクスポートの現行または以前のバージョンおよびリリース、あるいは従来型パス・エクスポートによって生成されるダンプ・ファイル・データは、Oracle Server の将来のバージョンおよびリリースとの上位互換性を持ちます。

### 下位互換性の技術および制約

バージョン 8 のエクスポート・ユーティリティでは、バージョン 8 より前のインポート・ユーティリティと下位互換性のないダンプ・ファイルが作成されます。エクスポートしたデータは、バージョン 8 以前のインポート・ユーティリティではインポートできません。

バージョン 8 のデータをバージョン 7 (またはそれ以前) のデータベースにエクスポートするには、CATEXP7.SQL スクリプトの実行後にバージョン 7 のエクスポート・ユーティリティを使用しなければなりません。

## NCHAR および NLS の使用

バージョン 8 において DBA は、特定の列および属性、PL/SQL 変数、パラメータ、戻り値に各国文字 (NCHAR) を使用することを宣言できます。そのような明示的な宣言がない限り、NCHAR および NLS の使用はほとんどが透過的に処理され、バージョン 8 の他の機能には影響しません。例外としては、props\$ または VALUES ディクショナリ・ビューにある SELECT 文で、CHARACTER\_SET\_NAME 列または NLS\_NCHAR\_CHARACTERSET 行を戻す場合があります。

### 移行および NCHAR と NLS

props\$ ディクショナリ表には、CREATE DATABASE 文で指定されたキャラクタ・セットを記述する 2 つの行があります。NAME='NLS\_CHARACTERSET' を含む行では、VALUES 内にデータベース・キャラクタ・セットの名前があります。NAME='NLS\_NCHAR\_CHARACTERSET' を含む行では、VALUES 内に各国文字セットの名前があります。

NLS\_DATABASE\_PARAMETERS には、CREATE DATABASE 文で指定された NCHAR キャラクタ・セット名を含む新しい行が入っています。新しいパラメータ、NLS\_NCHAR\_CHARACTERSET があります。

リリース 7.3 と比較すると、さまざまなビューに、次の値を持つ新規列 CHARACTER\_SET\_NAME が含まれます。

```
DECODE (x$.CHARSETFORM,  
        1, 'CHAR_CS',  
        2, 'NCHAR_CS',
```

ここで、x\$ は実表の 1 つを表します。ビューの DATA\_TYPE または COLTYPE 列値は、キャラクタ・セットの選択項目を表示するようには変更されません。

## NCHAR と NLS の互換性および相互運用性

リリース7.1以降のクライアントは、データを各国文字キャラクタ・セットで保持するバージョン8のサーバーとやりとりできます。出力された各国文字データがバインド・ハンドルまたは定義ハンドルを通る場合、クライアントのデータベース・キャラクタ・セットへの変換はOCIによって透過的に処理されます。データが入力バインド値として必要で、各国文字セットの文字列だけが許可されている場所で使用する場合、その値を使用するSQLまたはPL/SQLコードでバインド変数（データベース・キャラクタ・セットが使用されていると見なされる）を使用している箇所をCSCONVERT()へのコールで囲み、各国文字セットに変換する必要があります。この場合、クライアントは、クライアントのデータベース・キャラクタ・セットでしかデータを渡せません。クライアントのデータベース・キャラクタ・セットには、各国文字セットの全種類が揃っていない可能性があります。

バージョン6やバージョン7のクライアントは、ANY\_CSパラメータを使用して、サブプログラムに対するリモート・プロシージャ・コール（RPC）を行えます。ANY\_CSパラメータは、その実引数にサーバーのデータベース・キャラクタ・セットが使用されているものと解釈します。バージョン6やバージョン7のクライアントは、各国文字キャラクタ・セットの使用を宣言しているパラメータを持つサブプログラムに対してはRPCを行えません。また、戻り値に各国文字キャラクタ・セットが使用されている関数に対してもRPCを行えません。このような禁止事項の違反は、コンパイル時ではなく実行時に検出されます。

PL/SQLに埋め込まれたSQLコードを使用しているバージョン6またはバージョン7のクライアントは、SQL文でCSCONVERT()関数または2番目の引数を持つCHR()を直接使用することができません。これは、バージョン6やバージョン7のクライアントのPL/SQLパッケージであるSTANDARDに、これらの関数が含まれていないためです。この問題に対処するために、サーバー上に次のような2つのストアド・プロシージャを作成することができます。

```
create function to_nchar_cs(t varchar2) return nchar varying
is begin
    return cscnvert(t, nchar_cs);
end;
create function to_char_cs(t nchar varying) return varchar2
is begin
    return cscnvert(t, char_cs);
end;
```

---

## バージョン 8 の ROWID に関する移行の問題

バージョン 8 の ROWID には、バージョン 8 の 2 つの新機能であるパーティション表および相対表領域データ・ブロック・アドレス (DBA) を使用可能にする新しい内部形式および外部形式が含まれています。

関連項目: 詳細は、『Oracle8 Serverアプリケーション開発者ガイド』および『Oracle8 Server 概要』を参照してください。

この章では、新しいバージョン 8 の ROWID に関連する次のような移行の問題について説明します。

- アプリケーションおよびデータの移行
- DBMS\_ROWID パッケージ
- 変換プロシージャの例
- スナップショット・リフレッシュ
- バージョン 8 より前のクライアントとの互換性の問題
- ROWID 関連の移行に関する質問・回答

## アプリケーションおよびデータの移行

ROWIDはROWID型および文字型の列に格納できます。ただし、格納されていたバージョン7のROWIDは、バージョン7からバージョン8へのデータベースの移行後には無効になります。したがって、バージョン7のROWIDはバージョン8に変換する必要があります。

### アプリケーション

ROWIDのアセンブルおよび逆アセンブルを手動で行わないアプリケーションは、変更や再コンパイルの必要がありません。というのは、新しいROWIDがホスト変数についての現行の格納要件を満たしているからです。

ROWIDの作成やその内容分析を試みるアプリケーションでは、バージョン8に付属の新規パッケージDBMS\_ROWIDを使用して、新しいバージョン8のROWIDの形式や内容进行操作する必要があります。DBMS\_ROWIDには、バージョン7のROWIDから直接使用できる情報（ファイルおよびブロック・アドレスを含む）を抽出する関数、およびデータ・オブジェクト番号が含まれます。

### データ

ROWID値（ROWID形式または文字形式）を含む列は、値が参照する表をバージョン8へ移行する場合と一緒に移行する必要があります。そうしないと、格納された値を使って行を取り出すことができなくなります。一方、バージョン8の表に格納されたROWID値がバージョン8より前の表を参照している場合、列の移行は必要ありません。

列の移行には、定義の移行とデータの移行の2つの段階があります。

列の定義は、バージョン7からバージョン8へのディクショナリの移行時に自動的に調整されます。ROWIDユーザー列の最大サイズは拡張ディスクROWIDのサイズにまで増やされ、ROWID列用のCOL\$のLENGTH列を6バイトから10バイトに変更します。

データの移行は、システムがバージョン8で立ち上がった後でないと実行できません。ユーザーは、異なる表を別々に移行したり、複数の表を並行して一度に移行したりできます。ただし、移行は、バージョン7のデータベース・ファイル制限を超える前に行う必要があります。これは、あいまいなブロック・アドレスが作成される場合があるためです。

ユーザーのインストレーション環境で利用できる既存のROWIDリフレッシュ手順またはバージョン8のDBMS\_ROWID機能を使用して、格納済みのROWIDをバージョン7形式からバージョン8形式に移行できます。

この移行ユーティリティによるデータの移行は、ユーザー定義列に格納されたROWIDにだけ適用されます。システムによって格納されたROWID（索引内のROWIDなど）はすべて、移行ユーティリティによる移行の後でも有効であるため、移行のための特別な操作は必要ありません。

---

---

注意：ROWID が含まれる列をインポートすると、ROWID の妥当性を再設定するために特別な注意が必要になることを示す警告メッセージが出力されます。そのような特別な注意は、インポートするすべての ROWID について必要になります。したがって、エクスポート/インポートによるバージョン 8 への移行には、ユーザー定義列だけでなく ROWID が含まれるすべての列について特別な注意が必要です。

---

---

## DBMS\_ROWID パッケージ

バージョン 8 では、DBMS\_ROWID PL/SQL パッケージを提供しています。このパッケージには、次の機能が含まれます。

- バージョン 7 形式およびバージョン 8 形式での ROWID の作成。
- バージョン 7 の ROWID およびバージョン 8 の ROWID の解釈。
- バージョン 7 の ROWID およびバージョン 8 の ROWID 間での変換。

格納済み ROWID の移行は、次の節で説明するように変換関数を使用して実行できます。

### ROWID 変換型

ユーザーは、変換する ROWID の型を指定しなければなりません。ROWID が ROWID 型のユーザー列に格納されているのか (VAR)CHAR 型のユーザー列に格納されているのかによって、ROWID 変換関数による変換の実行方法が異なるからです。

ROWID 型の列の場合、変換プロシージャの呼出し元は、プロシージャ・パラメータとして次の値を渡す必要があります。

```
rowid_convert_internal constant integer := 0;
```

(VAR)CHAR 型の列の場合、変換プロシージャの呼出し元は、プロシージャ・パラメータとして次の値を渡す必要があります。

```
rowid_convert_external constant integer := 1;
```

### ROWID 変換関数

次の関数は、ROWID の変換を行います。

- ROWID\_TO\_EXTENDED は、ROWID をバージョン 7 (制限) 形式からバージョン 8 (拡張) 形式に変換する。
- ROWID\_TO\_RESTRICTED は、ROWID をバージョン 8 (拡張) 形式からバージョン 7 (制限) 形式に変換する。
- ROWID\_VERIFY は、指定の ROWID をバージョン 7 形式からバージョン 8 形式に変換できるかどうかをチェックする。

次に、ROWID\_TO\_EXTENDED プロシージャおよび ROWID\_VERIFY プロシージャについて詳しく説明します。

#### ROWID\_TO\_EXTENDED 変換プロシージャ

ROWID\_TO\_EXTENDED は、次のパラメータを使います。

- 変換する ROWID (外部文字形式)
- スキーマ名 - ROWID を拡張形式に変換する行を含む表の所有者 (スキーマ) 名
- 表名 - ROWID を拡張形式に変換する行を含む表の名前
- 変換型 - 変換する ROWID の型。

関連項目: 詳細は、7-3 ページの「ROWID 変換型」を参照してください。

ROWID\_TO\_EXTENDED は、バージョン 8 (拡張) の ROWID を外部文字形式で戻します。そのパラメータには、次のような意味があります。

- ターゲット表のスキーマ名および表名が指定されていない場合 (NULL) ROWID\_TO\_EXTENDED は、変換する ROWID によって指定されたページを取り出そうと試みる。ROWID\_TO\_EXTENDED は、この ROWID に格納されたファイル番号を絶対ファイル番号として扱うが、移行前にファイルが削除され、その番号が再利用されている場合には、問題が発生する可能性がある。取り出されたページが有効な表に属している場合、ROWID はその表のデータ・オブジェクト ID を使用して拡張形式に変換される。ただし、これは非常に非効率的な変換方法であり、ターゲット表がわからない場合など、最終手段としてだけ使用することをお勧めする。ユーザーは、変換された値を使用するときに、正しい表名を知っていなければならない。
- スキーマ名および表名が指定されている場合 (好ましいアプローチ) ROWID\_TO\_EXTENDED は表に対する SELECT 権限を検証し、この表のデータ・オブジェクト番号を使用して ROWID を拡張形式に変換する。変換された ROWID がこの表内の実際の行を参照するかどうかは、ROWID の変換時にも ROWID の使用時にも保証されない。
- ROWID に NULL 値が指定されている場合、プロシージャによって表の指定が無視され、NULL 値が戻される。
- ROWID に '0' の値または、より一般的な '<n>0.<m>0.<p>0' という値が指定されている場合、表名は無視され、'00000000.0000.0000' という形式の制限 ROWID が戻される。
- バージョン 8 の ROWID が指定されている場合は、ROWID 内のデータ・オブジェクトが実際のデータ・オブジェクト番号 (表名の指定によって決まる) に対して検証される。これら 2つの数字が一致しない場合は、"INVALID ROWID" エラーが発生する。一致した場合は、元の ROWID が戻される。

#### ROWID\_VERIFY

バージョン 8 では、ROWID\_VERIFY という ROWID 検証プロシージャが提供されています。このプロシージャは ROWID\_TO\_EXTENDED と同じパラメータを使用し、ROWID を拡張形式に正常に変換できる場合は 0 を返し、変換できない場合は 1 を戻します。

ただし、ユーザーが基礎表のSELECT権限を持っていない場合や、表が存在しない場合には、ROWID\_VERIFY はセキュリティ違反エラーまたは「オブジェクトが見つかりません」というエラーを戻します。ROWID\_VERIFY は不正な ROWID をすべて識別できるため、ROWID\_TO\_EXTENDED プロシージャを使用して移行する前に不正な ROWID を識別します。

## 変換プロセスの例

以下に、変換プロセスの例を示します。

### 例 1

表 SCOTT.T には、ROWID 形式の列 C が入っていると仮定します。これらの ROWID はすべて単一の表、SCOTT.T1 を参照します。

列 C の値は、次の文を使用して拡張形式に変換できます。

```
UPDATE SCOTT.T SET C = DBMS_ROWID.ROWID_TO_EXTENDED(C, 'SCOTT', 'T1', 0);
```

### 例 2

より一般的な状況では、列 C に格納された ROWID は別のさまざまな表を参照する可能性があります。表名は同じ行内にある他の列の値を基に見つけられます。たとえば、表 T の列 TNAME に、列 C の ROWID によって参照される表の名前が入っているとします。

この場合、列 C の値は、次の文を使用して拡張形式に変換できます。

```
UPDATE SCOTT.T SET C = DBMS_ROWID.ROWID_TO_EXTENDED(C, 'SCOTT', TNAME, 0);
```

### 例 3

ROWID\_TO\_EXTENDED関数は、CREATE AS SELECT文の中で使うことができます。たとえば、変換によってROWID型のユーザー列のサイズが増え（特定のポートによって異なりますが、通常は6～10バイト増えます）それによって間接行が作成されることもあるので、場合によってはROWID\_TO\_EXTENDED関数を使用すべきです。

この場合、UPDATE よりも CREATE AS SELECT を選択した方がよいでしょう。

```
CREATE TABLE SCOTT.TNEW (A, B, C) AS SELECT A, B, DBMS_ROWID.ROWID_TO_EXTENDED(C, 'SCOTT', 'T1', 0) FROM SCOTT.T;
```

### 例 4

列 C に格納された ROWID のターゲット表が不明な場合、変換は次の文を使って実行できます。

```
UPDATE SCOTT.T SET C = DBMS_ROWID.ROWID_TO_EXTENDED(C, NULL, NULL, 0);
```

### 例 5

次の SQL 文を使用して、変換の前にいくつかの不正な ROWID を検索できます。

```
SELECT ROWID, C FROM SCOTT.T WHERE DBMS_ROWID.ROWID_VERIFY(C, NULL, NULL, 0) = 1;
```

## スナップショット・リフレッシュ

バージョン8のROWID形式では、マスター・サイトおよびスナップショット・サイトの両方をバージョン8へアップグレードするとき、すべてのROWIDスナップショットの完全リフレッシュを実行しなければなりません。

