

Oracle9i

データベース移行ガイド

リリース 2 (9.2)

2002 年 12 月

部品番号 : J06260-02

ORACLE®

Oracle9i データベース移行ガイド, リリース 2 (9.2)

部品番号 : J06260-02

原本名 : Oracle9i Database Migration, Release 2 (9.2)

原本部品番号 : A96530-02

原著者 : Tony Morales

グラフィック・デザイナー : Valarie Moore

原本協力者 : Nipun Agarwal, Sanjay Agarwal, Rick Anderson, Vikas Arora, Rae Burns, Ben Chang, Lakshminaray Chidambaran, Eugene Chong, George Claborn, David Colello, Jay Davison, Alan Downing, Sreenivas Gollapudi, Brajesh Goyal, Tom Graves, Michael Hartstein, Jeffrey Hebert, Thuvan Hoang, Wei Huang, Robert Jenkins, Sanjeev Jhala, Christopher Jones, Mark Jungerman, Sanjay Kaluskar, Garrett Kaminaga, Dhiraj Kapoor, Vishwanath Karra, Mark Kennedy, Susan Kotsovolos, Viswanathan Krishnamurthy, Muralidhar Krishnaprasad, Paul Lane, Gordon Larimer, Simon Law, Jing Liu, Juan Loaiza, J. Bill Lee, Bill Maimone, Raghu Mani, Shailendra Mishra, Ari Mozes, Kannan Muthukkaruppan, Subramanian Muralidhar, Ravi Murthy, Karuna Muthiah, Mark Niebur, Peter Ogilvie, Naresh Pamnani, Jenn Polk, Greg Pongracz, Franco Putzolu, N. C. Ramesh, Paul Raveling, Ann Rhee, Ajay Sethi, Carol Sexton, Helen Slattery, James Stamos, Debbie Steiner, Alex Tsukerman, Randy Urbano, Guhan Viswanathan, Steven Wertheimer, Rick Wessman, Andrew Witkowski, Lik Wong, Aravind Yalamanchi, Qin Yu

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

Printed in Japan.

制限付権利の説明

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

当プログラムのリバース・エンジニアリング等は禁止されています。

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

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

危険な用途への使用について

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

Restricted Rights Notice

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

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

目次

はじめに	xi
対象読者	xii
このマニュアルの構成	xii
関連文書	xiv
表記規則	xiv
1 概要	
用語	1-2
Oracle リリース	1-2
データベースの移行	1-3
データベースの移行の概要	1-3
アップグレード手順の概要	1-4
アップグレードにおける DBA の役割	1-7
アップグレードにおけるアプリケーション開発者の役割	1-7
複数リリースの Oracle の実行	1-8
同じコンピュータ上の複数の Oracle ホームへのデータベースのインストール	1-9
異なるコンピュータ上の複数の Oracle ホームへのデータベースのインストール	1-9
現行リリースへのデータベースのアップグレード	1-9
現行リリースへのクライアントのアップグレード	1-9
Optimal Flexible Architecture (OFA) の使用	1-10
ワード・サイズの変更	1-10
ローリング・アップグレード	1-11
オプションの削除	1-11

2 アップグレードの準備

アップグレードの準備	2-2
新しいリリースの機能の理解	2-2
新しいリリースへのアップグレード・パスの決定	2-2
アップグレード方法の選択	2-4
新しいリリースに対する Oracle ホーム・ディレクトリの選択	2-8
バックアップ計画の準備	2-8
テスト計画の作成	2-9
アップグレード処理のテスト	2-12
アップグレードしたテスト・データベースのテスト	2-12

3 Oracle9i の新しいリリースへのデータベースのアップグレード

Oracle9i リリース 2 (9.2) ソフトウェアのインストール	3-2
Database Upgrade Assistant の単独での実行	3-4
Database Upgrade Assistant を使用したデータベースのアップグレード	3-5
データベースの手動でのアップグレード	3-9
システムの考慮点および要件	3-9
アップグレードするデータベースの準備	3-11
データベースのアップグレード	3-14
特定のコンポーネントのアップグレード	3-26
手動でのアップグレードのトラブルシューティング	3-28

4 データベースのアップグレード後の作業

データベースのアップグレード後に行う作業	4-2
データベースのバックアップ	4-2
オラクル社が提供するアカウント用のパスワードの変更	4-2
Standard Edition から Enterprise Edition へのアップグレード	4-2
Oracle Managed Files (OMF) の移行	4-3
Oracle OLAP のアップグレード	4-5
サーバー・パラメータ・ファイルへの初期化パラメータ・ファイルの移行	4-7
LONG から LOB への表の移行	4-7
listener.ora ファイルの変更	4-8
スタンバイ・データベースのアップグレード	4-8
新機能の適宜追加	4-9

必要な新しい管理手順の作成	4-10
新しいリリース用のパラメータ・ファイルの調整	4-10
リリース 8.1.7 以下のデータベースのアップグレード後のみに行う作業	4-11
ユーザの NCHAR 列のアップグレード	4-11
Server Manager ラインモード・スクリプトの SQL*Plus への移行	4-12
リリース 8.0.6 以下のデータベースのアップグレード後のみに行う作業	4-13
パラレル実行での問題の回避	4-13
Windows オペレーティング・システムでのファイル名の正規化	4-13
使用禁止のファンクション索引の再構築	4-15
マテリアライズド・ビューのアップグレード	4-16
キュー表のアップグレード	4-16
リカバリ・カタログのアップグレード	4-17
DBMS_STATS パッケージで作成された統計表のアップグレード	4-18
リリース 7.3.4 のデータベースのアップグレード後のみに行う作業	4-19
使用不可となったビットマップ索引の再構築	4-19
パーティション・ビューからパーティション表への移行	4-19
無効な日付制約のチェック	4-19
Oracle Net Services の新しいリリースへのアップグレード（オプション）	4-20
データベースのテストと結果の比較	4-21
アップグレードされたデータベースのチューニング	4-21
現行リリースのワード・サイズの変更	4-21

5 互換性および相互運用性

互換性の概要	5-2
COMPATIBLE 初期化パラメータ	5-2
COMPATIBLE 初期化パラメータの設定	5-6
COMPATIBLE 設定を必要とする機能	5-9
相互運用性の概要	5-12
リリース 2 (9.2) とリリース 1 (9.0.1) 間の互換性および相互運用性の問題	5-12
ローカル管理 SYSTEM 表領域	5-13
新しい AnyData データ型	5-13
ディクショナリ管理表領域	5-14
自動セグメント領域で管理される表領域の互換性の変更	5-14
互換性およびオブジェクト型	5-14
Oracle Managed Files	5-14

Oracle OLAP	5-15
パラレル REDO を使用したログ形式の変更	5-15
Oracle Dynamic Services	5-15
Oracle Syndication Server	5-16
リリース 2 (9.2) と以前のリリース間の互換性および相互運用性の問題	5-16
アプリケーション	5-17
STARTUP コマンド	5-25
表領域およびデータ・ファイル	5-26
データ・ディクショナリ	5-28
スキーマ・オブジェクト	5-29
データ型	5-29
ユーザー定義データ型	5-32
SQL および PL/SQL	5-34
アドバンスト・キューイング (AQ)	5-35
プロシージャおよびパッケージ	5-36
Oracle オプティマイザ	5-36
Oracle9i Real Application Clusters	5-37
データベース・セキュリティ	5-39
データベース・バックアップおよびリカバリ	5-41
分散データベース	5-45
SQL*Net、Net8 または Oracle Net	5-46
その他の互換性および相互運用性の問題	5-48

6 アプリケーションのアップグレード

アプリケーションのアップグレードの概要	6-2
アプリケーションの互換性の問題	6-2
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	6-3
ソフトウェアのアップグレードおよびクライアント / サーバー構成の理解	6-3
Oracle ソフトウェアをアップグレードする場合のアプリケーションの互換性規則	6-4
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選択肢	6-6
SQL*Plus スクリプトのアップグレード	6-10
Oracle7 Forms または Oracle Developer アプリケーションのアップグレード	6-10

7 以前の Oracle リリースへのデータベースのダウングレード

ダウングレードがサポートされているリリース	7-2
全オフライン・バックアップの実行	7-2
非互換性の削除	7-2
データベースの互換性レベルの調査	7-2
非互換性の識別	7-3
リリース 2 (9.2) の非互換性の削除	7-4
リリース 1 (9.0.1) の非互換性の削除	7-8
データベース互換性のリセット	7-19
特定のコンポーネントのダウングレード	7-20
Oracle Spatial のダウングレード	7-21
interMedia のダウングレード	7-21
Oracle Visual Information Retrieval のダウングレード	7-21
Oracle Text のダウングレード	7-21
Oracle Ultra Search のダウングレード	7-22
データベースのダウングレード	7-22

8 エクスポート/インポートによるデータベースの移行

エクスポート・ダンプ・ファイルの互換性	8-2
以前のリリースと互換性がないデータでのエクスポート / インポートの使用法	8-3
ソース・データベースおよびターゲット・データベース	8-3
エクスポート・ユーティリティ要件	8-3
インポート・ユーティリティ要件	8-3
エクスポート/インポートを使用したソース・データベースのアップグレード	8-4

A 初期化パラメータおよびデータ・ディクショナリの変更

初期化パラメータの変更	A-2
非推奨の初期化パラメータ	A-2
廃止された初期化パラメータ	A-3
初期化パラメータの互換性の問題	A-5
DB_BLOCK_CHECKSUM の新規デフォルト値	A-5
ジョブ・キュー・プロセスの最大数	A-5
ORACLE_TRACE_ENABLE パラメータ	A-6
SERIALIZABLE パラメータ	A-6

SORT_AREA_SIZE および SORT_DIRECT_WRITES パラメータ	A-6
LOG_CHECKPOINT_TIMEOUT の新規デフォルト値	A-7
O7_DICTIONARY_ACCESSIBILITY パラメータ	A-7
DML_LOCKS パラメータ	A-7
DB_DOMAIN パラメータ	A-8
ラージ・プールから割り当てられたパラレル実行	A-8
アーカイブ・ログの宛先のパラメータ	A-11
静的データ・ディクショナリ・ビューの変更	A-14
非推奨の静的データ・ディクショナリ・ビュー	A-14
廃止された静的データ・ディクショナリ・ビュー	A-16
列の名前が変更された静的データ・ディクショナリ・ビュー	A-17
列が削除された静的データ・ディクショナリ・ビュー	A-17
NULL を戻す可能性のある列を持つ静的データ・ディクショナリ・ビュー	A-19
動的パフォーマンス・ビューの変更	A-21
非推奨の動的パフォーマンス・ビュー	A-21
廃止された動的パフォーマンス・ビュー	A-23
列の名前が変更された動的パフォーマンス・ビュー	A-24
列が削除された動的パフォーマンス・ビュー	A-25

B Oracle Net Services のアップグレードに関する考慮点

サポートされない Oracle Net Services の機能の概要	B-2
サポートされないパラメータおよび制御ユーティリティ・コマンド	B-4
クライアントおよびデータベースの共存問題	B-4
Oracle9i データベースへの接続	B-4
Oracle8 または Oracle7 データベースへの接続	B-5
Oracle Names	B-6
Oracle Net Manager を使用した互換性の問題の処理	B-7
Oracle Net Services へのアップグレード	B-8
手順 1: サービス名およびインスタンス名の検証	B-8
手順 2: データベース・サーバーのソフトウェアのアップグレードの実行	B-9
手順 3: クライアントのソフトウェアのアップグレードの実行	B-9
手順 4: 機能のアップグレードの実行	B-9
Oracle Names バージョン 9 の使用	B-12
データベースを使用した Oracle Names バージョン 2 からのアップグレード	B-12
動的検出オプションを使用した Oracle Names バージョン 2 からのアップグレード	B-14

ROSFILES からのアップグレード	B-17
ドメインおよびトポロジ・チェックポイント・ファイルへのリージョン・チェックポイント・ ファイルのアップグレード	B-20
アップグレード・チェックリストの確認	B-21

C Server Manager から SQL*Plus への移行

起動の違い	C-2
Server Manager の起動	C-2
SQL*Plus の起動	C-2
コマンド	C-3
SQL*Plus リリース 8.1 で導入されたコマンド	C-3
Server Manager と SQL*Plus に共通のコマンド	C-4
Server Manager コマンドと同等の SQL*Plus コマンド	C-6
SET TIMING コマンドの違い	C-7
SQL*Plus で使用できない Server Manager コマンド	C-7
構文の違い	C-7
コメント	C-8
空白行	C-10
ハイフン継続文字	C-11
アンパサンド	C-12
CREATE TYPE および CREATE LIBRARY コマンド	C-13
COMMIT コマンド	C-14

D 移行ユーティリティによる Oracle7 データベースのアップグレード

移行ユーティリティの概要	D-2
移行ユーティリティを使用したアップグレード処理の概要	D-2
移行ユーティリティ使用時のシステムの考慮点と要件	D-3
領域要件	D-3
ブロック・サイズの考慮点	D-4
SQL*Net に関する考慮点	D-4
レプリケーション環境に関する考慮点	D-5
ConText から Oracle Text への移行に関する考慮点	D-5
分散データベースの考慮点	D-5
アップグレードする Oracle7 データベースの準備	D-5
移行ユーティリティ・コマンドライン・オプションの検討	D-9

移行ユーティリティの実行	D-10
UNIX オペレーティング・システムでの移行ユーティリティの実行	D-10
Windows プラットフォームでの移行ユーティリティの実行	D-12
移行ユーティリティの実行結果の確認	D-13
Oracle7 データベースの保存	D-14
移行ユーティリティ・メッセージ	D-15
移行ユーティリティ・エラーのトラブルシューティング	D-25
移行ユーティリティの使用に関する問題	D-25
ALTER DATABASE CONVERT 文における問題	D-27
Oracle7 のアップグレードの中止	D-32
物理 ROWID の移行について	D-33
アプリケーションのアップグレードおよびデータの移行	D-33
DBMS_ROWID パッケージ	D-34
スナップショット・リフレッシュ	D-37
Oracle7 クライアントの互換性の問題	D-38
ROWID に関する移行および互換性の問題	D-38
リリース 8.0 での初期化パラメータおよびデータ・ディクショナリの変更	D-39
リリース 8.0 での初期化パラメータの変更	D-39
リリース 8.0 での静的データ・ディクショナリ・ビューの変更	D-40

E レプリケーション環境のデータベース移行および互換性

レプリケーションのデータベース移行の概要	E-2
すべてのサイトの同時アップグレード	E-3
段階的アップグレード	E-6
段階的アップグレードのための Oracle7 マスター・サイトの準備	E-7
マテリアライズド・ビュー・サイトの段階的アップグレード	E-8
マスター・サイトの段階的アップグレード	E-10
主キー・マテリアライズド・ビューへのアップグレード	E-14
マスター・サイトでの主キー・マテリアライズド・ビューの変換	E-15
マテリアライズド・ビュー・サイトでの主キー・マテリアライズド・ビューの変換	E-15
Oracle の上位リリースへのアップグレードを必要とする機能	E-17
Oracle9i を必要とする機能	E-17
Oracle8i 以上を必要とする機能	E-17
Oracle8 以上を必要とする機能	E-18
Oracle7 以上で利用できる機能	E-18

廃止されたプロシージャ E-19

索引

はじめに

このマニュアルでは、Oracle データベース・サーバーにおけるデータベースの移行の計画と実行の手引きを示します。また、このマニュアルでは互換性、今回のリリースの Oracle へのアプリケーションのアップグレード、および初期化パラメータの変更やデータ・ディクショナリの変更などの今回のリリースにおける重要な変更について説明します。

また、このマニュアルでは、Oracle9i (Standard Edition ともいう) および Oracle9i Enterprise Edition 製品の様々な機能についても説明します。Oracle9i および Oracle9i 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 Oracle9i 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、アプリケーション・フェイルオーバーを使用するには、Oracle9i Real Application Clusters が備えられた Oracle9i Enterprise Edition が必要です。

参照： Oracle9i と Oracle9i Enterprise Edition の違い、および有効な機能やオプションなどの詳細は、『Oracle9i データベース新機能』を参照してください。

ここでは、次の項目について説明します。

- [対象読者](#)
- [このマニュアルの構成](#)
- [関連文書](#)
- [表記規則](#)

対象読者

このマニュアルは、Oracle データベースの移行の計画および実行を担当する、データベース管理者 (DBA)、アプリケーション開発者、セキュリティ管理者、システム・オペレータを対象としています。

このマニュアルを使用するには、次の内容について詳しく理解しておく必要があります。

- リレーショナル・データベースの概要
- ご使用のリリースの Oracle データベース・サーバー
- ご使用のオペレーティング・システム環境

このマニュアルの構成

このマニュアルは、次のように構成されています。

第 1 章「概要」

この章では、データベースの移行の概要、および複数のリリースの Oracle を実行する方法について説明します。また、アップグレードまたはダウングレード中に、データベースのワード・サイズを変更する方法についても説明します。

第 2 章「アップグレードの準備」

この章では、本番データベースをアップグレードする前に行う手順について説明します。

第 3 章「Oracle9i の新しいリリースへのデータベースのアップグレード」

この章では、データベースを Oracle9i の新しいリリースにアップグレードする処理について説明します。

第 4 章「データベースのアップグレード後の作業」

この章では、データベースを Oracle9i の新しいリリースにアップグレードした後で行う手順について説明します。また、データベースのワード・サイズを変更する (32 ビットと 64 ビットを切り替える) 方法についても説明します。

第 5 章「互換性および相互運用性」

この章では、Oracle の異なるリリース間の互換性および相互運用性について、COMPATIBLE 初期化パラメータの詳細を含めて説明します。また、Oracle の機能とそれに必要な互換性レベルを示し、互換性および相互運用性に関連する固有の問題についても説明します。

第 6 章「アプリケーションのアップグレード」

この章では、ユーザーのアプリケーションおよびツール製品を Oracle9i の新しいリリースでできるようにアップグレードする方法の概要を説明します。

第7章「以前の Oracle リリースへのデータベースのダウングレード」

この章では、データベースを Oracle の以前のリリースにダウングレードする処理について説明します。

第8章「エクスポート / インポートによるデータベースの移行」

この章では、エクスポート / インポート・ユーティリティを使用して Oracle データベース間でデータを移行する処理について説明します。

付録 A「初期化パラメータおよびデータ・ディクショナリの変更」

この付録では、Oracle の異なるリリース間での初期化パラメータおよびデータ・ディクショナリの変更点を示します。また、特定の初期化パラメータにおける互換性の問題についても説明します。

付録 B「Oracle Net Services のアップグレードに関する考慮点」

この付録では、Oracle Net Services の共存およびアップグレードについて説明します。

付録 C「Server Manager から SQL*Plus への移行」

この付録では、Server Manager のラインモード・スクリプトを SQL*Plus で使用するための変更処理について説明します。

付録 D「移行ユーティリティによる Oracle7 データベースのアップグレード」

この付録では、移行ユーティリティを使用して Oracle7 データベースを Oracle9i の新しいリリースへ手動でアップグレードする方法について説明します。

付録 E「レプリケーション環境のデータベース移行および互換性」

この付録では、Oracle7 データベースの Oracle Replication システムを Oracle9i にアップグレードする方法について説明します。Oracle Replication の異なるリリース間の互換性の問題についても説明します。

関連文書

詳細は、次の Oracle ドキュメントを参照してください。

- このマニュアルで使用される概念および用語の詳細は、『Oracle9i データベース概要』を参照してください。
- Oracle データベース・サーバーの管理の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。
- Oracle の SQL コマンドおよび関数の詳細は、『Oracle9i SQL リファレンス』を参照してください。
- Oracle データベース・サーバーにバンドルされるユーティリティ（エクスポート、インポートおよび SQL*Loader）の詳細は、『Oracle9i データベース・ユーティリティ』を参照してください。
- Oracle Net Services の詳細は、『Oracle9i Net Services 管理者ガイド』を参照してください。

このマニュアルの多くの例で、Oracle のインストール時にデフォルトとしてインストールされるシード・データベースのサンプル・スキーマを使用しています。スキーマの作成方法および使用方法の詳細は、『Oracle9i サンプル・スキーマ』を参照してください。

リリース・ノート、インストール・マニュアル、ホワイト・ペーパーまたはその他の関連文書は、OTN-J（Oracle Technology Network Japan）に接続すれば、無償でダウンロードできます。OTN-J を使用するには、オンラインでの登録が必要です。次の URL で登録できます。

<http://otn.oracle.co.jp/membership/>

すでに OTN-J のユーザー名およびパスワードを取得済であれば、次の OTN-J Web サイトの文書セクションに直接接続できます。

<http://otn.oracle.co.jp/document/>

表記規則

この項では、このマニュアルの本文およびコード例で使用される表記規則について説明します。この項の内容は次のとおりです。

- [本文中の表記規則](#)
- [コード例中の表記規則](#)
- [Microsoft Windows オペレーティング・システムの表記規則](#)

本文中の表記規則

本文では、特別な用語をより迅速に識別できるように、様々な表記規則を使用します。次の表に、それらの表記規則を説明し、その使用例を示します。

規則	意味	例
太字	太字は、本文中で定義されている用語または用語集に記載されている用語（あるいはその両方）を示します。	この句を指定すると、 索引構成表 が作成されます。
固定幅フォントの大文字	固定幅フォントの大文字は、システムが提供する要素を示します。このような要素には、パラメータ、権限、データ型、Recovery Manager キーワード、SQL キーワード、SQL*Plus またはユーティリティ・コマンド、パッケージおよびメソッドが含まれます。また、システムが提供する列名、データベース・オブジェクト、データベース構造、ユーザー名およびロールも含まれます。	NUMBER 列に対してのみに、この句を指定できます。 BACKUP コマンドを使用して、データベースのバックアップを取ることができます。 USER_TABLES データ・ディクショナリ・ビュー内の TABLE_NAME 列を問い合わせます。 DBMS_STATS.GENERATE_STATS プロシージャを使用します。
固定幅フォントの小文字	固定幅フォントの小文字は、実行可能ファイル、ファイル名、ディレクトリ名およびユーザーが提供する要素のサンプルを示します。このような要素には、コンピュータ名およびデータベース名、ネット・サービス名および接続識別子が含まれます。また、ユーザーが提供するデータベース・オブジェクトとデータベース構造と列名、パッケージとクラス、ユーザー名とロール、プログラム・ユニットおよびパラメータ値も含まれます。 注意： 大文字と小文字を組み合わせて使用するプログラム要素もあります。これらの要素は、記載されているとおり入力してください。	sqlplus と入力して、SQL*Plus をオープンします。 パスワードは、orapwd ファイルで指定します。 /disk1/oracle/dbs ディレクトリ内のデータ・ファイルおよび制御ファイルのバックアップを取ります。 hr.departments 表には、department_id、department_name および location_id 列があります。 QUERY_REWRITE_ENABLED 初期化パラメータを true に設定します。 oe ユーザーとして接続します。 JRepUtil クラスが次のメソッドを実装します。 parallel_clause を指定できます。 Uold_release.SQL を実行します。ここで、old_release とはアップグレード前にインストールしたリリースを示します。

コード例中の表記規則

コード例は、SQL、PL/SQL、SQL*Plus または他のコマンドライン文を説明します。コード例は、固定幅フォントで表示され、この例に示すとおり通常のテキストと区別されます。

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

次の表に、コード例で使用される表記規則を説明し、その使用例を示します。

規則	意味	例
[]	大カッコは、任意に選択する 1 つ以上の項目を囲みます。大カッコは、入力しないでください。	DECIMAL (<i>digits</i> [, <i>precision</i>])
{ }	中カッコは、2 つ以上の項目を囲み、そのうちの 1 つの項目は必須です。中カッコは、入力しないでください。	{ENABLE DISABLE}
	縦線は、大カッコまたは中カッコ内の 2 つ以上のオプションの選択項目を表します。オプションのうちの 1 つを入力します。縦線は、入力しないでください。	{ENABLE DISABLE} [COMPRESS NOCOMPRESS]
...	水平の省略記号は、次のいずれかを示します。 <ul style="list-style-type: none">■ 例に直接関連しないコードの一部が省略されている。■ コードの一部を繰り返すことができる。	CREATE TABLE ... AS <i>subquery</i> ; SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM employees;
.	垂直の省略記号は、例に直接関連しない複数の行が省略されていることを示します。	SQL> SELECT NAME FROM V\$DATAFILE; NAME ----- /fs1/dbs/tbs_01.dbf /fs1/dbs/tbs_02.dbf . . . /fs1/dbs/tbs_09.dbf 9 rows selected.
その他の句読点	大カッコ、中カッコ、縦線および省略記号以外の句読点は、表示されているとおり入力する必要があります。	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
イタリック体	イタリック体は、特定の値を指定する必要があるプレースホルダや変数を示します。	CONNECT SYSTEM/ <i>system_password</i> DB_NAME = <i>database_name</i>

規則	意味	例
大文字	大文字は、システムが提供する要素を示します。これらの用語は、ユーザー定義の用語と区別するために大文字で示されます。用語が大カッコ内にかぎり、表示されているとおりの順序および綴りで入力します。ただし、これらの用語は大 / 小文字が区別されないため、小文字でも入力できます。	<pre>SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;</pre>
小文字	<p>小文字は、ユーザー定義のプログラム要素を示します。たとえば、表名、列名またはファイル名などです。</p> <p>注意：大文字と小文字を組み合わせ使用するプログラム要素もあります。これらの要素は、記載されているとおりの入力してください。</p>	<pre>SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjjones IDENTIFIED BY ty3MU9;</pre>

Microsoft Windows オペレーティング・システムの表記規則

次の表に、Microsoft Windows オペレーティング・システムの表記規則を説明し、その使用例を示します。

規則	意味	例
「スタート」>	プログラムを起動する方法を示します。	Database Configuration Assistant を起動するには、「スタート」>「プログラム」>「Oracle - HOME_NAME」>「Configuration and Migration Tools」>「Database Configuration Assistant」を選択します。
ファイル名およびディレクトリ名	ファイル名およびディレクトリ名は大 / 小文字が区別されません。特殊文字の左山カッコ (<)、右山カッコ (>)、コロン (:)、二重引用符 (")、スラッシュ (/)、縦線 () およびダッシュ (-) は使用できません。円記号 (¥) は、引用符で囲まれている場合でも、要素のセパレータとして処理されます。Windows では、ファイル名が ¥¥ で始まる場合、汎用ネーミング規則が使用されていると想定されます。	<pre>c:¥winnt"¥"system32 は C:¥WINNT¥SYSTEM32 と同じです。</pre>

規則	意味	例
C:¥>	<p>現在のハードディスク・ドライブの Windows コマンド・プロンプトを表します。コマンド・プロンプトのエスケープ文字はカレット (^) です。プロンプトには、作業中のサブディレクトリが表示されます。このマニュアルでは、コマンド・プロンプトと呼びます。</p> <p>円記号 (¥) は、Windows コマンド・プロンプトの二重引用符 (") のエスケープ文字として必要な場合があります。カッコおよび一重引用符 (') にはエスケープ文字は必要ありません。エスケープ文字および特殊文字の詳細は、Windows オペレーティング・システムのドキュメントを参照してください。</p>	<p>C:¥oracle¥oradata></p> <p>C:¥>exp scott/tiger TABLES=emp QUERY=¥"WHERE job='SALESMAN' and sal<1600¥"</p> <p>C:¥>imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept)</p>
HOME_NAME	<p>Oracle ホームの名前を表します。ホーム名には、英数字で 16 文字まで使用できます。ホーム名に使用可能な特殊文字は、アンダースコアのみです。</p>	<p>C:¥> net start OracleHOME_NAME_TNSListener</p>

規則	意味	例
ORACLE_HOME および ORACLE_BASE	<p>Oracle8 リリース 8.0 以前のリリースでは、Oracle コンポーネントをインストールすると、すべてのサブディレクトリが最上位の ORACLE_HOME ディレクトリ（デフォルト名は次のいずれか）に配置されました。</p> <ul style="list-style-type: none">■ C:¥orant（Windows NT の場合）■ C:¥orawin98（Windows 98 の場合） <p>今回のリリースは、Optimal Flexible Architecture（OFA）のガイドラインに準拠します。最上位の ORACLE_HOME ディレクトリに配置されないサブディレクトリもあります。ORACLE_BASE と呼ばれる最上位のディレクトリ（デフォルトは C:¥oracle）があります。他の Oracle ソフトウェアがインストールされていないコンピュータに Oracle9i リリース 2（9.2）をインストールした場合、最初の Oracle ホーム・ディレクトリは、デフォルトで C:¥oracle¥ora90 に設定されます。Oracle ホーム・ディレクトリは、ORACLE_BASE の直下に配置されます。</p> <p>このマニュアルに示すすべてのディレクトリ・パスの例は、OFA の表記規則に準拠しています。</p>	<p>%ORACLE_HOME%¥rdbms¥admin ディレクトリへ移動します。</p>

この章では、データベースの移行の概要、および複数のリリースの **Oracle** を実行する方法について説明します。また、アップグレードまたはダウングレード中に、データベースのワード・サイズを変更する方法についても説明します。

この章では、次の項目について説明します。

- [用語](#)
- [データベースの移行の概要](#)
- [複数リリースの Oracle の実行](#)
- [Optimal Flexible Architecture \(OFA\) の使用](#)
- [ワード・サイズの変更](#)
- [ローリング・アップグレード](#)
- [オプションの削除](#)

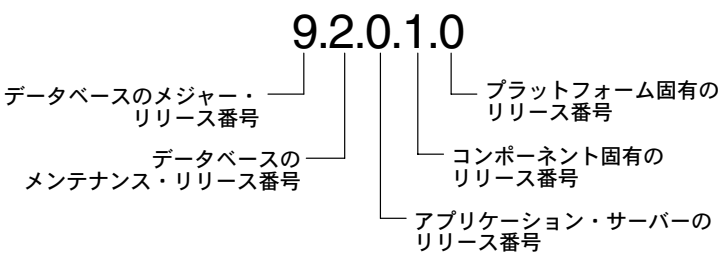
用語

次の用語が、このマニュアル全体で使用されます。

Oracle リリース

このマニュアルでは、Oracle データベース・サーバーの異なるリリース間での移行について説明します。図 1-1 に、リリース番号の各部分の意味を示します。

図 1-1 Oracle のリリース番号の例



参照： Oracle のリリース番号の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

注意： Oracle9i リリース 2 (9.2) 以上では、リリース番号の 2 番目の数字が、Oracle のメンテナンス・リリースを表します。以前のリリースでは、3 番目の数字がメンテナンス・リリースを表していました。

このマニュアルの本文にデータベースのメジャー・リリースが記載されている場合、その文は、そのメジャー・リリースのすべてのリリースに適用されます。Oracle9i の説明は、リリース 1 (9.0.1) およびリリース 2 (9.2) のすべてのリリースに適用されます。Oracle8 および Oracle8i の説明は、リリース 8.0 およびリリース 8.1 のすべてのリリースに適用されます。Oracle7 の説明は、7.0、7.1、7.2 および 7.3 を含む Oracle7 のすべてのリリースに適用されます。

同様に、このマニュアルの本文にメンテナンス・リリースが記載されている場合、その文は、そのメンテナンス・リリースのすべてのコンポーネント固有のリリースおよびプラットフォーム固有のリリースに適用されます。したがって、リリース 1 (9.0.1) の説明は、リリース 1 (9.0.1.1)、リリース 1 (9.0.1.1.2)、およびリリース 1 (9.0.1) のその他すべてのプラットフォーム固有のリリースに適用されます。

データベースの移行

データベースの移行は、Oracle データベース内のデータを変換し、Oracle データベース・サーバーの特定のリリースを反映させる一連の処理および手順です。データベースの移行には、次の処理が含まれます。

- アップグレード処理: データベースが Oracle の新しいリリースにアップグレードされます。

参照: アップグレード処理の詳細は、[第3章「Oracle9i の新しいリリースへのデータベースのアップグレード」](#)を参照してください。

- ダウングレード処理: データベースがアップグレード前の Oracle リリースのデータベースにダウングレードされます。

参照: ダウングレード処理の詳細は、[第7章「以前の Oracle リリースへのデータベースのダウングレード」](#)を参照してください。

注意: このマニュアルでは、Oracle の異なるリリース間で実行されるアップグレードおよびダウングレードについて説明しています。そのため、前述のデータベースの移行の定義が適用されます。ただし、他の Oracle マニュアルでは、**移行**という用語をより幅広い定義で使用する場合があります。たとえば、移行は、Oracle 以外のデータベースから Oracle データベースにデータを移動する処理を示すこともあります。