関連項目：詳細は、『Oracle8 Server レプリケーション』の付録B「移行および互換性」を参照してください。

## バージョン8より前のクライアントとの互換性の問題

バージョン8より前のクライアントはバージョン8のデータベースにアクセスできますし、バージョン8のクライアントもバージョン8より前のデータベースにアクセスできます。バージョン8より前のデータベースからバージョン8のデータベースへ戻される疑似列ROWIDおよびROWID型の列のバイナリ値と文字値は、常にバージョン7の（制限付きの）形式となります。これは、バージョン8より前のシステムでは拡張形式ROWIDを認識できないためです。

バージョン8に付属のDBMS\_ROWIDパッケージは、バージョン7のROWIDの内容解釈およびバージョン7形式でのROWIDの作成に使用できます。

バージョン8より前のクライアントがバージョン8のデータベースにアクセスすると、ROWIDをバージョン8の拡張形式で受け取ります。したがって、バージョン8より前のクライアントは、バージョン8のサーバーが戻すROWIDの内容を解釈できません。

バージョン8のスナップショットの互換性は、バージョン7リリース7.1.4以降に制限されます。さらに、マスター・サイトがアップグレードされた場合は、バージョン8のアップグレード・スクリプトによってログを無効にする必要があります。これにより、スナップショットでは、再度高速リフレッシュが可能になる前に完全リフレッシュが強制的に実行されます。

## ROWID 関連の移行に関する質問・回答

Q: バージョン8には、バージョン7のインポート・クライアントに対する制限はありますか？

A: バージョン8の表内に拡張ROWID値が入っている行が1行でもあれば、バージョン7のクライアントがROWIDユーザー列のあるバージョン8の表をインポートすることはできません。

Q: Forms3（およびForms4）では、実表の更新時に新しいROWID形式は認識されますか？

A: バージョン8のデータベースにアクセスしようとするフォーム・アプリケーションは、パッチ #380655を使用して再リンクしなければなりません。このパッチについての詳細は、Oracle8 リリース 8.0.4 の README ファイルで説明しています。

Q: バージョン8のROWIDを変更した場合、PRO\* プリコンパイル済みプログラムに影響はありますか？

A: ROWIDは使用しても、その形式に依存しないプログラムには影響がありません。バージョン7のROWID形式に依存するプログラムは、新しいパッケージ DBMS\_ROWID を使用するように変更しなければなりません。

Q: 「 WHERE CURRENT of CURSOR 」命令はまだ動作しますか？

A: 動作します。バージョン8より前のクライアントからバージョン8のサーバーにアクセスする場合や、バージョン8のクライアントからバージョン8より前のサーバーにアクセスする場合でも動作します。

Q: 現在、内部ROWID形式として動的SQLとバインドを使用しています。 malloc() を使用して領域を拡大する必要はありますか？

A: バージョン8のROWIDは、ホスト変数についてのバージョン7の記憶域要件を満たしていますので、変更したり領域を追加したりする必要はありません。

Q: 自分の表の列をROWID型に定義できますか？

A: 列をROWID型に定義することはできます。バージョン7ではROWID列に6バイト必要でしたが、バージョン8では10バイト必要になります。

Q: 現在、バージョン7のROWID形式に依存しています。変換アルゴリズムは、将来、文書化されるでしょうか？

A: バージョン8の新しいROWID形式は、そのような用途については文書化されていません。しかし、バージョン8では、バージョン8のROWIDの内容を解釈するためのDBMS\_ROWID ( PL/SQL ) パッケージが提供されています。

Q: 将来、自分の索引を再作成する必要がでてくるでしょうか？

A: 古いROWID形式を格納している列に対して作成された索引だけは、データの移行後に再作成する必要があります。

Q: ROWIDデータ型を、バージョン8より前のPL/SQLまたはRPC、FORMSで使用しています。この方法で今後も作業が続けられるでしょうか？

A: ROWIDがROWID型のホスト変数に戻される際の形式は同じですし、一般的に変更は必要ありません。ただし、次の特例を除きます。

dblink ( 異機種間dblink ) を介したバージョン7のサーバーからバージョン8のデータベースへのリモート・マップ問合せは、OCIによるDTYRID型 ( CHR変換なし ) としてのROWIDフェッチ時に、エラーORA3116で終了する。

- DTYCHR型として代わりにROWIDフェッチを使用すると、暗黙的変換を起動して問題を回避する。
- バージョン7のサーバーでSQLT\_RIDおよびパッチ ( Oracle から利用可能 ) を使用すると、CHR変換を起動せずに問題を回避する。



---

## アップグレードおよびダウングレード

この章では、バージョン 8 の現行リリースからバージョン 8 の新リリースへのアップグレードについて説明します。また、バージョン 8 のデータベース・システムからバージョン 7 へのダウングレードについても説明します。

この章で説明する内容は、Oracle バージョン 8 のインストール環境だけに適用されません。現行のリリースがバージョン 7 やバージョン 6 といったバージョン 8 より前のリリースで、そこからバージョン 8 へ移行する場合は、このマニュアルの最初に記載されている指示に従い、第 1 章「移行の概要」から始めてください。

この章で説明するトピックは、次のとおりです。

- バージョン 8 の新リリースへのアップグレード
- 製品の構成およびアップグレード
- アドバンスト・キューイング・オプションのアップグレード
- ダウングレード

関連項目：アップグレードおよびダウングレードの一部には、プラットフォームに固有の部分があります。使用しているプラットフォームでのこれらの操作の詳細は、プラットフォームに固有の Oracle マニュアルを参照してください。また、この章ではインストール・ユーティリティによるアップグレード方法について説明していません。これはオプションになっています。詳細は、使用しているプラットフォームに固有の Oracle マニュアルを参照してください。

## バージョン 8 の新リリースへのアップグレード

このセクションでは、リリース 8.0.3 からバージョン 8 の新しいリリースへのアップグレード・プロセスについて説明します。

---

警告: Oracle8 Enterprise Edition から Oracle8 (以前は Workgroup Server) へアップグレードする場合は、アップグレードの実行前に、Oracle8 Enterprise Edition の拡張機能を使うアプリケーションを変更し、それらの拡張機能を使わないようにしなければなりません。エディション間の違いに関する詳細は、『Oracle8 と Oracle8 Enterprise Edition の解説』および 8-4 ページの「製品の構成およびアップグレード」を参照してください。

---

バージョン 8 の現行のデータベースをアップグレードするには、次のステップを実行します。

1. 次のように入力し、Server Manager を起動します。

```
SVRMGRL
```

2. バージョン 8 のデータベース上で、SHUTDOWN NORMAL を実行します。

```
SVRMGR> SHUTDOWN NORMAL
```

3. データベースの全オフライン・バックアップを実行します。
4. バージョン 8 の新しいリリースをインストールします。

---

注意: インストレーションは、プラットフォームに固有の作業です。インストレーションの指示については、バージョン 8 のプラットフォーム固有のインストレーション・マニュアルおよび使用しているプラットフォームでのバージョン 8 の README ファイルを参照してください。

---

5. INIT.ORA ファイル内の COMPATIBLE パラメータを設定します。

関連項目: 詳細は、C-2 ページの「COMPATIBLE パラメータ」を参照してください。

6. CONNECT INTERNAL を実行します。

```
SVRMGR> CONNECT INTERNAL
```

7. STARTUP RESTRICT を実行します。

```
SVRMGR> STARTUP RESTRICT
```

- 成功したかどうかを後で確認するために、結果をログ・ファイルにスプールするようにシステムを設定します。

```
SVRMGR> SPOOL CATOUTU.LOG
```

- CAT8004.SQL を実行します。

```
SVRMGR> @CAT8004.SQL
```

CAT8004.SQL スクリプトは、一定のディクショナリ表を作成および変更します。また、システム・カタログ・ビューおよびPL/SQLの使用に必要なすべてのパッケージを作成する、CATALOG.SQL スクリプトと CATPROC.SQL スクリプトを実行します。

関連項目：その他のデータ・ディクショナリ構造体を作成する場合は、使用可能なスクリプトの完全な一覧と説明について、『Oracle8 Server リファレンス・マニュアル』を参照してください。

- Oracle システムにアドバンスド・レプリケーションをインストールしている場合は、新しいリリースに付属する次のカタログ・スクリプトを実行します。

```
SVRMGR> @CATREP.SQL
```

- Oracle システムに Parallel Server をインストールしている場合は、新しいリリースに付属する次のカタログ・スクリプトを実行します。

```
SVRMGR> @CATPARR.SQL
```

- スクリプト結果のログ・ファイルへのスプーリングをオフにします。

```
SVRMGR> SPOOL OFF
```

次にスプール・ファイルをチェックして、すべてのパッケージとプロシージャのコンパイルが成功したかどうかを検証してください。ステップ 8 でスプール・ファイルに名前を付けましたが、その名前は CATOUTU.LOG でした。このファイルに問題を見つけたら、それを訂正してください。

- ALTER SYSTEM DISABLE RESTRICTED SESSION を実行します。

```
SVRMGR> ALTER SYSTEM DISABLE RESTRICTED SESSION
```

- UTLCONST.SQL を実行して、無効な日付制約をチェックします。

---

注意：バージョン 8 の以前のリリースへ移行した後、UTLCONST.SQL をすでに実行済みの場合は、再度実行する必要はありません。ただし、このスクリプトを複数回実行してもシステムに害を与えることはありませんので、実行したかどうか分からない場合は、ここで実行すべきです。

---

## 製品の構成およびアップグレード

---

UTLCONST.SQL を実行するには、次のコマンドを発行します。

```
SVRMGRL> SPOOL UTLRESULT.LOG
SVRMGRL> @UTLCONST.SQL
SVRMGRL> SPOOL OFF
```

無効な日付の制約は、無効日付操作を必要とします。無効日付操作は、日付の中の世紀を暗黙的に想定するものであり、西暦 2000 年に問題が発生します。UTLCONST.SQL スクリプトは、データベース内のすべてのチェック制約で実行され、無効日付操作を組み込む場合は制約を無効に設定します。このスクリプトは、最後にすべての無効な制約を選択します。

スクリプトを実行した後、UTLRESULT.LOG ログ・ファイルには、無効な日付制約を持つすべての制約が入っています。

---

---

注意：UTLCONST.SQL は無効な制約を訂正せずに、使用禁止にします。必要な変更を加えた後に、無効な制約を削除するか、再作成する必要があります。

---

---

## 製品の構成およびアップグレード

リリース 8.0.4 の Oracle Server は、次のような 2 つの製品構成で提供されます。

- Oracle8
- Oracle8 Enterprise Edition

関連項目：Oracle8 と Oracle8 Enterprise Edition の違い、役に立つオプションや機能などの詳細は、『Oracle8 と Oracle8 Enterprise Edition の解説』を参照してください。

Oracle8 Enterprise Edition ではなく Oracle8 へアップグレードする場合は、前のリリースで役に立った特定のオプションと機能を失うことがあります。次の表に、そのようなアップグレードでの機能変更を示します。

---

なくなったオプションまたは機能	機能変更
パーティション化	CREATE TABLE / INDEX / CLUSTER の PARTITION 句が使用禁止になりました。PDML がオフになっています。これにより、すべての操作は並列ではなく直列に行われます。
オブジェクト	CREATE TYPE および ALTER TYPE REPLACE が使用禁止になりました。CREATE VIEW ... OF (オブジェクト・ビューなど) も使用禁止になっています。

---

なくなったオプションまたは機能	機能変更
Parallel Server	起動時の SHARED が使用禁止になりました。
アドバンスト・レプリケーション	更新可能なスナップショットが使用禁止になりました。また、N-way マスター・スナップショットが使用禁止になりました。
ビットマップ索引	CREATE BITMAP INDEX 文が使用禁止になりました。 ビットマップ索引は使っていません。 ランタイム副問合せの最適化がオフになっています。
接続の多重化	INIT.ORA ファイルで多重化を指定すると、起動時に Oracle エラーが発生します。
接続プーリング	INIT.ORA ファイルでプーリングを指定すると、起動時に Oracle エラーが発生します。
データベース・キューイング	キュー・プロシージャとデキュー・プロシージャが使用禁止になりました。
Instead-of トリガー	CREATE TRIGGER 文の INSTEAD OF 句が使用禁止になりました。
増分バックアップおよびリカバリ	Recovery Manager (rman) が、増分バックアップおよびリカバリを使用禁止にします。
パラレル・バックアップおよびリカバリ	Recovery Manager (rman) が、パラレル・バックアップおよびリカバリを使用禁止にします。
パラレル実行	CREATE INDEX および CREATE TABLE AS SELECT のパラレル実行が使用禁止になりました。
パラレル・ロード	パラレル・ロードが使用禁止になりました。
時点表領域回復	ALTER DATABASE MOUNT CLONE 文が使用禁止になりました。

## アドバンスト・キューイング・オプションのアップグレード

Oracle アドバンスト・キューイング (AQ) 8.0.4 の操作インタフェースには、8.0.3 Oracle AQ インタフェースとの下位互換性があります。

---

注意: COMPATIBLE パラメータの設定が 8.0.4.0.0 でリリース 8.0.4 のデータベースを起動していなければ、このセクションに記載の機能を使うために今すぐ起動してください。詳細は、C-2 ページの「COMPATIBLE パラメータ」を参照してください。

---

## AQ\$\_AGENT データ型で使用可能な新しいフィールド

最新のリリースでは、AQ\$\_AGENT データ型でアドレス・フィールドが使用可能です。したがって、このフィールドは、インタフェースが引数としてエージェントをとる場所ならどこでも指定できます。たとえば、メッセージ・プロパティの受信者リストや DBMS\_AQADM.ADD\_SUBSCRIBER 管理インタフェースなどがあります。

## 拡張アドレス・フィールド

AQ\$\_AGENT データ型のアドレス・フィールドは、1024 バイトに拡張されています。拡張アドレス・フィールドを使うときは、次のステップを実行してください。

1. エクスポート・ユーティリティを使って既存のキューの内容をエクスポートします。
2. CATNOQUEUE.SQL を実行して、既存のディクショナリとキュー表を削除します。

```
SVRMGRL> @CATNOQUEUE.SQL
```

3. CATQUEUE.SQL を実行して、新しい型とディクショナリ表を再定義します。

```
SVRMGRL> @CATQUEUE.SQL
```

4. インポート・ユーティリティを使って、エクスポート済みのキューをインポートします。

---

---

注意：アプリケーションでアドレス・フィールドを拡張する必要がない場合は、これらのステップを実行する必要もありません。

---

---

## 新しいディクショナリ表

リリース 8.0.4 のアップグレード・スクリプト ( CAT8004.SQL ) は、次のような他のディクショナリ表を作成します。

- SYSTEM.AQ\$\_SCHEDULES
- SYS.AQ\$\_MESSAGE\_TYPES
- SYS.AQ\$\_QUEUE\_STATISTICS

## ダウングレード

「ダウングレード」という用語は、Oracle データベースを同じバージョンの前のリリースへ変換するときに使います。たとえば、データベースをリリース 8.0.4 からリリース 8.0.3 へ変換するときなどです。また、データベースをバージョン 8 からバージョン 7 へ変換するといった、以前のバージョンへの Oracle データベースの変換にも「ダウングレード」を使います。以下のセクションでは、その両方について説明します。

---

---

警告: Oracle8 Enterprise Edition から Oracle8 ( 以前は Workgroup Server )  
へダウングレードする場合は、ダウングレードの実行前に、Oracle8  
Enterprise Edition の拡張機能を使うアプリケーションを変更し、それらの  
拡張機能を使わないようにしなければなりません。エディション間の違い  
に関する詳細は、『Oracle8 と Oracle8 Enterprise Edition の解説』を参照し  
てください。

---

---

## リリース 8.0.4 からリリース 8.0.3 へのダウングレード

リリース 8.0.4 のデータベースをリリース 8.0.3 へダウングレードするときは、次のステップを  
実行してください。

1. 次のコマンドを発行して、リリース 8.0.4 のデータベースを停止します。

```
SVRMGR> SHUTDOWN NORMAL
```

Parallel Server を実行している場合は、すべてのインスタンスについてこのコマンドを発  
行します。

2. 全オフライン・バックアップを実行します。

関連項目: 詳細は、『Oracle8 Server バックアップおよびリカバリ』を参照  
してください。

3. リリース 8.0.4 のデータベースをオープンします。
4. 次のコマンドを発行して、COMPATIBLE パラメータの設定をチェックします。

```
SVRMGR> SELECT name, value, description FROM v$parameter  
WHERE name="compatible";
```

COMPATIBLE が 8.0.3.0.0 または 8.0.0.0.0 に設定されている場合は、ステップ 10 へ進ん  
でください。COMPATIBLE が 8.0.4.0.0 に設定されている場合は、ステップ 5 へ進んで  
ください。

5. アドバンスト・キューイング・オプションの伝播を使う場合は、伝播を使用禁止にしま  
す。使わない場合はステップ 6 へ進みます。

関連項目: この機能の詳細は、C-3 ページの「アドバンスト・キューイング  
( AQ ) での伝播」を参照してください。

アドバンスト・キューイング・オプションの伝播を使用禁止にするには、次の手順のど  
れかを実行します。

- すべてのメッセージが正常に伝播されるのを待ってから、DBMS\_AQDM.UN-  
SCHEDULE\_PROPAGATION() を使ってすべての伝播をスケジュール解除します。

- DBMS\_AQDM.UNSCHEDULE\_PROPAGATION() を使ってすべての伝播をスケジュール解除した後に、キュー表から伝播されるはずのメッセージをすべて削除します。

関連項目：DBMS\_AQDM.UNSCHEDULE\_PROPAGATION() の詳細は、『Oracle8 Server アプリケーション開発者ガイド』を参照してください。

6. ALTER DATABASE RESET COMPATIBILITY を実行します。

```
SVRMGR> ALTER DATABASE RESET COMPATIBILITY
```

7. SHUTDOWN NORMAL を実行します。

```
SVRMGR> SHUTDOWN NORMAL
```

---

警告：ステップの完了後に COMPATIBLE=8.0.4.0.0 を使ってデータベースをオープンしないでください。

---

8. INIT.ORA ファイル内の COMPATIBLE パラメータを次のように設定します。

```
COMPATIBLE=8.0.0.0.0
```

9. データベースをオープンして、リリース 8.0.3 と互換性があることを確認します。

10. CAT8004D.SQL を実行します。

```
SVRMGR> @CAT8004D.SQL
```

11. SHUTDOWN NORMAL を実行します。

```
SVRMGR> SHUTDOWN NORMAL
```

12. リリース 8.0.3 のインストレーション・メディアを使ってリリース 8.0.3 をインストールします。

---

注意：インストレーションは、プラットフォームに固有の作業です。インストレーションの指示については、リリース 8.0.3 のプラットフォーム固有のインストレーション・マニュアルおよび使用しているプラットフォームでのリリース 8.0.3 の README ファイルを参照してください。

---

13. CONNECT INTERNAL を実行します。

```
SVRMGR> CONNECT INTERNAL
```

14. STARTUP RESTRICT を実行します。

```
SVRMGR> STARTUP RESTRICT
```

15. 成功したかどうかを後で確認するために、結果をログ・ファイルにスプールするようにシステムを設定します。

```
SVRMGR>SPOOL CATOUTD.LOG
```

16. CATALOG.SQL を実行します。

```
SVRMGR>@CATALOG.SQL
```

17. CATPROC.SQL を実行します。

```
SVRMGR>@CATPROC.SQL
```

18. Oracle システムにアドバンスド・レプリケーションをインストールしている場合は、リリース 8.0.3 に付属する次のカタログ・スクリプトを実行します。

```
SVRMGR> @CATREP.SQL
```

19. Oracle システムに Parallel Server をインストールしている場合は、リリース 8.0.3 に付属する次のカタログ・スクリプトを実行します。

```
SVRMGR> @CATPARR.SQL
```

20. スクリプト結果のログ・ファイルへのスプーリングをオフにします。

```
SVRMGR> SPOOL OFF
```

次にスプール・ファイルをチェックして、すべてのパッケージとプロシージャのコンパイルが成功したかどうかを検証してください。ステップ 15 でスプール・ファイルに名前を付けましたが、その名前は CATOUTD.LOG でした。このファイルに問題を見つけたら、それを訂正してください。

## バージョン 8 からリリース 7.x へのダウングレード

バージョン 8 のデータベースは、バージョン 7 のデータベース（リリース 7.3 など）にダウングレードできます。ただし、ダウングレードの方法はあまり多くありません。そのために必要な手順は操作が難しく、時間もかかります。

オラクル社では、バージョン 8 の移行ユーティリティを使ったバージョン 8 からバージョン 7 へのダウングレードはサポートしていません。また、バージョン 8 には、特にダウングレードのための他の機能は付属していません。

ダウングレードの手順は、保存しなければならない新しいデータがバージョン 8 のデータベースに含まれているかどうかによって決まります。実際のバージョン 8 のデータベースに該当する次の手順を使います。

- 新しいデータが含まれないバージョン 8 のデータベースをダウングレード
- 新しいデータを含むバージョン 8 のデータベースをダウングレード

### 新しいデータが含まれないバージョン 8 のデータベースをダウングレード

保存する必要のある新しいデータがバージョン 8 のデータベースにまだ入力されていない場合は、以前のリリース 7.x ソース・データベースの完全なバックアップを復元し、それを再オープンできます。その際、必ず以前のリリース 7.x データベースで使用していた元の初期化パラメータを含む完全な復元結果を使用してください。

### 新しいデータを含むバージョン 8 のデータベースをダウングレード

バージョン 8 のデータベースに、保存する必要のある新しいデータが含まれている場合は、次のステップを実行してください。

1. CATEXP7.SQL で作成したビューをインストール済みのバージョン 7 のエクスポート・ユーティリティを使って、新しいデータを含むバージョン 8 のデータベースの一部をエクスポートします。

関連項目：バージョン 8 からバージョン 7 のエクスポート・ユーティリティ  
を実行する方法は、『Oracle8 Server ユーティリティ』を参照してください。

2. 以前のバージョン 7 のデータベースの完全なバックアップを復元し、以前の初期化パラメータが復元結果に含まれていることを確認します。
3. 復元されたデータベースをオープンします。
4. バージョン 7 のインポート・ユーティリティを使って、以前にバージョン 8 のデータベースからエクスポートしたファイルを、復元されたバージョン 7 のデータベースへインポートします。

バージョン 8 のデータベースからバージョン 7 のデータベースへ表データを送る方法として、他にもいくつかの方法が利用できます。バージョン 7 へデータを戻すためのこれらの方法は、バージョン 8 を使って更新した表の数が少ない場合であれば、比較的簡単に行えます。ただし、表の全データベースをコピーすることは、時間のかかる複雑なタスクであるため、バージョン 8 を使って多くの表を更新する前に、バージョン 7 へ戻す必要があるかどうか判断しなければなりません。

バージョン 8 のデータベースをバージョン 7 へダウングレードする方法として、次のような代替方法を利用できます。

- SQL\*Plus を使用して新しいバージョン 8 のデータから Oracle 形式でないテキスト・ファイルを作成し、次に SQL\*Loader を使用してそのデータをバージョン 7 のデータベースにロードし直す。

- SQL\*Plus の COPY コマンドを使用して、バージョン 8 のデータベース表から以前のバージョン 7 のデータベース表へ新しいデータをコピーする。

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

- まだ以前のバージョン 7 のデータベースを使用している場合は、データベース・リンクを用いたバージョン 7 のデータベースからの分散問合せによりバージョン 8 のデータベース表内のデータを選択することで、データを追加または変更した表を再作成する (CREATE TABLE ... AS SELECT を使用)。

関連項目: CREATE TABLE コマンドの AS 句の詳細は、『Oracle8 Server SQL リファレンス』を参照してください。

ダウングレード

---

---

## 移行ユーティリティのメッセージ

移行ユーティリティは、移行処理中にエラー・メッセージと情報メッセージを表示することがあります。この付録では、移行ユーティリティの使用中表示されるエラーについて説明します。また、それぞれのエラーについて、考えられる原因とエラー訂正のための処置を示します。情報メッセージも示していますが、対処措置は必要ありません。

移行時に DBA のファイル番号ビットの圧縮はできません。

原因：移行ユーティリティがデータ・ブロック・アドレスで使用するファイル番号ビットの数を減らそうとしました。

処置：オラクル社カスタマー・サポートに連絡してください。

レコードが number バイトを超えているため変換ファイルを作成できません。

原因：内部エラーが発生しました。バージョン 7 の制御ファイルから有効な変換ファイルを作成できませんでした。

処置：バージョン 7 の制御ファイルが壊れていないか確認し、問題を解決して移行ユーティリティを再実行してください。

CHECK\_ONLY - V8 カタログ領域の要件だけを見積ります (デフォルト = FALSE)。

原因：これは、CHECK\_ONLY コマンド行引数に関する情報メッセージです。

処置：ユーザーの処置は不要です。

CHECK\_ONLY と NO\_SPACE\_CHECK は相互に排他オプションです。

原因：これら 2 つの相互排他のコマンド行オプションが移行ユーティリティに渡されました。

処置：どちらか 1 つだけを使用して、移行ユーティリティを再実行してください。

コマンド行引数の値は、TRUE または FALSE でなければなりません (string)。

原因：コマンド行引数に TRUE または FALSE 以外の値が入力されました。

処置：コマンド行引数の構文をチェックし、文を訂正して操作をもう一度やり直してください。

---

コマンド行の引数が < キーワード >=< 値 > で指定されていません (string)。

原因：コマンド行引数の指定方法が間違っています。

処置：コマンド行引数の構文をチェックし、文を訂正して操作をもう一度やり直してください。

コマンド行引数：

原因：この情報メッセージは、コマンド行引数を表示しています。

処置：ユーザーの処置は不要です。

コマンド名が見つかりません (string)。

原因：内部エラーが発生しました。MIGRATE.BSQ スクリプトが壊れている可能性があります。

処置：移行ユーティリティのバージョンおよび MIGRATE.BSQ のバージョン、バージョン 8 のターゲット・ソフトウェアのバージョンに互換性があるか確認してください。また、MIGRATE.BSQ が壊れていないか確認してください。問題を解決して移行ユーティリティを再実行してください。

コマンドが CMD(ARG1, ARG2, ...) で指定されていません。

原因：内部エラーが発生しました。MIGRATE.BSQ スクリプトが壊れている可能性があります。

処置：移行ユーティリティのバージョンおよび MIGRATE.BSQ のバージョン、バージョン 8 のターゲット・ソフトウェアのバージョンに互換性があるか確認してください。また、MIGRATE.BSQ が壊れていないか確認してください。問題を解決して移行ユーティリティを再実行してください。

Copy long コマンドの形式が間違っています。正しい形式は、COPYLONG (U1,T1,C1,U2,T2,C2,K1<,K2>) です。

原因：内部エラーが発生しました。MIGRATE.BSQ スクリプトが壊れている可能性があります。

処置：移行ユーティリティのバージョンおよび MIGRATE.BSQ のバージョン、バージョン 7 のターゲット・ソフトウェアのバージョンに互換性があるか確認してください。また、MIGRATE.BSQ が壊れていないか確認してください。問題を解決して移行ユーティリティを再実行してください。

c\_file#\_block# に対する number バイトの単一の連続エクステンが見つかりません。

原因：SYSTEM 表領域に連続したエクステンが十分ありません。

処置：SYSTEM 表領域に空き領域を追加し、移行ユーティリティを再実行してください。

c\_ts# に対する number バイトの単一の連続エクステンが見つかりません。

原因：SYSTEM 表領域に連続したエクステンが十分ありません。

処置：SYSTEM 表領域に空き領域を追加し、移行ユーティリティを再実行してください。

---

i\_file#\_block# に対する number バイトの単一の連続エクステントが見つかりません。

原因：SYSTEM 表領域に連続したエクステントが十分ありません。

処置：SYSTEM 表領域に空き領域を追加し、移行ユーティリティを再実行してください。

i\_ts# に対する number バイトの単一の連続エクステントが見つかりません。

原因：SYSTEM 表領域に連続したエクステントが十分ありません。

処置：SYSTEM 表領域に空き領域を追加し、移行ユーティリティを再実行してください。

論理名 name を変換できませんでした。

原因：内部エラーが発生しました。

処置：論理名が正しく定義されているかを確認して、移行ユーティリティを再実行してください。

現行バージョン：str -- データベースはバージョン 7.1 以降でなければなりません。

原因：現行のデータベースが、バージョン 7 リリース 7.1 より前のバージョンです。

処置：現行のデータベースを、お使いのプラットフォームでの移行ユーティリティがサポートするリリースに移行またはアップグレードしてください。そして、移行ユーティリティを再実行します。使用しているプラットフォームでの移行ユーティリティでサポートされるリリースについては、プラットフォームに固有の Oracle マニュアルを参照してください。

列 :name に対するデータ型は LONG 型でなければなりません。

原因：内部エラーが発生しました。MIGRATE.BSQ スクリプトが壊れている可能性があります。

処置：移行ユーティリティのバージョンおよび MIGRATE.BSQ のバージョン、バージョン 7 のターゲット・ソフトウェアのバージョンに互換性があるかを確認してください。また、MIGRATE.BSQ が壊れていないかを確認してください。問題を解決して移行ユーティリティを再実行してください。

データファイルに不整合な状態が見つかりました（内部エラー）

原因：内部エラーが発生しました。データファイルが不整合な状態になっています。

処置：オラクル社カスタマー・サポートに連絡してください。

表領域はオンラインですがデータファイルがオフラインです。- 移行の前にメディア回復を行ってデータファイルをオンラインにしてください。

原因：表領域はオンラインですが、表領域内のデータファイルがオフラインです。データファイルと表領域の両方がオンラインまたは通常オフラインになるまで、移行を行うことはできません。

処置：移行を再実行する前に、メディア回復を行ってデータファイルをオンラインにしてください。

---

DBNAME - 現行のデータベース名 ( init.ora 内の db\_name ) DBNAME -

原因: これは、DBNAME コマンド行引数に関する情報メッセージです。

処置: ユーザーの処置は不要です。

ディクショナリ定数が見つかりません。 -

原因: 内部エラーが発生しました。MIGRATE.BSQ スクリプトが壊れている可能性があります。

処置: 移行ユーティリティのバージョンおよび MIGRATE.BSQ のバージョン、バージョン 8 のターゲット・ソフトウェアのバージョンに互換性があるか確認してください。また、MIGRATE.BSQ が壊れていないか確認してください。問題を解決して移行ユーティリティを再実行してください。

system.def\$\_call または def\$\_calldest、def\$\_error に、エントリが見つかりました。 - 移行前にすべての遅延トランザクションをプッシュします。