データベースの移行の概要

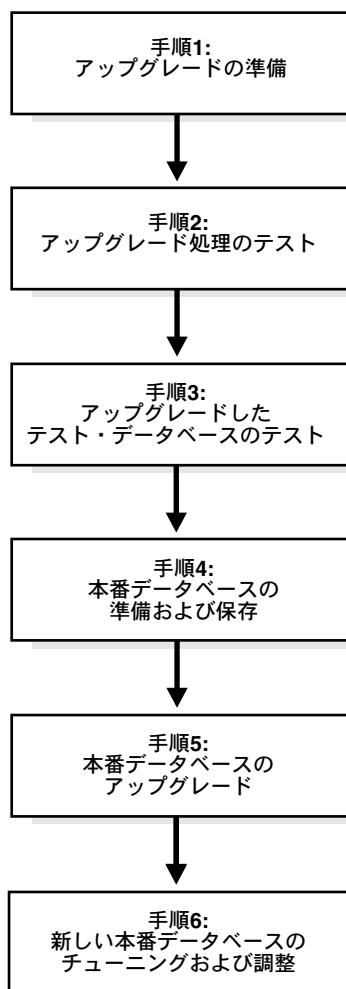
この項では、既存の Oracle データベースを Oracle9i の新しいリリースにアップグレードするために必要な主な手順の概要を説明します。Oracle9i は、以前のすべての Oracle リリースと互換性があります。したがって、このマニュアルで説明する手順によってアップグレードされたデータベースは、以前のリリースと同様に機能し、Oracle9i の新機能を利用できます。

現行の本番データベースをアップグレードする前に、いくつかの準備処理を実行する必要があります。アップグレード後、さらにテスト手順をいくつか実行します。さらに別の手順によって、既存のアプリケーションに Oracle9i の新機能を追加できます。

アップグレード手順の概要

データベースをアップグレードする前に、アップグレード処理の主な手順について理解しておく必要があります。これらの手順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、例外として一部のオペレーティング・システム固有の詳細項目があり、それについては、オペレーティング・システム固有の Oracle マニュアルに示されています。

図 1-2 主なアップグレード手順



Oracle9i のツール製品を慎重に計画して使用することにより、Oracle9i の新しいリリースへのデータベースのアップグレード処理を容易に行えます。データベースをアップグレードするには、次のいずれかの方法を実行します。

- Database Upgrade Assistant (DBUA) は、データベースをアップグレードする最も簡単な方法です。Oracle9i の新しいリリースのインストール中またはインストールの完了後のいずれかに Database Upgrade Assistant を実行するように選択できます。

注意： Database Upgrade Assistant は、データベースのアップグレード方法として適切です。Database Upgrade Assistant を使用して Oracle9i の新しいリリースにアップグレードすることをお勧めします。

- 手動アップグレードは、少し複雑になりますが、データベースをアップグレードする処理をより詳細に制御できます。
- エクスポート / インポート・ユーティリティおよびデータのコピーを使用すると、データベースを部分的に移行できます。

次の項では、[図 1-2](#) に示される主な手順の概要について説明します。これらの説明は、アップグレード処理の主な手順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。

手順 1: アップグレードの準備

- Oracle9i の新しいリリースの機能を理解します。これらの機能の概要は、『Oracle9i データベース新機能』を参照してください。
- 現行の本番データベースに関する考慮点に基づいて、使用するアップグレード方法を決定します。
- アップグレードに必要なシステム・リソースを見積もり、確保します。
- Oracle9i のテスト・データベースでアップグレードをテストする計画と、アップグレードした Oracle9i 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延が発生した場合にすぐにリカバリできるように、バックアップ計画を準備します。

手順 2: アップグレード処理のテスト

- テスト・データベースを使用して、アップグレードのテストを実行します。アップグレードのテストは、テスト用に作成した環境で行う必要があり、実際の本番データベースに影響しないようにします。

手順 3: アップグレードしたテスト・データベースのテスト

- テスト・データベースおよび Oracle9i の新しいリリースへアップグレードされたテスト・データベースに対して、手順 1 で計画したテストを実行します。
- 結果を比較して、テスト・データベースに対するテスト結果と、アップグレードされたデータベースに対するテスト結果の相違点を記録します。
- 発見した相違点を解決する方法を調べて、その解決方法を実装します。
- アップグレードのテストが問題なく完了し、必要なアプリケーションで正常に動作するまで、必要に応じて手順 1、手順 2、および手順 3 の最初の部分を繰り返します。

第 2 章「アップグレードの準備」では、手順 1 ～ 3 までを詳しく説明します。

手順 4: 本番データベースの準備および保存

- Oracle9i の新しいリリースへのアップグレード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。
- 本番データベースのバックアップ処理とアップグレード処理に必要な停止時間をスケジュールリングします。
- 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。

手順 5: 本番データベースのアップグレード

- 本番データベースを Oracle9i の新しいリリースへアップグレードします。
- アップグレードの後、本番データベースの全体バックアップおよびその他のアップグレード後の処理を行います。

第 3 章では、手順 4 および手順 5 での Database Upgrade Assistant の使用方法、および手動アップグレードの実行方法について説明します。第 8 章では、手順 4 および手順 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 4 章では、アップグレード後のバックアップ手順およびその他のアップグレード後の処理について説明します。

参照： Oracle Replication をインストールしたデータベースをアップグレードする場合は、[付録 E「レプリケーション環境のデータベース移行および互換性」](#)を参照してください。

手順 6: 新しい本番データベースのチューニングおよび調整

- 新しい Oracle9i の本番データベースをチューニングします。新しい Oracle9i の本番データベースのパフォーマンスは、アップグレード前のデータベースと同等以上にする必要があります。第 4 章では、これらのチューニングについて説明します。
- Oracle9i の新しいリリースのどの機能を使用するかを決定し、それに応じてアプリケーションを更新します。

- 必要に応じて、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 Oracle9i データベースにアップグレードしないでください。第 6 章では、アプリケーションを更新する際の考慮点について説明します。

アップグレードの際、マルチバージョン機能を使用することをお勧めします。これによって、同じデータベースの複数のコピーを 1 台のコンピュータに保管できます。新しいリリースをテストする間、既存のリリースを本番環境として使用できます。

アップグレードにおける DBA の役割

通常、DBA には、アップグレード処理を確実に成功させる責任があります。DBA はアップグレード処理の各手順に関係しますが、アップグレードしたデータベースでのアプリケーションのテストにはかかりません。

DBA 固有の役割は、次のとおりです。

- アップグレード処理に関連するすべてのスタッフと打合せを持ち、各スタッフの役割を明確に決定する
- アップグレードのテストを実行する
- テストおよび本番アップグレードをスケジューリングする
- 本番データベースのバックアップを作成する
- 本番データベースのアップグレードを完了する
- 新しくアップグレードした Oracle9i の本番データベースのバックアップを作成する

アップグレード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には、現行のデータベース用に設計されたアプリケーションが、アップグレードされた Oracle9i 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、アップグレードされた Oracle9i 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、使用する Oracle9i の新機能を決定します。

本番データベースをアップグレードする前に、DBA またはアプリケーション開発者は、Oracle9i 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。その後、アプリケーションが元の機能（または拡張された Oracle9i 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および変更できます。

次に、アップグレード後の Oracle9i データベースでの相違点を判断するための参照先を示します。これらの相違点は、特定のアプリケーションに影響する場合があります。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5章「互換性および相互運用性」](#)では、Oracle のリリースの違いによって発生する可能性のある互換性および相互運用性の問題について説明しています。
- [第6章「アプリケーションのアップグレード」](#)では、既存のアプリケーションが Oracle9i データベースにアクセスするために必要な変更点について説明します。また、Oracle9i の機能を利用するために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。
- [付録 A「初期化パラメータおよびデータ・ディクショナリの変更」](#)では、廃止および非推奨の初期化パラメータおよびデータ・ディクショナリ・ビューを示します。
- [付録 B「Oracle Net Services のアップグレードに関する考慮点」](#)では、SQL*Net および Net8 を Oracle Net Services へアップグレードする手順を説明しています。
- [付録 E「レプリケーション環境のデータベース移行および互換性」](#)では、Oracle Replication をインストールしたデータベースをアップグレードする手順を説明しています。
- 『Oracle9i データベース新機能』では、Oracle9i の新しいリリースで使用可能な新機能について説明しています。
- 『Oracle9i Real Application Clusters 概要』および『Oracle9i SQL リファレンス』では、Oracle9i の新しいリリースでの変更点と新機能について説明しています。
- 『Oracle9i アプリケーション開発者ガイド - 基礎編』、『Oracle9i アプリケーション開発者ガイド - ラージ・オブジェクト』および『Oracle9i アプリケーション開発者ガイド - アドバンスト・キューイング』では、アプリケーションの計画および実装について説明しています。

Oracle9i には、既存のアプリケーションを Oracle9i にアップグレードするための機能があります。次に例を示します。

- Oracle Net 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は未変更のままです。
- Oracle の下位互換性は、異なるリリース間のわずかな非互換に対応しています。

複数リリースの Oracle の実行

同一のコンピュータ上で異なるリリースの Oracle を同時に実行できます。ただし、それぞれのリリースは、そのリリースに対応するデータベースにのみアクセスできます。たとえば、Oracle8i および Oracle9i を同一のコンピュータにインストールしている場合は、Oracle8i のサーバーは Oracle8i データベースにはアクセスできますが、Oracle9i データベースにはアクセスできません。また、Oracle9i のサーバーは Oracle9i データベースにはアクセスできますが、Oracle8i データベースにはアクセスできません。次の項では、複数のリリースの Oracle を実行する際の概要を説明します。

注意： 既存の Oracle ホームに Oracle9i リリース 2 (9.2) 製品をインストールすることはできません。この機能は、以前の一部のリリースでのみ使用可能であり、以降はサポートされていません。Oracle9i は、以前のリリースとは別の新しい Oracle ホームにインストールする必要があります。また、1 つの Oracle ホームに複数のリリースをインストールすることはできません。複数の Oracle ホームを作成する場合、OFA を採用することをお勧めします。詳細は、1-10 ページの「[Optimal Flexible Architecture \(OFA\) の使用](#)」を参照してください。

参照： ご使用のオペレーティング・システムで、複数のリリースの Oracle を実行する方法については、各オペレーティング・システム固有の Oracle マニュアルを参照してください。オペレーティング・システムによっては制限がある場合があります。

同じコンピュータ上の複数の Oracle ホームへのデータベースのインストール

同じコンピュータ上の複数の異なる Oracle ホームに Oracle7、Oracle8、Oracle8i および Oracle9i データベースをインストールし、Oracle7、Oracle8、Oracle8i および Oracle9i クライアントからいずれかまたはすべてのデータベースに接続できます。

異なるコンピュータ上の複数の Oracle ホームへのデータベースのインストール

異なるコンピュータ上の複数の異なる Oracle ホームに Oracle7、Oracle8、Oracle8i および Oracle9i データベースをインストールし、Oracle7、Oracle8、Oracle8i および Oracle9i クライアントからいずれかまたはすべてのデータベースに接続できます。

現行リリースへのデータベースのアップグレード

Oracle7、Oracle8、Oracle8i または Oracle9i データベースを現行の Oracle9i にアップグレードして、Oracle7、Oracle8、Oracle8i または Oracle9i クライアントからアップグレードされたデータベースに接続できます。同じ Oracle ホーム内のデータベースはアップグレードできません。

現行リリースへのクライアントのアップグレード

Oracle7、Oracle8、Oracle8i または Oracle9i のいずれかまたはすべてのクライアントを現行の Oracle9i リリースにアップグレードできます。後で、Oracle7、Oracle8、Oracle8i または Oracle9i のいずれかまたはすべてのデータベースを現行の Oracle9i リリースへアップグレードすることもできます。

Optimal Flexible Architecture (OFA) の使用

Oracle9i のインストール用に Optimal Flexible Architecture (OFA) 規格をお薦めします。OFA 規格とは、メンテナンスの手間がほとんどない、効率的で信頼性のある Oracle データベース用の構成ガイドラインです。

OFA には、次のメリットがあります。

- ディスク上の大量の複雑なソフトウェアおよびデータを、デバイスのボトルネックまたはパフォーマンスの低下を避けながら構成する。
- ソフトウェアやデータのバックアップ機能など、データ破損の危険性のある日常業務を簡単にする。
- 複数の Oracle データベース間での切替えを軽減する。
- データベースの成長を処理および管理できる。
- データ・ディクショナリにおける空き領域の断片化を排除し、その他の断片化を特定し、リソース競合を最小限に抑える。

現在、OFA 規格を使用していない場合、OFA 規格への切替えには、ディレクトリ構造の修正とデータベース・ファイルの再配置が伴います。

参照：

- OFA の詳細は、ご使用のオペレーティング・システム固有の Oracle マニュアルを参照してください。
- データベース・ファイルの再配置の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

ワード・サイズの変更

アップグレードまたはダウングレード中に、データベースのワード・サイズを変更することができます。ワード・サイズの変更には、次のような状況が考えられます。

- 64 ビット・ハードウェアにインストールされている 32 ビット Oracle ソフトウェアを、64 ビット Oracle ソフトウェアへ変更する。
- 64 ビット・ハードウェアにインストールされている 64 ビット Oracle ソフトウェアを、32 ビット Oracle ソフトウェアへ変更する。

アップグレードまたはダウングレード中にワード・サイズを変更する場合、追加の処置は必要ありません。ワード・サイズは、アップグレードまたはダウングレード中に自動的に変更されます。ただし、同じリリース間でワード・サイズを変更する場合は、4-21 ページの「[現行リリースのワード・サイズの変更](#)」の指示に従ってください。たとえば、32 ビットの Oracle9i リリース 2 (9.2) を、64 ビットの Oracle9i リリース 2 (9.2) に切り替える場合に、この手順を実行します。

ハードウェアを 32 ビットから 64 ビットへ切り替える場合、または 64 ビットから 32 ビットへ切り替える場合は、次のことがいえます。

- ハードウェアを 32 ビットから 64 ビットへ切り替えても、問題なく既存の 32 ビット Oracle ソフトウェアを使用できます。
- ハードウェアを 64 ビットから 32 ビットへ切り替える場合は、32 ビットの Oracle ソフトウェアをインストールしてから、64 ビットから 32 ビットに切り替える必要があります。

データベース・データ、REDO、UNDO のディスク上の形式は、Oracle の 32 ビットおよび 64 ビットのインストールも同様の形式です。32 ビットと 64 ビットのインストールで異なるのは、次の内部構造のみです。

- PL/SQL のコンパイルされた形式。PL/SQL の再コンパイルの方法とタイミングに関する情報は、このマニュアルの該当する章で説明します。
- ユーザー定義型の記憶形式。これは、データベースを作成した Oracle のリリースによって決定されます。既存の記憶形式は、必要に応じて、正しい形式へ透過的に変換されます。ユーザー定義型には、オブジェクト型、REF、VARRAY およびネストした表が含まれます。

ローリング・アップグレード

ローリング・アップグレードとは、Oracle9i Real Application Clusters 内の異なるデータベースまたは同じデータベースの異なるインスタンスを、データベースを停止することなく一度に 1 つずつアップグレードすることを意味します。Oracle9i Real Application Clusters では、ローリング・アップグレードはサポートされていません。

オプションの削除

Oracle9i の新しいリリースにアップグレードするときに古いオプションを削除する場合は、Oracle Universal Installer で削除します。アップグレードの前後に削除できますが、削除する項目に対応したリリースのインストーラを使用する必要があります。

たとえば、Oracle Text をインストールした Oracle9i リリース 1 (9.0.1) を実行していて、Oracle9i の新しいリリースへのアップグレード後にこのオプションが必要なくなる場合、次のいずれかの方法で Oracle Text を削除する必要があります。

- Oracle9i の新しいリリースにアップグレードする前に、Oracle9i リリース 1 (9.0.1) の Oracle Universal Installer を使用して Oracle Text を削除します。Oracle9i の新しいリリースのインストール時に Oracle Text をインストールしないでください。
- Oracle9i の新しいリリースへのアップグレード時に、Oracle Text をインストールおよびアップグレードします。その後、Oracle9i の新しいリリースの Oracle Universal Installer を使用して、Oracle Text を削除します。

注意： オプションを削除した後、データベース内に付加的なデータ・ディクショナリ表が残る場合があります。

参照： Oracle Universal Installer の使用方法是、ご使用のオペレーティング・システム固有の Oracle インストレーション・ガイドを参照してください。

アップグレードの準備

この章では、本番データベースをアップグレードする前に完了しておく必要がある作業について説明します。また、この章では、アップグレード処理（1-3 ページの「[データベースの移行の概要](#)」で説明済）の手順 1～3 までを詳しく説明します。

この章では、次の項目について説明します。

- [アップグレードの準備](#)
- [アップグレード処理のテスト](#)
- [アップグレードしたテスト・データベースのテスト](#)

参照：

- Oracle Net Services のアップグレードに関する考慮点については、[付録 B「Oracle Net Services のアップグレードに関する考慮点」](#)を参照してください。
- Oracle Replication をインストールしたデータベース・システムをアップグレードする場合は、[付録 E「レプリケーション環境のデータベース移行および互換性」](#)を参照してください。

注意： アップグレード処理の一部は、オペレーティング・システム固有のもので、アップグレードの準備の詳細は、ご使用のオペレーティング・システム固有の Oracle マニュアルを参照してください。

アップグレードの準備

アップグレードの準備として、次の作業を行います。

- [新しいリリースの機能の理解](#)
- [新しいリリースへのアップグレード・パスの決定](#)
- [アップグレード方法の選択](#)
- [新しいリリースに対する Oracle ホーム・ディレクトリの選択](#)
- [バックアップ計画の準備](#)
- [テスト計画の作成](#)

新しいリリースの機能の理解

アップグレード処理を計画する前に、Oracle9i の新しいリリースの機能を理解する必要があります。Oracle リリース間の相違点については、『Oracle9i データベース新機能』を参照してください。各コンポーネントの新機能については、Oracle9i ドキュメント・セットにある各コンポーネント固有のドキュメントを参照してください。たとえば、Oracle9i Real Application Clusters の変更点については、『Oracle9i Real Application Clusters 概要』を参照してください。

新しいリリースへのアップグレード・パスの決定

Oracle9i の新しいリリースへアップグレードするために必要なパスは、ご使用のデータベースのリリースによって異なります。表 2-1 に、リリースごとに必要なアップグレード・パスを示します。ご使用のデータベースに固有のアップグレード・パスおよびドキュメントを使用してアップグレードします。

表 2-1 アップグレード・パス

現行リリース	アップグレード・パス
7.3.3 以下	<p>直接のアップグレードはサポートされていません。新しいリリースへアップグレードするには、次の手順を実行します。</p> <ol style="list-style-type: none"> 1. リリース 7.3 の『Oracle7 Server 移行ガイド』およびリリース 7.3.4 の README の指示に従って、リリース 7.3.4 へアップグレードします。 2. 第 3 章「Oracle9i の新しいリリースへのデータベースのアップグレード」 および 付録 D「移行ユーティリティによる Oracle7 データベースのアップグレード」 の指示に従って、リリース 7.3.4 のデータベースを新しいリリースへアップグレードします。
7.3.4	<p>直接のアップグレードがサポートされています。第 3 章「Oracle9i の新しいリリースへのデータベースのアップグレード」 および 付録 D「移行ユーティリティによる Oracle7 データベースのアップグレード」 の指示に従って、新しいリリースへアップグレードします。</p>
8.0.3	直接のアップグレードはサポートされていません。新しいリリースへアップグレードするには、次の手順を実行します。
8.0.4	
8.0.5	<ol style="list-style-type: none"> 1. リリース 8.0.6 の README.MIG.doc ファイルの指示に従って、リリース 8.0.6 へアップグレードします。 2. 第 3 章「Oracle9i の新しいリリースへのデータベースのアップグレード」 の指示に従って、リリース 8.0.6 のデータベースを新しいリリースへアップグレードします。
8.0.6	<p>直接のアップグレードがサポートされています。第 3 章「Oracle9i の新しいリリースへのデータベースのアップグレード」 の指示に従って、新しいリリースへアップグレードします。</p>
8.1.5	直接のアップグレードはサポートされていません。新しいリリースへアップグレードするには、次の手順を実行します。
8.1.6	<ol style="list-style-type: none"> 1. 『Oracle8i 移行ガイド』の指示に従って、リリース 8.1.7 へアップグレードします。 2. 第 3 章「Oracle9i の新しいリリースへのデータベースのアップグレード」 の指示に従って、リリース 8.1.7 のデータベースを新しいリリースへアップグレードします。
8.1.7	直接のアップグレードがサポートされています。 第 3 章「Oracle9i の新しいリリースへのデータベースのアップグレード」 の指示に従って、新しい
1 (9.0.1)	リリースへアップグレードします。

直接のアップグレードがサポートされていないリリースのデータベースをご使用の場合、Oracle9i の新しいリリースへアップグレードする前に、まずサポートされている Oracle リリースへデータベースをアップグレードする必要があります。

アップグレード方法の選択

データベースを Oracle9i の新しいリリースへアップグレードするには、次のいずれかの方法を選択します。

- Database Upgrade Assistant (DBUA) を使用する。

Database Upgrade Assistant は、(特定の形式のインストールを選択した場合に) Oracle Universal Installer から起動でき、これを使用すると Graphical User Interface (GUI) による指示に従ってデータベースをアップグレードできます。インストール中に、Database Upgrade Assistant を使用しない方法を選択し、かわりに、将来データベースをアップグレードする際に、Database Upgrade Assistant をスタンドアロン・ツールとして起動することを選択できます。

- 手動アップグレードを実行する。

手動アップグレードでは、SQL スクリプトおよびユーティリティを使用して、コマンドラインでデータベースをアップグレードできます。

- データベースからの全体または部分エクスポートを実行した後、新しい Oracle9i データベースへ全体または部分インポートを実行する。

エクスポート / インポートによって、データベース内のデータのサブセットをコピーできます。エクスポート / インポートでは、データベースを変更せずにデータのコピーが作成されます。

- SQL*Plus の COPY コマンドまたは SQL の CREATE TABLE 文の AS 句を使用し、あるデータベースから Oracle9i データベースへデータをコピーする。

データのコピーによって、データベース内のデータのサブセットをコピーできます。データのコピーでは、データベースを変更せずにデータのコピーが作成されます。

次の項では、それぞれのアップグレード方法について詳しく説明し、各方法のメリットおよびデメリットについて説明します。

Database Upgrade Assistant

Database Upgrade Assistant は、データベースを Oracle9i の新しいリリースへ簡単にアップグレードできる Graphical User Interface (GUI) ツールです。Database Upgrade Assistant の使用に役立つオンライン・ヘルプが使用可能です。

Database Upgrade Assistant は、次のすべてのアップグレード前の手順を実行します。

- アップグレードするデータベースを分析し、アップグレードが必要なデータベースのコンポーネントを判断します。データベースのアップグレードが完了した後、これらのコンポーネントは自動的にアップグレードされます。

- SYSTEM 表領域内の使用可能な領域を確認します。
- オプションで、すべての必要なデータベース・ファイルのバックアップを取ります。
- アップグレードが正常に行われるために必要なパラメータ・ファイルを調整します。
アップグレードが完了した後、一時的に調整された初期化パラメータを元の値に戻します。

アップグレード処理中に、Database Upgrade Assistant によって、すべての必要な SQL スクリプトおよびユーティリティの実行、廃止された初期化パラメータの削除、非推奨の初期化パラメータの調整、およびアップグレード中に実行されたすべての SQL スクリプトおよびユーティリティの詳細ログの作成が行われます。

アップグレードが完了した後、Database Upgrade Assistant によって、アップグレード結果の詳細を示すダイアログ・ボックスが表示されます。

Oracle9i リリース 2 (9.2) 以上では、Database Upgrade Assistant はクラスタ・データベースのアップグレードをサポートします。

Database Upgrade Assistant を使用するメリット

Database Upgrade Assistant を使用するメリットは、次のとおりです。

- アップグレード処理中に予測されるすべてのエラーが削除されます。
- 使用可能なリソースが十分あることが保証されます。

Database Upgrade Assistant では、アップグレードを完了するために、いくつかの手順を実行します。アップグレード・スクリプトが原因で発生することが予測されるすべてのエラーが削除されます。

アップグレード中に予測しないエラーが発生した場合、現在の手順をスキップし、次の手順に進むこともできます。アップグレードの完了後、エラーの原因を修正し、Database Upgrade Assistant を再起動できます。Database Upgrade Assistant は、スキップされたすべての手順を完了させることによってアップグレードを再開します。

たとえば、Oracle Spatial のアップグレード中に予測しないエラーが発生した場合、Oracle Spatial のアップグレードをスキップし、次のコンポーネントのアップグレードへ進むことができます。Database Upgrade Assistant がすべてのコンポーネントのアップグレードを完了した後、Database Upgrade Assistant を再起動して Oracle Spatial をアップグレードできます。

手動アップグレード

手動アップグレードでは、コマンドラインから SQL スクリプトおよびユーティリティを実行して、データベースを **Oracle9i** の新しいリリースへアップグレードします。

データベースを手動でアップグレードする場合、次のアップグレード前の手順を実行する必要があります。

- **SYSTEM** 表領域に十分な領域があることを確認し、十分な領域がない場合は、空き領域を追加します。
- アップグレード用のパラメータ・ファイルを調整して、アップグレードの問題が発生する可能性のある初期化パラメータを使用不可にします。アップグレードが完了した後、変更された初期化パラメータを元の値に戻す必要があります。

また、廃止された初期化パラメータをパラメータ・ファイルから削除し、非推奨の初期化パラメータなどの他の初期化パラメータの変更を考慮する必要もあります。

- データベースのバックアップを実行します。
- データベースのアップグレード中に、自動的にアップグレードされないデータベースのコンポーネントがあります。データベースのアップグレードが完了した後、これらのコンポーネントを手動でアップグレードする必要があります。

アップグレードしているデータベースのリリースに応じて、追加のアップグレード前の手順の実行が必要な場合があります。

手動アップグレードでは、アップグレード処理をより詳細に制御できますが、実行されなかったアップグレード手順またはアップグレード前の手順がある場合、または不適切な順序で実行された場合にエラーが発生する可能性が高くなります。**Database Upgrade Assistant** を使用すると、すべての必要なアップグレード前の手順およびアップグレード手順が実行されます。

エクスポート / インポート

Database Upgrade Assistant または手動アップグレードとは異なり、エクスポート / インポート・ユーティリティは、現行のデータベース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。現行のデータベースのエクスポート・ユーティリティは、データベースの特定部分をエクスポート・ダンプ・ファイルにコピーします。次に、**Oracle9i** の新しいリリースのインポート・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 **Oracle9i** データベースにロードします。ただし、エクスポート・ダンプ・ファイルをコピーする前に、新しい **Oracle9i** データベースを準備しておく必要があります。

以前のリリースからデータをインポートする場合、**Oracle9i** のインポート・ユーティリティでは、以前のリリースのエクスポート・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。

次の項では、データベースのアップグレードにエクスポート / インポートを使用するかどうかを決定する際に有効な、エクスポート / インポートの特長について説明します。

エクスポート/インポートによってアップグレードされたデータベースへの影響 エクスポート/インポートによるアップグレード方法では、現行のデータベースが変更されません。したがって、データベースは、アップグレード処理を通して常に使用可能な状態です。ただし、一貫性のあるデータベースの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）、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。現行のデータベースを使用可能な状態にしておくことができるため、たとえば、既存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 Oracle9i 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。このアップグレード処理中にデータベースの完全な一貫性を維持するには、データベースのデータを変更する場合に、新しい Oracle9i 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。

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、新しいデータベースが作成されることです。現行の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、アップグレードしたデータベースは、元のデータベースとは異なる方法で運用される場合があります。たとえば、エクスポート/インポート操作によってデータベースと同一のコピーを作成しても、データのディスク配置やチューニング・パラメータの解除など、その他の要素が予期しないパフォーマンスの劣化につながる場合があります。

エクスポート/インポートのメリット エクスポート/インポートを使用したアップグレードには、次のようなメリットがあります。

- データの断片化を解消できます。インポートされたデータを圧縮することによって、パフォーマンスを向上できます。
- データベースを再構成できます。新しい表領域を作成するか、または既存の表や表領域、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ます。
- 特定のデータベース・オブジェクトまたはユーザーをコピーできます。オブジェクト、ユーザー、およびその他希望する項目のみをインポートできます。
- バックアップ・アーカイブとして機能します。全データベース・エクスポートを現行のデータベースのアーカイブとして使用できます。

エクスポート/インポートの時間要件 エクスポート/インポートを使用する全データベースのアップグレードは、Database Upgrade Assistant を使用する場合または手動アップグレードを実行する場合と比較して、時間が長くなる場合があります。したがって、ピーク時を外してアップグレードするようにスケジュールするか、またはアップグレード中に現行のデータベースに対して行われた変更を新しい Oracle9i データベースに反映させる準備が必要です。

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 コマンドを使用して 1 つのデータベースの表から別のデータベースの表にデータをコピーしたり、INSERT INTO 文および CREATE TABLE...AS 文を使用して新しい表を作成し、その表にデータを書き込むことができます。

データのコピーを使用するアップグレード方法には、エクスポート / インポートと同様のメリットがあります。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のみをコピー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新しい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のみをコピーする場合は、データのコピーを使用します。逆に、エクスポート / インポートを使用すると、表全体のみのコピーが可能です。

新しいリリースに対する Oracle ホーム・ディレクトリの選択

Oracle9i の新しいリリース用に、現行のリリースの Oracle ホーム・ディレクトリとは別の Oracle ホーム・ディレクトリを選択する必要があります。新しい Oracle9i ソフトウェアは、現行のリリースと同じ Oracle ホーム・ディレクトリにはインストールできません。

別のインストール・ディレクトリを使用すると、既存のソフトウェアをインストールしたままで、新しい Oracle9i ソフトウェアを使用できます。この方法によって、本番環境全体を置き換える前に、テスト・データベース上でアップグレード処理をテストできます。

バックアップ計画の準備

アップグレード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、本番データベースの実行不可能状態の許容範囲がどの程度の期間か。
- 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る必要があるか。
- 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リストアできるか（オフサイト記憶域でのバックアップを含む）。
- リカバリ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リカバリするための手順を備えている必要があります。

参照： Oracle7 データベースについては、『Oracle7 Server 管理者ガイド』を参照してください。Oracle9i データベースについては、『Oracle9i ユーザー管理バックアップおよびリカバリ・ガイド』を参照してください。

テスト計画の作成

アップグレード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する必要があります。緻密なテストを正常に完了できれば、本番データベースのアップグレード処理を十分に理解し、予測できるようになり、アップグレード処理の成功が確実になります。本番データベースをアップグレードする前に、できるだけ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。

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。

アップグレード・テスト

アップグレード・テストでは、現行のデータベースから新しい Oracle9i データベースへのアップグレード・パスを計画およびテストする必要があります。Database Upgrade Assistant の使用、手動アップグレードの実行、エクスポート / インポートの使用またはデータのコピーのいずれかの方法を使用します。

どのアップグレード方法を選択する場合でも、アップグレード計画を作成、テストおよび検証する必要があります。

最小テスト

最小テストでは、現行のデータベースからアプリケーションの全部または一部を新しい Oracle9i データベースに移動し、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、現実の本番環境で発生するような潜在的問題は検出されません。ただし、アプリケーションの起動または呼出しに関する問題がある場合は、すぐに検出されます。

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アップグレード後にテストする一連のテストです。機能テストには、すべてのデータベース、ネットワーキングおよびアプリケーション・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アップグレード前と同様に機能し、新機能が正常に動作していることを検証することです。

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テストします。統合テストを計画するときは、次のことを考慮する必要があります。

- 新しい Oracle9i 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。
- Graphical User Interface を他のコンポーネントでテストする必要があります。
- 新しい Oracle9i 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）などの新しい Oracle9i 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ります。
- 2つのコンポーネントが SQL*Net、Net8 または Oracle Net Services を使用して接続されている場合は、ストレス・テストとともにその接続のテストも行う必要があります。

パフォーマンス・テスト

新しい Oracle9i データベースのパフォーマンス・テストでは、新しい Oracle9i データベースでの様々な SQL 文のパフォーマンスを、現行データベースでの SQL 文のパフォーマンスと比較します。アップグレードする前に、現行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する必要があります。

たとえば、Oracle9i Real Application Clusters を使用している場合、Oracle9i の新しいリリースへアップグレードしたときにキャッシュ・フュージョンによってもたらされるパフォーマンスの向上を測定するには、アップグレード前にシステム統計を記録する必要があります。キャッシュ・フュージョン用に、V\$SYSSTAT、V\$LOCK_ACTIVITY および V\$LOCK_CLASS_PING ビューからの統計を記録します。それによって、キャッシュ・フュージョン前後のパフォーマンス統計を比較できます。

最高の結果を得るために、SQL スクリプト utlbstat.sql および utlestat.sql を実行して、特定の期間の V\$SYSSTAT 統計を収集します。一貫性のあるトランザクション・アクティビティ・レベルのピーク時の負荷を、最も一貫して反映している収集タイムフレームを使用します。V\$LOCK_ACTIVITY および V\$LOCK_CLASS_PING からデータを取得するには、統計収集期間の最初と最後で SELECT * 文を使用します。キャッシュ・フュージョンを Oracle9i の新しいリリースで実行した後にこの処理を繰り返し、『Oracle9i Real Application Clusters 配置およびパフォーマンス』の説明に従って、システムのパフォーマンスを見積もります。