原因: system.def\$\_call または def\$\_calldest、def\$\_error に、エントリが存在します。

処置: エントリが system.def\$\_call 中にある場合は、system.def\$\_call が空になるまですべての遅延トランザクションをプッシュしてください。エントリが system.def\$\_error 中にある場合は、ローカル・キューが空になるまでその中のエラーを解決および再実行してください。移行ユーティリティを再実行してください。

slgtd を呼び出すエラー

原因: slgtd から現在の時間を取得する際のエラーであり、内部エラーです。移行ユーティリティが壊れている可能性があります。

処置: 移行ユーティリティのバージョンおよび MIGRATE.BSQ のバージョン、バージョン 8 のターゲット・ソフトウェアのバージョンに互換性があるか確認してください。また、MIGRATE.BSQ が壊れていないか確認してください。問題を解決して移行ユーティリティを再実行してください。

ファイル name をクローズ中にエラーが発生しました。

原因: 内部エラーが発生しました。データをディスクに書き込めませんでした。

処置: ファイルのアクセス権限が正しいか確認してください。ファイル書込みに十分な領域や割当てが確保されているか確認してください。また、ディスクが破損していないか確認してください。問題を解決して移行ユーティリティを再実行してください。

object の必要領域の見積りは number ブロックです。

原因: この情報メッセージの中で、移行ユーティリティはオブジェクトに必要な領域を表示します。

処置: ユーザーの処置は不要です。

ファイル filename が、DBA 変換には大きすぎます。

原因: 内部エラーが発生しました。DBA 変換を行うには、ファイル名が大きすぎます。

処置: オラクル社カスタマー・サポートに連絡してください。

---

ファイル・ヘッダーを number バイト中に入れることができません。

原因：内部エラーが発生しました。

処置：制御ファイルが壊れていないか確認し、問題を解決して移行ユーティリティを再実行してください。

制御ファイルの固定領域を number バイト中に入れることができません。

原因：内部エラーが発生しました。

処置：制御ファイルが壊れていないか確認し、問題を解決して移行ユーティリティを再実行してください。

NULL の SQL 文があります。

原因：内部エラーが発生しました。MIGRATE.BSQ スクリプトが壊れている可能性があります。

処置：移行ユーティリティのバージョンおよび MIGRATE.BSQ のバージョン、バージョン 8 のターゲット・ソフトウェアのバージョンに互換性があるかを確認してください。また、MIGRATE.BSQ が壊れていないか確認してください。問題を解決して移行ユーティリティを再実行してください。

システム表領域内の空き領域は、number ブロックです。

原因：この情報メッセージは、SYSTEM 表領域内にある空き領域の大きさを通知します。

処置：ユーザーの処置は不要です。

空き領域：

原因：この情報メッセージは、SYSTEM 表領域内にある空き領域の大きさを通知します。

処置：ユーザーの処置は不要です。

書き込みが未完全です。

原因：内部エラーが発生しました。データをディスクに書き込めませんでした。

処置：ファイルのアクセス権限が正しいかを確認してください。ファイル書込みに十分な領域や割当てが確保されているかを確認してください。また、ディスクが破損していないかを確認してください。問題を解決して移行ユーティリティを再実行してください。

新ディクショナリに領域が不足しています (number バイト必要、number 検出)。

原因：新しいデータ・ディクショナリ情報用の領域が SYSTEM 表領域に不足しています。

処置：SYSTEM 表領域に必要とされる領域を加えてから、移行ユーティリティを再実行してください。

---

無効な NLS\_NCHAR 値が指定されました。

原因：コマンド行に指定した NLS\_NCHAR 値が無効です。

処置：コマンド行に指定した NLS\_NCHAR 値を訂正し、移行ユーティリティを再実行してください。

移行処理を続行できません。 - データベースのブロックサイズがOracle8の最小ブロックサイズ 2k に達していません。

原因：既存のデータベースのブロックサイズが 2K バイトに達していません。

処置：バージョン 7 のデータベースのブロックサイズが最小でも 2K バイトあることを確認してください。バージョン 7 のデータベースの再作成を検討することもできます。そして、移行ユーティリティを再実行します。

表領域がオフラインだと、データファイルがオンラインでも移行処理を続行できません。

原因：表領域内のデータファイルはオンラインですが、表領域がオフラインです。データファイルと表領域の両方がオンラインまたは通常オフラインになるまで、移行を行うことはできません。

処置：データファイルのオンライン状態が表領域のオンライン状態と同じであることを確認し、移行ユーティリティを再実行してください。

未処理の取消しがあると、アクティブ・トランザクションまたはオフラインの表領域の移行処理を続行できません。

原因：移行ユーティリティがデータベースを移行しようとしたときに、1 つまたは複数の表領域がオフラインで未処理の保存取消しがありました。

処置：3-7 ページに記載の「移行のためのバージョン 7 のソース・データベースの準備」のステップ 7 に戻り、すべてのオフライン表領域が確実にオフラインになっていることを確認してください。その後、バージョン 8 の移行ユーティリティを再実行してください。

データベースをマウントしています ...

原因：これは情報メッセージです。移行ユーティリティがバージョン 7 のデータベースをマウントしています。

処置：ユーザーの処置は不要です。

MULTIPLIER - seg\$/uet\$ クラスタ化索引サイズの増加係数 ( デフォルト =15 )

原因：これは、MULTIPLIER コマンド行設定に関する情報メッセージであり、移行ユーティリティが表示します。

処置：ユーザーの処置は不要です。

MULTIPLIER 値は 2 以上でなければなりません。

原因：コマンド行の中でバージョン 8 の i\_file#\_block# の初期サイズを指定する MULTIPLIER 値が、2 よりも小さな値です。

処置：MULTIPLIER 値を 2 以上の値に変更し、移行ユーティリティを再実行してください。

---

NEW\_DBNAME name が長過ぎます。 - 最大長は 8 文字です。

原因：指定した新しいデータベース名の長さが 8 文字を超えていました。

処置：新しいデータベースの名前を 8 文字以下に変更し、移行ユーティリティを再実行してください。

NEW\_DBNAME - データベースの新しい名前 (最大 8 文字)

原因：この情報メッセージは、NEW\_DBNAME コマンド行引数に関する情報を表示しています。

処置：ユーザーの処置は不要です。

NLS\_NCHAR - nchar キャラクタ・セット値を指定します。

原因：この情報メッセージは、NLS\_NCHAR コマンド行引数に関する情報を表示しています。

処置：ユーザーの処置は不要です。

NO\_SPACE\_CHECK - 領域チェックを行いません (デフォルト=FALSE)

原因：これは、NO\_SPACE\_CHECK コマンド行引数に関する情報メッセージです。

処置：ユーザーの処置は不要です。ただし、このオプションを指定して移行ユーティリティを実行する前に、十分な領域があるか確認してください。

変換ファイルの作成時に処理する表領域 / データファイルの数値が不正です。

原因：変換ファイルの作成時に内部エラーが発生しました。

処置：オラクル社カスタマー・サポートに連絡してください。

データベースをオープンしています ...

原因：これは情報メッセージです。移行ユーティリティがバージョン 7 のデータベースをオープンしています。

処置：ユーザーの処置は不要です。

ORA\_NLS33 環境変数が設定されていないか、不正に設定されています。

原因：ORA\_NLS33 環境変数が NLS データファイルを指定していません。

処置：ORA\_NLS33 環境変数が正しいファイルを指定するように設定し、移行ユーティリティを再実行してください。

ORA-number:

原因：移行ユーティリティが ORA エラーを受け取りましたが、メッセージ・テキストを取り出せません。

処置：Oracle エラー番号に応じた適切な処置を行ってください (『Oracle8 Server エラー・メッセージ』を参照)。

パラメータ・バッファがオーバーフローしました。

原因：初期化パラメータ・ファイルが大きすぎて、バッファに格納できません。

処置：廃止されたパラメータなどを削除してパラメータ・ファイルのサイズを小さくし、移行ユーティリティを再実行してください。

---

パラメータ・ファイルが number バイトを超えました。

原因：バージョン 7 のデータベース用のパラメータ・ファイルが所定の最大サイズを超えています。

処置：可能ならば、廃止されたパラメータを除去することによってパラメータ・ファイルのサイズを縮小してください。それ以外の場合は、オラクル社カスタマー・サポートに連絡してください。

PFILE - init.ora ファイルを代替使用します。

原因：これは、PFILE コマンド行引数に関する情報を表示する情報メッセージです。

処置：ユーザーの処置は不要です。

ファイル name でシーク・エラーが発生しました。

原因：ファイル名の読取り時に内部エラーが発生しました。

処置：ファイルやディスクが壊れていないか確認してください。移行ユーティリティを再実行する前に、問題を解決してください。

読み込みの不足。 number バイト要求されましたが、 number バイトしか読み込まれませんでした。

原因：制御ファイルの読取りに問題があります。

処置：制御ファイルが壊れていないか確認し、問題を解決して移行ユーティリティを再実行してください。

データベースを強制終了します (異常終了)...

原因：内部エラーが発生しました。

処置：停止の原因を通知するために追加エラー・メッセージが表示されます。そのメッセージで提案される処置を行ってください。

データベースを停止しています ...

原因：これは情報メッセージです。移行ユーティリティがバージョン 7 のデータベースを停止しています。

処置：ユーザーの処置は不要です。

SPOOL - ファイルにスプール出力します

原因：これは、SPOOL コマンド行引数に関する情報を表示する情報メッセージです。

処置：ユーザーの処置は不要です。

データベースを起動しています ...

原因：これは情報メッセージです。移行ユーティリティがバージョン 7 のインスタンスを起動しています。

処置：ユーザーの処置は不要です。

---

文字列引数が長すぎます。最大長

原因：移行ユーティリティに渡したコマンド行引数の文字列が最大サイズを超えています。

処置：コマンド行引数の文字列を短縮し、移行ユーティリティを再実行してください。

データファイルの表領域が通常オフラインになっていません。移行前に表領域をオンラインまたは通常オフラインにするか、または削除してください。

原因：表領域が IMMEDIATE または TEMPORARY を使ってオフラインにされました。

処置：表領域をオンラインにしてから、NORMAL を使ってオフラインにするか削除してください。その後、移行ユーティリティを再実行してください。

コマンドの引数が多すぎます (number max)。

原因：コマンド行に指定した引数の数が多すぎます。

処置：コマンドの構文をチェックして、コマンド行に適切な数のオプションを指定してください。

LONG をコピーするのにバッファ領域を割り当てられません。

原因：移行ユーティリティが、データベースの LONG 列にコピーするためのバッファにメモリーを割り当てられません。

処置：移行ユーティリティに使用できる十分なマシン・リソースがあるかを確認して、移行ユーティリティを再実行してください。

ファイル name をオープンできません。

原因：内部エラーが発生しました。あるいは、バージョン 8 の移行ユーティリティを起動したときに、ファイルが所定の位置に存在しませんでした。

処置：ファイルが存在するか、ファイルのオープンと読取りの権限が Oracle にあるか確認してください。または、ファイルやファイルが存在するディスクが壊れていないかを確認してください。問題を解決して移行ユーティリティを再実行してください。

ファイル name を読み取れません。

原因：内部エラーが発生しました。あるいは、移行ユーティリティを起動したときにファイルが所定の位置に存在しませんでした。

処置：ファイルが存在するか、ファイルのオープンと読取りの権限が Oracle にあるかを確認してください。または、ファイルやディスクが壊れていないかを確認してください。問題を解決して移行ユーティリティを再実行してください。

ファイル name を書き込めません。

原因：内部エラーが発生しました。

処置：ファイルの書き込み権限が Oracle にあるかを確認してください。または、ファイルを書き込むディスクが破損していないかを確認してください。問題を解決してから、移行ユーティリティを再実行してください。

---

V8 カタログ領域要件：

原因：これは、移行ユーティリティを正常に実行するために SYSTEM 表領域に必要な空領域の大きさを表示する情報メッセージです。

処置：移行ユーティリティを実行する前に、追加の空領域がどの程度あるかを確認してください。

---

## 制御ファイル固定ビューの変更

この付録では、制御ファイル固定ビューの、バージョン7リリース7.3からバージョン8のサーバーへの変更点について簡単に説明します。

### 制御ファイル・ビュー内の日付列

バージョン7では、制御ファイル・ビューの日付列はすべて VARCHAR2(20) 文字列で、'MM/DD/YY HH24:MI:SS' という書式をとります。

バージョン8では、新しい日付列はすべて実際の DATE 列です。以前の VARCHAR2(20) 文字列とは対照的に、DATE によってユーザーは次のような利点を得ることができます。

- すべての日付列が DATE データ型なので、一貫性がある。
- SQL および PL/SQL では、日付を用いて容易に日付算術（ソートを含む）ができる。
- ユーザーは NLS\_DATE\_FORMAT を使用して、日付書式を設定できる。
- 以前の書式で日付を表示したいユーザーは、NLS\_DATE\_FORMAT を MM/DD/YY HH24:MI:SS 書式に設定できる。
- 西暦 2000 年以降の問題を避けるために、2 桁の年数値の使用を避けられる。

この付録では、バージョン8で変更または追加されたビューの名前およびビュー内の列の名前を示しています。

関連項目：表内の各列の説明および列の完全なリストについては、『Oracle8 Server リファレンス・マニュアル』を参照してください。

## バージョン 8 で保持されている廃止されたビュー

V\$ARCHIVE では、アーカイブする必要のあるログが表示されます。

V\$LOG は V\$LOG\_HISTORY に置き換えられ、V\$LOG\_HISTORY はバージョン 8 の新しいビューである V\$ARCHIVED\_LOG に置き換えられました。しかし、バージョン 8 では現在でも V\$LOG を提供しており、ARCHIVED 列内のすべてのアーカイブ・ログの一覧を示します。

V\$LOGHIST ビューは廃止されましたが、互換性のために保持されています。

### V\$LOG\_HISTORY の保持とアップグレード

バージョン 8 では、次の表に示すように、複数の V\$LOG\_HISTORY 列が改名されました。さらに、バージョン 7 の ARCHIVE\_NAME 列は、バージョン 8 にはありません。

バージョン 8 の列	バージョン 7 の列
FIRST_TIME	TIME
FIRST_CHANGE#	LOW_CHANGE#
NEXT_CHANGE#	HIGH_CHANGE#
[ なし ]	ARCHIVE_NAME

### V\$ARCHIVED\_LOG による V\$LOG\_HISTORY の置換え

バージョン 8 の新しいビュー V\$ARCHIVED\_LOG では、アーカイブされたログとその名前が表示されます。ログがアーカイブされていない場合、V\$ARCHIVED\_LOG ビューは、そのログについての行を戻しません。ログが 2 度アーカイブされた場合、V\$ARCHIVED\_LOG はそのログについて 2 つの行を戻します。

この新しいビュー内の特定の列は、バージョン 7 の V\$LOG\_HISTORY 名に対応し、バージョン 8 の他のビュー内にある列名と一致する形式を使用します。V\$ARCHIVED\_LOG ビューでは、制御ファイルからのアーカイブ・ログ情報が表示されます。

列名	列型
RECID	NUMBER
STAMP	NUMBER
NAME	VARCHAR2(512)
THREAD#	NUMBER
SEQUENCE#	NUMBER

列名	列型
RESETLOGS_CHANGE#	NUMBER
RESETLOGS_TIME	DATE
FIRST_CHANGE#	NUMBER
FIRST_TIME	DATE
NEXT_CHANGE#	NUMBER
NEXT_TIME	DATE
BLOCKS	NUMBER
BLOCK_SIZE	NUMBER
COMPLETION_TIME	DATE
DELETED	VARCHAR2(3)

アーカイブ・ログ・レコードは、オンライン REDO ログが正常にアーカイブまたは消去されるたびに挿入されます。ログが消去される場合、名前列は NULL です。ログを 2 度アーカイブした場合、2 つのアーカイブ・ログ・レコードが同じスレッド番号および順序番号、最初の変更番号とともに挿入されますが、それぞれ異なる名前を持ちます。アーカイブ・ログをバックアップ・セットまたはコピーから復元した場合も、アーカイブ・ログ・レコードが挿入されます。

## V\$BACKUP\_CORRUPTION

V\$BACKUP\_CORRUPTION ビューでは、制御ファイルから、データファイル・バックアップ内の破損に関する情報が表示されます。制御ファイルのバックアップとアーカイブされたログのバックアップ内には、破損があつてはなりません。

列名	列型
RECID	NUMBER
STAMP	NUMBER
SET_STAMP	NUMBER
SET_COUNT	NUMBER
PIECE#	NUMBER
FILE#	NUMBER

## V\$BACKUP\_DATAFILE

---

列名	列型
BLOCK#	NUMBER
BLOCKS	NUMBER
CORRUPTION_CHANGE#	NUMBER
MARKED_CORRUPT	VARCHAR2(3)

## V\$BACKUP\_DATAFILE

V\$bBACKUP\_DATAFILE ビューでは、制御ファイルからのバックアップ・データファイル情報およびバックアップ制御ファイル情報が表示されます。データファイル・バックアップ・セットには、1つの制御ファイルと複数のデータファイルを入れることができます。

列名	列型
RECID	NUMBER
STAMP	NUMBER
SET_STAMP	NUMBER
SET_COUNT	NUMBER
FILE#	NUMBER
CREATION_CHANGE#	NUMBER
CREATION_TIME	DATE
RESETLOGS_CHANGE#	NUMBER
RESETLOGS_TIME	DATE
INCREMENTAL_LEVEL	NUMBER
INCREMENTAL_CHANGE#	NUMBER
CHECKPOINT_CHANGE#	NUMBER
CHECKPOINT_TIME	DATE
ABSOLUTE_FUZZY_CHANGE#	NUMBER
MARKED_CORRUPT	NUMBER
MEDIA_CORRUPT	NUMBER

列名	列型
LOGICALLY_CORRUPT	NUMBER
DATAFILE_BLOCKS	NUMBER
BLOCKS	NUMBER
BLOCK_SIZE	NUMBER
OLDEST_OFFLINE_RANGE	NUMBER

## V\$BACKUP\_DEVICE

バージョン 8 の新しいビュー V\$BACKUP\_DEVICE では、サポートされているバックアップ・デバイスに関する情報が表示されます。デバイス・タイプが指定のデバイスをサポートしていない場合、そのデバイス・タイプについて、デバイス・タイプと NULL のデバイス名を示す行が 1 行戻されます。デバイス・タイプが指定のデバイスをサポートしている場合は、そのタイプの使用可能な各デバイスについて 1 行ずつ戻されます。特殊なデバイス・タイプである DISK は、常に使用可能な状態にあるため、このビューによって戻されません。

列名	列型
DEVICE_TYPE	VARCHAR2(17)
DEVICE_NAME	VARCHAR2(512)

## V\$BACKUP\_PIECE

バージョン 8 の新しいビュー V\$BACKUP\_PIECE では、制御ファイル内に格納される、バックアップ項目に関する情報が表示されます。各バックアップ・セットは、1 つ以上のバックアップ項目からなります。

列名	列型
RECID	NUMBER
STAMP	NUMBER
SET_STAMP	NUMBER
SET_COUNT	NUMBER
PIECE#	NUMBER
DEVICE_TYPE	VARCHAR2(17)

列名	列型
HANDLE	VARCHAR2(513)
COMMENTS	VARCHAR2(81)
MEDIA	VARCHAR2(65)
CONCUR	VARCHAR2(3)
TAG	VARCHAR2(32)
DELETED	VARCHAR2(3)

## V\$BACKUP\_REDOLOG

V\$BACKUP\_REDOLOG ビューでは、制御ファイルからの、バックアップ・セット内のアーカイブされたログに関する情報が表示されます。オンライン REDO ログは、直接バックアップすることはできません。最初にディスクへアーカイブしてからバックアップする必要があります。アーカイブ・ログ・バックアップ・セットには、アーカイブされたログを1つ以上入れることができます。

## V\$BACKUP\_SET

V\$BACKUP\_SET ビューでは、制御ファイルからのバックアップ・セット情報が表示されます。バックアップ・セット・レコードは、バックアップ・セットが正常に完了した後に挿入されます。

列名	列型
RECID	NUMBER
STAMP	NUMBER
SET_STAMP	NUMBER
SET_COUNT	NUMBER
BACKUP_TYPE	VARCHAR2(1)
CONTROLFILE_INCLUDED	VARCHAR2(3)
INCREMENTAL_LEVEL	NUMBER
PIECES	NUMBER
COMPLETION_TIME	DATE
BLOCK_SIZE	NUMBER

## V\$CONTROLFILE\_RECORD\_SECTION

バージョン8の新しいビュー V\$CONTROLFILE\_RECORD\_SECTION では、制御ファイルのレコード・セクションに関する情報が表示されます。

列名	列型
TYPE	VARCHAR2(17)
RECORD_SIZE	NUMBER
RECORDS_TOTAL	NUMBER
RECORDS_USED	NUMBER
FIRST_INDEX	NUMBER
LAST_INDEX	NUMBER
LAST_RECID	NUMBER

## V\$COPY\_CORRUPTION

バージョン8の新しいビュー V\$COPY\_CORRUPTION では、制御ファイルからデータファイル・コピーの破損に関する情報が表示されます。

列名	列型
RECID	NUMBER
STAMP	NUMBER
COPY_RECID	NUMBER
COPY_STAMP	NUMBER
FILE#	NUMBER
BLOCK#	NUMBER
BLOCKS	NUMBER
CORRUPTION_CHANGE#	NUMBER
MARKED_CORRUPT	VARCHAR2(3)

## V\$DATABASE の新規列

V\$DATABASE ビューでは、制御ファイルからのデータベース情報が表示されます。次の V\$DATABASE 列が、バージョン 8 に新たに加わりました。

列名	値
DBID	NUMBER
RESETLOGS_CHANGE#	NUMBER
RESETLOGS_TIME	DATE
CONTROLFILE_TYPE	VARCHAR2(7)
CONTROLFILE_CREATED	DATE
CONTROLFILE_SEQUENCE#	NUMBER
CONTROLFILE_CHANGE#	NUMBER
nCONTROLFILE_TIME	DATE
OPEN_RESETLOGS	VARCHAR2(11)

## V\$DATAFILE の新規列

V\$DATAFILE ビューでは、制御ファイルからのデータファイル情報が表示されます。バージョン 8 の V\$DATAFILE ビューでは、既存のバージョン 7 の V\$DATAFILE ビューの列に加えて、次の新規列も表示されます。データファイル・ヘッダーからの情報を表示する V\$DATAFILE\_HEADER ビューも参照してください。

列名	列型
CREATION_CHANGE#	NUMBER
CREATION_TIME	DATE
TS#	NUMBER
RFILE#	NUMBER
CHECKPOINT_TIME	DATE
UNRECOVERABLE_CHANGE#	NUMBER
UNRECOVERABLE_TIME	DATE
LAST_CHANGE#	NUMBER

列名	列型
LAST_TIME	DATE
OFFLINE_CHANGE#	NUMBER
ONLINE_CHANGE#	NUMBER
ONLINE_TIME#	DATE
BLOCKS	NUMBER
BLOCK_SIZE	NUMBER

## V\$DATAFILE\_COPY

V\$DATAFILE\_COPY ビューでは、制御ファイルからのデータファイル・コピー情報が表示されます。データファイル・コピー・レコードは、データファイル（またはデータファイル・コピー）がバックアップ・セットから正常にコピーまたは復元された後に挿入されます。

列名	列型
RECID	NUMBER
STAMP	NUMBER
NAME	VARCHAR2(512)
TAG	VARCHAR2(32)
FILE#	NUMBER
RFILE#	NUMBER
CREATION_CHANGE#	NUMBER
CREATION_TIME	DATE
RESETLOGS_CHANGE#	NUMBER
RESETLOGS_TIME	DATE
CHECKPOINT_CHANGE#	NUMBER
CHECKPOINT_TIME	DATE
ABSOLUTE_FUZZY_CHANGE#	NUMBER
RECOVERY_FUZZY_CHANGE#	NUMBER

## V\$DATAFILE\_HEADER

---

列名	列型
RECOVERY_FUZZY_TIME	DATE
ONLINE_FUZZY	VARCHAR2(3)
BACKUP_FUZZY	VARCHAR2(3)
MARKED_CORRUPT	NUMBER
MEDIA_CORRUPT	NUMBER
LOGICALLY_CORRUPT	NUMBER
BLOCKS	NUMBER
BLOCK_SIZE	NUMBER
OLDEST_OFFLINE_RANGE	NUMBER
COMPLETION_TIME	DATE
DELETED	VARCHAR2(3)

## V\$DATAFILE\_HEADER

バージョン 8 の新しいビュー V\$DATAFILE\_HEADER では、データファイル・ヘッドからのデータファイル情報が表示されます。

列名	列型
FILE#	NUMBER
STATUS	VARCHAR2(7)
ERROR	VARCHAR2(18)
RECOVER	VARCHAR2(3)
FUZZY	VARCHAR2(3)
CREATION_CHANGE#	NUMBER
CREATION_TIME	DATE
TABLESPACE_NAME	VARCHAR2(30)
TS#	NUMBER
RFILE#	NUMBER

列名	列型
RESETLOGS_CHANGE#	NUMBER
RESETLOGS_TIME	DATE
CHECKPOINT_CHANGE#	NUMBER
CHECKPOINT_TIME	DATE
CHECKPOINT_COUNT	NUMBER
BYTES	NUMBER
BLOCKS	NUMBER
NAME	VARCHAR2(512)

## V\$DELETED\_OBJECT

バージョン 8 の新しいビュー V\$DELETED\_OBJECT では、削除済みのアーカイブされたログおよびデータファイル・コピー、制御ファイルからのバックアップ項目に関する情報が表示されます。このビューの目的は、リカバリ・カタログの再同期を最適化することにあります。アーカイブされたログまたはデータファイル・コピー、バックアップ項目が削除された場合、それらに対応するレコードに削除済みのマークが付けられます。

列名	列型
RECID	NUMBER
STAMP	NUMBER
TYPE	VARCHAR2(13)
OBJECT_RECID	NUMBER
OBJECT_STAMP	NUMBER

## V\$INSTANCE

バージョン 8 の新しいビュー V\$INSTANCE では、データベース・インスタンスの状態情報が表示されます。バージョン 8 の V\$INSTANCE は 1 つの行だけを戻し、1 つの値につき行を 1 つ戻すバージョン 7 のビュー V\$INSTANCE とは互換性がありません。STARTUP\_TIME および LOGINS、SHUTDOWN\_PENDING の各列は、バージョン 7 の V\$INSTANCE と同じ情報を戻します。

## V\$INSTANCE

---

```
SVRMGR> select startup_time, logins, shutdown_pending from v$instance;
```

STARTUP_TIME	LOGINS	SHUTDOWN_PENDING
96.05.29 20:29:38	ALLOWED	NO

```
SQLDBA> select * from v$instance;
```

KEY	VALUE
RESTRICTED MODE	0
SHUTDOWN PENDING	0
STARTUP TIME - JULIAN	2450215
STARTUP TIME - SECONDS	50407
4つの行を選択	

バージョン 8 の V\$INSTANCE ビューはバージョン 7 の V\$INSTANCE ビューと互換性がありますが、同等またはそれ以上の情報を提供します。

列名	列型
INSTANCE_NUMBER	NUMBER
INSTANCE_NAME	VARCHAR2(16)
HOST_NAME	VARCHAR2(64)
VERSION	VARCHAR2(17)
STARTUP_TIME	DATE
STATUS	VARCHAR2(7)
PARALLEL	VARCHAR2(3)
THREAD#	NUMBER
ARCHIVER	VARCHAR2(7)
LOG_SWITCH_WAIT	VARCHAR2(11)
LOGINS	VARCHAR2(10)
SHUTDOWN_PENDING	VARCHAR2(3)

インスタンスが起動されていない場合に、このビューから問合せを行うとORA-1034が発生します。

## V\$OFFLINE\_RANGE

V\$OFFLINE\_RANGEビューでは、制御ファイルからのオフライン範囲情報が表示されます。各データファイルの最新のオフライン範囲は、データファイル・レコードに格納されています。V\$OFFLINE\_RANGE: 制御ファイルからのデータファイル・オフライン範囲情報は、次のとおりです。

列名	値
RECID	NUMBER
STAMP	NUMBER
FILE#	NUMBER
OFFLINE_CHANGE#	NUMBER
ONLINE_CHANGE#	NUMBER
ONLINE_TIME#	DATE

各データファイルの最後のオフライン範囲は、データファイル・レコードに保持されます。V\$DATAFILE ビューを参照してください。

## V\$RESOURCE\_LIMIT

バージョン 8 の新しいビュー V\$RESOURCE\_LIMIT は、システム・リソースのグローバル・リソース使用率に関する情報を表示します。

列名	列型
RESOURCE_NAME	VARCHAR2(30)
CURRENT_UTILIZATION	NUMBER
MAX_UTILIZATION	NUMBER
INITIAL_ALLOCATION	VARCHAR2(10)
LIMIT_VALUE	VARCHAR2(10)

## V\$TABLESPACE

バージョン8の新しいビュー V\$TABLESPACE では、制御ファイルからの表領域情報が表示されます。

列名	値	説明
TS#	NUMBER	表領域番号
NAME	VARCHAR2(30)	表領域名

## V\$THREAD

V\$THREAD ビューには、制御ファイルのスレッド情報が含まれます。

列名	値	説明
ENABLE_CHANGE#	NUMBER	スレッドを使用可能にしたときの SCN
ENABLE_TIME	DATE	使用可能 SCN の時刻
DISABLE_CHANGE#	NUMBER	スレッドを使用不能にしたときの SCN
DISABLE_TIME	DATE	使用不能 SCN の時刻

## 変更された列型

バージョン8では、次のタイム・スタンプ列のデータ型が、バージョン7の VARCHAR2(20) から DATE に変更されました。

ビュー名	列
V\$BACKUP	TIME
V\$DATABASE	CREATED
V\$LOG	FIRST_TIME
V\$LOG_HISTORY	FIRST_TIME ( 以前は TIME )
V\$LOGHIST	FIRST_TIME
V\$RECOVER_FILE	TIME
V\$RECOVERY_STATUS	RECOVERY_CHECKPOINT
V\$RECOVERY_STATUS	TIME_NEEDED