参照： チューニングの詳細は、『Oracle9i データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』を参照してください。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、TKPROF で SQL トレース機能およびプロファイルを使用可能にしてください。

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、アップグレード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・ストレス・テストの目的は、様々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。そのかわりに、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、パフォーマンスおよび統合に関する問題点を検証するために使用するものであり、ボリューム / ロード・ストレス・テストのかわりになるものではありません。

データベースをアップグレードした後、すべてのデータがアクセス可能で、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、これらのテスト処理を自動化してください。

テスト計画は、サイトで実行される作業を反映する必要があります。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。

特定のアップグレード前後のテスト

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・ディクショナリの成長の監視
- ロールバック、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って、現行のデータベースと新しい Oracle9i データベースを簡単に比較できます。

Oracle で各 SQL 文を実行するための実行計画を決定するには、以前のデータベースと新しい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 句を使用します。

アップグレード後に、新しい Oracle9i データベースの実行計画と現行の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、新しい Oracle9i データベース上でその文を実行し、そのパフォーマンスを、現行のデータベース上で実行した文のパフォーマンスと比較します。

参照： EXPLAIN PLAN の詳細は、『Oracle9i データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』を参照してください。

アップグレード処理のテスト

現行の本番データベースに影響しないテスト環境を作成します。テスト環境は、選択したアップグレード方法によって異なります。

- Database Upgrade Assistant を使用する場合は、手動アップグレードを実行する場合は、現行の本番データベースのテスト・バージョン（通常はサブセット）を作成してアップグレードをテストします。
- エクスポート / インポートを使用する場合は、現行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。

テスト環境を使用してデータベースのアップグレードを行います。最適なアップグレードのテストとは、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、アップグレード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。

注意： このデータベースのテスト・サブセットを正常にアップグレードして、次の手順で説明するアプリケーションでテストするまでは、実際の本番データベースをアップグレードしないでください。

新しい Oracle9i データベースで使用する 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。それによって、現行の本番データベースをアップグレードする前に、それらのアプリケーションをサンプルのデータベースでテストできます。詳細は、6-3 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。

アップグレードしたテスト・データベースのテスト

現行のデータベースおよび Oracle9i の新しいリリースへアップグレードされたテスト・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、アップグレードのテストを繰り返します。

新しい Oracle9i 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を確認するために、この新しくアップグレードされた Oracle9i のテスト・データベースをテストします。また、利用可能な Oracle9i の機能を使用する機能を追加して、機能拡張についてもテストします。ただし、アプリケーションが現行のデータベースの場合と同様に動作するかを最初に確認してください。

参照： Oracle9i におけるアプリケーションの使用の詳細は、[第 6 章「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。

Oracle9i の新しいリリースへのデータベースのアップグレード

この章では、データベースを Oracle9i の新しいリリースへアップグレードする処理について説明します。この章では、次の項目について説明します。

- [Oracle9i リリース 2 \(9.2\) ソフトウェアのインストール](#)
- [Database Upgrade Assistant を使用したデータベースのアップグレード](#)
- [データベースの手動でのアップグレード](#)

参照： アップグレード処理の一部は、オペレーティング・システム固有のもので、ご使用のオペレーティング・システムでのアップグレードに関するその他の指示は、オペレーティング・システム固有の Oracle マニュアルを参照してください。

Oracle9i リリース 2 (9.2) ソフトウェアのインストール

Oracle9i リリース 2 (9.2) ソフトウェアをインストールするには、次の手順を実行します。

1. オペレーティング・システムが UNIX の場合は、Oracle ホーム・ディレクトリ、Oracle ベース・ディレクトリおよびその配下のすべてのサブディレクトリに対する書込み権限を持つユーザーとしてログインしていることを確認します。
2. オペレーティング・システム固有のマニュアルを参照して、インストールの準備を行い、Oracle Universal Installer を起動します。

クラスタ・データベースをアップグレードする場合のインストールの詳細は、『Oracle9i Real Application Clusters セットアップおよび構成』を参照してください。

3. Oracle Universal Installer の「Welcome」画面で、「Next」をクリックします。「File Locations」画面が表示されます。

ヘルプが必要な場合、または Oracle Universal Installer の詳細を参照する場合は、「Help」ボタンをクリックしてオンライン・ヘルプを開きます。

4. 「File Locations」画面で、次の手順を実行します。
 - a. 「Source」フィールドのテキストを変更しないでください。これは、インストールするファイルの場所です。
 - b. 「Destination Name」フィールドに、新しい Oracle ホームの名前を入力します。
 - c. 「Destination Path」フィールドに、新しいリリースをインストールする Oracle ホーム・ディレクトリのフルパスを入力します。「Browse」ボタンをクリックして、ディレクトリにナビゲートします。

注意： 新しい Oracle9i リリース 2 (9.2) は、古いリリースとは別の Oracle ホームにインストールする必要があります。

- d. 「Next」をクリックします。

「Available Products」画面が表示されます。

5. 「Available Products」画面で、Oracle9i Database を選択します。その後、「Next」をクリックします。

「Installation Types」画面が表示されます。

6. 「Installation Types」画面で、「Enterprise Edition」、「Standard Edition」または「Custom」を選択します。その後、「Next」をクリックします。

「Enterprise Edition」または「Standard Edition」を選択した場合、「Database Configuration」画面が表示されます。次の手順を実行します。

- a. 「Software Only」を選択します。
- b. 「Next」をクリックします。

注意： 通常、既存のデータベースをアップグレードする場合は、初期データベースをインストールしないでください。

「Custom」を選択した場合、「Available Product Components」画面が表示されます。次の手順を実行します。

- a. インストールする製品コンポーネントを選択します。その後、「Next」をクリックします。

Database Upgrade Assistant をインストールする場合は Oracle Utilities、Oracle7 からアップグレードする場合は移行ユーティリティがインストールされていることを確認します。

特定のオプションの使用を中止しないかぎり、以前のデータベースにインストール済のすべてのオプションをインストールします。たとえば、以前のデータベースに Oracle Text をインストールしていた場合、新しい Oracle9i データベースには Oracle Text をインストールします。

- b. Oracle9i Real Application Clusters をインストールする場合、「Cluster Node Selection」画面で、ソフトウェアをインストールするノードを選択します。その後、「Next」をクリックします。
- c. 「Upgrading an Existing Database」画面が表示されるまで、画面の指示に従って Custom インストール設定を行います。
- d. 「Create Database」画面が表示されたら、「No」オプションを選択します。これは、既存のデータベースをアップグレード中であるため、データベースを作成しないことを示します。その後、「Next」をクリックします。

7. 「Upgrading an Existing Database」画面で、次の手順を実行します。
 - a. Database Upgrade Assistant を使用してデータベースをアップグレードするには、「Upgrade an Existing Database」チェック・ボックスを選択します。

データベースを手動でアップグレードする場合、またはインストールが完了した後に Database Upgrade Assistant を単独で実行する場合は、「Upgrade an Existing Database」チェック・ボックスを選択しないでください。
 - b. 「Next」をクリックします。
8. 「Summary」画面で、インストールの設定および選択が正しいかどうかを確認します。その後、「Install」をクリックします。Oracle Universal Installer がインストールを実行します。

インストールが完了したら、1 つ以上のアシスタントが起動する場合があります。インストール中に Database Upgrade Assistant を実行するように選択した場合は、Database Upgrade Assistant が起動するため、アップグレードを続行できます。3-5 ページの「[Database Upgrade Assistant を使用したデータベースのアップグレード](#)」を参照してください。

インストールが正常に完了したら、「Exit」ボタンをクリックして Oracle Universal Installer を閉じます。

Database Upgrade Assistant の単独での実行

既存のデータベースのアップグレードを指定しないで Oracle9i の新しいリリースをインストールした場合、インストール完了後、Database Upgrade Assistant を単独で実行できます。

Database Upgrade Assistant を単独で実行するには、次の手順を実行します。

1. Database Upgrade Assistant を起動します。

UNIX プラットフォームの場合、システム・プロンプトに次のコマンドを入力します。

```
dbua
```

Windows オペレーティング・システムの場合、次のように選択します。

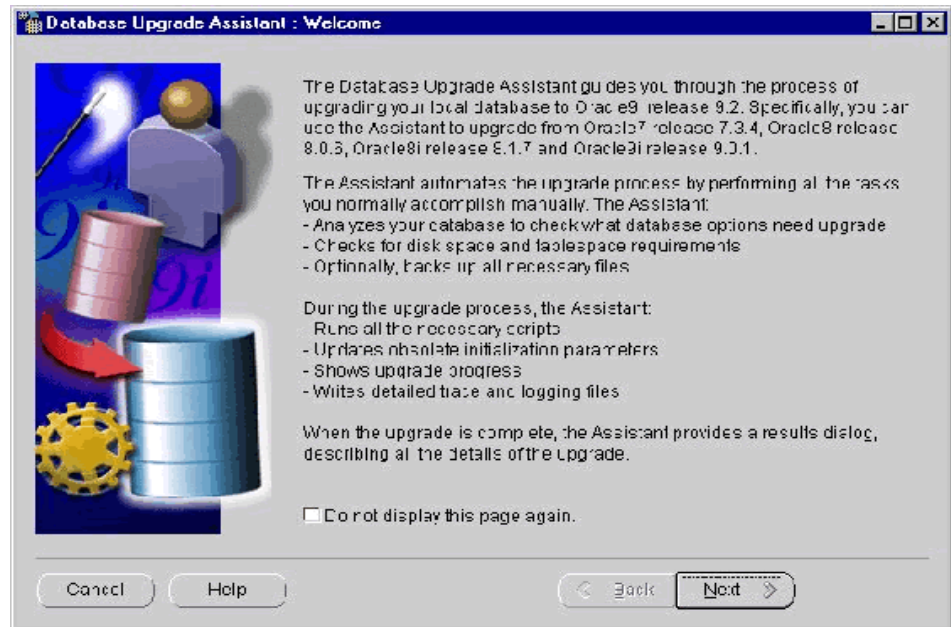
「スタート」>「プログラム」>「Oracle - HOME_NAME」>「Configuration and Migration Tools」>「Database Upgrade Assistant」

Database Upgrade Assistant が起動すると、アップグレードを続行できます。

Database Upgrade Assistant を使用したデータベースのアップグレード

Database Upgrade Assistant が起動すると、「Welcome」画面が表示されます。

図 3-1 Database Upgrade Assistant の「Welcome」画面



Database Upgrade Assistant を使用してデータベースをアップグレードするには、次の手順を実行します。

1. Database Upgrade Assistant の「Welcome」画面（図 3-1 を参照）で、アップグレードするデータベースが指定した条件を満たすことを確認します。その後、「Next」をクリックします。

ヘルプが必要な場合、または Database Upgrade Assistant の詳細を参照する場合は、「Help」ボタンをクリックしてオンライン・ヘルプを開きます。

2. 「Select a Database」画面で、アップグレードするデータベースを選択します。その後、「Next」をクリックします。

オペレーティング・システムの認証がない場合は、SYSDBA 権限でユーザー名およびパスワードを入力する必要があります。

3. Oracle7 データベースをアップグレードする場合は、「Database Name」画面で新しいデータベース名を指定できます。その後、「Next」をクリックします。
4. 「Password」画面で、表示された各ユーザーに対して有効なパスワードを入力します。その後、「Next」をクリックします。

この画面は、Database Upgrade Assistant が、アップグレードするユーザーにパスワードを要求する場合のみ表示されます。
5. 「Backup」画面には、次の 2 つのオプションがあります。
 - Database Upgrade Assistant を実行する前にバックアップを取った場合は、「I have already backed up my database」を選択します。
 - バックアップを取っていない場合は、「I would like this tool to back up the database」を選択します。このオプションを選択した場合、「Browse」ボタンをクリックしてバックアップ・ディレクトリを選択できます。

選択後、「Next」をクリックします。

6. 「Network Configuration for the database」画面には、次の 2 つのタブがあります。

リリース 2 (9.2) の Oracle ホームに複数のリスナーが存在する場合、「Listeners」タブが表示されます。リリース 2 (9.2) の Oracle ホームで、アップグレードされたデータベースを登録するリスナーを選択します。

リリース 2 (9.2) の Oracle ホームにディレクトリ・サービスが構成されている場合、「Directory Service」タブが表示されます。ディレクトリ・サービスを使用して、アップグレードされたデータベースを登録するかどうかを選択できます。
7. 「Summary」画面で、すべての設定が正しいことを確認します。不適切な設定がある場合、その設定を修正できる画面が表示されるまで「Back」をクリックします。すべて正しい場合は、「Finish」をクリックします。

Database Upgrade Assistant によって、アップグレード中にデータベースに設定される初期化パラメータが表示されます。COMPATIBLE 初期化パラメータは、8.1.0 以上に設定されます。

参照： アップグレード後の COMPATIBLE 初期化パラメータの設定の詳細は、[第 5 章「互換性および相互運用性」](#)を参照してください。

8. 「Progress」ダイアログ・ボックスが表示され、Database Upgrade Assistant がアップグレードを開始します。

「Ignore」、「Abort」および「Skip the Step」を選択できるエラー・メッセージが表示される場合があります。その他のエラー・メッセージが表示された場合、それに応じて問題を解決する必要があります。エラーが重大で、アップグレード中に処理できない場合、次のいずれかを選択します。

- 「Skip the Step」が表示された場合、そのボタンをクリックすると、現在のアップグレード手順がスキップされます。

これによって、Database Upgrade Assistant は、この手順およびすべての関連する手順をスキップして、次の手順へ進みます。アップグレードが完了した後、問題を修正して、Database Upgrade Assistant を再起動し、スキップされた手順を完了できます。

- 「Skip the Step」が表示されない場合、「Abort」ボタンをクリックして処理を中断する必要があります。

これによって、アップグレード処理が中断されます。Database Upgrade Assistant によってデータベースのバックアップが取られた場合は、データベースのリストアが要求されます。

データベースがリストアされた後、エラーの原因を修正して Database Upgrade Assistant を再起動し、アップグレードを再実行する必要があります。

データベースをリストアしない場合は、手動アップグレードを続行できるように、データベースの現在の状態が保持されます。

アップグレードが完了した後、「Progress」ダイアログ・ボックスに次のメッセージが表示されます。

Upgrade has been completed. Click the "OK" button to see the results of the upgrade.

「OK」ボタンをクリックします。

9. 「Results」ダイアログ・ボックスで、アップグレードの詳細を確認できます。ユーザー・アカウントのロックを解除して、パスワードを設定することもできます。

アップグレードの結果に問題がある場合は、データベースを以前のリリースへリストアできます。

アップグレードの結果に問題がない場合は、「Done」ボタンをクリックします。Database Upgrade Assistant は、アップグレードされたデータベースのエントリを古い listener.ora ファイルから削除し、古いデータベースのリスナーを再ロードします。

- a. Database Upgrade Assistant は、次のいずれかの方法で、Oracle9i の listener.ora ファイルのアップグレードされたデータベースに対する SID_DESC エントリを変更します。

簡単な場合: 古い listener.ora に、次の SID_DESC エントリがあるとします。

```
...
(SID_DESC =
(SID_NAME = ORCL)
)
...
```

データベース名が SAL、ドメイン名が COM、Oracle ホームが /oracle/product/9.2 の場合、Database Upgrade Assistant は次のエントリを追加します。

```
...
(SID_DESC =
  (GLOBAL_DBNAME = sal.com)
  (ORACLE_HOME = /oracle/product/9.2)
  (SID_NAME = SAL)
)
...
```

複雑な場合: 古い listener.ora に、次の SID_DESC エントリがあるとします。

```
...
(SID_DESC =
  (GLOBAL_DBNAME = an_entry)
  (SID_NAME = ORCL)
)
...
```

an_entry が移行したデータベースの GLOBAL_DBNAME と一致しないとき、データベース名が SAL、ドメイン名が COM、Oracle ホームが /oracle/product/9.2 の場合、Database Upgrade Assistant は次のエントリを追加します。

```
...
(SID_DESC =
  (GLOBAL_DBNAME = sal.com)
  (ORACLE_HOME = /oracle/product/9.2)
  (SID_NAME = SAL)
)
...
```

このエントリは簡単な場合のエントリと同じですが、Database Upgrade Assistant はエントリ *an_entry* の listener.ora ファイルにある SERVICE_NAMES パラメータへの追加も行います。したがって、Database Upgrade Assistant は次のように SERVICE_NAMES パラメータを変更します。

```
SERVICE_NAMES = sal.com, an_entry
```

- b. Database Upgrade Assistant は、アップグレードしたデータベースのエントリを古い listener.ora ファイルから削除します。
- c. Database Upgrade Assistant は、古い環境と新しい Oracle9i 環境で listener.ora ファイルを再ロードします。

10. 第4章「データベースのアップグレード後の作業」に示す手順を実行します。

注意： 古い Oracle ソフトウェアを保持している場合、古いソフトウェアを使用してアップグレードしたデータベースを起動しないでください。新しい Oracle9i インストールの実行可能ファイル以外で、データベースを起動しないでください。また、古い Oracle 環境を削除する前に、Oracle 環境のデータ・ファイルをすべて新しい Oracle9i 環境に再配置してください。データ・ファイルの再配置の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

データベースの手動でのアップグレード

次の項では、手動でのアップグレードを実行手順について説明します。

- [システムの考慮点および要件](#)
- [アップグレードするデータベースの準備](#)
- [データベースのアップグレード](#)
- [特定のコンポーネントのアップグレード](#)
- [手動でのアップグレードのトラブルシューティング](#)

システムの考慮点および要件

次の項では、システムに関する考慮点および要件について説明します。

クラスタ・データベースのアップグレード

クラスタ・データベースをアップグレードする場合、この項で説明するほとんどの処理は、システムの 1 つのノードのみで実行する必要があります。したがって、特に指示がないかぎり、この項に説明する処理は 1 つのノードのみで実行してください。

異なるオペレーティング・システムへの移行

アップグレード処理中に、異なるオペレーティング・システムが存在するコンピュータ・システムへデータベースを移行することはできません。たとえば、Solaris 上の Oracle7 を Windows 2000 上の Oracle9i に移行できません。ただし、エクスポート / インポートを使用することによって、異なるオペレーティング・システムのデータベースの移行が可能です。

注意： アップグレード処理の際、ワード・サイズの変更がサポートされています。ワード・サイズの変更には、同じオペレーティング・システム内で 32 ビットから 64 ビット・アーキテクチャへの切替えが伴います。詳細は、1-10 ページの「[ワード・サイズの変更](#)」を参照してください。

リリース 8.1.7 以下のデータベース・キャラクタ・セットに関する考慮点

Oracle9i では、SQL NCHAR データ型 (NCHAR、NVARCHAR2 および NCLOB) は、Unicode キャラクタ・セット・コード (UTF8 および AL16UTF16) のみです。アジア言語のキャラクタ・セット (たとえば、JA16SJISFIXED など) を含む、NCHAR データ型で使用可能だった Oracle8 および Oracle8i のその他のキャラクタ・セットは、今後サポートされません。

SQL NCHAR 型のデータを新しい Unicode NCHAR へ移行する前に、無効なキャラクタ・セットの変換またはデータの切捨てを識別するために、キャラクタ・セット・スキャナを使用してご使用の SQL NCHAR データを分析することをお勧めします。

参照： キャラクタ・セット・スキャナの詳細は、『Oracle9i Database グローバリゼーション・サポート・ガイド』を参照してください。

Oracle9i へアップグレードする場合、アップグレードされたデータベースの各国語キャラクタ・セットの値は、アップグレードされる Oracle8 および Oracle8i データベースの各国語キャラクタ・セットの値を基に設定します。

アップグレード前の各国語キャラクタ・セットが UTF8 の場合、新しい各国語キャラクタ・セットも UTF8 となります。それ以外の場合は、各国語キャラクタセットを AL16UTF16 に変更します。

アップグレード中、データ・ディクショナリの既存の NCHAR 列が変更され、新しい Oracle9i の書式が使用可能になります。また、各国語キャラクタ・セットが AL16UTF16 に変更されている場合、データ・ディクショナリの NCHAR 列は AL16UTF16 キャラクタ・セットに変換されます。

注意： ユーザー表の NCHAR 列は、アップグレード中では変更されません。ユーザー表の NCHAR 列の変更の詳細は、4-11 ページの「[ユーザーの NCHAR 列のアップグレード](#)」を参照してください。

レプリケーション環境に関する考慮点

Oracle9i で CHAR 列長セマンティクスを使用する場合、あるいは NCHAR 列または NVARCHAR2 列を持つ表がレプリケーション・データベースに含まれている場合は、この項に示す Oracle9i へのレプリケーション環境のアップグレードに関する説明を参照してください。

CHAR 列長セマンティクス レプリケーション・データベースを Oracle9i へアップグレードした後で、CHAR 列長セマンティクスを使用する場合は、レプリケーション環境にあるデータベースに関連するすべてのデータベースで、CHAR 列長セマンティクスを使用する必要があります。この場合、レプリケーション環境に関連するすべてのデータベースを同時にアップグレードすることをお勧めします。これは、レプリケーション環境にあるマスター・サイトおよびマテリアライズド・ビュー・サイトの両方に適用されます。

レプリケーション環境にあるすべてのデータベースを同時にアップグレードできない場合は、Oracle9i より前のすべてのデータベースでシングルバイト・キャラクタ・セットが使用されている場合にのみ、Oracle9i データベースで CHAR 列長セマンティクスを使用できます。それ以外の場合は、レプリケーション環境にある他のすべてのデータベースを Oracle9i へアップグレードするまで、Oracle9i データベースで CHAR 列長セマンティクスは使用しないでください。

NCHAR 列または NVARCHAR2 列 NCHAR 列または NVARCHAR2 列を持つ表がレプリケーション・データベースに含まれている場合は、レプリケーション環境に関連するすべてのデータベースを同時にアップグレードすることをお勧めします。これは、レプリケーション環境にあるマスター・サイトおよびマテリアライズド・ビュー・サイトの両方に適用されます。Oracle9i では、NCHAR データ型または NVARCHAR2 データ型として指定したすべての列は、Unicode 書式で格納されます。

レプリケーション環境にあるすべてのデータベースを同時にアップグレードできない場合、Oracle9i より前のすべてのデータベースが固定幅の各国語キャラクタ・セットを使用している場合にのみ、相互運用性がサポートされます。Oracle9i より前のデータベースで可変幅キャラクタ・セットを使用している場合、レプリケーション環境にある他のデータベースを Oracle9i へアップグレードする前に、それらのデータベースを固定幅キャラクタ・セットに変換する必要があります。

参照：

- 列長セマンティクスおよび Unicode のレプリケーション・サポートの詳細は、『Oracle9i アドバンスド・レプリケーション』を参照してください。
- 列長セマンティクスおよび Unicode の概要は、『Oracle9i Database グローバリゼーション・サポート・ガイド』を参照してください。
- リリース 8.1 のキャラクタ・セットの変換については、『Oracle8i NLS ガイド』を参照してください。

アップグレードするデータベースの準備

データベースを Oracle9i の新しいリリースへアップグレードする前に、いくつかの準備処理を実行する必要があります。アップグレードするデータベースのリリースによっては、次の手順の一部またはすべての実行が必要になる場合があります。

1. SQL*Net、Net8 および Oracle Net Services に関するアップグレードの問題を確認します。

参照： 詳細は、付録 B「Oracle Net Services のアップグレードに関する考慮点」を参照してください。

- アップグレードするデータベースの Oracle ホーム・ディレクトリの所有者として、システムにログインします。
- SQL*Plus を起動します。

注意： アップグレードするデータベースがリリース 8.0.6 以下の場合、Server Manager を起動します。SQL*Plus は起動しないでください。

- SYSDBA 権限を持つユーザーとして、データベース・インスタンスに接続します。
- アップグレードするデータベースがリリース 8.0.6 以下の場合、次の手順を実行します。アップグレードするデータベースがリリース 8.1.7 以上の場合、手順 6 にスキップします。

- OUTLN という名前のユーザーまたはロールが存在しないことを確認します。これは、Oracle9i をインストールするときにこのスキーマが自動的に作成されるためです。OUTLN という名前のユーザーまたはロールが存在する場合、そのユーザーまたはロールを削除し、別の名前で再作成する必要があります。

OUTLN という名前のユーザーを確認するには、次の SQL 文を入力します。

```
SELECT username FROM dba_users WHERE username = 'OUTLN';
```

OUTLN という名前のユーザーが存在しない場合、どの行も選択されません。

OUTLN という名前のロールを確認するには、次の SQL 文を入力します。

```
SELECT role FROM dba_roles WHERE role = 'OUTLN';
```

OUTLN という名前のロールが存在しない場合、どの行も選択されません。

- アップグレードするデータベースがリリース 7.3.4 の場合、D-5 ページの「[アップグレードする Oracle7 データベースの準備](#)」に示す追加の準備手順を実行します。
- 必要に応じて、SYSTEM 表領域、およびロールバック・セグメントを保存している表領域に空き領域を追加します。

新しいリリースへアップグレードするには、SYSTEM 表領域およびロールバック・セグメントを保存している表領域に、より大きな空き領域が必要です。システムに十分な空き領域がある場合は、これらの表領域へより大きな領域を追加することを考慮してください。表 3-1 に、サポートされている各 Oracle リリースから新しい Oracle9i へアップグレードするために必要な、SYSTEM 表領域の追加の空き領域のサイズを示します。アップグレード中に領域が不足した場合は、アップグレードを再実行する必要があります。

表 3-1 SYSTEM 表領域の要件

リリース	追加の SYSTEM 表領域	追加の SYSTEM 表領域 (Oracle JVM がインストール されている場合)
9.0.1	16MB	30MB
8.1.7	52MB	80MB
8.0.6	70MB	なし
7.3.4	85MB	なし

次の例では、SYSTEM 表領域に領域を追加する方法を示します。

```
ALTER TABLESPACE system
  ADD DATAFILE '/home/user1/mountpoint/oradata/db1/system02.dbf'
  SIZE 16M
  AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED;

ALTER ROLLBACK SEGMENT rbl
  STORAGE (MAXEXTENTS UNLIMITED);
```

大規模なパブリック・ロールバック・セグメントは、多数のデータベース・オブジェクト（パッケージ、表、型など）を持つデータベースをアップグレードする場合に必要です。データベース・オブジェクトの総数が 5000 を超える場合は、70MB 以上のロールバック・セグメントを使用することをお勧めします。

データベース・オブジェクトの総数を判断するには、次の SQL 文を発行します。

```
SQL> SELECT COUNT(*) FROM dba_objects;
```

7. 次の SQL 文を発行して、手順 10 でバックアップを取るファイルを決定します。

```
SQL> SPOOL files.log
SQL> SELECT member FROM v$logfile;
SQL> SELECT name FROM v$datafile;
SQL> SELECT name from v$controlfile;
SQL> SPOOL OFF
```

files.log スプール・ファイルには、手順 10 でバックアップを取る必要があるすべてのファイルがリストされています。

8. データベース上で、SHUTDOWN IMMEDIATE を実行します。

```
SQL> SHUTDOWN IMMEDIATE
```

クラスタ・データベースをアップグレードする場合は、すべてのインスタンスを停止します。

9. ご使用のオペレーティング・システムが Windows の場合、すべての Oracle サービスが停止していることを確認します。

参照： サービスの停止方法については、『Oracle9i Database for Windows 管理者ガイド』を参照してください。

10. 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。手順 7 で作成した files.log スプール・ファイルにリストされたファイルは、必ずバックアップを取ってください。

注意： アップグレード中に問題が発生した場合、このバックアップからデータベースをリストアする必要があります。そのため、予防措置として、ここでデータベースのバックアップを取ってください。

参照：

- リリース 8.1 データベースのバックアップの詳細は、『Oracle8i バックアップおよびリカバリ・ガイド』を参照してください。
- リリース 8.0 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』を参照してください。

11. SQL*Plus を終了します。

データベースのアップグレード

データベースをアップグレードするには、次の手順を実行します。

1. リリース 7.3.4 からアップグレードする場合は、次の手順を実行します。
 - a. D-9 ページの「[移行ユーティリティ・コマンドライン・オプションの検討](#)」および D-10 ページの「[移行ユーティリティの実行](#)」に示す手順を実行します。
 - b. データベースの制御ファイルを削除または名前を変更するか、CONTROL_FILES 初期化パラメータを使用して、新しい制御ファイル名を指定します。
CONTROL_FILES 初期化パラメータは、通常、初期化パラメータ・ファイルで設定しますが、クラスタ・データベースをアップグレードする場合は、initdb_name.ora ファイルで設定する場合があります。

アップグレード処理の後半で ALTER DATABASE CONVERT 文を発行します。この文は、新しい制御ファイルを自動的に作成します。CONTROL_FILES 初期化パラメータを使用しない場合、この文は以前のデータベースの（CONVERT ファイルから派生した）制御ファイル名を使用し、その制御ファイルがすでに存在していたときはエラーを戻します。エラーが発生した場合は、制御ファイルを削除するか、または制御ファイル名を変更する必要があります。

ただし、CONTROL_FILES 初期化パラメータを使用して新しい制御ファイルを指定する場合は、指定した名前を持つ新しい制御ファイルが ALTER DATABASE CONVERT 文によって作成されるため、古い制御ファイルを削除する必要はありません。既存の制御ファイルの完全なリストは、3-13 ページの手順 7 で作成した files.log スプール・ファイルで確認できます。

制御ファイルは、Oracle7 よりも Oracle9i の方がかなり大きくなります。たとえば、Oracle7 の数百 KB の制御ファイルが、Oracle9i では数十 MB のサイズに増大する場合があります。Oracle9i においてサイズが大きくなるのは、バックアップや表領域レコードなど、より多くの情報が制御ファイル内に記憶されるためです。制御ファイルが RAW デバイス上にある場合、または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。

注意： CONTROL_FILES 初期化パラメータは、1 つ以上の制御ファイルの名前をカンマで区切って指定します。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用するか、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、『Oracle9i データベース管理者ガイド』を参照してください。

注意： Oracle9i Real Application Clusters に対して、すべてのノードでこの手順を実行します。

2. ご使用のオペレーティング・システムが Windows の場合、次の手順を実行します。
- a. アップグレードするデータベースの Oracle サービス OracleServiceSID を停止します。SID はインスタンスの名前です。たとえば、SID が ORCL の場合、コマンド・プロンプトで次のように入力します。

C:¥> NET STOP OracleServiceORCL
 - b. コマンド・プロンプトで、ORADIM を使用して Oracle サービスを削除します。次の表に、各 Oracle リリースに対して実行するコマンドを示します。

Oracle リリース	コマンド・プロンプトで入力するコマンド
7.3	C:¥> ORADIM73 -DELETE -SID SID
8.0	C:¥> ORADIM80 -DELETE -SID SID
8.1 以上	C:¥> ORADIM -DELETE -SID SID

たとえば、Oracle リリースがリリース 8.0.6 で *SID* が ORCL の場合、次のコマンドを入力します。

```
C:¥> ORADIM80 -DELETE -SID ORCL
```

Oracle リリースがリリース 8.1.7 で *SID* が ORCL の場合、次のコマンドを入力します。

```
C:¥> ORADIM -DELETE -SID ORCL
```

- c. コマンド・プロンプトで、Oracle9i の新しいリリースの ORADIM コマンドを使用して新しい Oracle9i データベース・サービスを作成します。

```
C:¥> ORADIM -NEW -SID SID -INTPWD PASSWORD -MAXUSERS USERS  
-STARTMODE AUTO -PFILE ORACLE_HOME¥DATABASE¥INITSID.ORA
```

この構文には次の変数が含まれます。

<i>SID</i>	アップグレードするデータベースの <i>SID</i> と同じ <i>SID</i> 名。
<i>PASSWORD</i>	新しい Oracle9i リリース 2 (9.2) のデータベース・インスタンスのパスワード。SYSDBA 権限で接続したユーザー用のパスワードです。-INTPWD オプションは必須ではありません。これを指定しない場合、オペレーティング・システムの認証が使用され、パスワードは不要です。
<i>USERS</i>	SYSDBA 権限および SYSOPER 権限を付与できるユーザーの最大数。
<i>ORACLE_HOME</i>	Oracle9i リリース 2 (9.2) の Oracle ホーム・ディレクトリ。 -PFILE オプションを使用して、(Oracle ホーム・ディレクトリのドライブを含む) フルパス名を指定する必要があります。

たとえば、*SID* が ORCL、*PASSWORD* が TWxy579、*USERS* の最大数が 10、*ORACLE_HOME* ディレクトリが C:¥ORA92 の場合、次のコマンドを入力します。

```
C:¥> ORADIM -NEW -SID ORCL -INTPWD TWxy579 -MAXUSERS 10  
-STARTMODE AUTO -PFILE C:¥ORA92¥DATABASE¥INITORCL.ORA
```

- 以前の Oracle ホームから新しい Oracle9i の Oracle ホームへ構成ファイルをコピーします。
 - UNIX プラットフォーム上でリリース 7.3.4 からアップグレードする場合は、変換ファイルを Oracle7 の Oracle ホーム・ディレクトリから新しい Oracle9i の Oracle ホーム・ディレクトリへ移動またはコピーします。ほとんどの UNIX プラットフォームでは、変換ファイル *convsid.dbf* (*sid* は Oracle9i データベース名) は、Oracle7 および Oracle9i 環境ともに \$*ORACLE_HOME*/dbs に存在する必要があります。