ビュー名	列
V\$THREAD	OPEN_TIME
V\$THREAD	CHECKPOINT_TIME

## データベース・スケジュール機能

データベース・スケジュール機能の主な目的は、次の事柄をさらに制御できるようにすることです。

- 変更された固定ビュー
- 新しい固定ビュー

### 変更された固定ビュー

- V\$SESSION では、SCHEDULER\_CLASS 列が追加された。
- V\$MYSESSION では、SCHEDULER\_CLASS 列が追加された。
- DBA\_USERS では、DEFAULT\_SCHEDULER\_CLASS 列が追加された。
- USER\_USERS では、DEFAULT\_SCHEDULER\_CLASS 列が追加された。

### 新しい固定ビュー

- V\$SERVER\_THRESHOLD: 現行のサーバーしきい値に関するデータ
- V\$SCHEDULER\_CLASS: 現在アクティブなスケジューラ・クラスに関するデータ
- V\$COMMON\_SERVER: 現在アクティブな共通サーバーに関するデータ
- V\$COMMON\_SERVER\_REQUEST: 現在処理中のリクエストのタイプに関するデータ
- V\$COMMON\_SERVER\_REQUEST\_HISTORY: すでに処理されたリクエストのタイプに関するデータ
- V\$COMMON\_SERVER\_QUEUE: 現在アクティブな共通サーバーのキューに関するデータ
- DBA\_SCHEDULER\_CLASS\_PRIVS: 全スケジューラ・クラスおよびそれらが付与されたユーザーまたはロールを表示する、新しい DBA ビュー
- DBA\_SCHEDULER\_CLASSES: データベース内にある全スケジューラ・クラスを表示する新しい DBA ビュー
- DBA\_SCHEDULER\_PLANS: データベース内にある全スケジューラ計画を表示する新しい DBA ビュー
- DBA\_SCHEDULER\_PLAN\_ENTRIES: データベース内にある全スケジューラ計画エントリを表示する新しい DBA ビュー

## 表（ビュー）名の変更

---

- SCHEDULER\_CLASS\_POLICIES: スケジューラ・クラスについて使用可能な、全スケジューリング方法を表示する新しい DBA ビュー
- SCHEDULER\_PLAN\_POLICIES: スケジューラ計画について使用可能な、全スケジューリング方法を表示する新しい DBA ビュー
- USER\_SCHEDULER\_CLASS\_PRIVS: ユーザーに付与された全スケジューラ・クラスを表示する新しい PUBLIC ビュー

## 表（ビュー）名の変更

バージョン 8 より前のビュー名	バージョン 8 での表名
ALL_HISTOGRAMS	ALL_TAB_HISTOGRAMS
DBA_HISTOGRAMS	DBA_TAB_HISTOGRAMS
USER_HISTOGRAMS	USER_TAB_HISTOGRAMS

---

## バージョン 8 の INIT.ORA の変更点

バージョン 8 では、新しい INIT.ORA 初期化パラメータをサポートする一方で、リリース 7.3 のパラメータの一部は変更または廃止されています。

この付録では、次のトピックについて説明します。

- COMPATIBLE パラメータ
- データ・ディクショナリの保護
- DML\_LOCKS
- NCHAR パラメータと NLS パラメータ、および互換性
- バージョン 8 で改名されたバージョン 8 より前のパラメータ
- バージョン 8 で廃止されたリリース 7.3 のパラメータ
- レプリケーション・サーバー用の REPLICATION\_DEPENDENCY\_TRACKING
- バージョン 8 ではサポートされなくなっている機能

## COMPATIBLE パラメータ

COMPATIBLE パラメータは、Oracle Server が互換性を維持しなければならないリリースを指定します。INIT.ORA ファイル内の COMPATIBLE パラメータ設定により、Oracle Server の機能が決まります。したがって、これを正しく設定することが重要です。

ただし、COMPATIBLE パラメータは、移行またはアップグレード、ダウングレードの各処理において特定の時点で設定する必要があります。したがって、適切な章に記載の手順に従い、設定を指示された場合にだけ COMPATIBLE パラメータを設定するようにしてください。操作が完了すれば、必要なときはいつでも COMPATIBLE パラメータを設定できます。

現行の COMPATIBLE パラメータの設定をチェックするときは、次のコマンドを発行します。

```
SVRMGR> SELECT name, value, description FROM v$parameter  
WHERE name="compatible";
```

---

警告：バージョン 8 へ移行する場合は、COMPATIBLE パラメータがバージョン 7 のリリースに設定されていないことを確認してください。

---

### リリース 8.0.4 への移行またはアップグレード

リリース 8.0.4 へ移行またはアップグレードする場合、COMPATIBLE パラメータはデフォルトで次のように設定されています。

```
COMPATIBLE=8.0.0.0.0
```

ユーザーが COMPATIBLE パラメータをまったく設定しなければ、この値に設定されます。この設定には COMPATIBLE を 8.0.3.0.0 に設定するのと同じ効果があり、後でリリース 8.0.3 へのダウングレードを選択した場合には、簡単にダウングレードできます。

ただし、デフォルト設定 ( 8.0.0.0.0 ) を使うと、次のようなリリース 8.0.4 の新機能は使えなくなります。

- アドバンスド・キューイング ( AQ ) での伝播
- SCN 生成の改善

これらリリース 8.0.4.0.0 の新機能を使う場合は、COMPATIBLE パラメータを次のように設定します。

```
COMPATIBLE=8.0.4.0.0
```

---

注意：COMPATIBLE パラメータを 8.0.4.0.0 に設定すると、前のリリースへダウングレードする際に必要な作業が増えることがあります。しかし、前のリリースへダウングレードする予定がなければ、オラクル社では 8.0.4.0.0 の設定をお勧めします。

---

以下の節では、COMPATIBLE パラメータを 8.0.4.0.0 に設定する必要がある新機能の概要について簡単に説明します。

#### アドバンスト・キューイング (AQ) での伝播

リリース 8.0.4 では分散オブジェクト型をサポートしていませんので、標準データベース・リンクを使ってリモート・エンキューまたはデキューをインプリメントすることはできません。ただし、リリース 8.0.4 では、AQ のメッセージ伝播を使って、リモート・キューにエンキューすることができます。

たとえば、データベース X に接続し、その中にあるたとえば "DROPBOX" というキューにメッセージをエンキューできます。"DROPBOX" キューにエンキューされたメッセージのすべてが、データベース Y の中の別のキューに自動的に伝播されるように AQ を構成することができます。このとき、データベース Y がローカルであるかリモートであるかは関係ありません。AQ は、データベース Y の中のリモート・キューの型がデータベース X の中のローカル・キューの型と構造的に等しいかどうかを自動的にチェックし、メッセージを伝播します。

伝播されたメッセージの受信側は、アプリケーションまたはキューのどちらでもかまいません。受信側がキューの場合、実際の受信側は受信側キューに対応する加入リストによって決まります。キューがリモートの場合、メッセージは指定された DBLINK を使って伝播されます。AQ から AQ へのメッセージ伝播だけがサポートされています。

関連項目: AQ における伝播の詳細は、『Oracle8 Server アプリケーション開発者ガイド』を参照してください。

#### SCN 生成の改善

リリース 8.0.4 では、システム変更番号 (SCN) の生成用に改善されたアルゴリズムをインプリメントしています。INIT.ORA パラメータの MAX\_COMMIT\_PROPAGATION\_DELAY を 7 秒より小さい値に設定して Oracle Parallel Server を実行する場合、COMPATIBLE を 8.0.4.0.0 に設定することをお勧めします。

関連項目: SCN の生成に関する詳細は、『Oracle8 Server 概要』を参照してください。

## データ・ディクショナリの保護

O7\_DICTIONARY\_ACCESSIBILITY は INIT.ORA のパラメータ・スイッチであり、バージョン 7 のデータ・ディクショナリの動作を継承しています。これは一時的な機能であり、将来のバージョン 8 のリリースでは提供されません。

## DML\_LOCKS

バージョン 8 のシステムでは、通常、DDL 操作の実行中にバージョン 7 のシステムで必要とする以上の DML ロックを使用します。それにもかかわらず、バージョン 7 の DML\_LOCKS パラメータのデフォルト設定はバージョン 8 のシステムにも適しており、DML を非常に多く使用するアプリケーションにも適しています。

DML\_LOCKS のデフォルト値はトランザクションの数の倍数であり、ロールバック・セグメントの数を基に計算されます。ただし、バージョン 8 ではロールバック・セグメント当たりの使用トランザクション数がバージョン 7 よりも少なくなっています。したがって、バージョン 8 では DML\_LOCKS のデフォルト値が低くなっています。ある極端なテスト条件の下では、バージョン 8 のシステムは (バージョン 7 の) DML ロック制限を超え、DML\_LOCKS パラメータ値を増やす必要がありました。

また、TRANSACTION\_PER\_ROLLBACK\_SEGMENT パラメータ設定の調整が必要になる場合があります。それはプラットフォームに固有の設定によって決まります。この変更に関する情報メッセージが、データベースの起動操作中に表示されることがあります。たとえば、3-13 ページの「バージョン 8 の環境での移行ステップ」のステップ 12 などです。

## NCHAR パラメータと NLS パラメータ、および互換性

次に示すように、使用環境に NLS\_LANG を設定する必要があります。

- リリース 7.3.x の環境では ORA\_NLS32 を設定する。
- リリース 8.0 の環境では ORA\_NLS33 を設定する。

クライアント側に正しい NLS キャラクタ・セットの環境変数があることを確認してください。Oracle 7.3 の NLS コードがバージョン 8 のキャラクタ・セットをロードしようとする、エラーが発生します。

## バージョン 8 で改名されたバージョン 8 より前のパラメータ

バージョン 8 では、次の初期化パラメータが改名されました。

バージョン 8 より前の名前	バージョン 8 での名前
ASYNC_READ	DISK_ASYNC_IO
ASYNC_WRITE	DISK_ASYNC_IO
DB_FILE_STANDBY_NAME_CONVERT	DB_FILE_NAME_CONVERT
DB_WRITERS	DBWR_IO_SLAVES
LOG_FILE_STANDBY_NAME_CONVERT	LOG_FILE_NAME_CONVERT
SNAPSHOT_REFRESH_INTERVAL	JOB_QUEUE_INTERVAL
SNAPSHOT_REFRESH_PROCESS	JOB_QUEUE_PROCESSES
USE-ASYNC_IO	DISK_ASYNC_IO

関連項目：初期化パラメータに関する詳細は『Oracle8 Server リファレンス・マニュアル』を参照してください。

---

注意：上記のパラメータの一部は、プラットフォームに固有のもので、お使いのプラットフォームでの改名された初期化パラメータの詳細は、プラットフォーム固有の Oracle マニュアルを参照してください。

---

## バージョン 8 で廃止されたリリース 7.3 のパラメータ

次のパラメータは、バージョン 8 では廃止されています。

CCF\_IO\_SIZE  
CHECKPOINT\_PROCESS  
GC\_DB\_LOCKS  
GC\_FREELIST\_GROUPS  
GC\_ROLLBACK\_SEGMENTS  
GC\_SAVE\_ROLLBACK\_LOCKS  
GC\_SEGMENTS  
GC\_TABLESPACES  
IO\_TIMEOUT  
INIT\_SQL\_FILES  
IPQ\_ADDRESS  
IPQ\_NET  
LM\_DOMAINS  
LM\_NON\_FAULT\_TOLERANT  
OPTIMIZER\_PARALLEL\_PASS  
PARALLEL\_DEFAULT\_MAX\_SCANS  
PARALLEL\_DEFAULT\_SCAN\_SIZE  
SEQUENCE\_CACHE\_HASH\_BUCKETS  
SERIALIZABLE  
SESSION\_CACHED\_CURSORS  
UNLIMITED\_ROLLBACK\_SEGMENTS  
USE\_IPQ  
USE\_READV

USE\_SIGIO

V733PLANS\_ENABLED

関連項目：初期化パラメータに関する詳細は、『Oracle8 Server リファレンス・マニュアル』を参照してください。

---

注意：この項に示すパラメータの一部は、プラットフォームに固有のもので、使用しているプラットフォームで廃止されている初期化パラメータの詳細は、プラットフォーム固有の Oracle マニュアルを参照してください。

---

## レプリケーション・サーバー用の REPLICATION\_DEPENDENCY\_TRACKING

REPLICATION\_DEPENDENCY\_TRACKING パラメータは、DSCN 追跡のオンまたはオフを調整するときに使います。

TRUE（デフォルト値）では、データベースに対する読取り/書込み操作について依存性追跡を ON にします。依存性追跡は、レプリケーション・サーバーが変更を並列に伝播するのに必須です。

FALSE では、依存性追跡を OFF にします。この条件によってデータベースに対する読取り/書込み操作が高速になりますが、レプリケーション・サーバーが並列伝播を実行するための依存情報は作成されません。

---

警告：アプリケーションが任意の複製表に対して読取り/書込み操作を実行する場合、REPLICATION\_DEPENDENCY\_TRACKING を FALSE には設定しないでください。

---

## バージョン 8 ではサポートされなくなっている機能

この項では、バージョン 8 で廃止された以前の Oracle バージョンまたはリリースの INIT.ORA ファイル・パラメータに対応する機能について説明します。Oracle の今後のリリースでは、これらの機能はサポートされない可能性もあります。したがって、バージョン 8 においても、これらの機能はサポートされていないものとお考えください。

### SERIALIZABLE=TRUE または \_SERIALIZABLE

バージョン 8 では、INIT.ORA ファイルのパラメータ SERIALIZABLE=TRUE または \_SERIALIZABLE は、自動表レベルロック ON READ を意味します。これは、バージョン 8 のどのインストールでも使うべきではありません。8.1 より後のメンテナンス・リリースからは永久に削除される可能性があります。

# D

---

## 新しい SQL キーと予約語

この付録では、バージョン 8 に新たに追加されたいくつかのキーワードと 1 つの予約語の一覧を示します。

これらの特殊語をデータベース・オブジェクト(表、列など)の名前として使用した場合は、問題が発生する可能性があります。加えて、表や列の名前には、インストール時に使用する可能性のある他の言語またはユーティリティによって予約されている用語の使用は避けるべきです。予約語の各種一覧については、さまざまな言語マニュアルおよびリファレンス・マニュアルとこの付録を参照してください。

SQL によって予約されているワードの一覧は、『Oracle8 Server SQL リファレンス』を参照してください。予約語を表または列の名前として使用する場合は、その予約語を二重引用符で囲んで指定する必要があります。バージョン 8 で使用される特殊文字および予約語、キーワードの一覧を、以下に示します。バージョン 8 で使用される新しいワードは、1 つの予約語 ( VALUE ) を除いてすべてキーワードです。

ACCOUNT	NCLOB
ARRAY	NESTED
BFILE	NOLOGGING
BLOB	NOPARALLEL
CAST	NOREVERSE
CFILE	NORMAL
CHAR_CS	NVARCHAR2
CHUNK	OBJECT
CLOB	OBJNO_REUSE
CLONE	OID
DANGLING	OIDINDEX
DATAOBJNO	ORGANIZATION

---

DEFERRABLE	OVERFLOW
DEFERRED	PASSWORD
DEREF	PCTTHRESHOLD
DIRECTORY	PCTVERSION
ENFORCE	PRESERVE
EXCHANGE	PURGE
EXPIRE	QUEUE
EXTENT	REF
FLOB	REPLACE
GLOBALLY	RETURN
HASH	RETURNING
HASHKEYS	REVERSE
HEAP	SCOPE
IDGENERATORS	SEG_BLOCK
INITIALLY	SEG_FILE
LIBRARY	SKIP
LOCKED	SYS_OP_NTCIMG
LOGGING	THAN
LOGICAL_READS_PER_CALL	THE
LOGICAL_READS_PER_SESSION	TOPLEVEL
MASTER	UNLOCK
NATIONAL	USAGE
NCHAR	VALUE (予約語)
NCHAR_CS	VARYING