Windows オペレーティング・システムでは、変換ファイル `convert.ora` は、Oracle9i 環境の `%ORACLE_HOME%\rdbms` に存在する必要があります。ただし、移行ユーティリティによってこのディレクトリの中に自動的に配置されるので、移動する必要はありません。

- b. パラメータ・ファイルが古い環境の Oracle ホームに存在する場合は、新しい Oracle ホームへコピーします。デフォルトでは、Oracle はパラメータ・ファイルを UNIX プラットフォームの場合は `$ORACLE_HOME/dbs` で、Windows オペレーティング・システムの場合は `%ORACLE_HOME%\database` で検索します。パラメータ・ファイルは任意の場所に格納できます。ただし、Oracle9i の新しいリリースへアップグレードした後は、古い環境の Oracle ホームには格納できません。
- c. パラメータ・ファイルが IFILE (インクルード・ファイル) エントリを含み、IFILE エントリ内に指定されたファイルが古い環境の Oracle ホームに存在する場合、IFILE エントリで指定されているファイルを新しい Oracle ホームへコピーします。IFILE エントリ内に指定されたファイルには、追加の初期化パラメータがあります。
- d. 古い環境の Oracle ホームに存在するパスワード・ファイルがある場合は、そのパスワード・ファイルを Oracle9i の Oracle ホームに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、オペレーティング・システムによって異なります。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。Windows オペレーティング・システムでは、デフォルトのパスワード・ファイルは `%ORACLE_HOME%\database\pwdsid.ora` です。両方の場合とも、`sid` は Oracle インスタンスの ID です。
- e. クラスタ・データベースをアップグレードしている場合で、`initdb_name.ora` ファイルが古い環境の Oracle ホームに存在する場合、`initdb_name.ora` ファイルを新しい Oracle ホームに移動またはコピーします。

注意： クラスタ・データベースをアップグレードしている場合は、このクラスタ・データベースがインスタンスを構成しているすべてのノードでこの手順を実行してください。

4. 次の手順を実行して、Oracle9i の新しいリリースで使用するパラメータ・ファイルを調整します。
 - a. 廃止された初期化パラメータを削除して、非推奨の初期化パラメータを調整します。特定の初期化パラメータは、Oracle9i の新しいリリースでは廃止されています。廃止されたパラメータを、Oracle9i リリース 2 (9.2) インスタンスを起動するパラメータ・ファイルからすべて削除します。廃止されたパラメータは、Oracle9i の新しいリリースではエラーの原因となります。また、Oracle9i の新しいリリース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。非推奨の初期化パラメータまたは廃止された初期化パラメータのリストは、[付録 A「初期化パラメータおよびデータ・ディクショナリの変更」](#)を参照してください。

また、クラスタ・データベースをアップグレードしている場合、廃止されたクラスタ・データベースの初期化パラメータの詳細は、『Oracle9i Real Application Clusters セットアップおよび構成』を参照してください。

- b. SHARED_POOL_SIZE 初期化パラメータが、48MB 以上に設定されていることを確認します。
- c. PGA_AGGREGATE_TARGET 初期化パラメータが、24MB 以上に設定されていることを確認します。
- d. LARGE_POOL_SIZE 初期化パラメータが、8MB 以上に設定されていることを確認します。
- e. COMPATIBLE 初期化パラメータが、Oracle9i の新しいリリース用に正しく設定されていることを確認します。COMPATIBLE が 8.1.0 未満に設定されている場合、後述の手順 11 に従ってリリース 2 (9.2) データベースを起動すると、次のエラーが発生します。

ORA-00401: パラメータ compatible の値はこのリリースではサポートされません。

アップグレードするデータベースのリリースに応じて、次の手順で COMPATIBLE を設定します。

リリース	COMPATIBLE に設定
7.3.4 8.0.6	パラメータ・ファイルから COMPATIBLE を削除するか、または COMPATIBLE に 8.1.0 を設定します。
8.1.7	COMPATIBLE が 8.0.x に設定されている場合は、パラメータ・ファイルから COMPATIBLE を削除するか、または COMPATIBLE に 8.1.0 を設定します。 COMPATIBLE が 8.1.x に設定されている場合は、設定を変更する必要はありません。
9.0.1	1 つ以上の自動セグメント領域管理表領域がデータベースに存在する場合、COMPATIBLE に 9.0.1.3 を設定します。その他の場合は、設定を変更する必要はありません。

- f. DB_DOMAIN 初期化パラメータが正しく設定されていることを確認します。

参照： この初期化パラメータの設定方法の詳細は、A-8 ページの「DB_DOMAIN パラメータ」を参照してください。

- g. Windows オペレーティング・システムで、RDBMS80 またはその他の環境変数を指す BACKGROUND_DUMP_DEST および USER_DUMP_DEST 初期化パラメータを、次のディレクトリを指すように変更します。

初期化パラメータ	設定
BACKGROUND_DUMP_DEST	%ORACLE_BASE%\%oradata%\DB_NAME
USER_DUMP_DEST	%ORACLE_BASE%\%oradata%\DB_NAME%\archive

この設定では、ORACLE_BASE は Oracle ベースのフルパスに、DB_NAME はデータベース名に置き換えます。

- h. パラメータ・ファイルのすべてのパス名が完全に指定されていることを確認します。パラメータ・ファイルには相対パス名を使用しないでください。
- i. パラメータ・ファイルに IFILE エントリがある場合、パラメータ・ファイルの IFILE エントリを変更して、手順 3 の c で指定したインクルード・ファイルの新しい場所を指定するようにします。次に、手順 a ～ h でパラメータ・ファイルを編集したときと同じ方法で、IFILE エントリに指定されているファイルを編集します。
- j. クラスタ・データベースをアップグレードしている場合は、パラメータ・ファイルを変更したときと同じ方法で initdb_name.ora ファイルを変更します。

これらの調整後、変更したすべてのファイルを保存します。

注意： クラスタ・データベースをアップグレードしている場合は、このクラスタ・データベースがインスタンスを構成しているすべてのノードでこの手順を実行してください。

- 5. アップグレード中に、次の初期化パラメータを調整します。アップグレードが完了した後、これらのパラメータをアップグレード前の設定に戻す必要があります。
 - a. 廃止された初期化パラメータを削除して、非推奨の初期化パラメータを調整します。
 - b. パスワード・ファイルを使用している場合は、パラメータ・ファイルの REMOTE_LOGIN_PASSWORDFILE を NONE に設定します。
 - c. クラスタ・データベースをアップグレードしている場合は、CLUSTER_DATABASE 初期化パラメータを false に設定します。アップグレードが完了した後、この初期化パラメータの設定を true に戻す必要があります。
 - d. NLS_LENGTH_SEMANTICS 初期化パラメータが CHAR に設定されている場合は、BYTE に設定します。アップグレードが完了した後、この初期化パラメータの設定を CHAR に戻す必要があります。

6. ご使用のオペレーティング・システムが UNIX の場合、次の環境変数が、新しい Oracle9i リリース 2 (9.2) のディレクトリを指定していることを確認します。

- ORACLE_HOME
- PATH
- ORA_NLS33
- LD_LIBRARY_PATH

リリース 7.3.4 からアップグレードしているときに ORACLE_HOME が Oracle7 の実行可能ファイルを指している場合、アップグレード処理の後半で ALTER DATABASE CONVERT 文を発行すると、次のエラーが表示されます。

ORA-00223: 変換ファイルが無効またはバージョンが正しくありません。

注意： クラスタ・データベースをアップグレードしている場合は、このクラスタ・データベースがインスタンスを構成しているすべてのノードでこの手順を実行してください。

参照： ご使用のオペレーティング・システムでのその他の重要な環境変数の設定は、オペレーティング・システム固有の Oracle9i インストレーション・ガイドを参照してください。

7. 新しいリリースの Oracle ホーム・ディレクトリの所有者としてシステムにログインします。
8. システム・プロンプトで、ORACLE_HOME/rdbms/admin ディレクトリへ移動します。
9. SQL*Plus を起動します。
10. SYSDBA 権限を持つユーザーとして、データベース・インスタンスに接続します。
11. アップグレードしているデータベースがリリース 8.0.6 以上の場合、次のコマンドを実行してデータベースを起動します。

```
SQL> STARTUP MIGRATE
```

PFILE オプションを使用して、初期化パラメータ・ファイルの場所を指定する必要がある場合があります。

廃止された初期化パラメータをリストするエラー・メッセージが表示される場合があります。その場合、廃止された初期化パラメータを記録し、通常のアップグレードを継続します。次にデータベースを停止するときに、廃止された初期化パラメータを削除します。

12. アップグレードしているデータベースがリリース 7.3.4 の場合、次の手順を実行してデータベースを起動します。
- a. 新しい Oracle9i データベースをマウントせずに、Oracle9i データベースのインスタンスを起動します。

```
SQL> STARTUP RESTRICT NOMOUNT
```

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

PFILE オプションを使用して、初期化パラメータ・ファイルの場所を指定する必要がある場合があります。

廃止された初期化パラメータをリストするエラー・メッセージが表示される場合があります。その場合、廃止された初期化パラメータを記録し、通常のアップグレードを継続します。次にデータベースを停止するときに、廃止された初期化パラメータを削除します。

- b. Oracle9i 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 Oracle9i 形式に変換します。そのためには、次のコマンドを発行します。

```
SQL> ALTER DATABASE CONVERT;
```

この文の実行が成功すると、このデータベースを Oracle7 に戻せなくなります。ただし、必要に応じてバックアップから Oracle7 データベースをリストアすることはできます。

この手順でエラーが発生した場合は、エラーの原因となっている状態を解決し、移行ユーティリティを再実行してください。そうでない場合は、移行ユーティリティを実行した後に実行したバックアップをリストアします。

参照： この手順で発生するエラーと問題解決のために実行する処置の詳細は、D-27 ページの「[ALTER DATABASE CONVERT 文における問題](#)」を参照してください。

- c. 次の文で Oracle9i データベースをオープンします。

```
SQL> ALTER DATABASE OPEN RESETLOGS MIGRATE;
```

Oracle9i データベースがオープンされると、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 Oracle9i 形式に変換されます。

この文を発行したときエラーが発生した場合は、始めから移行処理をやりなおし、移行ユーティリティ実行後にデータベースを Oracle7 環境でオープンしていないことを確認してください。この章の最初から開始します。ただし、第 2 章で説明した移行前の手順を完了していることを確認してください。

13. 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。

```
SQL> SPOOL upgrade.log
```

14. uold_release.sql を実行します。ここで、old_release はアップグレード前にインストールしたリリースを指します。表 3-2 を参照して、適切なスクリプトを選択してください。各スクリプトで、Old Release 列に指定されたリリースからの直接のアップグレードが行えます。Old Release はアップグレード元のリリースです。

スクリプトを実行するには、次のように入力します。

```
SQL> @uold_release.sql
```

表 3-2 アップグレード・スクリプト

リリース	実行スクリプト
7.3.4	u0703040.sql
8.0.6	u0800060.sql
8.1.7	u0801070.sql
9.0.1	u0900010.sql

参照： アップグレードの前にインストールしたリリースが表 3-2 に示されていない場合は、2-2 ページの「新しいリリースへのアップグレード・パスの決定」を参照してください。

スクリプトを実行する場合、次のガイドラインに従ってください。

- 新しい Oracle9i リリース 2 (9.2) のインストールで提供されるスクリプトを使用する必要があります。
- 新しい Oracle9i リリース 2 (9.2) の環境でスクリプトを実行する必要があります。
- アップグレードが複数のリリースにわたる場合でも、1 つのスクリプトのみを実行する必要があります。たとえば、古いリリースが 8.1.7 であった場合、u0801070.sql のみを実行します。

実行するスクリプトは、一定のディクショナリ表を作成および変更します。また、新しいリリース 2 (9.2) に付属する `catalog.sql` および `catproc.sql` スクリプトを実行します。これらのスクリプトは、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します。

次のコンポーネントは、`uold_release.sql` スクリプトを実行するとアップグレードされます。

- Oracle9i Catalog Views
- Oracle9i Packages and Types

15. コンポーネント・レジストリの内容を表示して、アップグレードする必要があるコンポーネントを確認します。

```
SQL> SELECT comp_name, version, status
        FROM dba_registry;
```

次に、問合せを発行したときに表示される出力例を示します。

COMP_NAME	VERSION	STATUS
-----	-----	-----
Oracle9i Catalog Views	9.2.0.1.0	VALID
Oracle9i Packages and Types	9.2.0.1.0	VALID
JServer JAVA Virtual Machine	9.0.1	LOADED
Java Packages	9.0.1	LOADED
Oracle XDK for Java	9.0.1	LOADED
Oracle Text	9.0.1	LOADED
Oracle Workspace Manager	9.0.1.0.0	LOADED
Oracle interMedia	9.0.0.0.0	LOADED
Oracle Spatial	9.0.0.0.0 BETA	LOADED
Ultrasearch	9.0.1.0.0	LOADED
OLAP Catalog	9.0.1.0.0	LOADED

11 rows selected.

16. `cmpdbmig.sql` スクリプトを実行して、SYSDBA 権限での接続中にアップグレードできるコンポーネントをアップグレードします。

```
SQL> @cmpdbmig.sql
```

次のコンポーネントは、`cmpdbmig.sql` スクリプトを実行するとアップグレードされます。

- Oracle JVM
- Oracle9i Java Packages
- Oracle XDK for Java
- Messaging Gateway

- Oracle9i Real Application Clusters
- Oracle Workspace Manager
- Oracle Data Mining
- OLAP Catalog
- OLAP Analytic Workspace
- Oracle Label Security

17. コンポーネント・レジストリの内容を表示して、アップグレードされたコンポーネントを確認します。

```
SQL> SELECT comp_name, version, status
        FROM dba_registry;
```

次に、問合せを発行したときに表示される出力例を示します。

COMP_NAME	VERSION	STATUS
-----	-----	-----
Oracle9i Catalog Views	9.2.0.1.0	VALID
Oracle9i Packages and Types	9.2.0.1.0	VALID
JServer JAVA Virtual Machine	9.2.0.1.0	VALID
Oracle9i Java Packages	9.2.0.1.0	VALID
Oracle XDK for Java	9.2.0.2.0	UPGRADED
Oracle Text	9.0.1	LOADED
Oracle Workspace Manager	9.2.0.1.0	VALID
Oracle interMedia	9.0.0.0.0	LOADED
Oracle Spatial	9.0.0.0.0 BETA	LOADED
Ultrasearch	9.0.1.0.0	LOADED
OLAP Catalog	9.2.0.1.0	VALID
OLAP Analytic Workspace	9.2.0.1.0	LOADED

12 rows selected.

18. 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。

```
SQL> SPOOL OFF
```

次に、スプール・ファイルを確認して、すべてのパッケージおよび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ます。手順 13 で、スプール・ファイル名を upgrade.log としました。このファイルで見つけた問題进行处理し、必要に応じて適切なアップグレード・スクリプトを再実行します。必要に応じて、この章で説明されているスクリプトのいずれかを実行できます。

19. インスタンスを停止して再起動し、通常の操作用にシステム・パラメータを再初期化します。再起動によって、Oracle JVM およびその他のコンポーネント用のリリース 2 (9.2) の初期化も実行されます。

```
SQL> SHUTDOWN IMMEDIATE
```

この正常な停止によって、すべてのキャッシュがフラッシュされ、バッファが消去されて、その他の内部的な不整合が解消されます。これらの手段は、新規にアップグレードした Oracle9i データベースの整合性との一貫性を保証するための重要な最終ステップです。

また、手順 11 でデータベースを起動したときに、廃止された初期化パラメータをリストしたメッセージが表示された場合、ここで初期化パラメータ・ファイルから廃止された初期化パラメータを削除します。

20. 以前のデータベースに存在した残りのすべてのコンポーネントをアップグレードします。3-26 ページの「[特定のコンポーネントのアップグレード](#)」を参照してください。

次のコンポーネントには、個別のアップグレード手順が必要です。

- Oracle Text
- Oracle Ultra Search
- Oracle Spatial
- Oracle *interMedia*
- Oracle Visual Information Retrieval

21. utlrlp.sql を実行して、残りのすべての PL/SQL モジュールおよび Java コードを再コンパイルします。

```
SQL> @utlrlp.sql
```

予測されるすべてのパッケージおよびクラスが有効であることを確認します。

```
SQL> SELECT count(*) FROM dba_objects WHERE status='INVALID';
SQL> SELECT distinct object_name FROM dba_objects WHERE status='INVALID';
```

すべてのコンポーネントが有効で、リリース 2 (9.2) にアップグレードされていることを確認します。

```
SQL> SELECT comp_name, version, status
        FROM dba_registry;
```

これで、ご使用のデータベースが新しい Oracle9i リリース 2 (9.2) へアップグレードされました。第 4 章「[データベースのアップグレード後の作業](#)」に示す手順を実行します。

注意： 古い Oracle ソフトウェアを保持している場合は、古いソフトウェアを使用してアップグレードしたデータベースを起動しないでください。新しい Oracle9i リリース 2 (9.2) のインストール・ディレクトリにある実行可能ファイルでのみ、データベースを起動してください。また、古い Oracle 環境を削除する前に、Oracle 環境のデータ・ファイルをすべて新しい Oracle9i 環境に再配置してください。データ・ファイルの再配置の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

特定のコンポーネントのアップグレード

Oracle データベース・サーバーのいくつかのコンポーネントには、データベースの通常のアップグレードとは異なるアップグレードが必要です。次の項に示す処理を実行して、自動的にアップグレードされなかったコンポーネントをアップグレードします。

注意： この章の前半の指示に従ってデータベースをアップグレードした後、次に示す処理を実行します。

Oracle Spatial のアップグレード

Oracle システムに Oracle Spatial がインストールされている場合、Oracle Spatial をリリース 2 (9.2) へアップグレードする方法については、『Oracle Spatial ユーザーズ・ガイドおよびリファレンス』を参照してください。

Oracle *interMedia* のアップグレード

Oracle *interMedia* のアップグレード方法については、UNIX プラットフォームでは `$ORACLE_HOME/ord/im/admin/README.txt` を、Windows プラットフォームでは `%ORACLE_HOME%\ord\im\admin\README.txt` を参照してください。

Oracle Visual Information Retrieval のアップグレード

Oracle Visual Information Retrieval のアップグレード方法については、UNIX プラットフォームでは `$ORACLE_HOME/ord/vir/admin/README.txt` を、Windows プラットフォームでは `%ORACLE_HOME%\ord\vir\admin\README.txt` を参照してください。

Oracle Text のアップグレード

Oracle システムに Oracle Text がインストールされている場合は、次の手順を実行します。

1. 新しいリリースの Oracle ホーム・ディレクトリの所有者としてシステムにログインします。
2. システム・プロンプトで、`ORACLE_HOME/ctx/admin` ディレクトリへ移動します。

3. SQL*Plus を起動します。
4. SYSDBA 権限を持つユーザーとして、データベース・インスタンスに接続します。
5. インスタンスが実行されている場合は、SHUTDOWN IMMEDIATE を使用してインスタンスを停止します。

```
SQL> SHUTDOWN IMMEDIATE
```

6. RESTRICT モードでインスタンスを起動します。

```
SQL> STARTUP RESTRICT
```

PFILE オプションを使用して、初期化パラメータ・ファイルの場所を指定する必要がある場合があります。

7. 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。

```
SQL> SPOOL text_upgrade.log
```

8. リリース 8.1.7 からアップグレードする場合は、次の手順を実行します。リリース 1 (9.0.1) からアップグレードする場合は、手順 9 へ進みます。

- a. s0900010.sql を実行します。

```
SQL> @s0900010.sql
```

このスクリプトは、データベースに必要な新しい権限をユーザー CTXSYS に付与します。

- b. ユーザー CTXSYS としてデータベース・インスタンスに接続します。

- c. u0900010.sql を実行します。

```
SQL> @u0900010.sql
```

- d. SYSDBA 権限を持つユーザーとして、データベース・インスタンスに接続します。

9. リリース 8.1.7 またはリリース 1 (9.0.1) からアップグレードする場合は、次の手順を実行します。

- a. s0902000.sql を実行します。

```
SQL> @s0902000.sql
```

このスクリプトは、データベースに必要な新しい権限をユーザー CTXSYS に付与します。

- b. ユーザー CTXSYS としてデータベース・インスタンスに接続します。

- c. u0902000.sql を実行します。

```
SQL> @u0902000.sql
```

このスクリプトは、CTXSYS スキーマをリリース 2 (9.2) にアップグレードします。

- d. SYSDBA 権限を持つユーザーとして、データベース・インスタンスに接続します。

10. 無効になった CTXSYS オブジェクトを調べて、必要に応じて再コンパイルします。

11. 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。

```
SQL> SPOOL OFF
```

次に、スプール・ファイルを確認して、すべてのパッケージおよび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ます。手順 7 で、スプール・ファイル名を text_upgrade.log としました。このファイルで見つけた問題を処理し、必要に応じて適切なアップグレード・スクリプトを再実行します。

12. インスタンスを停止します。

```
SQL> SHUTDOWN IMMEDIATE
```

13. SQL*Plus を終了します。

これで、Oracle Text は新しいリリースへアップグレードされました。

Oracle Ultra Search のアップグレード

Oracle システムに Oracle Ultra Search がインストールされている場合、Oracle Ultra Search をリリース 2 (9.2) へアップグレードする方法については、Oracle Ultra Search のオンライン・ドキュメントを参照してください。

手動でのアップグレードのトラブルシューティング

通常、次の 3 つのリソースは、新しい Oracle リリース用に増やす必要があります。

- SYSTEM 表領域
- 共有メモリー
- ロールバック・セグメント

アップグレード実行中にこれらのリソースの 1 つが不足した場合、リソースの割当てを増やし、適切なアップグレード・スクリプトを再実行します。

SYSTEM 表領域

通常、SYSTEM 表領域のサイズが不十分な場合、アップグレード中に次のいずれかのメッセージが表示されます。

ORA-01650: ロールバック・セグメント *string* を拡張できません (*string* 分、表領域 *string*)。
ORA-01651: ロールバック・セグメントを拡張できません (*string* 分、表領域 *string*)。
ORA-01652: 一時セグメントを拡張できません (*string* 分、表領域 *string*)。
ORA-01653: 表 *string.string* を拡張できません (*string* 分、表領域 *string*)。
ORA-01654: 索引 *string.string* を拡張できません (*string* 分、表領域 *string*)。
ORA-01655: クラスタ *string.string* を拡張できません (*string* 分、表領域 *string*)。

共有メモリー

データベースに Oracle JVM がインストールされている場合は、特に大きい共有メモリーのプール・サイズが必要です。増やす必要がある共有メモリーの初期化パラメータが、エラー・メッセージに示されます。

ORA-04031: 共有メモリーの *string* バイトを割当てできません
(*"string"*, *"string"*, *"string"*, *"string"*)

共有メモリーの初期化パラメータの詳細は、『Oracle9i データベース・リファレンス』を参照してください。

パブリック・ロールバック・セグメント

ロールバック・セグメントを使用している場合、アップグレード・スクリプトの実行中は、1 つの大きい (70MB の) パブリック・ロールバック・セグメントをオンラインにする必要があります。小さいパブリック・ロールバック・セグメントは、アップグレード中はオフラインにする必要があります。通常、ロールバック・セグメントのサイズが不十分な場合は、次のエラーが表示されます。

ORA-01562: ロールバック・セグメント番号 *string* の拡張に失敗しました

データベースのアップグレード後の作業

この章では、データベースのアップグレード後に行う手順について説明します。この章では、次の項目について説明します。

- データベースのアップグレード後に行う作業
- リリース 8.1.7 以下のデータベースのアップグレード後のみに行う作業
- リリース 8.0.6 以下のデータベースのアップグレード後のみに行う作業
- リリース 7.3.4 のデータベースのアップグレード後のみに行う作業
- データベースのテストと結果の比較
- 現行リリースのワード・サイズの変更

データベースのアップグレード後に行う作業

データベースをアップグレードした後、次の作業を行います。

データベースのバックアップ

必ず本番データベースの完全なバックアップを作成してください。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 スクリプトなどをすべて含む完全なバックアップである必要があります。ホット・バックアップは完全なリカバリ能力を提供できないため、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。後続の手順で、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、このバックアップを使用して、この時点まで戻すことができます。

参照： データベースのバックアップの詳細は、『Oracle9i ユーザー管理 バックアップおよびリカバリ・ガイド』を参照してください。

オラクル社が提供するアカウント用のパスワードの変更

アップグレード元のリリースによっては、新しいアカウントが提供されている場合があります。オラクル社が提供するすべてのアカウント（SYS および SYSTEM を除く）をロックして、パスワードを期限切れにし、アカウントのロックを解除する際には、新しいパスワードを指定することをお勧めします。

次の SQL 文を発行して、すべてのアカウントの状態を確認できます。

```
SQL> SELECT username, account_status  
       FROM dba_users  
       ORDER BY username;
```

次の SQL 文を発行して、パスワードをロックまたは期限切れにします。

```
SQL> ALTER USER username PASSWORD EXPIRE ACCOUNT LOCK;
```

Standard Edition から Enterprise Edition へのアップグレード

Oracle の Standard Edition を Enterprise Edition にアップグレードする場合は、次の手順を実行します。

1. Standard Edition サーバー・ソフトウェアと Enterprise Edition サーバー・ソフトウェアのリリース番号が同じかどうかを確認します。

たとえば、Standard Edition サーバー・ソフトウェアがリリース 9.2.0.1.0 の場合、Enterprise Edition サーバー・ソフトウェアもリリース 9.2.0.1.0 にアップグレードする必要があります。

2. データベースを停止します。
3. オペレーティング・システムが Windows の場合、`OracleServiceSID` (`SID` はインスタンス名) を含むすべての Oracle サービスを停止します。
4. Standard Edition サーバー・ソフトウェアを削除します。
5. Oracle Universal Installer を使用して、Enterprise Edition サーバー・ソフトウェアをインストールします。

削除された Standard Edition で使用されていたのと同じ Oracle ホームを選択します。インストール中は、必ず Enterprise Edition を選択します。プロンプトが表示されたら、「Database Configuration」画面から「Software Only」を選択します。
6. データベースを起動します。

これで、ご使用のデータベースが Enterprise Edition へアップグレードされました。

Oracle Managed Files (OMF) の移行

リリース 9.0.1.2.0 より前のリリースの Oracle9i からアップグレードする場合、Oracle Managed Files (OMF) を移行する必要があります。リリース 9.0.1.2.0 より前のリリースの Oracle9i では、非 OMF ファイルが誤って OMF とみなされる場合があります。これによって、データベースにデータ・ファイル、制御ファイルまたはログ・ファイルを追加する場合に、次のエラーが発生します (`string` には文字列が入ります)。

ORA-01276: ファイル `string` は追加できません。ファイルに Oracle Managed Files のファイル名があります。

また、表領域または REDO ログを削除すると、表領域または REDO ログに対応付けられたオペレーティング・システム・ファイルが誤って削除されることもあります。

リリース 9.0.1.2.0 以上では、Windows および UNIX オペレーティング・システム上の Oracle Managed Files のファイル名の書式が変更されています。以前のリリースの Oracle9i で作成された OMF ファイルは、OMF ファイル名の新しい書式に準拠するように名前が変更されないかぎり、OMF ファイルとして認識されません。

以前のリリースの Oracle9i では、ベース・ファイル名が次の書式に準拠しているファイルは OMF とみなされていました。

- 接頭辞 `ora_`
- 拡張子 `.dbf`、`.tmp`、`.log` または `.ctl`

リリース 9.0.1.2.0 以上では、ベース・ファイル名が次の書式に準拠しているファイルは OMF とみなされます。

- 接頭辞 `ol_mf_`
- 拡張子 `.dbf`、`.tmp`、`.log` または `.ctl`
- 拡張子の直前のアンダースコア (`_`)

古い OMF データ・ファイル、一時ファイルおよびログ・ファイルは、ファイル・システムおよび制御ファイル内で名前を変更することによって移行できます。次の手順を実行します。

1. 次の SQL 文を発行して、OMF ファイルを検索します。

```
SQL> SELECT name FROM v$datafile;
SQL> SELECT name FROM v$tempfile;
SQL> SELECT member FROM v$logfile;
```

2. インスタンスを停止します。

```
SQL> SHUTDOWN IMMEDIATE
```

3. ファイル・システムのファイル名を変更します。

- `ora_` を `ol_mf_` に変更します。
- 拡張子の前に `_` を追加します。

たとえば、`ora_tbs1_2ixfh90q.dbf` という名前のファイルの場合、新しい名前は `ol_mf_tbs1_2ixfh90q_.dbf` になります。

4. データベースをマウントします。
5. 制御ファイルのファイル名を変更します。次に例を示します。

```
SQL> ALTER DATABASE RENAME FILE 'old_filename' TO 'new_omf_filename';
```

6. データベースをオープンします。

OMF 制御ファイルは、ファイル・システムおよび `CONTROL_FILES` 初期化パラメータで名前を変更することによって移行できます。次の手順を実行します。

1. `CONTROL_FILES` 初期化パラメータを確認して、OMF ファイルを検索します。
2. インスタンスを停止します。

```
SQL> SHUTDOWN IMMEDIATE
```


3. ファイル・システムのファイル名を変更します。
 - ora_ を ol_mf_ に変更します。
 - 拡張子の前に _ を追加します。

たとえば、ora_cmr7t90p.ct1 という名前のファイルの場合、新しい名前は ol_mf_cmr7t90p_.ct1 になります。
4. 新しい名前を参照するように CONTROL_FILES 初期化パラメータを変更します。
5. データベースをマウントし、オープンします。

Oracle OLAP のアップグレード

この項では、Oracle OLAP をアップグレードする方法について説明します。

リリース 8.1.7 以上のリリースからのアップグレード

Oracle OLAP では、SQL を介して分析作業領域へアクセスできます。COMPATIBLE 初期化パラメータが 8.1.6 以上に設定されている場合、標準のアップグレード手順でこの機能を使用できます。追加手順は必要ありません。

Oracle OLAP では、OLAP API (Java インタフェース) および OLAP Catalog Metadata も提供されます。COMPATIBLE が 8.1.6 以上の場合にこれらの機能を組み込むには、次の手順を実行します。

1. 標準のアップグレード手順を実行します。
2. COMPATIBLE を 9.2.0 に設定します。
3. データベースを再起動します。
4. 次のスクリプトを実行します。


```
ORACLE_HOME/olap/admin/olapapi.sql
```
5. リリース 1 (9.0.1) より前のリリースからアップグレードする場合は、次の追加手順を実行して、固有の表領域に OLAP Catalog Metadata を作成します。
 - a. 次のような文を実行して、表領域を作成します。任意の有効な表領域名およびデータベース・ファイル名を指定できます。

```
CREATE TABLESPACE OLAPCAT LOGGING
  DATAFILE 'ORACLE_HOME/rdbms/dbs/olap01.dbf'
  SIZE 20M REUSE AUTOEXTEND ON NEXT 640K
  MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL
  SEGMENT SPACE MANAGEMENT AUTO;
```

- b. 作成した表領域の名前およびデータベースの一時表領域を指定して、次のスクリプトを実行します。この例の場合、名前は OLAPCAT および TEMP です。

```
ORACLE_HOME/cwmlite/admin/oneinst1.sql OLAPCAT TEMP
```

リリース 8.0.6 より前のリリースからのアップグレード

OLAP API および OLAP Catalog Metadata を使用せずに、SQL を介して分析作業領域にアクセスのみする場合、次の手順を実行します。

1. 標準のアップグレード手順を実行します。
2. COMPATIBLE を 8.1.6 以上に設定します。
3. データベースを再起動します。
4. 次のスクリプトを実行します。

```
ORACLE_HOME/olap/admin/olapaw.sql
```

SQL を介した分析作業領域へのアクセスに加えて、OLAP API および OLAP Catalog Metadata のサポートが必要な場合は、次の手順を実行します。

1. 標準のアップグレード手順を実行します。
2. COMPATIBLE を 9.2.0 に設定します。
3. データベースを再起動します。
4. 次のスクリプトを実行します。

```
ORACLE_HOME/olap/admin/olap.sql
```

5. 次のような文を実行して、表領域を作成します。任意の有効な表領域名およびデータベース・ファイル名を指定できます。

```
CREATE TABLESPACE OLAPCAT LOGGING
  DATAFILE 'ORACLE_HOME/rdbms/dbs/olap01.dbf'
  SIZE 20M REUSE AUTOEXTEND ON NEXT 640K
  MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL
  SEGMENT SPACE MANAGEMENT AUTO;
```

6. 作成した表領域の名前およびデータベースの一時表領域を指定して、次のスクリプトを実行します。この例の場合、名前は OLAPCAT および TEMP です。

```
ORACLE_HOME/cwmlite/admin/oneinst1.sql OLAPCAT TEMP
```

サーバー・パラメータ・ファイルへの初期化パラメータ・ファイルの移行

初期化パラメータ・ファイルを使用している場合は、次の手順でサーバー・パラメータ・ファイルへ移行します。

1. 初期化パラメータ・ファイルがクライアント・マシンにある場合は、クライアント・マシンからサーバー・マシンへ転送します。

注意： Oracle9i Real Application Clusters を使用している場合は、インスタンス固有のすべての初期化パラメータ・ファイルを、単一の初期化パラメータ・ファイルに結合する必要があります。インスタンス固有の初期化パラメータ・ファイルの結合方法およびクラスター・データベース用のサーバー・パラメータ・ファイルの使用方法については、次のマニュアルを参照してください。

- 『Oracle9i Real Application Clusters セットアップおよび構成』
 - 『Oracle9i Real Application Clusters 管理』
-

2. CREATE SPFILE 文を使用して、サーバー・パラメータ・ファイルを作成します。この文は、初期化パラメータ・ファイルを読み込み、サーバー・パラメータ・ファイルを作成します。CREATE SPFILE 文を発行するために、データベースを起動する必要はありません。
3. 新しく作成されたサーバー・パラメータ・ファイルを使用して、インスタンスを起動します。

参照：

- サーバー・パラメータ・ファイルの作成の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。
- CREATE SPFILE 文の詳細は、『Oracle9i SQL リファレンス』を参照してください。

LONG から LOB への表の移行

LOB データ型 (BFILE、BLOB、CLOB および NCLOB) には、LONG データ型以上のメリットがあります。LONG データ型と LOB データ型の違いの詳細は、『Oracle9i データベース概要』を参照してください。

Oracle9i では、ALTER TABLE 文を使用して、LONG 列のデータ型を CLOB に、LONG RAW 列のデータ型を BLOB に変更できます。

次の例では、long_tab 表の long_col という LONG 列が、CLOB データ型に変更されます。

```
SQL> ALTER TABLE Long_tab MODIFY ( long_col CLOB );
```

LONG 列を LOB へ変換するメソッドを使用した後も、表に対するすべての制約およびトリガーは使用可能です。ただし、表のすべての列で、ドメイン索引およびファンクション索引を含むすべての索引が使用不可となるため、ALTER INDEX ... REBUILD 文を使用してすべての索引を再構築する必要があります。また、LONG 列上のドメイン索引は、LONG 列を LOB に変更する前に削除する必要があります。

参照： LOB データを使用するためのアプリケーションの変更の詳細は、『Oracle9i アプリケーション開発者ガイド - ラージ・オブジェクト』を参照してください。

listener.ora ファイルの変更

次のいずれかに該当する場合のみ、listener.ora ファイルを変更する必要があります。

- Database Upgrade Assistant を使用しないで、データベースをアップグレードした場合。
- データベースのアップグレードに Database Upgrade Assistant を使用したが、listener.ora ファイルが自動的に更新されるように選択しなかった場合。

これらのいずれにも該当しない場合は、この項は省略してください。これらのいずれかに該当する場合は、listener.ora ファイルを変更する必要があります。

参照： listener.ora ファイルの変更の詳細は、B-11 ページの「[listener.ora](#)」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

この項では、1 つ以上のスタンバイ・データベースを含む構成に対して、現行リリースの Oracle から Oracle9i の新しいリリースへアップグレードする方法について示します。

アップグレードの準備

複数のスタンバイ・データベースが存在する場合、アップグレードする各スタンバイ・データベースに対して、この項の手順を繰り返します。

1. ロギングなしの操作が実行されているかどうかを確認します。ロギングなしの操作が実行されている場合、スタンバイを更新する必要はありません。詳細は、『Oracle9i Data Guard 概要および管理』を参照してください。
2. 即時にオフラインになったためにリカバリが必要な表領域またはデータ・ファイルを記録します。アップグレードする前に、表領域またはデータ・ファイルをリカバリして、オンラインまたはオフラインにする必要があります。

本番サイトのアップグレード

本番サイトに Oracle9i の新しいリリースをインストールし、指示に従って本番データベースをアップグレードします。

アップグレードする前に、パラメータ・ファイルに次の追加調整を行います。

- 本番データベースのパラメータ・ファイル内でのリモート・アーカイブがまだ使用可能でない場合は、使用可能にしないでください。リモート・アーカイブが使用可能な場合は、リモートの宛先を **defer** に設定します。
- スタンバイ・データベース上で実行されている管理リカバリを取り消します。
- リリース 8.1.7 以前のリリースからアップグレードする場合で、Oracle Parallel Fail Safe を実行している場合は、**PARALLEL_SERVER** 初期化パラメータをコメントアウトし、本番サイトで **CLUSTER_DATABASE = true** を設定します。

アップグレードする前に、すべてのアーカイブ REDO ログがスタンバイに適用されていることを確認します。

アップグレードが完了した後、ログ・ファイルを切り替え、最後のログに残っているすべての REDO をアーカイブします。

```
SQL> ALTER SYSTEM SWITCH LOGFILE;
```

アップグレードからのアーカイブ・ログを、本番サイトのプライマリ・アーカイブ先から、スタンバイ・ホストのスタンバイ・アーカイブ先へ手動で転送します。

スタンバイ・データベースおよびリスナーを停止します。

スタンバイ・データベースを起動して、マウントします。

スタンバイ・データベースを管理リカバリ・モードにします。SUGGESTION プロンプトで、**AUTO** を入力し、アップグレード処理中に生成されたすべてのアーカイブ・ログを適用します。

スタンバイ・データベースが、スタンバイ・ホストへ転送された最後のログまでリカバリされていることを確認します。本番とスタンバイ間のアーカイブ・ログの差異を解消します。

スタンバイ宛先を **defer** から **enable** へ変更することによって、プライマリ・サイト上のリモート・アーカイブを再度使用可能にします。

スタンバイをリカバリ状態にします。

新機能の適宜追加

『Oracle9i データベース新機能』では、Oracle9i の新しいリリースで使用可能な多くの新機能について説明しています。どの新機能がデータベースおよびアプリケーションに有効かを判断して、これらの機能を使用する計画を立ててください。

新しい Oracle9i 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ることもできます。

第 6 章「[アプリケーションのアップグレード](#)」では、Oracle9i 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、Oracle9i の新機能を実装する前にアプリケーションをテストし、アップグレードしたデータベース上でそれらを正常に動作させる必要があります。

必要な新しい管理手順の作成

Oracle9i の新機能をよく理解したうえで、データベース管理用のスクリプトおよび手順を再確認し、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このデータ整合機能のいくつかを取り除けます。

新しいリリース用のパラメータ・ファイルの調整

Oracle の新しいリリースごとに新しい初期化パラメータが導入され、いくつかの初期化パラメータが使用不可または廃止されます。ご使用のシステムに有効な新しい初期化パラメータを使用するために、これらの変更に対してパラメータ・ファイルを調整する必要があります。

参照：

- Oracle9i リリース 2 (9.2) の新しい初期化パラメータのリストおよび各パラメータの詳細は、『Oracle9i データベース・リファレンス』を参照してください。
- リリース 2 (9.2) で廃止された初期化パラメータおよび使用不可にされた初期化パラメータのリストは、[付録 A「初期化パラメータおよびデータ・ディクショナリの変更」](#)を参照してください。

COMPATIBLE 初期化パラメータは、ご使用のデータベースの互換レベルを制御します。COMPATIBLE 初期化パラメータを、新しいデータベースに必要な互換レベルに基づいて設定します。

参照： 詳細は、5-6 ページの「[COMPATIBLE 初期化パラメータの設定](#)」を参照してください。

リリース 8.1.7 以下のデータベースのアップグレード後のみに行う作業

データベースをリリース 8.1.7 以下のリリースからアップグレードした場合のみ、次の追加作業を行います。リリース 1 (9.0.1) からアップグレードする場合は、これらの作業は必要ありません。

ユーザーの NCHAR 列のアップグレード

Oracle8 および Oracle8i からアップグレードしたデータベースに NCHAR 列を持つユーザー表が含まれている場合は、Oracle9i で使用する前に NCHAR 列をアップグレードする必要があります。

次の手順で、古い書式の NCHAR 列およびキャラクタ・セットを Oracle9i の新しい書式に変換します。また、古い各国語キャラクタ・セットが UTF8 の場合は、Oracle9i でも UTF8 のまま残ります。ただし、以前のリリースの各国語キャラクタ・セットが UTF8 でなかった場合は、AL16UTF16 へ変換されます。

デフォルトでアップグレードに選択された各国語キャラクタ・セットを上書きできます。つまり、Oracle8 の UTF8 各国語キャラクタ・セットを Oracle9i の AL16UTF16 各国語キャラクタ・セットに、また、Oracle8 の UTF8 以外の各国語キャラクタ・セットを Oracle9i の UTF8 各国語キャラクタ・セットに変換できます。

この項の手順を実行するまで、Oracle9i で NCHAR 列を使用すると、次のエラーが発生します。

ORA-12714: 無効な各国語キャラクタ・セットが指定されました。

注意： NCHAR 列をアップグレードすると、すべての NCHAR 列を削除するまで Oracle の以前のリリースへはダウングレードできません。

NCHAR 列を持つユーザー表をアップグレードするには、次の手順を実行します。

1. Oracle ホーム・ディレクトリの所有者としてシステムにログインします。
2. システム・プロンプトで、ORACLE_HOME/rdbms/admin ディレクトリへ移動します。
3. SQL*Plus を起動します。
4. SYSDBA 権限を持つユーザーとして、データベース・インスタンスに接続します。

5. インスタンスが実行されている場合は、SHUTDOWN IMMEDIATE を使用してインスタンスを停止します。

```
SQL> SHUTDOWN IMMEDIATE
```

6. RESTRICT モードでインスタンスを起動します。

```
SQL> STARTUP RESTRICT
```

PFILE オプションを使用して、初期化パラメータ・ファイルの場所を指定する必要があります。

7. utlnchar.sql を実行します。

```
SQL> @utlnchar.sql
```

デフォルトでアップグレードに選択された各国語キャラクタ・セットを上書きするには、n_switch.sql を実行します。

```
SQL> @n_switch.sql
```

8. インスタンスを停止します。

```
SQL> SHUTDOWN IMMEDIATE
```

9. SQL*Plus を終了します。

SQL NCHAR 列のダウングレード

SQL NCHAR 列（NCHAR、NVARCHAR2 および NCLOB）を Oracle9i にアップグレードすると、すべての SQL NCHAR 列を削除するまで、以前のリリースへはダウングレードできません。Oracle8 および Oracle8i の SQL NCHAR データをリカバリする必要がある場合は、以前のリリースのバックアップからデータを再度インポートする必要があります。

Server Manager ラインモード・スクリプトの SQL*Plus への移行

Oracle9i では、Server Manager は使用できません。Server Manager ラインモードを使用して SQL スクリプトを実行する場合は、SQL スクリプトを変更して SQL*Plus と互換性を持つようにする必要があります。SQL*Plus で作業をするための Server Manager ラインモード・スクリプトの変更の詳細は、[付録 C 「Server Manager から SQL*Plus への移行」](#) を参照してください。

リリース 8.0.6 以下のデータベースのアップグレード後のみに行う作業

データベースをリリース 8.0.6 以下のリリースからアップグレードした場合のみ、次の追加作業を行います。リリース 8.1.7 以上のリリースからアップグレードする場合は、これらの作業は必要ありません。

パラレル実行での問題の回避

Oracle8i 以上では、パラレル実行のメッセージ・バッファはラージ・プールから割り当てられます。以前のリリースでは、共有プールから割り当てられていました。この変更により発生する問題を避けるために、初期化パラメータ・ファイルの次のパラメータを調整する必要があります。

- SHARED_POOL_SIZE
- LARGE_POOL_SIZE

参照： これらのパラメータの調整については、A-8 ページの「[ラージ・プールから割り当てられたパラレル実行](#)」を参照してください。

Windows オペレーティング・システムでのファイル名の正規化

Oracle を Windows オペレーティング・システムで実行している場合のみ、ファイル名の正規化が必要です。UNIX プラットフォームでは、これらの手順を実行する必要はありません。

制御ファイルおよびリカバリ・カタログは、データベースが必要とする次のようなファイルにアクセスするためにファイル名を格納しています。

- データ・ファイル
- 制御ファイル
- Oracle が使用するオンラインおよびアーカイブ REDO ログ
- Recovery Manager が使用するデータ・ファイルのコピーおよびディスク上のバックアップ・ピース

Windows オペレーティング・システムで実行しているリリース 8.1.6 より前のリリースでは、ファイル名の正規化メカニズムの不具合によって、異なる 2 つのファイル名が同じ物理ファイルを参照することができました。このため、たとえば、Oracle はファイルのフルパス名を制御ファイルに記録しない場合があります。つまり、Oracle は、`c:\¥oracle¥oradata¥dbfile1.dbf` ではなく `dbfile1.dbf` のみを記録する場合があります。この場合、`c:\¥oracle¥oradata¥dbfile1.dbf` を変更する後続の文で、Oracle はこのファイルを `dbfile1.dbf` とは異なるファイルであると認識します。

同様にこの動作によって、既存のファイルを参照する SQL 文および Recovery Manager のコマンドは入力されたとおりに指定されるか、または認識されません。既存のファイルを参照する SQL 文には、ALTER DATABASE RENAME FILE 文などがあります。

リリース 8.1.6 以上では、このファイル名の正規化メカニズムの不具合が修正されています。ただし、制御ファイルおよびリカバリ・カタログの既存のファイル名は、新しいファイル名の正規化メカニズムで正規化する必要があります。

注意： リリース 8.1.6 より前の Oracle リリースでは、次の手順は行わないでください。

これらのファイル名を正規化するには、次の手順を行います。

1. SQL*Plus を使用して、SYSDBA 権限を持つユーザーとしてデータベースに接続します。
2. SHUTDOWN NORMAL または SHUTDOWN IMMEDIATE を使用して、データベースを停止します。

```
SQL> SHUTDOWN IMMEDIATE
```

3. 制御ファイルのオペレーティング・システム・バックアップを作成します。

参照： オペレーティング・システム・バックアップの詳細は、『Oracle9i ユーザー管理バックアップおよびリカバリ・ガイド』を参照してください。

4. STARTUP MOUNT を実行して、データベースをオープンせずにマウントします。

```
SQL> STARTUP MOUNT
```

5. DBMS_BACKUP_RESTORE.RENORMALLIZEALLFILENAMES プロシージャを実行して、制御ファイルのファイル名を正規化します。

```
SQL> EXECUTE DBMS_BACKUP_RESTORE.RENORMALLIZEALLFILENAMES;
```

6. DBMS_BACKUP_RESTORE.RENORMALLIZEALLFILENAMES プロシージャが正常に終了したら、データベースをオープンします。

```
SQL> ALTER DATABASE OPEN;
```

7. SQL*Plus を終了します。

8. Recovery Manager にログインして、ターゲット・データベースおよびリカバリ・カタログに接続します。

たとえば、ターゲット・データベースのネットワーク・サービス名が TGT_DB で、リカバリ・カタログ・データベースのネットワーク・サービス名が CAT_DB の場合、適切なスキーマ名およびパスワードで次のように入力します。

```
rman target sys/password@tgt_db catalog rcat_schema/rcat_password@cat_db
```

9. RENORMALIZE CATALOG コマンドを発行して、このターゲット・データベースのリカバリ・カタログのファイル名を正規化します。

```
RMAN> renormalize catalog;
```

注意： RENORMALIZE CATALOG コマンドは Recovery Manager 構文の一部ではないため、『Oracle9i Recovery Manager ユーザーズ・ガイド』には記述されていません。このコマンドは、Windows プラットフォームのリリース 8.1.6 より前のリリースから移行またはアップグレードされたデータベースのみで使用されるものです。

10. このリカバリ・カタログに登録されたリリース 8.1.6 以上の各ターゲット・データベースで、手順 8～9 を繰り返します。

これで、ファイル名が正規化されました。

注意： 前述のファイル名を正規化する手順を行う前に作成したバックアップから Point-in-Time リカバリのために制御ファイルをリストアする必要がある場合は、最初にバックアップ制御ファイルをリストアし、次に手順 1～7 を実行し、最後にリカバリを実行します。

使用禁止のファンクション索引の再構築

アップグレード中に、一部のファンクション索引が使用禁止になることがあります。これらの索引を検索するには、次の SQL 文を発行します。

```
SELECT owner, index_name, funcidx_status  
FROM dba_indexes WHERE funcidx_status = 'DISABLED';
```

リストされた使用禁止のファンクション索引を再構築します。

マテリアライズド・ビューのアップグレード

注意： スナップショットとマテリアライズド・ビューは同義語です。

リリース 8.0 からアップグレードされたマテリアライズド・ビュー、またはリリース 8.0 データベースからインポートされたマテリアライズド・ビューでは、リリース 8.1 以上で利用できる新しいサマリー管理機能を使用できません。これらの新しい機能を使用する場合は、アップグレードされたマテリアライズド・ビューおよびリリース 8.0 からインポートされたマテリアライズド・ビューのそれぞれに次の手順を実行します。

1. `QUERY REWRITE` 権限をマテリアライズド・ビューの所有者に付与します。ローカルのマテリアライズド・ビューのみでクエリー・リライトができます。

マテリアライズド・ビューが、所有者のスキーマ以外のスキーマ・オブジェクトを参照する場合、`GRANT GLOBAL QUERY REWRITE` 文を発行する必要があります。
2. アップグレードするマテリアライズド・ビューに `ALTER MATERIALIZED VIEW ... ENABLE QUERY REWRITE` 文を発行します。

たとえば、`SSORDERS` という名前のマテリアライズド・ビューには、次の文を発行します。

```
ALTER MATERIALIZED VIEW ssorders ENABLE QUERY REWRITE;
```

また、マテリアライズド・ビューに対して `ENABLE QUERY REWRITE` を発行しなかった場合、マテリアライズド・ビューに対して `ALTER MATERIALIZED VIEW ... COMPILE` 文を発行するまで、`DBMS_MVIEW.REFRESH` プロシージャの `ATOMIC=FALSE` オプションは無効になります。たとえば、`SSCUST` という名前のマテリアライズド・ビューには、次の文を発行します。

```
ALTER MATERIALIZED VIEW sscust COMPILE;
```

マテリアライズド・ビューに `ALTER MATERIALIZED VIEW ... ENABLE QUERY REWRITE` 文など、別の `ALTER MATERIALIZED VIEW` 文を発行した場合は、この文を発行する必要はありません。

キュー表のアップグレード

既存のキュー表をアップグレードした場合のみ、次のリリース 8.1 以上の AQ 拡張機能が利用できます。

- 伝播されたメッセージのためのオリジナル・メッセージ ID 列の追加
- 送信者の ID 列の追加
- キューおよびシステム・レベル権限

- ルール・ベースのサブスクリプション
- リリース 8.0 の VARRAY に格納された履歴管理情報の個別の記憶域

既存のキュー表をアップグレードするには、DBMS_AQADM.MIGRATE_QUEUE_TABLE プロシージャを実行して、オプションに 8.1 を指定します。たとえば、ユーザー scott が所有する tb_queue というキュー表の場合は、次のコマンドを発行します。

```
EXECUTE dbms_aqadm.migrate_queue_table (  
    queue_table => 'scott.tb_queue',  
    compatible => '8.1');
```

リリース 8.1 以上と互換性のある新しいキュー表を作成するには、キュー表の所有者として接続し、DBMS_AQADM.CREATE_QUEUE_TABLE プロシージャを実行し、次の例のように COMPATIBLE オプションに 8.1 を指定します。

```
EXECUTE dbms_aqadm.create_queue_table(  
    queue_table => 'scott.tkaqqtpeqt',  
    queue_payload_type => 'message',  
    sort_list => 'priority,enq_time',  
    multiple_consumers => true,  
    comment => 'Creating queue with priority and enq_time sort order',  
    compatible => '8.1');
```

注意： キュー表をアップグレードし、新しいリリース 8.1 互換キュー表を作成するには、COMPATIBLE 初期化パラメータを 8.1.0 以上に設定する必要があります。

リカバリ・カタログのアップグレード

アップグレードされたデータベースのリカバリ・カタログ・スキーマが、アップグレードしたデータベースとは別のデータベースに存在する場合があります。Recovery Manager 実行可能ファイルをリリース 8.1 へアップグレードした場合は、リカバリ・カタログもリリース 8.1 へアップグレードする必要があります。

また、単一のリカバリ・カタログによって管理される異なるリリースの複数のデータベースがある場合は、特定の Recovery Manager リリースとリカバリ・カタログ・リリース間での互換性の問題を考慮する必要があります。たとえば、Recovery Manager のリリース 8.1.3 と 8.1.4 は、リリース 8.1.5 以上のリカバリ・カタログにアクセスできません。したがって、この場合は、リリース 8.1.5 以上へのリカバリ・カタログによって管理されるデータベースのすべてをアップグレードする必要があります。Recovery Manager とリカバリ・カタログの互換性の詳細は、5-41 ページの「[Recovery Manager](#)」を参照してください。

リカバリ・カタログを手動でアップグレードするには、次の手順を実行します。

1. Recovery Manager にログインして、リカバリ・カタログに接続します。

たとえば、RCAT/RCAT がリカバリ・カタログの所有者のユーザー名およびパスワードで、RECDDB がネットワーク・サービス名の場合、次のように入力します。

```
rman rcvcat rcat/rcat@recdb
```

Recovery Manager のリリース 8.1 を使用して古いリカバリ・カタログへ最初に接続するときは、メッセージ RMAN-06186 が表示されます。これは、リカバリ・カタログをアップグレードする必要があることを表します。

2. UPGRADE CATALOG コマンドを使用して、リカバリ・カタログを最新のリリースへアップグレードします。Recovery Manager によって、カタログ・アップグレードの確認のためにコマンドを 2 回入力するように指示されます。アップグレード中にエラーが発生した場合は、Recovery Manager ログ内に表示されます。

次に、リリース 8.0.4 からリカバリ・カタログをアップグレードするセッションのログを示します。

Recovery Manager: Release 8.1.5.0.0

RMAN-06008: リカバリ・カタログ・データベースに接続されました。

RMAN-06186: PL/SQL パッケージ rcat.DBMS_RCVCAT バージョン 8.0.4 (RCVCAT データベース内) が古すぎます。

database is too old

RMAN> upgrade catalog

RMAN-06435: リカバリ・カタログの所有者は rcat です。

RMAN-06442: UPGRADE CATALOG コマンドを再入力して、カタログのアップグレードを確認してください。

RMAN> upgrade catalog

RMAN-06408: リカバリ・カタログがバージョン 8.1.5 にアップグレードしました。

DBMS_STATS パッケージで作成された統計表のアップグレード

DBMS_STATS.CREATE_STAT_TABLE プロシージャを使用して統計表を作成した場合、次のプロシージャを実行してこれらの表をアップグレードします。

```
EXECUTE DBMS_STATS.UPGRADE_STAT_TABLE('scott', 'stat_table');
```

ここで、SCOTT は統計表の所有者で、STAT_TABLE は統計表の名前です。各統計表にこのプロシージャを実行します。

リリース 7.3.4 のデータベースのアップグレード後のみに行う作業

データベースをリリース 7.3.4 からアップグレードした場合のみ、次の作業を行います。データベースをリリース 8.0.6 以上のリリースからアップグレードした場合は、これらの作業は必要ありません。

使用不可となったビットマップ索引の再構築

アップグレード中に、一部のビットマップ索引が使用不可になることがあります。これらの索引を検索するには、次の SQL 文を発行します。

```
SELECT index_name, index_type, table_owner, status
FROM dba_indexes
WHERE index_type = 'BITMAP'
AND status = 'UNUSABLE';
```

リストされた使用不可のビットマップ索引を再構築します。

参照： ビットマップ索引の使用の詳細は、『Oracle9i データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』および『Oracle9i データベース概要』を参照してください。

パーティション・ビューからパーティション表への移行

Oracle9i の新しいアプリケーションでのパーティション・ビューの使用はお薦めしません。また、既存のパーティション・ビューはパーティション表へ変換する必要があります。

ALTER TABLE 文の EXCHANGE PARTITION オプションを使用して、Oracle7 データベース用に作成されたパーティション・ビューをパーティション表へ変換できます。

参照： パーティション・ビューのパーティション表への変換方法の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。パーティション・ビューおよびパーティション表のバックグラウンド情報は、『Oracle9i データベース概要』を参照してください。

無効な日付制約のチェック

無効な日付制約は、無効な日付操作を含みます。無効な日付操作は、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 CHECK 制約で実行されるもので、無効な日付操作を組み込んでいる制約を無効とマークします。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。Oracle7 では、年号桁が 2 桁の日付定数を持つ制約を作成できます。ただし、Oracle8 以上では、CHECK 制約の日付定数に 4 桁の年号がない場合、エラーを戻します。

utlconst.sql スクリプトを実行するには、次の手順を実行します。

1. システム・プロンプトで、`ORACLE_HOME/rdbms/admin` ディレクトリへ移動します。
2. `SQL*Plus` を起動します。
3. `SYSDBA` 権限を持つユーザーとして、データベース・インスタンスに接続します。
4. 次のように入力します。

```
SQL> SPOOL utlresult.log
SQL> @utlconst.sql
SQL> 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` スクリプトは無効な制約を修正せずに、使用禁止にします。無効な制約を削除するか、必要な変更を加えた後に再作成する必要があります。

Oracle Net Services の新しいリリースへのアップグレード（オプション）

Oracle Net Services の新しいリリースへの移行またはアップグレードは必須ではありません。ただし、Oracle Net Services には `SQL*Net V2` に比べて非常に便利で、簡素化された構成および拡張された機能があります。また、Oracle Net Services の新しいリリースは、次の点で以前のリリースの `Net8` および `SQL*Net` よりも優れています。

- **サービス・ネーミング**によって、クライアントは特定のデータベースのインスタンスではなく、サービス名を使用して、全体としてサービスにアクセスできます。サービス・ネーミングはサービス名を論理的に特定なインスタンス名と区別し、`SID` パラメータを交換して、1つのインスタンスが複数のサービスで利用できるようにします。個々のインスタンスも複数のサービスの一部になることができます。
- **インスタンス登録**は、自動で行われます。インスタンスは起動時にそれ自体をリスナーに登録します。以前のリリースでは、インスタンスの情報は手動で `listener.ora` ファイルに設定されていました。
- **フェイルオーバー**は、自動で行われます。インスタンスが停止した場合は、クライアント接続要求は自動的に別のリスナーへ送信されます。
- **ロード・バランシング**によって、リスナーへ接続を分散します。

参照： Oracle Net Services のメリットの詳細は、『Oracle9i Net Services 管理者ガイド』を参照してください。Oracle Net Services の新しいリリースへの移行またはアップグレード方法の詳細は、付録 B「[Oracle Net Services のアップグレードに関する考慮点](#)」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

2-9 ページの「[テスト計画の作成](#)」で作成したテスト計画を使用して、新しい Oracle9i 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、新しい Oracle9i データベースでのパフォーマンスは、以前のデータベースでのパフォーマンスより向上します。新しい Oracle9i データベースでデータベースのパフォーマンスの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。初期化パラメータの設定が正しくない場合、パフォーマンスに影響することがあります。

アップグレードされたデータベースのチューニング

アップグレードしたデータベースのパフォーマンスを改善するには、データベースをチューニングします。以前の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理が原因で、アップグレードした Oracle9i データベースのパフォーマンスが低下することはありません。

参照： チューニングの詳細は、『Oracle9i データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』を参照してください。

現行リリースのワード・サイズの変更

この項では、現行リリースのワード・サイズを変更する手順について説明します（32 ビット・ソフトウェアから 64 ビット・ソフトウェア、または 64 ビット・ソフトウェアから 32 ビット・ソフトウェア）。

参照： ワード・サイズ変更の詳細は、1-10 ページの「[ワード・サイズの変更](#)」を参照してください。

現行リリースのワード・サイズを変更するには、次の手順を実行します。

1. SQL*Plus を起動します。
2. SYSDBA 権限を持つユーザーとして、データベース・インスタンスに接続します。
3. データベース上で、SHUTDOWN IMMEDIATE を実行します。

```
SQL> SHUTDOWN IMMEDIATE
```

注意： Oracle9i Real Application Clusters に対して、すべてのインスタンスにこの文を発行します。また、CLUSTER_DATABASE 初期化パラメータを false に設定します。ワード・サイズの変更が完了した後、true に戻すことができます。

4. データベースの完全なバックアップを実行します。

参照： 詳細は、『Oracle9i ユーザー管理バックアップおよびリカバリ・ガイド』を参照してください。

5. 現行リリースと切替え先のリリースに同じ Oracle ホームを使用している場合、Oracle Universal Installer を使用して現行リリースを削除します。異なる Oracle ホーム・ディレクトリを使用している場合は、現行リリースを削除する必要はありません。
6. 現在 32 ビットのリリースがインストールされている場合は、64 ビットのリリースをインストールします。または、現在 64 ビットのリリースがインストールされている場合は、32 ビット・リリースをインストールします。

注意： インストールおよび削除は、オペレーティング・システム固有の作業です。インストールおよび削除の詳細は、オペレーティング・システム固有の Oracle9i のインストール・ガイドおよびオペレーティング・システムの Oracle9i の README ファイルを参照してください。

7. 構成ファイルを古い Oracle ホーム以外の場所へコピーします。
 - a. 使用するパラメータ・ファイルが古い環境の Oracle ホーム内に存在する場合、古い環境の Oracle ホーム以外の場所へ移動またはコピーします。パラメータ・ファイルはどの場所にも置くことができます。ただし、新しいリリースへの切替え後は、古い環境の Oracle ホーム内に置くことはできません。
 - b. パラメータ・ファイルが IFILE（インクルード・ファイル）エントリを含み、IFILE 内に指定されたファイルが古い環境の Oracle ホーム内に存在する場合、IFILE エントリで指定されているファイルを古い環境の Oracle ホーム以外の場所へコピーします。IFILE エントリ内に指定されたファイルは、追加の初期化パラメータを持っています。このファイルをコピーしたら、パラメータ・ファイルの IFILE エントリを編集して新しい場所を指定します。
 - c. 古い Oracle ホーム内に存在するパスワード・ファイルがある場合、そのパスワード・ファイルを新しい Oracle9i の Oracle ホームに移動またはコピーします。パスワード・ファイルの名前および位置は、オペレーティング・システ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 オペレーティング・システムでは、%ORACLE_HOME%\database\pwsid.ora です。両方の場合とも、sid は Oracle インスタンスの ID です。

注意： Oracle9i Real Application Clusters に対して、すべてのノードでこの手順を実行します。また、ご使用の initdb_name.ora ファイルが古い環境の Oracle ホーム内に存在する場合、initdb_name.ora ファイルを古い環境の Oracle ホーム以外の場所へ移動またはコピーします。

8. システム・プロンプトで、`ORACLE_HOME/rdbms/admin` ディレクトリへ移動します。
9. `SQL*Plus` を起動します。
10. `SYSDBA` 権限を持つユーザーとして、データベース・インスタンスに接続します。
11. `STARTUP MIGRATE` を実行します。

```
SQL> STARTUP MIGRATE
```

`PFILE` オプションを使用して、初期化パラメータ・ファイルの場所を指定する必要がある場合があります。

12. 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。

```
SQL> SPOOL wordsize.log
```

13. `utlirp.sql` を実行します。

```
SQL> @utlirp.sql
```

`utlirp.sql` スクリプトは新しいデータベースが必要な形式で既存の `PL/SQL` モジュールを再コンパイルします。このスクリプトは、最初に特定のディクショナリ表を変更します。次に、`STANDARD` パッケージおよび `DBMS_STANDARD` パッケージを再ロードします。これは、`PL/SQL` を使用するために必要です。最後に、パッケージ、プロシージャ、型などのすべての `PL/SQL` モジュールを再コンパイルします。

14. 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。

```
SQL> SPOOL OFF
```

次に、スプール・ファイルを確認して、すべてのパッケージおよび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ます。手順 12 で、スプール・ファイル名を `wordsize.log` としました。このファイルで見つけた問題を処理します。

15. `ALTER SYSTEM DISABLE RESTRICTED SESSION` を実行します。