---

## 移行に関する一般的なシステム要件

この付録では、Oracle RDBMSの移行、アップグレード、ダウングレードを正常に行うために満たさなければならないシステム要件について説明します。バージョン7から（またはエクスポート/インポートによりバージョン6から）バージョン8にデータベースを移行する場合は、オペレーティング・システムとハードウェアに特定の構成が必要になります。

この付録では、次のトピックについて説明します。

- メモリー要件
- Oracle Parallel Server の使用
- バージョン 8 の新しいサイズおよび制限

## メモリー要件

バージョン 8 のメモリー要件は、基本メモリー要件および実行可能ファイルに必要なメモリー、同時アクセスの量からなります。

### 基本メモリー要件

バージョン 8 では、データベースを実行するのに最低でも 48 メガバイトの RAM を必要とします。これは、ディスクに大量のオペレーティング・システム・メモリーをスワッピングしない場合の最小構成です。複数ユーザーの接続をサポートするには、システムに増設メモリーが必要です。Enterprise Manager の同時使用では、20MB 増設する必要があります。

関連項目： 使用しているプラットフォームでのメモリー要件については、プラットフォームに固有の Oracle マニュアルを参照してください。

### バージョン 8 の実行可能ファイル

バージョン 8 の実行可能ファイルはバージョン 7 の実行可能ファイルの 3 倍の大きさがありますが、これは主にバージョン 8 に追加された新機能のためです。通常 3MB のバージョン 7 の実行可能ファイル・セットでは、バージョン 8 への移行時にこれらの実行可能ファイル用に必要となる領域は、約 9MB になります。必要な領域サイズが 3 倍に増えるということは、大規模なバッチ・システム（数十個または数百個単位の実行可能ファイルを生成する）においては特に注意が必要です。実行可能ファイルに必要な領域は、バージョン 8 の環境に組み込むオプションによっても異なります。たとえば、次のようなオプションがあります。

- Oracle Parallel Server（『Oracle8 Parallel Server 概要および管理』を参照）
- Net8 または SQL\*Net の使用（『Oracle Net8 管理者ガイド』を参照）

バージョン 7 からバージョン 8 へ移行するときは、オプションの組込みに対応してシステム・メモリーを調整する必要があります。

---

注意： 多数のプラットフォーム上で共有オブジェクトを使用すると、クライアント・アプリケーションのサイズが多少減少します。しかし、一部の共有オブジェクト（またはライブラリ）のサイズは、バージョン 8 の新機能のためにかなり大きくなる場合があります。

---

### 同時アクセス

バージョン 8 のシステムのメモリー・サイズも、同時アクセスとその実現方法によって決まります。バージョン 8 では、次の接続オプションをサポートしています。

オプション 1: 専用サーバー・アーキテクチャでローカル接続を使用する（「2 タスク Oracle」ともいう）。このオプションは、バージョン 7 と同じです。

オプション 2: SQL\*Net によるリモート接続を使用する。このオプションは、バージョン 7 と同じです。

オプション 3: ローカル接続とリモート接続の両方にマルチスレッド共有サーバーを使用する。

オプション 4: トランザクション・プロセッサ (TP) モニターを使用する。

オプション 1 は、オプション 2 またはオプション 3 よりも多くのメモリーを必要とします。オプション 1 では、クライアント・アプリケーションとその Oracle Server (またはシャドウ) プロセスの両方が同じマシン上に常駐する場合、その両方のためにメモリーが必要です。たとえば、100 個のクライアント・アプリケーション・プロセスをバージョン 8 に接続した場合、100 個の Oracle Server プロセスがシステムに追加される結果となり、プロセスは合計 200 個になります。

オプション 2 の場合、Oracle プロセスだけがシステムに常駐し、クライアント・プロセスはリモートで接続されます。したがって、考慮する必要があるのは、Oracle Server プロセスのサイズおよび使用可能な共有メモリーのサイズだけです。

オプション 3 はマルチスレッド・サーバー機能を使用し、バージョン 7 と同じです。マルチスレッド・サーバー機能を使用した場合、専用の Oracle シャドウ・プロセスを持たずに、ローカル・クライアントまたはリモート・クライアントの複数のプロセスを単一のディスパッチャ・プロセスに接続できます。マルチスレッド・サーバー構成はパフォーマンス向上を意図したものではありませんが、バージョン 8 のサーバー上で多数の同時接続を実現するようになるので、スループットが向上します。複数のクライアントが単一のディスパッチャに接続できるため、同時ユーザー接続のためのメモリー使用量は減少します。バージョン 8 のマルチスレッド・サーバー機能の詳細は、『Oracle8 Server 概要』を参照してください。

オプション 4 は TP モニターを使用し、多数の (数百人を越える) ユーザー全員が OLQP/OLTP タイプのトランザクションを実行する必要があるシステムのための選択肢です。通常、そのようなトランザクションは持続時間が短く、ユーザーはデータベースに直接接続する必要はありません。すべてのトランザクションは、TP (トランザクション・プロセッサ) モニター・サービスを経由したメッセージを使用して実行されます。TP レイヤーは名前付きサービスを提供し、Oracle を含めた多様な DBMS システムとサービス要求を調整します。

---

注意: TP モニターを使用するための要件は非常に多様であり、このマニュアルでは説明しません。システム要件については、該当の TP モニターのベンダーに問い合せてください。

---

単一システムの場合には、次の要因を考慮すればシステム・メモリー要件を見積もることができます。

- オープン・カーソル、および特定の Oracle アプリケーション・セッションでソートが発生させることのあるカーソルの平均数

- オープン・カーソルおよびソート領域を含む Oracle シャドウ・プロセスの平均サイズ
- システムの同時ユーザーのピーク数
- Oracle フロントエンド・アプリケーションの平均メモリー・サイズ

## Oracle Parallel Server の使用

Oracle Parallel Server (DLM を含む) を移行するには、各ノードにバージョン7およびバージョン8のソフトウェアのコピーがなければなりません。したがって、IBM RS6000、NCR 3XXX シリーズ、Pyramid MIS Server、Sequent Symmetry、Sun SPARCcenterなどの多数のプラットフォームで、プライベート・ディスクの追加が必要になる場合があります。

## バージョン 8 の新しいサイズおよび制限

バージョン 8 では、次の一覧に示すとおり、さまざまな制限の容量を増やしています。

データ型：

- CHAR は 2000 バイトに増加。
- VARCHAR/VARCHAR2 は 4000 バイトに増加。
- NCHAR は 2000 バイトまで格納可能。
- NCHAR VARYING は 4000 バイトまで格納可能。

列と索引：

- データベース・サイズは、プラットフォームの容量だけによって制限される。
- 表領域あたりのデータ・ファイル最大数は 1022。
- 表内の列の最大数が 1000 に増加。
- 主キーは 16 フィールドまでに制限される。

## CHAR および NCHAR の最大サイズのサポート

バージョン 8 では、いくつかのデータ型について、その最大文字列長を増やしました。

- CHAR の最大長は 2000 バイト ( RAW データ型も同様 )  
( Oracle 7.3 では、CHAR および RAW の長さとして最大 255 バイトをサポート )
- VARCHAR2 の最大長は 4000 バイト  
( Oracle 7.3 では、VARCHAR2 の長さとして最大 2000 バイトをサポート )

バージョン 8 のデータ型文字列の最大長は NTT/MIA の仕様に従っており、次の文字列長が必要とされます。

- CHAR の最大長は最低 255 バイト

- NCHAR の最大長は 127 文字
- NCHAR VARYING(NVARCHAR) の最大長は 2000 文字
- VARCHAR2 の最大長は 4000 バイト



---

# 索引

## A

Ada, SQL\*Module 6-4  
Ada 用の SQL\*Module 6-4  
ALTER DATABASE CONVERT コマンド  
    移行ユーティリティ 3-2  
ALTER DATABASE OPEN RESETLOGS コマンド  
    移行ユーティリティ 3-3  
ALTER TABLESPACE コマンド  
    OFFLINE IMMEDIATE または OFFLINE TEMPORARY 3-6

## AQ

aq\$\_agent データ型 8-6  
CATNOQUEUE.SQL 8-6  
CATQUEUE スクリプト 8-6  
新しいディクショナリ表 8-6  
アップグレード 8-5  
拡張アドレス・フィールド 8-6

## AS 句

領域要件 2-8

## B

BFILE キーワード  
    Oracle8 および Oracle7 での動作 6-8  
BLOB キーワード  
    Oracle8 および Oracle7 での動作 6-8

## C

CAST キーワード  
    Oracle8 および Oracle7 での動作 6-8  
CAT8000.SQL スクリプト 3-16  
CAT8004.SQL スクリプト 8-3

CAT8004D.SQL スクリプト 8-8  
CATALOG.SQL スクリプト 3-16, 8-3, 8-9  
    移行の中止 3-18  
CATALOG5.SQL、Oracle8 では廃止 6-6  
CATALOG6.SQL、Oracle8 では廃止 6-6  
CATEXP.SQL スクリプト 6-15  
CATNOQUEUE.SQL スクリプト 8-6  
CATOUTU.LOG 8-3, 8-9  
CATPROC.SQL スクリプト 3-16, 8-3, 8-9  
    移行の中止 3-18  
CATQUEUE.SQL スクリプト 8-6  
CATREP.SQL スクリプト 3-16, 3-19, 8-3, 8-9  
CATREP8M.SQL スクリプト 3-16  
CATREPARR.SQL スクリプト 3-17, 8-3, 8-9  
CHAR  
    サイズ E-4  
CHARACTER\_SET\_NAME 6-15  
CHARACTER キーワード  
    Oracle8 および Oracle7 での動作 6-8  
CHECK\_ONLY  
    オプションの使用 3-4  
CHECK\_ONLY 移行パラメータ 3-9  
CLOB キーワード  
    Oracle8 および Oracle7 での動作 6-8  
COMMIT キーワード  
    Oracle8 および Oracle7 での動作 6-8  
COMPATIBLE パラメータ C-2  
CONNECT INTERNAL 3-10  
COPY コマンド  
    大きいクラスタ表 2-8  
    クラスタ表 2-8

領域要件 2-8  
CREATE DATABASE コマンド  
文字のコード化 3-5

## D

DB\_BLOCK\_SIZE  
ブロック・サイズの考慮事項 3-4  
DB\_NAME 移行パラメータ 3-9  
DBA 1-4  
任務 1-4  
DBMS  
プリコンパイラ・コマンド行オプション 6-4  
DBMS\_LOB パッケージ  
LOADFROMFILE プロシージャ 6-5  
WRITE プロシージャ 6-5  
DEC キーワード  
Oracle8 および Oracle7 での動作 6-8  
DEREF キーワード  
Oracle8 および Oracle7 での動作 6-8  
Developer/2000 アプリケーション  
アップグレード 6-6  
DLM E-4  
DML\_LOCKS  
INIT.ORA において C-3  
DTYCHR 型 7-7

## E

EBU  
バックアップ管理 6-11  
EXPLAIN PLAN  
テスト 2-12

## F

FALSE キーワード  
Oracle8 および Oracle7 での動作 6-8  
Forms  
Oracle Forms アプリケーションの移行 6-6  
Forms アプリケーション  
アップグレード 6-6  
forms アプリケーション  
Oracle8 での実行 6-6

## I

INIT.ORA パラメータ  
廃止 C-5

INIT.ORA ファイル  
COMPATIBLE パラメータ C-2  
INTO パラメータ  
テスト 2-12  
INT キーワード  
Oracle8 および Oracle7 での動作 6-8

## L

LOADFROMFILE プロシージャ  
DBMS\_LOB パッケージ 6-5  
LOB データ型  
LONG の変換 6-5  
LONG データ型  
LOB データ型へのアップグレード 6-5

## M

MIGRATE.BSQ スクリプト 3-12  
MIGRATE ユーザーの削除 3-16  
MIGRATE ユーザー、回避 3-6

## N

NCHAR  
Oracle8 での使用 6-15  
移行 6-15  
互換性および相互運用性 6-16  
互換性の問題 6-4  
サイズ E-4  
NCHAR VARYING  
サイズ E-4  
NCHAR および NLS の使用 6-15  
NCHAR キーワード  
Oracle8 および Oracle7 での動作 6-8  
NCLOB キーワード  
Oracle8 および Oracle7 での動作 6-8  
Net8  
アップグレードされたアプリケーション 6-9  
NEW\_DBNAME 移行パラメータ 3-9  
NLS  
移行 6-15  
互換性および相互運用性 6-16  
NLS\_LANG 設定 C-4  
NLS\_NCHAR\_CHARACTERSET 6-15  
NLS\_NCHAR 移行パラメータ 3-9  
NLS および NCHAR の使用 6-15  
NLS キャラクタ・セット C-4

NO\_SPACE\_CHECK 移行パラメータ 3-9  
NUMERIC キーワード  
Oracle8 および Oracle7 での動作 6-8  
NVARCHAR2 キーワード  
Oracle8 および Oracle7 での動作 6-8

## O

O7\_DICTIONARY\_ACCESSIBILITY スイッチ 6-11  
OCI/UIP クライアント  
Net8 での再リンク 6-10  
OCIChangePassword コール 6-12  
OCILIB、OCI ライブラリ 6-4  
OCILobLoadFromFile コマンド 6-5  
OCILobWrite コマンド 6-5  
OCI-Net2 クライアント  
要件 6-9  
OCISessionBegin コール 6-12  
OCI アプリケーション  
Oracle8 のリンク・ライン 6-5  
リンク・ライン 6-5  
OLQP E-3  
OLTP E-3  
ORA\_NLS32 C-4  
ORA\_NLS33 3-10, C-4  
Oracle6  
Oracle コール・インタフェース ( OCI ) ライブラリ 6-5  
エクスポート/インポート・ユーティリティ 4-3  
Oracle6 の OCI ライブラリ  
未サポート 6-5  
Oracle8 および Oracle8 Enterprise Edition xi  
アップグレード 8-4  
Oracle8 の ROWID  
パーティション表および相対表領域データ・ブロック・アドレス 7-1  
Oracle コール・インタフェース ( OCI )  
Oracle8 の機能 6-10  
アプリケーション移行の準備 6-2  
アプリケーションの移行 6-4  
スレッド・セーフティの互換性の問題 6-13  
Oracle リリース  
移行に必要な 2-9

## P

Parallel Server のオプション

アップグレード 8-3  
システム要件 E-4  
ダウングレード 8-9  
PFILE 移行パラメータ 3-9  
PL/SQL  
V2 と V8.0.3 の間の互換性 6-7  
V2 の互換モード 6-7  
下位互換性 6-4  
PL/SQL 変数  
NCHAR および NLS 6-15  
PLSQL\_V2\_COMPATIBILITY フラグ 6-4, 6-7  
Pro\*Ada  
Ada 用 SQL\*Module へのアップグレード 6-4  
Pro\*C 3 6-3  
Pro\*C/C++  
統合テスト 2-11  
Pro\*C/C++ 2.2  
Oracle8 Server との互換性 6-3  
Pro\*C/C++ 3.0  
Oracle7 との互換性 6-3  
props\$ ビュー 3-9  
NCHAR および NLS 6-15