```
SQL> ALTER SYSTEM DISABLE RESTRICTED SESSION;
```

これで、ご使用のデータベースのワード・サイズが変更されました。データベースをオープンして通常どおり使用できます。

互換性および相互運用性

この章では、Oracle の異なるリリース間で発生する可能性のある互換性の問題、および相互運用性の問題について説明します。これらの違いは、一般的なデータベース管理および既存のアプリケーションに影響する可能性があります。

この章では、次の項目について説明します。

- [互換性の概要](#)
- [COMPATIBLE 設定を必要とする機能](#)
- [相互運用性の概要](#)
- [リリース 2 \(9.2\) とリリース 1 \(9.0.1\) 間の互換性および相互運用性の問題](#)
- [リリース 2 \(9.2\) と以前のリリース間の互換性および相互運用性の問題](#)

互換性の概要

Oracle の新しいリリースにアップグレードする場合、特定の新機能によって、以前のリリースとの互換性がなくなることがあります。次の機能については、アップグレードした Oracle データベースは以前のリリースと互換性がなくなります。

- 以前のリリースでは処理できないデータ形式をディスクに格納する新機能（データ・ディクショナリの変更を含む）。
- 以前のリリースと新しいリリースを比較すると、異なる動作をする既存の機能。このような非互換性は**言語的非互換性**として分類されます。

COMPATIBLE 初期化パラメータ

Oracle では、COMPATIBLE 初期化パラメータを使用して、データベースの互換性を制御できます。COMPATIBLE 初期化パラメータがパラメータ・ファイルに設定されない場合、そのリリースで可能な最も低い値（すべての Oracle9i リリースでは 8.1.0）がデフォルトで設定されます。COMPATIBLE 初期化パラメータをより高い値に設定しないかぎり、ご使用のデータベースに互換性の問題が発生するような新機能は使用できません。

このデフォルトの動作には、次のようなメリットがあります。

- デフォルトによって前回のリリースとの互換性が維持されるため、ダウングレードしやすくなります。
- 複数のデータベースを使用する環境で操作する場合、アップグレードされたデータベースは、アップグレードされていないデータベースとの互換性を保ちます。

ただし、デフォルト設定では、COMPATIBLE 初期化パラメータを設定しない場合、新しいリリースの機能の多くが使用できません。

参照： COMPATIBLE 初期化パラメータを必要とする機能のリストは、5-9 ページの「[COMPATIBLE 設定を必要とする機能](#)」を参照してください。

Oracle9i の新しいリリースのインストール中にインストールする製品に応じて、Oracle Universal Installer が COMPATIBLE 初期化パラメータをより高い値（9.2.0 など）に設定することがあります。現行の COMPATIBLE 初期化パラメータの設定が不明な場合、パラメータ・ファイルを確認してください。


 **図 5-1** に、リリース 8.0、リリース 8.1、リリース 1 (9.0.1) およびリリース 2 (9.2) の COMPATIBLE 初期化パラメータのデフォルトの設定およびその他の可能な設定を示します。

図 5-1 COMPATIBLE 初期化パラメータ

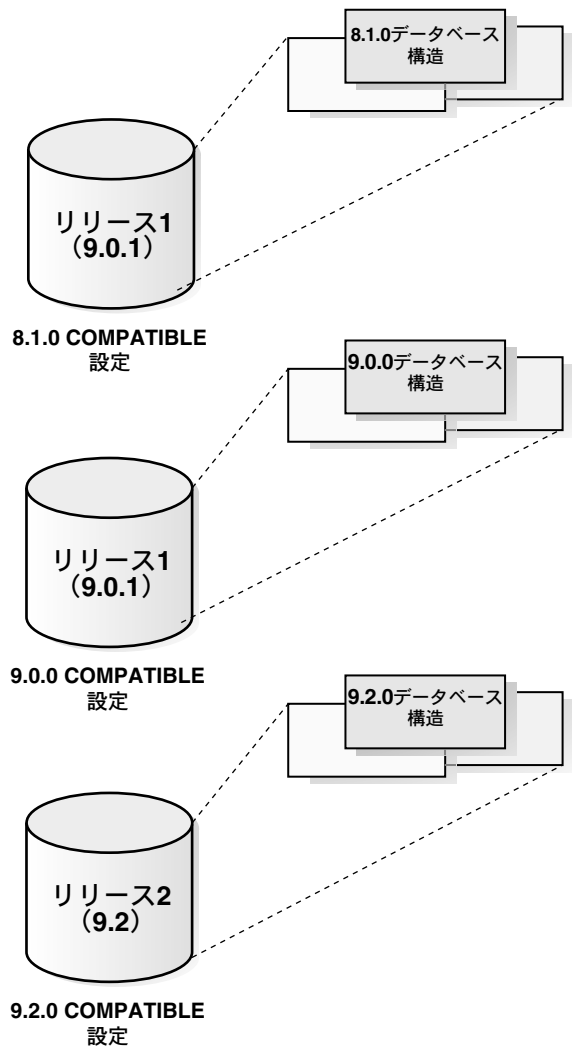
デフォルトは8.0.0  リリース8.0 8.0.xにのみ設定可能	デフォルトは8.0.0  リリース8.1 8.1.yまたは8.0.xに 設定可能	デフォルトは8.1.0  リリース1 (9.0.1) 9.0.yまたは8.1.xに 設定可能	デフォルトは8.1.0  リリース2 (9.2) 9.2.z、9.0.yまたは8.1.xに 設定可能
最小設定値: 8.0.0	最小設定値: 8.0.0	最小設定値: 8.1.0	最小設定値: 8.1.0
最大設定値: ご使用の 現行リリース	最大設定値: ご使用の 現行リリース	最大設定値: ご使用の 現行リリース	最大設定値: ご使用の 現行リリース
設定不可能な値: • Oracle7以下のリリース • 現行リリース以上の リリース (8.1.0以上を 含む)	設定不可能な値: • Oracle7以下のリリース • 現行リリース以上の リリース (9.0.0以上を 含む)	設定不可能な値: • Oracle8以下のリリース • 現行リリース以上の リリース (9.2.0以上を 含む)	設定不可能な値: • Oracle8以下のリリース • 現行リリース以上の リリース

COMPATIBLE 初期化パラメータの役割

COMPATIBLE 初期化パラメータは、次のように動作します。

- データベースの動作を制御します。たとえば、COMPATIBLE 初期化パラメータを 8.1.6 に設定してリリース 2 (9.2) のデータベースを実行する場合、リリース 2 (9.2) のデータベースはリリース 8.1.6 と互換性のあるデータベース構造をディスクに生成します。つまり、COMPATIBLE 初期化パラメータの設定によって、新機能を使用可能または使用禁止にできます。データベースと COMPATIBLE 初期化パラメータに設定されたリリースとの間の互換性を損う新機能を使用しようとすると、エラーが戻ります。ただし、互換性を損う変更をディスクに行わない新機能はすべて使用できます。
- データベースが、そのデータベースの COMPATIBLE の設定と互換性があることを確認します。データベースがその設定と互換性がない場合、データベースは起動せず、エラーを戻して終了します。この場合、COMPATIBLE 初期化パラメータを、データベースに対する適切な値に設定する必要があります。

図 5-2 COMPATIBLE 設定によるデータベース構造



参照： データベース構造の詳細は、『Oracle9i データベース概要』を参照してください。

ダウングレードおよび互換性

新しいリリースへアップグレードすると、COMPATIBLE 初期化パラメータを新しいリリースと一致するように設定できます。これによって、新しいリリースのすべての機能を使用できますが、前回のリリースにダウングレードすることがより難しく、または不可能になることがあります。ダウングレードする場合、ダウングレード先のリリースとの非互換性をすべて削除する必要があります。その処理には時間と手間がかかります。

参照： ダウングレードの詳細は、[第7章「以前の Oracle リリースへのデータベースのダウングレード」](#)を参照してください。

互換性のレベル

データベースの互換性レベルは、COMPATIBLE 初期化パラメータの値に対応します。たとえば、COMPATIBLE 初期化パラメータを 8.1.6 に設定すると、データベースは 8.1.6 の互換性レベルで実行されます。

現行の COMPATIBLE 初期化パラメータ値の確認

現行の COMPATIBLE 初期化パラメータ値を確認するには、次の SQL 文を発行します。

```
SQL> SELECT name, value, description FROM v$parameter
        WHERE name = 'compatible';
```

特定の機能の互換性レベルの確認

特定の機能の互換性レベルを確認するには、次の SQL 文を発行します。

```
SQL> SELECT * FROM v$compatibility;
```

互換性レベルが 0.0.0.0.0 の機能は現在使用されていません。

COMPATIBLE 初期化パラメータ設定のタイミング

COMPATIBLE 初期化パラメータは、アップグレードまたはダウングレード処理の特定の時点で設定する必要があります。該当する章の手順に従い、設定を指示された場合のみ、COMPATIBLE 初期化パラメータを設定してください。

注意： アップグレードまたはダウングレードが完了した後は、必要に応じて COMPATIBLE 初期化パラメータの設定を変更できます。

COMPATIBLE 初期化パラメータの設定

次のいずれかの手順を実行して、COMPATIBLE 初期化パラメータを設定します。

- COMPATIBLE 初期化パラメータ値を増やす
- COMPATIBLE 初期化パラメータ値を減らす

COMPATIBLE 初期化パラメータ値を増やす

COMPATIBLE 初期化パラメータ値を増やすには、次の手順を実行します。

1. COMPATIBLE 初期化パラメータ値を増やす前に、データベースのバックアップを取ります（オプション）。

COMPATIBLE 初期化パラメータ値を増やすことによって、データベースと Oracle の以前のリリースとの間の互換性が失われることがあります。バックアップを取っておくと、必要に応じて以前のリリースに戻すことができます。

参照： バックアップの詳細は、『Oracle9i バックアップおよびリカバリ概要』を参照してください。

2. サーバー・パラメータ・ファイルを使用している場合は、次の手順を実行します。
 - a. サーバー・パラメータ・ファイルを更新して、COMPATIBLE 初期化パラメータの値を設定または変更します。

たとえば、COMPATIBLE 初期化パラメータを 9.2.0 に設定するには、次の文を発行します。

```
SQL> ALTER SYSTEM SET COMPATIBLE = '9.2.0' SCOPE=SPFILE;
```

- b. インスタンスを停止し、再起動します。
3. 初期化パラメータ・ファイルを使用している場合は、次の手順を実行します。
 - a. インスタンスが実行している場合は、停止します。

```
SQL> SHUTDOWN IMMEDIATE
```

- b. 初期化パラメータ・ファイルを編集して、COMPATIBLE 初期化パラメータの値を設定または変更します。

たとえば、COMPATIBLE 初期化パラメータを 9.2.0 に設定するには、初期化パラメータ・ファイルに次のように入力します。

```
COMPATIBLE = 9.2.0
```

- c. STARTUP を使用してインスタンスを起動します。

COMPATIBLE 初期化パラメータ値を減らす

COMPATIBLE 初期化パラメータ値を減らすには、次の手順を実行します。

1. データベースが COMPATIBLE 初期化パラメータに設定する値のリリースと互換性があることを確認します。

参照： 非互換性の削除の詳細は、7-2 ページの「[非互換性の削除](#)」を参照してください。

2. COMPATIBLE 初期化パラメータに設定するリリースより高いリリースに追加された初期化パラメータを使用している場合は、そのパラメータをご使用のパラメータ・ファイルから削除します。

参照： Oracle9i の各リリースに追加された初期化パラメータのリストは、『Oracle9i データベース・リファレンス』を参照してください。

3. ALTER DATABASE RESET COMPATIBILITY 文を発行します。

```
SQL> ALTER DATABASE RESET COMPATIBILITY;
```

参照： 詳細は、5-8 ページの「[ALTER DATABASE RESET COMPATIBILITY について](#)」を参照してください。

4. サーバー・パラメータ・ファイルを使用している場合は、次の手順を実行します。

- a. サーバー・パラメータ・ファイルを更新して、COMPATIBLE 初期化パラメータの値を設定または変更します。

たとえば、COMPATIBLE 初期化パラメータを 9.0.0 に設定するには、次の文を発行します。

```
SQL> ALTER SYSTEM SET COMPATIBLE = '9.0.0' SCOPE=SPFILE;
```

- b. インスタンスを停止し、再起動します。

5. 初期化パラメータ・ファイルを使用している場合は、次の手順を実行します。

- a. インスタンスが実行している場合は、停止します。

```
SQL> SHUTDOWN IMMEDIATE
```

- b. 初期化パラメータ・ファイルを編集して、COMPATIBLE 初期化パラメータの値を設定または変更します。

たとえば、COMPATIBLE 初期化パラメータを 9.0.0 に設定するには、初期化パラメータ・ファイルに次のように入力します。

```
COMPATIBLE = 9.0.0
```

- c. STARTUP を使用してインスタンスを起動します。

ALTER DATABASE RESET COMPATIBILITY について

データベースの互換性レベルを下げるには、ALTER DATABASE RESET COMPATIBILITY 文を使用します。UNDO 表領域などの一部の Oracle の機能は 9.0.0 以上の互換性レベルを必要とします。COMPATIBLE 初期化パラメータを 9.0.0 以上に設定して UNDO 表領域を作成すると、その UNDO 表領域は、データベース内で互換性レベル 9.0.0 のオブジェクトになります。

ALTER DATABASE RESET COMPATIBILITY は、考えられる最も低い互換性レベル (Oracle9i のすべてのリリースで 8.1.0) とは互換性のないオブジェクトを作成する各機能を調査します。特定の機能に対して互換性のないオブジェクトが存在しないことが確認されると、その機能の互換性レベルは 0.0.0 に設定されます。これは、その機能が使用されていないことを意味します。ただし、特定の機能によって作成された互換性のないオブジェクトが確認されると、その機能に対する互換性レベルは、その機能を使用可能にするために考えられる最も低い互換性レベルに設定されます。

たとえば、1 つ以上の UNDO 表領域が存在する場合、9.0.0 がその UNDO 表領域の機能を使用可能にするための、考えられる最も低い互換性レベルであるため、UNDO 表領域の互換性レベルは 9.0.0 に設定されます。ただし、ALTER DATABASE RESET COMPATIBILITY では、データベースの互換性レベルを上げることができないことに注意してください。9.0.0 以上の互換性レベルを必要とするデータベース・オブジェクトを作成する前に、まず COMPATIBLE 初期化パラメータをより高い値 (9.0.0 など) に設定する必要があります。

データベースをクローズし、COMPATIBLE 初期化パラメータを低い値に変更してからデータベースをオープンすると、Oracle は各機能の互換性レベルを調査します。COMPATIBLE 初期化パラメータの値より上の互換性レベルの機能がある場合、データベースはオープンできず、互換性のない機能が存在することを示すエラー・メッセージを表示します。

ご使用のデータベースの非互換性をすべて削除しても、データベースを停止する前に ALTER DATABASE RESET COMPATIBILITY 文を発行しないと、非互換性がないにもかかわらず、データベースをオープンすることはできません。これは、各機能の互換性レベルの調査が、データベース内の各オブジェクトに対して行われるわけではないためです。Oracle は、互換性を持たないオブジェクトが、過去に作成されたことがあるという情報のみを記録しており、これらの機能の互換性レベルは、リセットされないかぎり、下がることはありません。ALTER DATABASE RESET COMPATIBILITY 文を実行すると、互換性を持たないオブジェクトが明示的に確認され、互換性を持たないオブジェクトがない場合、互換性レベルはリセットされます。

COMPATIBLE 設定を必要とする機能

表 5-1 にリストされている機能を使用するには、COMPATIBLE 初期化パラメータを指定された値に設定する必要があります。リストされている機能は、Oracle のすべての機能ではなく、互換性レベルを必要とする機能のみです。一部の機能は互換性レベルを必要としません。

参照：

- 次の項に示す機能の詳細、または Oracle9i リリース 2 (9.2) のその他の新機能の詳細は、『Oracle9i データベース新機能』を参照してください。
- Oracle9i リリース 2 (9.2) の新機能の一覧については、『Oracle9i Database Generic Documentation Master Index』を参照してください。

表 5-1 COMPATIBLE の設定を必要とする機能

機能の識別子	互換性のレベル	説明
DEFPART	9.2.0.0.0	リリース 2 (9.2) のデフォルト・パーティション <ul style="list-style-type: none">■ リスト・パーティション表のデフォルト・パーティション
MV92	9.2.0.0.0	リリース 2 (9.2) のマテリアライズド・ビュー
PARTMCLS	9.2.0.0.0	リリース 2 (9.2) のパーティション化メソッド <ul style="list-style-type: none">■ レンジ・リスト・メソッドを使用した表のパーティション化
KNL92	9.2.0.0.0	リリース 2 (9.2) の Streams
SPTEMPL	9.2.0.0.0	リリース 2 (9.2) のサブパーティション・テンプレート <ul style="list-style-type: none">■ コンポジット・パーティション表のサブパーティション・テンプレート
FGASYNPL	9.2.0.0.0	ファイングレイン・セキュリティ・シノニム・ポリシー
HSC	9.2.0.0.0	ヒープ・セグメント・ブロック圧縮
LOB_RET	9.2.0.0.0	LOB リテンション <ul style="list-style-type: none">■ LOB 列に格納されたリテンション
LMST	9.2.0.0.0	ローカル管理 SYSTEM 表領域
COLLOCT	9.2.0.0.0	表内の順序付けされたコレクション <ul style="list-style-type: none">■ 表に格納された、順序付けされたコレクション

表 5-1 COMPATIBLE の設定を必要とする機能（続き）

機能の識別子	互換性のレベル	説明
PMTMGDLB	9.2.0.0.0	LOB を含む、自動セグメント領域で管理される表領域 <ul style="list-style-type: none"> ■ 自動セグメント領域で管理される表領域の LOB 列
RLENG	9.2.0.0.0	ルール・エンジン
SYNUDC	9.2.0.0.0	型シノニム・コンストラクタまたはユーザー定義コンストラクタ
XMLSBSTR	9.2.0.0.0	XMLSchema ベースの XMLType 記憶域
PMTMGDTS	9.0.1.3.0	自動セグメント領域で管理される表領域
MV90	9.0.0.0.0	リリース 1 (9.0) のマテリアライズド・ビュー
PARTM82	9.0.0.0.0	リリース 1 (9.0) のパーティション化メソッド <ul style="list-style-type: none"> ■ リスト・メソッドを使用した表のパーティション化
APPROLE	9.0.0.0.0	アプリケーション・ロール
LGMR_B	9.0.0.0.0	LogMiner の基本情報
CPTLEN	9.0.0.0.0	コード・ポイントの長さ
EXTTAB	9.0.0.0.0	外部表の作成
WRDIR	9.0.0.0.0	ディレクトリの書き込み権限
DOMINDEA	9.0.0.0.0	埋込みユーザー定義型のドメイン索引
DOMINIOT	9.0.0.0.0	索引構成表のドメイン索引
DOMINDRM	9.0.0.0.0	行移動を伴うドメイン索引
EJTYPE	9.0.0.0.0	外部 Java 型
APPFGA	9.0.0.0.0	ファイングレイン監査
LGMR_F	9.0.0.0.0	LogMiner の完全な情報
FDOMIND	9.0.0.0.0	ファンクション・ドメイン索引
HASHPIOT	9.0.0.0.0	ハッシュ・パーティション索引構成表
IOTBULOG	9.0.0.0.0	索引構成表のバッチ更新ロギング
IOTCVLOG	9.0.0.0.0	索引構成表の列のベクター・ロギング
IOTWMAP	9.0.0.0.0	マッピング表を持つ索引構成表
URIDIND	9.0.0.0.0	UROWID の索引
JOININD	9.0.0.0.0	結合索引

表 5-1 COMPATIBLE の設定を必要とする機能（続き）

機能の識別子	互換性のレベル	説明
LGINDKEY	9.0.0.0.0	大きい索引キー
LDOMIND	9.0.0.0.0	ローカル・ドメイン索引
LOGSTDBY	9.0.0.0.0	ロジカル・スタンバイ
MLCTABLE	9.0.0.0.0	表のマルチレベル・コレクション
MULTBZ	9.0.0.0.0	マルチブロック・サイズ
NFSTABLE	9.0.0.0.0	表の NOT FINAL 型またはサブタイプ
PDMLITLS	9.0.0.0.0	不変 PDML ITL
PIOTLOBS	9.0.0.0.0	LOB を含むパーティション索引構成表 <ul style="list-style-type: none"> ■ パーティション索引構成表の LOB 列 ■ パーティション索引構成表の VARRAY 列
TXNAUDN	9.0.0.0.0	トランザクション名監査の REDO
NESTEDTX	9.0.0.0.0	ネストしたトランザクションの REDO/UNDO
ROWDEP	9.0.0.0.0	行レベル依存性
STAUTOFM	9.0.0.0.0	スタンバイ自動ファイル管理
TYPEVL	9.0.0.0.0	型進化
UNDOTBSP	9.0.0.0.0	UNDO 表領域
VWCONSTR	9.0.0.0.0	ビュー制約
ANYTABLE	9.0.0.0.0	表の XMLType/AnyType/AnyData
ALTERFRL	8.1.6.0.0	空きリストの変更 <ul style="list-style-type: none"> ■ ALTER 文の FREELIST 指定の変更
CARELOB	8.1.6.0.0	LOB のキャッシュ読み込みモード
EDTRIG	8.1.6.0.0	トリガーの拡張 DDL/DML のサポート
FASTDROP	8.1.6.0.0	セグメントの高速削除
OPQTYPE	8.1.6.0.0	不透明な型
TBSMIGTN	8.1.6.0.0	表領域の移行
TBSTRNSG	8.1.6.0.0	一時セグメント

相互運用性の概要

相互運用性とは、Oracle の異なるリリース間における分散環境で、相互に通信および動作する性能を示します。Oracle の分散データベース・システムは、異なるリリースの Oracle データベースを持つことができます。サポートされるすべての Oracle のリリースは、分散データベース・システムで実行できます。ただし、分散データベースとともに動作するアプリケーションは、システムの各ノードで使用できる機能を認識する必要があります。

たとえば、分散データベース・アプリケーションでは、リリース 7.3.4 データベースが、リリース 8.0 以上でのみ利用可能なオブジェクト SQL 拡張子を認識できません。

注意： このマニュアルでは、Oracle の異なるリリース間で実行されるアップグレードおよびダウングレードについて説明しています。そのため、前述の相互運用性の定義が適用されます。ただし、他の Oracle マニュアルでは、**相互運用性**という用語をより幅広い定義で使用する場合があります。たとえば、相互運用性は、異なるハードウェア・プラットフォームとオペレーティング・システム間の通信を示すこともあります。

リリース 2 (9.2) とリリース 1 (9.0.1) 間の互換性および相互運用性の問題

次の項では、互換性の問題および相互運用性の問題を示し、その問題によって発生する問題を防ぐ処置について説明します。これらの項で説明する問題は、リリース 2 (9.2) とリリース 1 (9.0.1) の違いによって発生します。

- ローカル管理 SYSTEM 表領域
- 新しい AnyData データ型
- ディクショナリ管理表領域
- 自動セグメント領域で管理される表領域の互換性の変更
- 互換性およびオブジェクト型
- Oracle Managed Files
- Oracle OLAP
- パラレル REDO を使用したログ形式の変更
- Oracle Dynamic Services
- Oracle Syndication Server

ローカル管理 SYSTEM 表領域

SYSTEM 表領域は、COMPATIBLE が 9.2.0 以上に設定されている場合のみ、ローカル管理できます。SYSTEM 表領域は、DBMS_SPACE_ADMIN.TABLESPACE_MIGRATE_TO_LOCAL プロシージャを使用して、ディクショナリ管理形式からローカル管理形式へ移行できます。

SYSTEM 表領域をローカル管理形式へ移行する前に、次のことを確認する必要があります。

- データベースには、SYSTEM 以外のデフォルトの一時表領域がある。
- ディクショナリ管理表領域にロールバック・セグメントがない。
- ローカル管理表領域に 1 つ以上のオンライン・ロールバック・セグメントがある。そうでない場合は、(自動 UNDO 管理モードを使用している場合に) UNDO 表領域がオンラインである必要があります。
- UNDO 領域 (UNDO 表領域またはロールバック・セグメントを含む表領域) を含む表領域およびデフォルトの一時表領域以外のすべての表領域が、読込み専用モードである。
- システムのコールド・バックアップがある。
- システムは制限モード状態である。

SYSTEM 表領域がローカル管理形式へ移行されると、以前のリリースへダウングレードすることはできません。次の問合せを実行すると、SYSTEM 表領域がローカル管理されているかどうかを判断できます。

```
SQL> SELECT ts# FROM ts$  
        WHERE ts# = 0 AND bitmapped <> 0;
```

0 rows が戻される場合、SYSTEM 表領域はディクショナリ管理されています。それ以外の場合、SYSTEM 表領域はローカル管理されています。

新しい AnyData データ型

COMPATIBLE が 9.2.0 以上に設定されている場合のみ、次のデータ型の AnyData 値の永続ストレージは使用可能です。

- TIMESTAMP
- TIMESTAMP WITH TIME ZONE
- TIMESTAMP WITH LOCAL TIME ZONE
- INTERVAL YEAR TO MONTH
- INTERVAL DAY TO SECOND
- NCHAR

- NVARCHAR2
- NCLOB

これらの値を表の列に永続的に格納した場合、以前のリリースにダウングレードすることはできなくなります。

ディクショナリ管理表領域

リリース 2 (9.2) 以上では、ディクショナリ管理表領域は非推奨です。SYSTEM 表領域がディクショナリ管理形式からローカル管理形式へ移行されると、既存のディクショナリ管理表領域は読み込み専用になります。SYSTEM 表領域がローカル管理されると、読み込み / 書き込みモードにできません。

(SYSTEM 表領域が新しくインストールされたか、または移行されたかによって) SYSTEM 表領域がローカル管理されると、新しいディクショナリ管理表領域は作成できません。

自動セグメント領域で管理される表領域の互換性の変更

リリース 1 (9.0.1.3.0) 以上では、自動セグメント領域で管理される表領域の互換性の要件が、最初にリリース 1 (9.0.1.0.0) で導入されたときの 9.0.0.0.0 から 9.0.1.3.0 へ変更されています。リリース 1 (9.0.1.3.0) より前のリリースの Oracle9i からアップグレードし、データベースに自動セグメント領域で管理される表領域が含まれている場合、データベースをオープンするには、COMPATIBLE 初期化パラメータを 9.0.1.3.0 以上に設定する必要があります。既存の表領域は削除する必要はありません。

COMPATIBLE を 9.0.1.3.0 以上に設定してデータベースをオープンすると、自動セグメント領域で管理される表領域が使用されている場合、リリース 1 (9.0.1.3.0) 以上にのみダウングレードできます。

互換性およびオブジェクト型

リリース 2 (9.2) 以上では、オブジェクト型は、以前の Oracle リリースの PL/SQL プログラムからは参照できない CONSTRUCTOR キーワードを使用して、ユーザー定義のコンストラクタをサポートします。このようなプログラムはコンパイルに失敗し、エラーが表示されます。

Oracle Managed Files

リリース 1 (9.0.1.2.0) 以上では、Oracle Managed Files で Oracle が使用するネーミング計画が変更されています。そのため、リリース 1 (9.0.1.2.0) より前のリリースの Oracle9i で作成された既存の Oracle Managed Files が、Oracle の通常のオペレーティング・システム・ファイルとして表示されます。Oracle Managed Files を新しいネーミング計画に移行する方法の詳細は、4-3 ページの「[Oracle Managed Files \(OMF\) の移行](#)」を参照してください。

Oracle OLAP

リリース 1 (9.0.1) で提供される OLAP API クライアントは、それより後のリリースの Oracle とは互換性がありません。同様に、リリース 2 (9.2) で提供される OLAP API クライアントは、それより前のリリースの Oracle とは互換性がありません。

アプリケーションが OLAP API を介して接続するために使用するプロシージャは、リリース 2 (9.2) で変更されています。以前のリリースでの接続は CORBA ソフトウェアに依存していましたが、リリース 2 (9.2) では、Java Database Connectivity (JDBC) を介して接続が確立されます。したがって、リリース 1 (9.0.1) で提供される OLAP API クライアントを使用して作成されたプログラムは、それより後のリリースでは実行されず、リリース 2 (9.2) で提供される OLAP API クライアントを使用して作成されたプログラムは、それより前のリリースの Oracle では実行されません。

リリース 1 (9.0.1) で実行するように設計された OLAP API アプリケーションをアップグレードするために、アプリケーション開発者は、リリース 2 (9.2) で提供された OLAP API クライアントを使用し、接続の確立および MetadataProvider の作成用にコードを変更する必要があります。

リリース 2 (9.2) の OLAP API を使用してこれらの処理を実行する方法の詳細は、『Oracle9i OLAP Developer's Guide to the OLAP API』およびリリース 2 (9.2) で提供される Oracle OLAP API のオンライン・ヘルプを参照してください。

パラレル REDO を使用したログ形式の変更

リリース 2 (9.2) 以上では、パラレル REDO 機能によって、新しい形式の REDO ログが生成されます。以前のリリースの Oracle では、パラレル REDO 生成ログは適用できません。ただし、Oracle9i リリース 1 (9.0.1) で Oracle9i リリース 2 (9.2) のパラレル REDO が適用されていることが検出されると、次のエラーが表示されます。

ORA-00303: パラレル REDO は処理できません

新しいログ形式では、ダウングレード前にデータベースを正しく停止する必要があります。完全なバックアップを取ることもお薦めします。ダウングレード後に ORA-00303 エラーが表示された場合、リリース 2 (9.2) へのアップグレード、リカバリの実行、データベースの正しい停止およびダウングレードの再実行を行う必要があります。

リリース 2 (9.2) は、リリース 2 (9.2) のパラレル REDO 形式のログに加えて、リリース 1 (9.0.1) 以前のリリース形式のログを処理できます。

Oracle Dynamic Services

リリース 2 (9.2) 以上では、Oracle Dynamic Services は非推奨です。様々な内容のソースからサービスを作成、集計および配置するための XML ベースのブローカである Oracle Dynamic Services は、Oracle9i データベース・リリース 1 (9.0.1) で導入されています。また、『Oracle Dynamic Services User's and Administrator's Guide』が付属しています。

Oracle9iAS リリース 2 (9.0.2) 以上では、統合された J2EE 対応の Web サービス・プラットフォームが提供されています。Oracle Dynamic Services は、XML/HTML ストリーム処理ツールとして、Oracle9iAS Web Services に統合されています。

参照： 詳細は、『Oracle9iAS Web Services 開発者ガイド』を参照してください。

Oracle9iAS リリース 2 (9.0.2) では、業界標準に準拠し、完全に統合された J2EE サービスおよび Web サービスの配置プラットフォームが提供されています。現行の Dynamic Services の機能は Oracle9iAS プラットフォームに統合され、Dynamic Services 端末は Oracle9i データベース・リリース 2 (9.2) で導入されています。

Oracle Syndication Server

リリース 2 (9.2) 以上では、Oracle Syndication Server は非推奨です。ファイル・システムおよびデータベースの内容を Information and Content Exchange (ICE) に準拠したサブスクリイバに配信するように設計された Oracle Syndication Server は、Oracle9i データベース・リリース 1 (9.0.1) で導入されています。また、『Oracle Syndication Server User's and Administrator's Guide』が付属しています。

Oracle9iAS リリース 2 (9.0.2) 以上では、Oracle Syndication Server は Oracle9iAS の機能になっています。現行の Syndication Server の機能はこのプラットフォームに統合され、Syndication Server 端末は Oracle9i データベース・リリース 2 (9.2) で導入されています。

Oracle9i Syndication Server は、Oracle9iAS Portal をインストールすると、自動的にインストールされます。現行のリリースの『Oracle Syndication Server User's and Administrator's Guide』は、Oracle9iAS Portal のマニュアルとともに、Oracle9iAS リリース 2 (9.0.2) のドキュメント CD-ROM にあります。

リリース 2 (9.2) と以前のリリース間の互換性および相互運用性の問題

次の項では、互換性の問題および相互運用性の問題を示し、その問題によって発生する問題を防ぐ処置について説明します。これらの項で説明する問題は、Oracle のリリースの違いによって発生します。

- [アプリケーション](#)
- [STARTUP コマンド](#)
- [表領域およびデータ・ファイル](#)
- [データ・ディクショナリ](#)
- [スキーマ・オブジェクト](#)
- [データ型](#)

- ユーザー定義データ型
- SQL および PL/SQL
- アドバンスド・キューイング (AQ)
- プロシージャおよびパッケージ
- Oracle オプティマイザ
- Oracle9i Real Application Clusters
- データベース・セキュリティ
- データベース・バックアップおよびリカバリ
- 分散データベース
- SQL*Net、Net8 または Oracle Net
- その他の互換性および相互運用性の問題

アプリケーション

リリース 2 (9.2) の新機能を使用しない既存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、同じまたは拡張された機能をリリース 2 (9.2) で実行できます。以前のリリースへダウングレードしても、リリース 2 (9.2) のデータベースで動作しているアプリケーションが継続して動作できるようにするには、COMPATIBLE 初期化パラメータを以前のリリースに一致するように設定します。

ただし、この COMPATIBLE 初期化パラメータは、ディスクの書式を変更するリリース 2 (9.2) の機能の使用を制限するのみで、リリース 2 (9.2) のその他の機能の使用は制限しません。したがって、COMPATIBLE 初期化パラメータを 9.2.0 未満に設定しても、データベースを以前のリリースへダウングレードした場合に、リリース 2 (9.2) で開発されたアプリケーションが正しく動作するという保証はありません。

参照： アプリケーションのアップグレードの詳細は、第 6 章「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。

アプリケーションの一般的な互換性の問題および相互運用性の問題

この項では、アプリケーションの一般的な互換性の問題および相互運用性の問題について説明します。

VARCHAR2、CHAR および RAW の最大サイズの変更 VARCHAR2、CHAR または RAW データ型を使用している Oracle7 クライアントでは、アプリケーションでバッファのオーバーフロー・エラーが発生する場合があります。リリース 8.0 以上では VARCHAR2 データ型の最大サイズが 2000 から 4000 へ増加され、CHAR データ型および RAW データ型の最大サイズが 255 から 2000 へ増加されたため、このエラーが発生します。

この問題が発生したクライアントでは、アプリケーションが大きいバッファ・サイズを受け入れることができるように修正するか、または問題のある問合せに SUBSTR() 演算子を使用して、バッファの戻りサイズをアプリケーションが処理できる長さに制限することができます。

次の例では、SIZE_TAB.SIZE_COL 列は VARCHAR(80) です。

```
SQL> CREATE VIEW v1 AS SELECT
      LPAD(' ',40-length(size_tab.size_col)/2,' ') size_col
    FROM size_tab;
Statement processed.
```

```
SQL> DESC v1
```

Column Name	Null?	Type
SIZE_COL		VARCHAR2(4000)

```
SQL> DROP VIEW v1;
View dropped.
```

```
SQL> CREATE VIEW v1 AS SELECT
      SUBSTR(lpad(' ',40-length(size_tab.size_col)/2,' '), 2000) size_col
    FROM size_tab;
Statement processed.
```

```
SQL> DESC v1;
```

Column Name	Null?	Type
SIZE_COL		VARCHAR2(2001)

アプリケーションにアクセスされる索引構成表 アプリケーションにアクセスされる表が、通常の表から索引構成表に変更された場合、アプリケーションも変更を必要とすることがあります。この変更が必要であるかどうかは、アプリケーションが物理 ROWID またはユニバーサル ROWID (UROWID) のどちらを使用しているかによって決定されます。

アプリケーションが変更を必要とするかどうかは、ROWID 値をバインドまたは定義するためにアプリケーションが使用しているホスト変数の種類によって異なります。

- アプリケーションがリリース 8.0 以上の OCI ROWID 記述子 (Pro*C の OCIROWID および Pro*COBOL の SQL-ROWID) を使用している場合、アプリケーションは変更しなくても正常に機能します。
- アプリケーションがホスト変数に対して常に DESCRIBE を実行している場合、アプリケーションは変更しなくても正常に機能します。アプリケーションは SQLT_RDD という新しいデータ型に対応できることを確認してください。

- アプリケーションが `SQLT_RID` ホスト変数を使用している場合、`VARCHAR` ホスト変数または `ROWID` 記述子を使用するようにアプリケーションを書きなおす必要があります。書き直す場合、`ROWID` 記述子の方が適切です。
- アプリケーションが `CHARACTER` ホスト変数を使用している場合、動作はそのホスト変数のサイズによって異なります。そのサイズによって主キーを保持でき、変数が可変長文字列である場合、アプリケーションは変更しなくても正常に機能します。ただし、アプリケーションが、固定サイズで 18 文字列を使用している場合、より長い可変文字列または `OCI` 記述子を使用するようにアプリケーションを変更する必要があります。

`UROWID` を使用するアプリケーションでは、`VARCHAR` ホスト変数が `ROWID` を保持できるほど大きくない場合があります。その場合、変数の最大サイズを増やすか、または `OCI ROWID` 記述子を使用するようにアプリケーションを変更します。`OCI ROWID` 記述子是不透明であり、自動的にサイズを変更するため、`OCI ROWID` 記述子を使用する方が適切です。

ANALYZE TABLE VALIDATE STRUCTURE 文の動作の変更 リリース 8.1 以上では、`ANALYZE TABLE VALIDATE STRUCTURE` 文は最初に発見されたエラーで実行を停止しません。この変更を考慮して、この動作に基づくアプリケーションを変更します。

OCI アプリケーション

この項では、OCI アプリケーションに関する互換性の問題および相互運用性の問題について説明します。

参照： 詳細は、『Oracle Call Interface プログラマーズ・ガイド』を参照してください。

共有構造および相互運用性 共有構造は、Oracle8i のライブラリとリンクした Oracle7 クライアントではサポートされていません。共有構造を利用するには、アプリケーションを Oracle8i 以上の OCI で作成する必要があります。また、Oracle8i 以上の Oracle データベース・サーバーと通信する必要があります。

Oracle8 の Oracle データベース・サーバーにアクセスしている Oracle8i の OCI クライアントは、共有構造の一部のみを利用できます。クライアントおよび Oracle データベース・サーバーの両方が Oracle8 以前の場合、共有構造はサポートされません。

スレッド・セーフティ ORLON コールおよび OLON コールは、Oracle8 および Oracle8i ではサポートされていません。シングル・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。

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、Oracle9i のリンク・ラインは Oracle7 のリンク・ラインとは異なります。Oracle9i の OCI アプリケーションをコンパイルする場合の Oracle9i のリンク・ラインの使用例については、`ORACLE_HOME/rdbms/demo/demo_rdbms.mk` ファイルを参照してください。

Oracle7 クライアント Oracle7 クライアントは、Oracle7 と Oracle9i のコールを組み合わせ、Oracle9i OCI の使用を選択できます。どちらのコールを使用するかによって機能性の程度が異なります。暗号化 API およびパスワード・リセット・コールも同様に、単独で使用できます。次の機能を使用可能にするには、処理された文のすべてのフェーズに Oracle9i の OCI を使用します。

- フェイルオーバー
- プリフェッチ
- ピギーバック方式のコミットまたは取消し
- クライアント側の変換

Oracle7 コードと Oracle9i のコードを組み合わせる場合、Oracle7 クライアントは Oracle9i コールを使用してログインする必要があります。

文の実行に対するバッチ・エラー・モードの使用 リリース 8.1 以上では、OCI アプリケーションは、`OCIStmtExecute` を使用して DML 配列を実行するときにバッチ・エラー・モードを使用できます。バッチ・エラー・モードを使用するには、OCI ライブラリおよびサーバー・ライブラリがリリース 8.1 以上である必要があります。

`OCI_BATCH_ERRORS` にモード・パラメータを設定し、この機能に必要な新規コードを追加することによって、既存のアプリケーションを変更してバッチ・エラー・モードを使用できます。その後、リリース 8.1 クライアント・ライブラリに、変更したアプリケーションを再コンパイルおよび再リンクします。

クライアントへの通知サポート リリース 8.1 以上では、パブリッシュ / サブスクライブ・インタフェースを使用する OCI アプリケーションでクライアント通知がサポートされます。クライアント通知によって、アプリケーションはデータベース・イベント・パブリッシュおよびアドバンスト・キューイング機能を利用できます。クライアント通知機能を使用するには、クライアント・アプリケーションがリリース 8.1 以上のクライアント・ライブラリとリンクしている必要があります。

アドバンスト・キューイング・オプションでの LISTEN コール・サポート リリース 8.1 以上では、OCI アプリケーションで LISTEN コールがサポートされます。LISTEN コールはアドバンスト・キューイング・オプションで使用でき、一連のキューに対してメッセージがないか監視するために使用されます。LISTEN コールを使用するには、クライアント・アプリケーションがリリース 8.1 以上のクライアント・ライブラリとリンクしている必要があります。

プリコンパイラ・アプリケーション

この項では、プリコンパイラ・アプリケーションに関する互換性の問題および相互運用性の問題について説明します。

参照： 詳細は、『Pro*C/C++ Precompiler プログラマーズ・ガイド』および『Pro*COBOL Precompiler プログラマーズ・ガイド』を参照してください。

Pro*C/C++ での SYSDBA の権限による接続 Pro*C/C++ で CONNECT 文を発行する場合、SYSDBA 権限はデフォルトで使用されなくなりました。リリース 8.0 では、次の CONNECT 文を使用して Pro*C/C++ の SYSDBA 権限と接続しました。

```
EXEC SQL CONNECT :sys IDENTIFIED BY :sys_passwd;
```

リリース 8.1 以上では、次の CONNECT 文を発行して Pro*C/C++ で SYSDBA の権限と接続します。

```
EXEC SQL CONNECT :sys IDENTIFIED BY :sys_passwd IN SYSDBA MODE;
```

Pro*COBOL での SYSDBA 権限による接続 Pro*COBOL で CONNECT 文を発行する場合、SYSDBA 権限はデフォルトで使用されなくなりました。リリース 8.0 では、次の CONNECT 文を使用して SYSDBA 権限と接続しました。

```
EXEC SQL  
    CONNECT :sys IDENTIFIED BY :SYS-PASSWD  
END-EXEC.
```

リリース 8.1 以上では、次の CONNECT 文を発行して SYSDBA の権限と接続します。

```
EXEC SQL  
    CONNECT :sys IDENTIFIED BY :SYS-PASSWD IN SYSDBA MODE  
END-EXEC.
```

Oracle8 および Oracle8i での ADA のサポート Pro*ADA 製品は、Oracle7 リリース 7.3 ではサポートされませんでした。Pro*ADA は、多くの新機能を持つ SQL*Module for ADA 8.1 の最新リリースへアップグレードできます。ただし、SQL*Module for ADA 8.1 は、オブジェクトをサポートしていません。

PL/SQL の下位互換性およびプリコンパイラ PLSQL_V2_COMPATIBILITY 下位互換性動作は、次のように DBMS プリコンパイラ・コマンドライン・オプション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=Oracle7
```

PL/SQL アプリケーション

この項では、PL/SQL のアプリケーションの互換性の問題および相互運用性の問題について説明します。

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

SQL 分析の統合 PL/SQL プログラムでの SQL 文の構文およびセマンティクスの分析は、現在、SQL エンジンに統合されています。その結果、SQL*Plus または OCI を介して使用可能な SQL の新機能は、PL/SQL でも使用可能です。

Oracle9i での SQL 文の構文およびセマンティクスの分析は、以前のリリースよりわずかに精度が上がっています。PL/SQL では、無効な SQL 文に対して実行時に例外を発生させるのではなく、コンパイル中に SQL 文の追加エラーを表示します。その結果、以前のリリースでは正常にコンパイルされた PL/SQL プログラムで、接頭辞が「PL/SQL: ORA-」のコンパイル・エラーが表示されることがあります。新しいエラー・メッセージでは、プログラムをコンパイルする前に修正する必要がある SQL 文の問題が示されます。

高精度の新しいチェックを満たすように SQL 文を直接変更できない場合のために、オラクル社では、PL/SQL コードの Oracle9i への移行を一時的に補助するイベントを提供しています。

```
ALTER SESSION SET events = '10933 trace name context forever, level 512';
```

このイベントは、一時的な移行の補助としてのみ提供されます。このイベントを長期間にわたって、使用しないでください。また、このイベントは Oracle の次の主要リリースではサポートされません。

リリース 8.1.7 からアップグレードし、このイベントがパラメータ・ファイル内に存在する場合、一時的な解決策として、このイベントの発生条件を event = '10933 trace name context forever, level 512' から event = '10933 trace name context forever, level 1024' に変更します。

仕様部と本体のファンクションまたはプロシージャに対するパラメータのデフォルト値の不一致 以前のリリースでは、PL/SQL を実行するとこのエラーが暗黙的に無視され、(本体の値が異なる場合も無視して) 仕様部に指定されたデフォルト値が使用されていました。また、仕様部にデフォルト値が指定されておらず、本体にデフォルト値が指定されている場合、本体のデフォルト値は無視されます。

Oracle9i では、PL/SQL を実行すると、このような不一致がエラーとしてフラグ付けされます。このようなエラーが報告される場合は、将来発生する可能性のあるエラーを回避するために、コードを修正することをお勧めします。

PL/SQL コードをすぐに変更できない場合、Oracle は、古いコンパイラの動作を一時的にリストアするためのイベントを提供します。

```
ALTER SESSION SET events = '10932 trace name context level 32768';
```

このイベントは、一時的な移行の補助としてのみ提供されます。このイベントを長期間にわたり、使用しないでください。また、このイベントは Oracle の次の主要リリースではサポートされません。

互換性およびオブジェクト型 Oracle9i では、NOT FINAL または NOT INSTANTIABLE であるオブジェクト型、サブタイプあるいは SQLJ 型は、以前のリリースの Oracle の PL/SQL プログラムからは参照できません。このようなプログラムはコンパイルに失敗し、エラーが表示されます。

PL/SQL V2 互換モード PL/SQL V2 互換モードは、リリース 8.0 以上の PL/SQL で使用できます。このモードは、PLSQL_V2_COMPATIBILITY 初期化パラメータによって使用可能になります。

PL/SQL V2 互換モードは、次のいずれかの方法で設定します。

- 次の行を初期化パラメータ・ファイルに追加します。

```
PLSQL_V2_COMPATIBILITY = true
```

- 次の SQL 文を発行します。

```
ALTER SYSTEM SET PLSQL_V2_COMPATIBILITY = true;
```

- 次の SQL 文を発行します。

```
ALTER SESSION SET PLSQL_V2_COMPATIBILITY = true;
```

PLSQL_V2_COMPATIBILITY 初期化パラメータによって、次のような状況でリリース 8.0 以上の PL/SQL と PL/SQL V2 の間に互換性が得られます。

- 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。リリース 8.0 以上の PL/SQL では、ソース内で型を参照する前に必ず型を定義し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、リリース 8.0 以上の PL/SQL は PL/SQL V2 と同様に動作します。

- 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
```

このような不正な構文が見つかり、リリース 8.0 以上の PL/SQL の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、リリース 8.0 以上の PL/SQL 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または削除できません。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。リリース 8.0 以上の PL/SQL では、IN パラメータの読み込み専用セマンティクスを正しく施行し、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、リリース 8.0 以上の PL/SQL 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、リリース 8.0 以上の PL/SQL 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、リリース 8.0 以上の PL/SQL は PL/SQL V2 と同様に動作し、このような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ことを許可します（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しないでください。リリース 8.0 以上の PL/SQL では、OUT パラメータを式のコンテキストに使用できません。ただし、PL/SQL V2 の互換モードを使用可能にすると、リリース 8.0 以上の PL/SQL 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。リリース 8.0 以上の PL/SQL 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、リリース 8.0 以上の PL/SQL は PL/SQL V2 と同様に動作します。

キーワード動作の違い: Oracle7 と Oracle8 以上 Oracle7 および Oracle8 以上のいずれにも含まれる様々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、Oracle7 および Oracle8 以上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。

表 5-2 キーワード動作の違い

キーワード	Oracle8 以上の動作	Oracle7 の動作
CHARACTER、COMMIT、 DEC、FALSE、INT、 NUMERIC、REAL、 SAVEPOINT、TRUE	エラーの生成 (ORA-06550 および PLS-00222)	エラーの生成 (ORA-06552 および PLS-00222)

STARTUP コマンド

この項では、STARTUP コマンドに関する互換性の問題および相互運用性の問題について説明します。

デフォルトで選択されるパラメータ・ファイルの変更

PFILE オプションなしで STARTUP コマンドを発行すると、デフォルトのパラメータ・ファイルを使用してインスタンスが起動されます。Oracle9i では、サーバー・パラメータ・ファイルを簡単に使用できるように、デフォルトのパラメータ・ファイルを選択する場合の検索基準が変更されています。

以前のリリースの Oracle では、STARTUP コマンドは、ORACLE_HOME/dbs/initSID.ora という名前の初期化パラメータ・ファイルを検索しました。この場合の SID はインスタンス名です。

Oracle9i では、次のようにデフォルトのパラメータ・ファイルが選択されます。

- STARTUP コマンドは、ORACLE_HOME/dbs/spfileSID.ora という名前のサーバー・パラメータ・ファイルを検索します。この場合の SID はインスタンス名です。
- STARTUP コマンドは、ORACLE_HOME/dbs/spfile.ora という名前のサーバー・パラメータ・ファイルを検索します。
- STARTUP コマンドがサーバー・パラメータ・ファイルを検索できない場合は、以前のリリースの STARTUP コマンドの動作と同様に、デフォルトで ORACLE_HOME/dbs/initSID.ora という名前の初期化パラメータ・ファイルを検索します。

参照： サーバー・パラメータ・ファイルの詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

表領域およびデータ・ファイル

この項では、表領域およびデータ・ファイルに関する互換性の問題および相互運用性の問題について説明します。

CREATE TABLESPACE: 新しい動作

Oracle8i では、CREATE TABLESPACE 文で EXTENT MANAGEMENT 句が指定されていない場合、作成される表領域のデフォルト形式は、ディクショナリ管理になります。

Oracle9i では、EXTENT MANAGEMENT 句のデフォルトは、COMPATIBLE 初期化パラメータの設定によって異なります。

- COMPATIBLE が 8.1.x に設定されている場合、Oracle8i の動作が維持されます。この場合、表領域はディクショナリ管理として作成されます。
- COMPATIBLE が 9.0.0 以上に設定されている場合、デフォルトはローカル管理になります。デフォルトの STORAGE 句が解析され、この表領域に AUTOALLOCATE 割当てポリシーまたは UNIFORM 割当てポリシーを使用するかどうかが決まります。

また、永続的なローカル管理表領域をユーザーの一時表領域として割り当てることが禁止されています。Oracle8i では、一時セグメントを表領域に作成する必要がある場合にのみエラーが発生していました。

デフォルトの一時表領域

データベースに対してデフォルトの一時表領域を使用することをお薦めします。将来のリリースでは、デフォルトの一時表領域が必須になります。CREATE TEMPORARY TABLESPACE 文を使用して、デフォルトの一時表領域を作成する必要があります。

UNDO 表領域

Oracle インスタンスは、次のいずれかの UNDO 領域管理モードで実行できます。

- 自動 UNDO 管理モード
- 手動 UNDO 管理モード

同じデータベースのすべてのインスタンスは、同じ UNDO 領域管理モードで実行する必要があります。

COMPATIBLE 初期化パラメータは、UNDO 領域の管理方法に影響します。COMPATIBLE が 9.0.0 以上に設定されている場合のみ、自動 UNDO 管理モードを使用できます。

UNDO_MANAGEMENT 初期化パラメータが指定されていない場合、または COMPATIBLE が 9.0.0 未満に設定されている場合、インスタンスは手動 UNDO 管理モードで起動されません。

COMPATIBLE が 9.0.0 以上に設定されている手動 UNDO 管理モードでは、表領域に対して CREATE、ALTER および DROP 操作を行うことができます。ロールバック・セグメントは、UNDO 表領域と共存できます。つまり、自動 UNDO 管理モードで実行中にロールバック・セグメントが存在し、手動 UNDO 管理モードで実行中に UNDO 表領域が存在できます。インスタンスが自動 UNDO 管理モードで実行されていないかぎり、UNDO 表領域はオンラインにできません。

自動 UNDO 管理モードでは、DROP ROLLBACK SEGMENT 操作を行うことができます。ロールバック・セグメントはオンラインにできません。

参照： UNDO 領域の管理については、『Oracle9i データベース管理者ガイド』を参照してください。

トランスポートブル表領域

2 つのデータベース間で表領域を移動させる場合、互換性の問題が発生します。

参照： 互換性の問題の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

一時ファイル

リリース 8.1 では、一時ファイルが導入されました。一時ファイルの情報は、データ・ファイルの情報とは異なり、静的データ・ディクショナリ・ビューおよび動的パフォーマンス・ビューにあります。一時ファイルの情報については、DBA_TEMP_FILES 静的データ・ディクショナリ・ビューおよび次に示す動的パフォーマンス・ビューを参照してください。

- V\$tempfile
- V\$temp_extent_map
- V\$temp_extent_pool
- V\$temp_space_header
- V\$tempstat
- V\$temp_ping

Oracle は、データ・ファイルと一時ファイルの両方に自動的に番号を割り当てます。2 つのデータ・ファイル間では同じ番号を共有できません。同様に、一時ファイル間でも同じ番号は共有できません。ただし、一時ファイルとデータ・ファイルの間では同じ番号を共有できます。

参照： 一時ファイルの詳細は、『Oracle9i SQL リファレンス』を参照してください。

データ・ディクショナリ

この項では、データ・ディクショナリの変更によって発生する互換性の問題および相互運用性の問題について説明します。

参照： 廃止されたディクショナリ・ビューおよび非推奨のディクショナリ・ビューの詳細は、[付録 A「初期化パラメータおよびデータ・ディクショナリの変更」](#)を参照してください。

データ・ディクショナリの保護

Oracle8 で導入されたデータ・ディクショナリの保護のメカニズムによって、SYS スキーマにユーザー表を作成し、'ANY' 権限を使用してその表にアクセスするアプリケーションに問題が発生する場合があります。たとえば、ユーザーは、AUD\$ 表内の監査レコードを消去する DELETE 文を使用するために DELETE CATALOG ROLE を所有する必要があります。

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 という初期化パラメータを使用します。ただし、このパラメータは暫定的にのみ使用されます。

アプリケーションから SYSDBA 権限なしでユーザー SYS に接続することは避けてください。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 DBA 権限を付与し、SYSDBA 権限を持つユーザーとしてデータベースに接続して SYS スキーマに接続できるようにします。

Oracle9i では、ユーザーに SELECT ANY DICTIONARY 権限を付与することができます。この権限を持つユーザーは、O7_DICTIONARY_ACCESSIBILITY の設定にかかわらず、SYS スキーマのオブジェクトにアクセスできます。

廃止されたデータ・ディクショナリ・ビュー

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 Oracle7 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、リリース 8.0 以上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を、データベース・ツールおよびアプリケーションから削除してください。

スキーマ・オブジェクト

この項では、スキーマ・オブジェクトに関する互換性の問題および相互運用性の問題について説明します。

ビットマップ索引の保護

リリース 8.1 より前のリリースでは、特定の SQL 文を発行することによって、ビットマップ索引が無効になることがありました。ビットマップ索引が無効になる最も一般的な原因は、次に示すタイプの文によるものです。

- あらかじめ主キーを持たない既存の表に主キーを設定する ALTER TABLE 文
 - あらかじめ NOT NULL 制約を持たない列に NOT NULL を設定する ALTER TABLE 文
- Oracle9i では、このように意図せずに索引が無効になることはありません。

データ型

この項では、データ型に関する互換性の問題および相互運用性の問題について説明します。

日時データ型および期間データ型

データベースが Oracle9i にアップグレードされると、データベースのタイムゾーンは、環境変数 ORA_SDTZ のタイムゾーンに設定されます。ORA_SDTZ が設定されていない場合、データベースのタイムゾーンはオペレーティング・システムのタイムゾーンに設定されます。オペレーティング・システムのタイムゾーンが設定されていない場合、または有効でない場合、データベースのタイムゾーンはデフォルトの協定世界時 (UTC) に設定されます。

時間部分を持つ以前の Oracle の DATE データは、秒の小数部分をサポートするために TIMESTAMP へ、またはレガシー・データを再書き込みせずに秒の小数部分およびタイムゾーンの調整をサポートするために TIMESTAMP WITH LOCAL TIME ZONE へ移行することができます。DATE 列を TIMESTAMP 列または TIMESTAMP WITH LOCAL TIME ZONE に変更するには、ALTER TABLE 文を明示的に発行する必要があります。

ラージ・オブジェクト (LOB)

この項では、LOB に関する互換性の問題および相互運用性の問題について説明します。

CLOB および NCLOB の可変幅キャラクタ・セット リリース 8.0 では、データベース・キャラクタ・セットが可変幅の場合、SYSTEM 以外のユーザーは CLOB または NCLOB のデータ型がある表を作成できませんでした。リリース 8.1 以上では、可変幅キャラクタ・セットを持つ表で CLOB および NCLOB データ型をサポートします。データは UCS2 (2 バイトの固定値ユニコード) として格納されます。

LOB 索引句 LOB 格納用の表領域以外の表領域で、LOB 索引データを格納するために LOB 索引句を使用した場合、索引データは、LOB と同じ表領域に属するように再配置されます。これには、リリース 8.1 以上で次のいずれかの操作を実行する必要があります。

- LOB についてエクスポート / インポートを実行します。
- LOB をパーティション表へ移動します。

Oracle7 から Oracle9i へアップグレードするために、エクスポート / インポートを使用した場合、索引データは移行中に自動的に再配置されます。ただし、移行ユーティリティまたは Database Upgrade Assistant を使用した場合、索引データは再配置されません。

また、リリース 8.1 以上で新しい表を作成し、非パーティション表の LOB 索引のために表領域を指定した場合、表領域の指定は無視され、LOB 索引は LOB と同じ表領域に配置されます。

LOB 索引の格納を確認するには、SYSDBA 権限を持つユーザーとして接続し、次の SQL 文を発行します。

```
SELECT index_name, index_type, tablespace_name
FROM dba_indexes
WHERE index_type = 'LOB';
```

動的パフォーマンス・ビューの日付列

Oracle7 では、動的パフォーマンス・ビューのすべての日付列は、MM/DD/YY HH24:MI:SS という書式の VARCHAR2 (20) 文字列です。リリース 8.0 以上では、すべての日付列は、DATE データ型を使用する実際の DATE 列です。以前の VARCHAR2 (20) 文字列と比較して、DATE データ型には次のようなメリットがあります。

- すべての日付列が DATE データ型であるため、一貫性を確立します。
- SQL および PL/SQL では、日付算術（ソートを含む）を簡易化します。
- NLS_DATE_FORMAT を使用して、日付書式を設定します。
- NLS_DATE_FORMAT を MM/DD/YY HH24:MI:SS に設定することにより、以前の書式で日付を見せることができます。
- 2 桁の年数値の使用を避けることによって、西暦 2000 年以降の問題を回避することができます。

注意： Oracle7 は、動的パフォーマンス・ビューの VARCHAR (20) データ型を使用して日付を表示しますが、完全には西暦 2000 年問題を避けることができません。Oracle7 は、REDO ログ・ファイルと制御ファイルにある最も近い秒にあわせて時間を格納します。

Oracle ROWID

この項では、ROWID に関する互換性の問題および相互運用性の問題について説明します。

UROWID データ型 リリース 8.1 では、UROWID (ユニバーサル ROWID) データ型が導入されました。リリース 8.1 より前のクライアントが UROWID データ型の列にアクセスする場合に使用できるのは、キャラクタ・ホスト変数のみです。つまり、他の種類の変数はサポートされません。

新規の物理 ROWID データ型書式 リリース 8.0 で、新規の物理 ROWID 書式が導入されました。列またはアプリケーション・コードに格納された物理 ROWID を使用する場合、以前の物理 ROWID は無効であるため、変換する必要があります。

参照： 新規の物理 ROWID 書式の詳細は、D-33 ページの「[物理 ROWID の移行について](#)」を参照してください。

AL24UTFSS キャラクタ・セットの非サポート

AL24UTFSS Unicode キャラクタ・セットは、Oracle9i ではサポートされていません。AL24UTFSS は、Unicode 1.1 規格に基づく UTF-8 コード体系をサポートする Unicode キャラクタ・セットとして Oracle7 で導入されましたが、現在は廃止されています。Oracle9i では、Unicode キャラクタ・セット AL32UTF8 および UTF8 が、Unicode 3.1 規格に基づく Unicode の拡張機能を備えています。

既存の AL24UTFSS データベースの移行パスは、Oracle9i へアップグレードする前に、データベース・キャラクタ・セットを UTF8 にアップグレードすることです。新しいデータベース・キャラクタ・セットへ移行する場合は、既存のデータベース・キャラクタ・セットを UTF8 へ移行する前に、キャラクタ・セット・スキャナを使用して、データを分析することをお勧めします。

参照： キャラクタ・セット・スキャナの詳細は、『Oracle9i Database グローバリゼーション・サポート・ガイド』を参照してください。

NCHAR および NLS の使用

Oracle8 および Oracle8i では、特定の列、属性、PL/SQL 変数、パラメータおよび戻り値に各国語キャラクタ・セット (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かぎり、NCHAR および NLS の使用はほとんどが透過的に処理され、Oracle8 および Oracle8i の他の機能には影響しません。例外としては、PROPS\$ または VALUE\$ 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。

NCHAR および NLS に伴う移行問題 PROPS\$ デictionary 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、VALUE\$ 列にデータベース・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、VALUE\$ 列に各国語キャラクタ・セットの名前があります。

リリース 7.3 と比較すると、様々なビューに、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\$ はベース表の 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 環境変数および互換性 次のように、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 の環境に環境変数 ORA_NLS32 を設定します。
- リリース 8.0 以上の環境に環境変数 ORA_NLS33 を設定します。

クライアントに正しい NLS キャラクタ・セットの環境変数があることを確認します。リリース 7.3 の NLS コードがリリース 8.0 以上のキャラクタ・セットをロードしようとすると、エラーが発生します。

ユーザー定義データ型

この項では、ユーザー定義データ型に関する互換性の問題および相互運用性の問題について説明します。

型進化

型進化には、COMPATIBLE 初期化パラメータを 9.0.0 以上に設定する必要があるため、以前のリリースの PL/SQL を使用しているクライアントは、進化した型にはアクセスできません。

サブタイプおよび NOT FINAL 型

リリース 8.1 以下で作成された型は、FINAL 型とみなされます。そのため、Oracle9i ではサブタイプとして使用できません。ただし、ALTER 文を明示的に使用して、型を NOT FINAL に変更することができます。

COMPATIBLE 初期化パラメータが 9.0.0 未満に設定されている場合、サブタイプは作成できません。また、NOT INSTANTIABLE 型および NOT FINAL 型も作成できません。したがって、サブ表、サブビューおよび代替可能な列も使用できません。

リリース 8.1 クライアントによるリリース 1 (9.0.1) 以上のサーバーへのアクセス NOT FINAL 型のデータを含む転送はエラーを戻します。型がサーバー上で NOT FINAL 型に変更されている場合、リリース 8.1 クライアントは、リリース 1 (9.0.1) 以上のサーバーにアクセスできません。

リリース 1 (9.0.1) 以上のクライアントによるリリース 8.1 サーバーへのアクセス リリース 8.1 サーバーには FINAL 型のみが存在するため、エラーは発生しません。

ユーザー定義データ型の新しい書式

リリース 8.1 では、ユーザー定義データ型に新しい書式が導入されました。新しい書式によって、リリース 8.0 で使用されていた書式よりもパフォーマンスが大幅に向上します。リリース 8.0 のユーザー定義データ型をリリース 8.1 以上のデータベースで使用しても互換性の問題はありませぬ。ただし、データベースは、新しい書式によるパフォーマンスの向上は得られません。

リリース 8.1 以上のクライアントによるリリース 8.0 のユーザー定義データ型へのアクセス ユーザー定義データ型の書式は、クライアントとサーバー間の互換性のやりとりの一部として交渉されます。リリース 8.0 サーバーを使用している場合、リリース 8.1 以上のクライアントはデータベースにアクセスできますが、リリース 8.0 に設定されます。

リリース 8.0 クライアントによるリリース 8.1 以上のユーザー定義データ型へのアクセス リリース 8.0 クライアントがリリース 8.1 以上のユーザー定義データ型を持つサーバーにアクセスする場合は、データベースがユーザー定義データ型をリリース 8.0 の書式に変換します。したがって、リリース 8.0 クライアントはデータにアクセスできますが、パフォーマンスの向上は得られないことがあります。

ネストした表

リリース 8.0 クライアントは、リリース 8.1 以上のネストした表の次の機能はサポートしません。

- コレクション・ロケータ
- 索引構成表にあるネストした表データの記憶域を含む、コレクション列のユーザー指定記憶域

したがって、リリース 8.0 のクライアントがリリース 8.1 以上のサーバーにアクセスしようとし、ネストした表がロケータとして戻されるように指定されている場合、またはネストした表の記憶域がユーザー指定の場合、非互換エラーとなりアクセスできません。

LOB として格納される VARRAY

リリース 8.0 のクライアントは、VARRAY を LOB として格納する記憶域パラメータの指定をサポートしません。したがって、リリース 8.0 のクライアントが、VARRAY を LOB として格納する記憶域パラメータの指定のあるリリース 8.1 以上のサーバーにアクセスしようとすると、非互換エラーとなりアクセスできません。

SQL および PL/SQL

この項では、SQL および PL/SQL に関する、互換性の問題および相互運用性の問題について説明します。

参照： SQL および PL/SQL の詳細は、『Oracle9i SQL リファレンス』および『PL/SQL 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。

GREATEST_LB、LEAST_UB および TO_LABEL ファンクションの非サポート

リリース 8.1 以上では、組込み PL/SQL ファンクション GREATEST_LB、LEAST_UB および TO_LABEL はサポートされません。

PL/SQL のネイティブ動的 SQL

次の項では、PL/SQL のネイティブ動的 SQL に関する相互運用性の問題について説明します。

サーバー側の PL/SQL リリース 8.1.0 以上の互換性レベルを持つ Oracle データベース・サーバーでは、リモート・サーバーのオブジェクトへの参照を含むネイティブ動的 SQL 文を、どの互換性レベルでも実行できます。

たとえば、次のプロシージャはネイティブ動的 SQL 文を持ち、リモートの Oracle データベース・サーバーにリンクします。