## R

REAL キーワード  
Oracle8 および Oracle7 での動作 6-8  
Recovery Manager  
バックアップ管理 6-11  
REF キーワード  
Oracle8 および Oracle7 での動作 6-8  
ROWID 2-9  
移行  
質問および回答 7-6  
互換性、クライアント・アクセス 7-6  
スナップショット・リフレッシュ 7-6  
ROWID 上の索引 7-7  
ROWID の使用方法  
DBMS\_ROWID 互換性パッケージ 7-2  
ROWID の変換 7-3, 7-4  
例 7-5

## S

SAVEPOINT キーワード  
Oracle8 および Oracle7 での動作 6-8  
SCN 生成

- 改善 C-3
- SELECT リスト
  - キーワード動作の違い
    - Oracle7 と Oracle8 6-8
- SERIALIZABLE=TRUE
  - 非サポート C-6
- SET COMPATIBILITY コマンド 6-6
- SHUTDOWN ABORT、使用しない 3-17
- SPOOL 移行パラメータ 3-9
- SQL
  - キーワード D-1
  - 特殊キャラクタ D-1
  - 予約語 D-1
- SQL\*Module
  - Ada 用 6-4
- SQL\*Net
  - Oracle8 で使用する SQL\*Net バージョン 2 および Net8 6-2
  - SQL\*Net バージョン 1 は Oracle8 では使用されない 6-2
  - アップグレードされたアプリケーション 6-9
  - 移行 6-9
  - 再リンク 6-2
  - バージョン 1 からバージョン 2 へのアップグレード 6-9
- SQL\*Net での再リンク 6-2
- SQL\*Plus
  - Oracle8 へのスクリプトのアップグレード 6-6
- SQL\*Plus スクリプト
  - 移行 6-6
- SQL\_TRACE
  - パフォーマンス・テスト 2-11
- SQLLIB コール
  - Oracle8 に対する動作 6-3
  - 再リンク 6-4
- SQL スクリプト
  - Oracle7 と Oracle8 の互換性 6-6
  - Oracle8 での実行 6-6
- STARTUP コマンド
  - NOMOUNT オプションとブロック・サイズ 3-15
- SYS\_OP\_NTCIMG\$ キーワード
  - Oracle8 および Oracle7 での動作 6-8
- SYSTEM 表領域
  - 移行ユーティリティ 2-4, 3-4
- SYS スキーマ

ユーザー作成オブジェクト 6-11

## T

- TP モニター E-3
- TRUE キーワード
  - Oracle8 および Oracle7 での動作 6-8
- Trusted Oracle7
  - エクスポート / インポート・ユーティリティによる移行 4-2
- Tuxedo アプリケーション
  - Oracle8 XA ライブラリとの互換性 6-2
  - 再リンク 6-2
- TWO\_TASK E-3
- TWO\_TASK 変数 3-10

## U

- US7ASCII 3-9
- UTLCONST.SQL スクリプト 5-2, 8-3

## V

- VALUES
  - NCHAR および NLS 6-15
- VALUE キーワード
  - Oracle8 および Oracle7 での動作 6-8
- VARCHAR
  - サイズ E-4
- VARCHAR2
  - サイズ E-4

## W

- W52DEC 3-9
- WRITE プロシージャ
  - DBMS\_LOB パッケージ 6-5

## X

- XA
  - コール 6-2
  - ライブラリ 6-2
- XA ライブラリ
  - Oracle7 および Oracle8 間での互換性 6-2

## あ

- アーキテクチャ
  - TWO-TASK E-2

- 異なるアーキテクチャへの移行 3-5
- アクセス
  - 同時 E-2
- 新しいディクショナリ表
  - アドバンスト・キューイング・オプション 8-6
- アップグレード 8-2
  - CAT8004.SQL 8-3
  - CAT8004.SQL スクリプト 8-3, 8-8, 8-9
  - Parallel Server のオプション 8-3
  - UTLCONST.SQL スクリプト 8-3
  - 拡張レプリケーション・オプション 8-3
  - 製品の構成 8-4
  - 定義 1-2
- アドバンスト・キューイング・オプション
  - aq\$\_agent データ型 8-6
  - CATNOQUEUE.SQL スクリプト 8-6
  - CATQUEUE スクリプト 8-6
  - 新しいディクショナリ表 8-6
  - アップグレード 8-5
  - 拡張アドレス・フィールド 8-6
- アプリケーション
  - 移行 6-1
  - テスト 2-13
- アプリケーション開発者
  - 移行における役割 1-5

## い

- 移行
  - CATREPARR.SQL スクリプト 3-17
  - NCHAR および NLS 6-15
  - OCI アプリケーション 6-2, 6-4
  - Oracle Forms アプリケーション 6-6
  - SQL\*Net 6-9
  - SQL\*Plus スクリプト 6-6
  - UTLCONST.SQL スクリプト 5-2
  - 新しい管理手順 5-4
  - アプリケーション 6-1
  - アプリケーション開発者の役割 1-5
  - 一般的コメント 1-2
  - エクスポート / インポートの使用 2-4
  - エラー 3-18
  - オフラインの表領域 3-6
  - 共通の問題 2-9
  - 異なるコンピュータ・アーキテクチャ 3-5
  - 処理 1-2

- ステップ 1-2
- ソース・データベースの保存 3-5
- 定義 1-2
- データベース管理者の役割 1-4
- テスト 2-13
- 必須の Oracle リリース 2-9
- プリコンパイラ・アプリケーション 6-2, 6-3
- 方法の選択 2-2
- 用語 1-2
- 領域要件 3-3
- 移行および互換性
  - スタンバイ・データベース 6-14
  - スレッド・セーフティ、OCI 6-13
- 移行時のエラー 3-18
- 移行ステップ
  - 概要 1-2
  - ステップ 1、移行のための準備 2-2
  - ステップ 2、移行のテスト 2-12
  - ステップ 3、アプリケーションのテスト 2-13
  - ステップ 4、ソース・データベースの保存 3-5
  - ステップ 4、ソース・データベースの保存と準備 1-3
  - ステップ 5、ソース・データベースの移行 1-4
  - ステップ 6、移行されたデータベースの初期調整 1-4, 5-1
  - テスト、移行テストの計画 2-10
- 移行のためのソース・データベースの準備 3-5
- 移行の中止 3-18
- 移行パラメータ
  - CHECK\_ONLY 3-9
  - DB\_NAME 3-9
  - NEW\_DBNAME 3-9
  - NLS\_NCHAR 3-9
  - NO\_SPACE\_CHECK 3-9
  - PFILE 3-9
  - SPOOL 3-9
- 移行ユーティリティ
  - ALTER DATABASE CONVERT コマンド 3-2
  - ALTER DATABASE OPEN RESETLOGS 3-3
  - SYSTEM 表領域に必要な領域 3-4
  - インストール 3-7
  - エクスポート / インポート・ユーティリティ 2-5
  - エラーとメッセージ A-1
  - 概要 2-4, 3-2
  - 異なるアーキテクチャへの移行 3-5

- コマンド行オプション 3-8
- 使用 3-10
- 使用するキャラクタ・セット 3-5, C-4
- 選択 2-4
- ソース・データベースの移行 2-5, 2-6, 3-14
- 必須の Oracle リリース 2-9
- 必要な権限 3-10
- ファイルの変換 3-2
- 用途 2-4
- 以前のデータベース
- ダウングレード 8-9
- インポート
- データ定義変換 2-7
- インポートによるデータ定義変換 2-7

## え

- エクスポート
- CATEXP7.SQL を使用する Oracle7 8-10
- エクスポート / インポート
- 基本ステップ 4-2
- パーティション・オブジェクト 6-13
- エクスポート / インポートによる移行
- データ定義変換 2-7
- エクスポート / インポート・ユーティリティ
- Trusted Oracle7 4-2
- 移行の制約 2-6
- 移行ユーティリティまたは 2-5
- 以前のバージョン 2-7, 4-3
- 互換性の問題 6-14
- 時間要件 2-6
- 使用 4-3
- 制約 2-5
- データ定義変換 2-7
- データ・ディクショナリ 2-7
- 用途 2-4
- 利点 2-5

## お

- オプション
- 移行ユーティリティ 3-8
- オフラインの表領域 3-6
- オペレーティング・システム
- 互換性 2-9

## か

- カーソル
- オープン数 E-3
- 各国文字キャラクタ・セット
- Oracle8 での ~ 6-15
- 拡張アドレス・フィールド
- アドバンスト・キューイング・オプション 8-6
- 拡張レプリケーション・オプション
- アップグレード 8-3
- ダウングレード 8-9

## 型

- Oracle8 の新しい型 6-8
- 環境変数
- NLS キャラクタ・セット C-4
- 管理手順
- 移行済みデータベースおよび 5-4
- 新規開発 5-4

## き

- キーワード D-1
- Oracle8 での動作 6-8
- Oracle8 の新しいキーワード 6-8
- 動作の違い
- Oracle7 と Oracle8 6-8
- 逆方向移行のサポート 8-9
- キャラクタ・セット
- 移行ユーティリティの用途 3-5, C-4

## く

- クラスタ表
- COPY コマンド 2-8

## け

- 権限
- 移行ユーティリティの要件 3-10

## こ

- 互換性 2-9
- Oracle7 と Oracle8 の SQL スクリプト 6-6
- 従来型パス・エクスポート 6-14
- ダイレクト・パス・エクスポート 6-14
- コマンド行オプション 3-8

## さ

サード・パーティー製ソフトウェア  
互換性 2-9

## 索引

ビットマップ 5-3

索引内の ROWID 7-7

索引の再作成 7-7

## し

### 時間要件

エクスポート / インポート 2-6

システム変更番号 (SCN) C-3

### システム要件

Parallel Server E-4

見積り 2-8

システム要件の見積り 2-8

### 実行可能ファイル

サイズ E-2

領域要件 E-2

### シャドウ・プロセス

オープン・カーソル E-4

### 従来型パス

エクスポート 6-15

### 主キー

フィールドの制限 E-4

### 新機能

追加 5-4

## す

### スタンバイ・データベース

移行および互換性 6-14

互換性およびアップグレード 6-14

互換性の問題 6-14

### スナップショットの互換性

ROWID 7-6

スナップショット・リフレッシュ、ROWID 7-6

### スレッド・セーフティ、OCI

互換性および移行 6-13

## せ

制御ファイル 3-14

### 製品の構成

アップグレード 8-4

### 接続

TWO\_TASK アーキテクチャ E-3

ローカルとリモート E-2

マルチスレッド共有サーバー E-3

### 接続マネージャ

OCI-Net2 クライアントによる使用 6-9

## そ

### ソース・データベース

定義 1-2

## た

### ターゲット・データベース

定義 1-2

### 対象読者 xii

### ダイレクト・パス・エクスポート

移行および互換性の問題 6-14

### ダウングレード 8-6

CAT8004D.SQL 8-8

CATALOG.SQL 8-9

CATPROC.SQL 8-9

Parallel Server のオプション 8-9

拡張レプリケーション・オプション 8-9

定義 1-2

リリース 7.x へ 8-9

リリース 8.0.4 からリリース 8.0.3 へ 8-7

## ち

### 調整 5-3

移行済みデータベース 5-3

## て

定義 1-2

### ディクショナリ

移行 3-19

### データベース

新しい名前の指定 3-9

移行済みの調整 5-3

移行済みのテスト 5-3

移行のためのソース・データベースの準備 3-5

調整 5-3

停止 3-10

テスト 5-3

テスト計画の作成 2-11

名前の指定 3-9

- バックアップ 3-12
- データベース管理者
  - 移行における役割 1-4
- データベースの停止 3-10
- データ・ディクショナリ
  - エクスポート / インポート・ユーティリティ 2-7
- データ・ファイル
  - 表領域あたりの制限 E-4
- テスト 2-11
  - EXPLAIN PLAN 2-12
  - INTO パラメータ 2-12
  - アプリケーション 2-13
  - 移行 2-10
    - 移行前 2-10
    - 機能 2-10
    - 計画の作成 2-10
    - 結果の比較 5-3
    - 最小限度 2-10
    - 統合 2-11
  - パフォーマンス
    - SQL\_TRACE 2-11
  - ボリューム / ロード・ストレス 2-11

## と

- 同時アクセス E-2
- 同時ユーザー E-4
- 特殊キャラクタ D-1

## は

- バ 6-15
- バージョン 7
  - Oracle8へのアプリケーションのアップグレード6-1
  - インストール 4-3
- バージョン7のクライアント
  - バージョン8のサーバー 6-13
- バージョン7のパラメータ
  - 廃止 C-5
- バージョン7をインストール 4-3
- バージョン8
  - クライアントとバージョン7のサーバー 6-13
- パーティション・オブジェクト
  - エクスポート / インポート 6-13
- 廃止されたパラメータ C-5
- パスワード管理

- Oracle8に必要なアプリケーションの変更 6-12
- パスワードの期限切れ 6-12
- バックアップ
  - EBU 6-11
  - Recovery Manager 6-11
  - 移行後 1-4, 3-12, 5-2
  - 移行前 3-12
  - 計画 2-9
    - ターゲット・データベース 3-12, 5-2
- パフォーマンス・テスト 2-11
- パラメータ
  - 移行ユーティリティ 3-8
  - 廃止された C-5
- パラメータ・ファイル (INIT.ORA)
  - 廃止されたパラメータ 3-14
  - ファイル名の指定 3-9

## ひ

- 日付の制約
  - 無効 5-2, 8-3
- ビットマップ索引
  - 無効 5-3

## 表

- 列の制限 E-4
- 表領域
  - オフライン 3-6
  - オフライン表領域の移行 3-6

## ふ

- プリコンパイラ
  - Oracle8 および Oracle7 間の互換性 6-3
- プリコンパイラ・アプリケーション
  - Oracle7 から Oracle8 へのアップグレード 6-3
  - Oracle8 へのアップグレード 6-2
  - 移行 6-3
- プロシージャ・オプション 3-6
- ブロック・サイズ
  - DB\_BLOCK\_SIZE 3-4
  - 最小値 3-4
- 分散ロック・マネージャ ( DLM ) E-4

## へ

- 変換
  - データ定義の 2-7

## ほ

ボリューム / ロード・ストレス・テスト 2-11

## ま

マルチスレッド・サーバー

共有 E-3

共有およびローカル / リモート接続 E-3

実行要件 6-9

## む

無効な日付の制約 5-2, 8-3

## め

メモリー要件 E-2

移行 3-3

同時アクセス E-2

## も

文字のコード化

CREATE DATABASE コマンド 3-5

## や

役割

アプリケーション開発者 1-5

データベース管理者 1-4

## ゆ

ユーザー作成オブジェクト

SYS スキーマ 6-11

ユーティリティ

移行ユーティリティ 2-4

インポート 2-4

エクスポート 2-4

## よ

要件

移行のためのリリース 2-9

用語 1-2

用語集 1-2

読取り専用表領域 3-2

予約語 2-9, D-1

## り

リモート接続

マルチスレッド共有サーバー E-2

領域要件

COPY コマンド 2-8

移行 3-3

実行可能ファイル E-2

リリース

移行に必要な 2-9

リンク・ライン

OCI アプリケーション用の ~ 6-5

## ろ

ローカル接続

マルチスレッド共有サーバー E-2

ロールバック・セグメント

変換 3-3

ロック

DML ロック制限、DML\_LOCKS C-4