```
PROCEDURE dyn1 is
BEGIN
    EXECUTE IMMEDIATE 'insert into tab@remote_link
        values ('a', 10)';
END;
```

前述の例の *remote_link* は、リリース 7.3、8.0、8.1 などのすべての Oracle リリースへのリンクとして使用できます。

ネイティブ動的 SQL および RPC コール RPC コールを作成するクライアントのリリースにかかわらず、RPC コールのターゲットである PL/SQL プログラムはネイティブ動的 SQL を使用できます。たとえば、リリース 7.3 または 8.0 のクライアントは、リリース 8.1.0 以上の互換性レベルで Oracle データベース・サーバーに RPC コールを発行することができます。

SQL スクリプト (utlchain.sql および utlchn1.sql)

Oracle9i のインストールでは、移行された行および連鎖された行を格納する表を作成するスクリプトとして、utlchain.sql および utlchn1.sql の 2 つが提供されます。utlchn1.sql スクリプトは通常の表および索引構成表の両方で実行できますが、utlchain.sql スクリプトは通常の表でのみ実行でき、索引構成表では実行できません。

Oracle9i では、常に utlchn1.sql スクリプトを実行する必要があります。

SQL スクリプト (utlexcpt.sql および utlexpt1.sql)

Oracle9i のインストールでは、制約を使用可能にした場合の例外を格納する表を作成するスクリプトとして、utlexcpt.sql および utlexpt1.sql の 2 つが提供されます。utlexpt1.sql スクリプトは通常の表および索引構成表の両方で実行できますが、utlexcpt.sql スクリプトは通常の表でのみ実行でき、索引構成表では実行できません。

Oracle9i では、常に utlexpt1.sql スクリプトを実行する必要があります。

AS 副問合せを持つパラレル CREATE TABLE 文の動作変更

リリース 8.0 以上で、AS 副問合せを持つ CREATE TABLE 文に PARALLEL 句を使用すると、INITIAL 記憶域パラメータは無視され、NEXT 記憶域パラメータが使用されます。Oracle7 では、INITIAL 記憶域パラメータは無視されません。

次に、SQL 文の例を示します。

```
CREATE TABLE tb_2 STORAGE (INITIAL 1M NEXT 500K)
  PARALLEL (DEGREE 2)
  AS SELECT * FROM tb_1;
```

リリース 8.0 以上では、INITIAL の値は 500KB となりますが、Oracle7 では 1MB となります。

アドバンスト・キューイング (AQ)

この項では、AQ の互換性の問題および相互運用性の問題について説明します。

参照： AQ の詳細は、『Oracle9i アプリケーション開発者ガイド - アドバンスト・キューイング』を参照してください。次の項では、新しい AQ 機能の互換性および相互運用性のみ説明します。使用方法の詳細は、『Oracle9i アプリケーション開発者ガイド - アドバンスト・キューイング』を参照してください。

キュー・レベル権限およびシステム・レベル権限

キュー・レベル権限およびシステム・レベル権限を使用するには、キュー表が 8.1.0 以上の互換性レベルである必要があります。特に、DBMS_AQADM パッケージにある次のプロシージャを使用してキュー・レベル権限を付与するには、キュー表の互換性レベルが 8.1.0 以上である必要があります。

- GRANT_QUEUE_PRIVILEGE
- REVOKE_QUEUE_PRIVILEGE

相互運用性および送信者の ID 列

リリース 8.1 以上では、送信者の ID は付加的な属性としてメッセージ・プロパティにマップされます。リリース 8.0 とリリース 1 (9.0.1) 以上のデータベース間で通信する場合、新しい属性は無視されます。

OCI のアプリケーション用に、メッセージ・プロパティの記述子の新しい属性として送信者の ID 属性を利用できます。リリース 8.1 以上の OCI クライアントは、メッセージ・プロパティをサーバーへ送信またはサーバーから受信するために新しい RPC コードを使用します。

ルールベースのサブスクリプション

DBMS_AQADM.MIGRATE_QUEUE_TABLE プロシージャを使用してリリース 8.0 からリリース 8.1 以上へキュー表を移行する場合は、既存のサブスクライバは NULL ルールを持つサブスクライバに自動的にアップグレードされます。

プロシージャおよびパッケージ

この項では、プロシージャおよびパッケージに関する互換性の問題および相互運用性の問題について説明します。

参照： パッケージの詳細は、『Oracle9i PL/SQL パッケージ・プロシージャおよびタイプ・リファレンス』を参照してください。

SET_SESSION_LONGOPS プロシージャの構文変更

リリース 8.0 では、DBMS_APPLICATION_INFO.SET_SESSION_LONGOPS プロシージャが変更されました。新しい構文の詳細は、dbmsapin.sql ファイルを参照してください。ご使用のアプリケーションがこのプロシージャを使用している場合は、それに応じてアプリケーションを変更します。

Oracle オプティマイザ

Oracle9i には、以前のリリースでは新規またはデフォルトで使用不可であった多くのオプティマイザ拡張機能が含まれています。

したがって、既存のアプリケーションを Oracle9i へアップグレードすると、実行計画が大幅に変更される可能性があります。長期間使用しているアプリケーションの場合、動作の変更がリスクを伴う場合があります。OPTIMIZER_FEATURES_ENABLE 初期化パラメータを使用すると、実行計画の変更を最小化できます。

このパラメータの値をリリース 8.1.7 などの以前のリリースに設定すると、Oracle は、そのリリースのデフォルトで使用可能なオプティマイザ機能のみを使用します。これによって、そのリリースからアップグレードする場合に実行計画が変更される可能性が減ります。

Oracle のプラン・スタビリティ機能を使用して、新しいリリースへアップグレードする場合に、以前の動作を保つこともできます。

Oracle9i Real Application Clusters

Oracle9i Real Application Clusters 環境で複数の Oracle リリースを使用する場合、リリースのサポートはオペレーティング・システムによって異なります。ご使用のオペレーティング・システムが、単一の Oracle9i Real Application Clusters 環境で異なるリリースの共存をサポートするかどうかについては、ご使用のオペレーティング・システム固有の Oracle マニュアルを参照してください。

PARALLEL 句の INSTANCES キーワード

リリース 8.1 以上で INSTANCES キーワードを使用することはできますが、旧リリースとは異なって解釈されます。Oracle7 およびリリース 8.0 では、次のような文の PARALLEL 句で INSTANCES キーワードを使用することができました。

- ALTER CLUSTER
- ALTER DATABASE ... RECOVER
- ALTER INDEX ... REBUILD
- ALTER TABLE
- CREATE CLUSTER
- CREATE INDEX
- CREATE TABLE

また、ヒントでも使用することができました。INSTANCES キーワードは、パラレル操作で使用される Oracle Parallel Server インスタンスの数を指定するために使用されました。

またリリース 8.1 以上では、PARALLEL 句に並列度を指定する構文が変更されています。次の例のように、PARALLEL の後に数値を置くことで簡単に並列度を指定できます。

```
ALTER TABLE emp PARALLEL 5;
```

ただし、使用できるように選択した場合、DEGREE キーワードは有効なままです。前述の構文は次の文と同等です。

```
ALTER TABLE emp PARALLEL (DEGREE 5 INSTANCES 1);
```

構文にかかわらず、パラレル操作で使用される問合せスレッド数と同じ数値を指定します。いずれの構文も、問合せを実行するために必要なインスタンス数に影響することはありません。いくつのインスタンスを使用するかは、使用可能なインスタンス数および各インスタンスのロードに基づいてシステムが決定します。したがって、どの構文でも結果は同じとなります。

リリース 8.1 以上での INSTANCES キーワードの使用 リリース 8.1 以上では、旧構文を引き続き使用して INSTANCES および DEGREE を指定することができますが、並列度を指定する単一のキーワードとして解釈されます。したがって、リリース 8.1 以上では廃止されたコマンド構文は引き続き受け入れられますが、これまでのリリースとは解釈が異なる場合があります。表 5-3 に、廃止された構文を引き続き使用した場合に、可能な INSTANCES および DEGREE の設定を Oracle が解釈する方法を示します。表 5-3 の各列は、次のことを示します。

- **並列度**は、PARALLEL 句の DEGREE キーワードの設定を示します。
- **インスタンス**は、PARALLEL 句の INSTANCES キーワードの設定を示します。
- **8.1 並列度**は、DEGREE および INSTANCES に基づいてリリース 8.1 以上での Oracle の並列度の解釈を示します。

表 5-3 リリース 8.1 での INSTANCES キーワードの変換

並列度	インスタンス	8.1 並列度
解除または 1	解除または 1	1
x	DEFAULT	x
x	解除または 1	x
解除または 1	DEFAULT	DEFAULT
DEFAULT	y	y
解除または 1	y	y
DEFAULT	解除または 1	DEFAULT
x	y	x * y

表中の x および y は、整数を表す変数です。

キーワードが解除された場合、1 を使用します。

これらの解釈のために、リリース 8.1 以上では Oracle は次のように動作します。

- DEGREE に x、INSTANCES に 1 を設定しても、パラレル操作で使用するインスタンスが 1 つのみになるとは保証されません。
- DEGREE に 1、INSTANCES に y を設定しても、パラレル操作で 1 つのインスタンスについて使用する問合せスレッドが 1 つのみになるとは保証されません。
- DEGREE に x、INSTANCES に y を設定しても、いずれの設定も保証されません。そのかわり、Oracle は x に y を掛けた値を並列度に設定しようとします。

予期しない動作を避けるため、INSTANCES キーワードを使用しないことをお勧めします。また、PARALLEL_INSTANCE_GROUP 初期化パラメータを使用することをお勧めします。

参照： PARALLEL 句の詳細は、『Oracle9i SQL リファレンス』を、PARALLEL_INSTANCE_GROUP 初期化パラメータの詳細は、『Oracle9i データベース・リファレンス』を参照してください。

データベース・セキュリティ

この項では、データベース・セキュリティに関する互換性の問題および相互運用性の問題について説明します。

パスワード管理

Oracle7 以下のアプリケーションを Oracle8 および Oracle8i のパスワード管理に対応させるには、アプリケーションに次の変更を行います。

- Oracle8 および Oracle8i の OCI コールである OCISessionBegin() を使用してサーバーに接続します。サーバーが SUCCESS_WITH_INFO を戻す場合、パスワードが有効でまだ猶予期間にあるかどうかを確認します。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、OCIChangePassword() コールを使用してユーザーのパスワードを変更します（パスワード変更コールはオプションですが、使用することをお勧めします）。
- 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、Oracle8 および Oracle8i の OCI コールである OCIChangePassword() を使用してユーザーのパスワードを変更する必要があります。OCIChangePassword() を使用してパスワードを変更しない場合は、パスワード検証ルーチンは、新しいパスワードが古いパスワードと等しいかどうかを確認せず、等しい場合でも変更を拒否しません。

Oracle7 のアプリケーションで前述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 などの Oracle のツール製品が必要になります。

この Oracle8 および Oracle8i 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。Oracle8 および Oracle8i のサーバー・システムでパスワード期限切れ機能を実装していない場合は、Oracle7 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。デフォルトのプロファイルでは、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 / インポートされます。ユーザーの履歴表も Oracle8 および Oracle8i でインポート / エクスポートできます。

Oracle8 以上のサーバーに接続する Oracle7 以前のクライアント Oracle7 クライアントは、Oracle7 の OCI コールを使用してサーバーに接続します。このため、Oracle8 以上のパスワードの有効期限は検出できません。ただし、Oracle8 以上のその他のパスワード管理機能は、Oracle7 クライアントでも動作します。パスワード期限切れの処理も含めた Oracle8 以上のパスワード管理の全機能は、Oracle7 のログイン・コールを Oracle8 以上のログイン・コールに置き換えるのみで、Oracle7 クライアントでも動作します。

Oracle7 以下のサーバーのある Oracle8 以上のクライアント Oracle8 以上のクライアントをコード化して、Oracle7 以前のサーバーで機能させることができます。このようなクライアント用のコード例を、次に示します。

```
OCISessionBegin(...) /* call release 8.0 and higher logon OCI call */
if (SUCCESS_WITH_INFO) then
{ /* Check for password expiration and take appropriate action*/
...
OCIChangePassword(...);
...
}
```

エンタープライズ・ユーザー管理

この項では、エンタープライズ・ユーザー管理に関する互換性の問題および相互運用性の問題について説明します。この機能は、Oracle Advanced Security の機能の一部です。

注意： Oracle Security Server (OSS) のコンポーネントは、Oracle8i には存在しません。ほとんどの機能は Oracle Advanced Security に統合されています。オラクル社は、OSS から Oracle Advanced Security に移行するためのツールを提供していません。

リリース 8.1.5 とリリース 8.0 との相互運用性 リリース 8.1.5 および 8.0 のサーバーは、グローバル・ユーザーおよびロールをリリース 8.1.6 以上のサーバーと共有できません。また、リリース 8.1.5 とリリース 8.1.6 以上の間での現行のユーザーのデータベース・リンクはサポートされません。リリース 8.0 とリリース 8.1.6 以上の間での現行のユーザーのデータベース・リンクもサポートされません。

Oracle7 およびバージョン 6 リリースとの相互運用性 Oracle7 またはバージョン 6 のサーバーではグローバル・ユーザーを作成または認可できないため、これらのサーバーはグローバル・ユーザーまたはロールを Oracle8 および Oracle8i と共有できません。また、Oracle8 および Oracle8i からバージョン 6 または Oracle7 への現行のユーザーのデータベース・リンクは、サポートされません。

データベース・バックアップおよびリカバリ

この項では、データベースのバックアップおよびリカバリに関する互換性および相互運用性の問題について説明します。

Recovery Manager

参照：

- Recovery Manager に関する互換性の問題および相互運用性の問題の詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- リカバリ・カタログのアップグレードの詳細は、4-17 ページの「[リカバリ・カタログのアップグレード](#)」を参照してください。

Recovery Manager コマンド

Recovery Manager リリース 8.1 では Recovery Manager コマンドがいくつか変更されました。ただし、以前のリリースで使用されていたすべてのコマンドは、Recovery Manager リリース 8.1 以上で同様に動作します。

たとえば、CLONE コマンドは DUPLICATE コマンドに変更されていますが、CLONE コマンドは引き続き実行できます。また、ALLOCATE コマンドおよび CONNECT コマンドの CLONE オプションは、現在 AUXILIARY オプションですが、CLONE オプションは引き続き実行できます。同様に、COPY コマンドおよび SET コマンドの CLONENAME キーワードは、現在 AUXNAME に変更されていますが、CLONENAME は引き続き実行できます。

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、Oracle8 および Oracle8i 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、Oracle8 および Oracle8i では使用できません。

EBU および Recovery Manager は両方とも、EMC などのサード・パーティの記憶域サブシステムと通信するためにメディア管理言語 (MML) を使用します。EBU および Oracle7 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、Oracle8 および Oracle8i 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、Oracle7 のバックアップは Oracle8 および Oracle8i でのリストアには一般的に役に立たないため、Oracle8 および Oracle8i で記憶域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 のスクリプト言語は、EBU のスクリプト言語とは大きく異なります。

データ・ファイル・バックアップ

Oracle7 で作成されたデータ・ファイル・バックアップは、それ以上のリリースの Oracle ではリストアすることができません。ただし、移行ユーティリティの実行後に取られた Oracle7 データベースのバックアップは、Oracle9i でリストアまたはリカバリできます。EBU を使用して Oracle7 データベースのバックアップを取り、そのデータベースを Oracle9i でリストアする必要がある場合、Oracle9i でリカバリする前に、EBU を使用してデータ・ファイルをリストアする必要があります。Oracle7 データベースが、オペレーティング・システム・コマンドでディスク・ファイルにバックアップされる場合、そのディスク・ファイルは、CATALOG DATAFILECOPY コマンドを使用して、Recovery Manager に登録できます。

ファイルのバックアップを取るリリースとファイルをリカバリするリリース間の直接のアップグレード・パスが、2-3 ページの表 2-1 「アップグレード・パス」でサポートされている場合、Oracle8 以上で取られたデータ・ファイル・バックアップは、それ以上のどのリリースでもリストアおよびリカバリできます。データ・ファイルの内容が以前のリリースと互換性がある場合、Oracle8 以上のバックアップをそれ以下のリリースにリストアおよびリカバリすることもできます。

スタンバイ・データベース

スタンバイ・データベースは Oracle7 リリース 7.3 以上で動作します。次の互換性に関する制限はスタンバイ・データベースに適用されます。

- プライマリ・データベースおよびスタンバイ・データベースは同じオペレーティング・システムで実行する必要があります。また、プライマリ・データベースとスタンバイ・データベースはオペレーティング・システムの同じリリースで実行することをお勧めします。
- プライマリ・データベースおよびスタンバイ・データベースは、Oracle の同じメンテナンス・リリースを実行する必要があります。たとえば、プライマリ・データベースがリリース 8.1.6 を実行する場合、スタンバイ・データベースは、8.1.5、8.1.6、8.1.7 などの Oracle8i を実行できます。ただし、この場合、スタンバイ・データベースは Oracle7 または Oracle8 を実行することはできません。

参照： スタンバイ・データベースに関するオペレーティング・システムの要件の詳細は、オペレーティング・システム固有の Oracle マニュアルを参照してください。

ファスト・スタート・オンデマンド・ロールバックおよびファスト・スタート・パラレル・ロールバック

セッションまたはインスタンスが正常に終了しない場合、リカバリ・プロセスとして、Oracle はコミットされていないトランザクションをロールバックします。Oracle9i には、ロールバック・パフォーマンスを向上する、ファスト・スタート・オンデマンド・ロールバックおよびファスト・スタート・パラレル・ロールバックの 2 つの新機能があります。

使用不可能のトランザクションが、他のトランザクションが必要とする行に行ロックをかけている場合、ファスト・スタート・オンデマンド・ロールバックは、自動的にその新規トランザクションが必要とするデータ・ブロックをリカバリします。新規トランザクションの進行をブロックしない他のデータ・ブロックおよびトランザクションは、バックグラウンドでロールバックされます。ファスト・スタート・オンデマンド・ロールバックは、COMPATIBLE 初期化パラメータを 8.1.0 以上に設定した場合にのみ使用可能です。

ファスト・スタート・パラレル・ロールバックは、複数のサーバー・プロセスを使用して使用不可能のトランザクションをリカバリすることによって、バックグラウンドのロールバック・パフォーマンスを向上させます。COMPATIBLE 初期化パラメータがリリース 8.0 または 8.1 に設定されている場合に、ファスト・スタート・パラレル・ロールバックを使用できます。ファスト・スタート・パラレル・ロールバックは、次の条件が満たされた場合にのみ、複数のサーバー・プロセスを使用して使用不可能のトランザクションをリカバリします。

- 使用不可能の各トランザクションに 1 つ以上のプロセスを割り当てるサーバー・プロセスが十分ある。
- COMPATIBLE がリリース 8.0 に設定された場合、トランザクションは複数のサーバー・プロセスで作成される。トランザクションが 1 つのサーバー・プロセスのみで作成される場合、1 つのサーバー・プロセスのみがロールバック操作で使用されます。この制限は COMPATIBLE が 8.1.0 以上に設定された場合には適用されません。

参照： ファスト・スタート・オンデマンド・ロールバックの詳細は、『Oracle9i データベース概要』を参照してください。

REDO ログのアーカイブ

リリース 8.1 以上では、オンライン REDO ログ・ファイルを、ローカル・ディスクベースのファイルや特定のスタンバイ・データベースなど、複数のアーカイブ先にアーカイブすることができます。この機能によって、この項で説明する互換性および相互運用性の問題が発生する可能性があります。

以前アーカイブされたオンライン REDO ログの再アーカイブ リリース 8.1 より前のリリースでは、正常かつ完全にアーカイブされたオンライン REDO ログ・ファイルを再アーカイブできました。REDO ログ・ファイルを、以前、正常にアーカイブしたアーカイブ先に再アーカイブもできました。

リリース 8.1 以上では、次の制限が適用されます。

- 正常にアーカイブされたオンライン REDO ログは再アーカイブできない。
- 正常にアーカイブしたアーカイブ先に再アーカイブできない。

アーカイブ操作エラー検出動作 リリース 8.1 より前のリリースでは、エラーが検出された場合、アーカイブ操作がただちに停止し、そのエラーをアラート・ログに報告して、ユーザーにエラーの警告をしていました。

リリース 8.1 以上では、アーカイブ操作は、アーカイブ先のすべてが処理されない場合のみ、プロセスを停止します。1 つ以上のアーカイブ先でのエラーのみではアーカイブ操作は停止しません。アーカイブ操作は、すべてのアーカイブ先が処理できない場合に停止します。特に、必須のアーカイブは再試行され、アーカイブの再試行が失敗するとプロセスが強制終了します。

LogMiner

LogMiner はリリース 8.1 以上のインスタンスで動作し、次の基準を満たすすべてのデータベースの REDO ログ・ファイルを分析します。

- 分析 Oracle インスタンスと同様のデータベース・キャラクタ・セット (DBCS) を持つ。
- 分析 Oracle インスタンスと同様のハードウェア・プラットフォームで実行する。
- リリース 8.0 以上のデータベースである。

LogMiner は、REDO ログ・ファイルを分析するために、マウントされたデータベースを必要としません。ただし、REDO ログ・ファイルの内容を完全に翻訳するには、LogMiner ディクショナリ (カタログ) へアクセスする必要があります。LogMiner はそのディクショナリを使用して、内部オブジェクト識別子およびデータ型をオブジェクト名および外部データ形式に翻訳します。PL/SQL パッケージ DBMS_LOGMNR_D を使用して、データベース・ディクショナリを外部ファイルに抽出します。そのファイルは、後で REDO ログ・ファイルの分析に使用できます。ディクショナリがなければ、LogMiner は内部オブジェクト識別子を戻し、データを 16 進で表示します。

他のデータベースのアーカイブ REDO ログ・ファイルの分析 異なるデータベースの REDO ログ・ファイルを分析すると同時に、データベースのインスタンスに LogMiner を実行できます。他のデータベースのアーカイブされた REDO ログ・ファイルを分析する場合、LogMiner は次の条件を備えている必要があります。

- REDO ログ・ファイルと同じデータベースおよび同じデータベースのキャラクタ・セットで作成されるディクショナリ・ファイルへアクセスする。
- 同じシステム上である必要はないが、ログ・ファイルを生成したのと同様のハードウェア・プラットフォームで実行する。
- Oracle8 以上のリカバリに適用される REDO ログ・ファイルを使用する。

Oracle Media Management API およびプロキシ・コピー

Oracle Media Management API バージョン 2 以上では、プロキシ・コピーの機能がサポートされます。Recovery Manager プロキシ・バックアップを行う場合、Oracle はプロキシ・コピー機能をサポートしない Oracle Media Management API リリース 1.1 またはバージョン 2 とリンクされます。その後 Recovery Manager はエラーを戻し、バックアップは失敗します。

分散データベース

この項では、分散データベースに関連する互換性および相互運用性の問題について説明します。

マテリアライズド・ビュー

リリース 8.1 より前のリリースでは、Oracle マテリアライズド・ビューは、常にマテリアライズド・ビュー・ベース表とそのベース表のビューで構成されていました。たとえば、SNAP_EMP マテリアライズド・ビューを作成することによって、SNAP_EMP ビューおよびベース表（通常、SNAP\$_SNAP_EMP と呼ばれる）が作成されました。リリース 8.1 以上では、ほとんどのマテリアライズド・ビューは、そのマテリアライズド・ビューと同じ名前のベース表のみを持ちます。ビューは作成されません。

ビューは次の場合にマテリアライズド・ビューに追加されます。

- リリース 8.0 などのリリース 8.1 より前のデータベースからマテリアライズド・ビューがインポートされた。
- COMPATIBLE パラメータが 8.1.0 未満に設定されている。
- マテリアライズド・ビューは非表示列を必要とする（副問合せを含む ROWID マテリアライズド・ビューおよび高速リフレッシュ可能なマテリアライズド・ビュー）。

注意： Oracle8 および Oracle8i のスナップショットの Oracle7 データベースへのインポートはサポートされていません。

Oracle Replication

レプリケート環境には、次の互換性に関する制限が適用されます。

- Oracle の異なるリリースがあるレプリケート環境の場合、古いリリースと互換性がないデータはレプリケートできません。たとえば、リリース 8.1.0 の互換性レベルのデータベースとリリース 8.0.0 の互換性レベルのデータベースがあるレプリケート環境では、リリース 8.0 と互換性のないデータは、これらのデータベース間でレプリケートできません。

- リリース 8.1 より前のリリースの環境では、Advanced Replication パッケージの多くは外部に存在しましたが、リリース 8.1 以上では内部に存在します。これは、パフォーマンスを向上し、データの整合性を保つためです。このような内部パッケージについては、『Oracle9i アドバンスド・レプリケーション』を参照してください。

マスター・サイトの 1 つ以上がリリース 8.1 より前のものであれば、次のプロシージャで GENERATE_80_COMPATIBLE フラグを解除または TRUE に設定する必要があります。

- GENERATE_REPLICATION_SUPPORT
- CREATE_SNAPSHOT_REPOBJECT
- GENERATE_SNAPSHOT_SUPPORT

異機種間サービス・エージェント

この項では、異機種間サービス・エージェントに関する互換性の問題および相互運用性の問題について説明します。

異なるリリースのサーバー間の相互運用性 リリース 8.0.3 以上のリリースのサーバーは、リリース 8.0.3 以上のリリースの異機種間サービス・エージェントに接続および使用できます。異なるリリースのサーバー間で接続する場合、リリースの低い方の機能に制限されます。

マルチスレッド・サービス・エージェント リリース 8.1 以上では、マルチスレッド異機種間サービス・エージェントがサポートされています。既存のエージェントが存在し、マルチスレッド機能を利用する場合は、エージェント初期化ファイルを作成し、エージェント制御ユーティリティを使用してそのエージェントを起動します。

参照： 異機種間サービスの概要、エージェント初期化ファイルの作成およびエージェント制御ユーティリティを使用したエージェントの起動については、『Oracle9i Heterogeneous Connectivity Administrator's Guide』を参照してください。

SQL*Net、Net8 または Oracle Net

Oracle7、Oracle8 および Oracle8i では、SQL*Net V2 または Net8 を使用できます。ただし、SQL*Net V1 は、異なるネットワーク・アドレッシング・スキームを使用しているため、Oracle8 以上では使用できません。したがって、アップグレード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 V2 または Net8 を起動する必要があります。
- 共有サーバーには、SQL*Net V2、Net8 または Oracle Net がサーバー上に必要です。したがって、共有サーバーを使用してクライアントと接続するには、クライアント上の SQL*Net V2、Net8 または Oracle Net を使用する必要があります。

SQL*Net V2 または Net8 への SQL*Net V1 のアップグレード

SQL*Net V1 を SQL*Net V2 または Net8 へアップグレードするには、次の変更を行います。

- SQL*Net V2 または Net8 をインストールします。
- 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します。SQL*Net V2 は『SQL*Net 管理者ガイド』で説明されている構文を使用し、Net8 は『Oracle8i Net8 管理者ガイド』で説明されている構文を使用します。
- SQL*Plus および SQL*Forms を含む、SQL*Net V2 または Net8 で使用するプリコンパイル・プログラムおよび Oracle 実行可能ファイルを再リンクします。

参照： V1 から V2 への SQL*Net のアップグレードの詳細は、『SQL*Net 管理者ガイド』および『SQL*Net V2 Migration Guide』を参照してください。SQL*Net V1 の Net8 へのアップグレードの詳細は、『Oracle8i Net8 管理者ガイド』を参照してください。

サービス・ネーミングおよび接続時ロード・バランシング

リリース 8.1 以上のリリースは、複数のデータベース・インスタンスを含むサービスのサービス・ネーミングおよび接続時ロード・バランシングをサポートします。各サービスは複数インスタンスを含むことができ、各インスタンスは複数ハンドラを含むことができます。このサポートによって、クライアントはある特定のデータベース・インスタンスではなくサービスにアクセスできるようになります。また、ある特定のインスタンス名からサービス名を論理的に分割することができます。

複数インスタンスを含むサービスをサポートするには、次のような接続記述子の新規パラメータを使用します。

- SERVICE_NAME
- INSTANCE_NAME

新規パラメータは、次のような処理で発生する要求を処理することで接続時ロード・バランシングを使用可能にします。

1. クライアント・プログラムが接続先のサービスの名前を指定します。
2. TNS リスナーがサービス内で最も負荷の少ないインスタンスを検索します。
3. TNS リスナーがサービス内で最も負荷の少ないハンドラを検索します。
4. 必要に応じて、TNS リスナーがクライアントを最適のハンドラにリダイレクトするか、クライアント接続をハンドラに渡します。

接続時ロード・バランシングを使用するには、次の処理を実行してください。

- 接続拡張子の SID パラメータの使用を中止します。
- SERVICE_NAMES および INSTANCE_NAME 初期化パラメータを初期化パラメータ・ファイルで使用します。

- `SERVICE_NAME` パラメータを `tnsnames.ora` ファイルで使用します。

注意： TNS リスナーが同じインスタンス名を持つ 2 つ以上のインスタンスを処理できるように設定する前に、どのクライアント・プログラムも SID パラメータに基づく接続を使用していないことを確認してください。

参照： 接続時ロード・バランシングおよび `SERVICE_NAME` パラメータの使用の詳細は、『Oracle9i Net Services 管理者ガイド』を参照してください。

その他の互換性および相互運用性の問題

この項では、Oracle インストールのその他の互換性の問題および相互運用性の問題について説明します。

2GB のファイル・サイズ依存性

リリース 8.0.4 以上では、2GB を超えるファイルにアクセスできます。ただし、このアクセスは次のようなオペレーティング・システム依存性に影響されます。

- **ファイル・モード：** そのファイルは、ファイル・システム・ファイルまたは RAW デバイス・ファイルですか？多くの UNIX システムは、2GB を超えるファイルサイズを RAW デバイスでのみサポートします。
- **非同期 I/O：** オペレーティング・システムは、ファイルの非同期 I/O をローおよびファイル・システム・ファイルの両方についてサポートしていますか？非同期 I/O は 2GB を超えるファイルをサポートしていますか？
- **オペレーティング・システムのリビジョン：** オペレーティング・システムのリリース番号は 2GB を超えるファイル・サイズをサポートしていますか？たとえば、Solaris 2.5.1 では、RAW デバイスについてのみ 2GB を超えるファイル・サイズをサポートしています。ただし、Solaris 2.6 では、2GB を超えるローおよびファイル・システム・ファイルをサポートします。
- **オペレーティング・システムの I/O サブシステムに関する問題：** 2GB を超えるファイル・サイズをサポートするために、オペレーティング・システムがファームウェアのアップグレードを必要としていますか？2GB を超えるファイル・サイズをサポートするようになったのは最近のことで、多くのディスク・アレイまたは I/O サブシステムは、2GB を超えるファイル・サイズをサポートするために、ファームウェアのアップグレードを必要としています。オペレーティング・システムの発売元に、大きいサイズのファイルをサポートするためにどのファームウェア・パッチが必要かを確認してください。

2GB を超えるサイズのファイルを使用する前に、これらのオペレーティング・システム依存性を確認してください。

アプリケーションのアップグレード

この章では、ご使用のアプリケーションのアップグレードについて説明します。この章では、次の項目について説明します。

- [アプリケーションのアップグレードの概要](#)
- [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](#)
- [SQL*Plus スクリプトのアップグレード](#)
- [Oracle7 Forms または Oracle Developer アプリケーションのアップグレード](#)

アプリケーションのアップグレードの概要

Oracle9i の新しいリリースで使用可能になった機能を使用しない既存のアプリケーションを変更する必要はありません。新しい Oracle9i データベースで実行される既存のアプリケーションは以前のリリースと同様に機能し、同等以上のパフォーマンスを得ることができます。

Oracle9i の新しいリリースにアップグレードすると、多くの新機能および拡張機能を使用できます。いくつかの機能が追加され、またパフォーマンスが向上します。アプリケーションをアップグレードする前に、これらの新機能を確認して、どの機能を使用するかを判断する必要があります。

参照： Oracle9i の新しいリリースで使用可能な機能の詳細は、『Oracle9i データベース新機能』を参照してください。

アプリケーションの互換性の問題

Oracle の異なるリリース間に互換性の問題があり、ご使用のアプリケーションに影響する場合があります。これらの互換性の問題は、異なるリリースの Oracle データベース・サーバーの違いによって発生します。また、Oracle の新しいリリースでは、Oracle の新しい予約語が追加され、初期化パラメータおよびデータ・ディクショナリが変更されている場合があります。

ご使用の Oracle データベース・サーバーを新しいリリースにアップグレードする場合は、アプリケーションが Oracle の予約語を使用していないか、アプリケーションがサーバーの初期化パラメータと互換性があるか、またはアプリケーションがサーバーのデータ・ディクショナリと互換性があるかを確認してください。Oracle ソフトウェアの新しいリリースには、特定のリリースのオペレーティング・システムまたはパッチの適用が必要な場合があります。

参照：

- アプリケーションに関する互換性の問題の詳細は、5-17 ページの「[アプリケーション](#)」を参照してください。
- 初期化パラメータおよびデータ・ディクショナリの変更の詳細は、[付録 A「初期化パラメータおよびデータ・ディクショナリの変更」](#)を参照してください。
- Oracle の予約語の詳細は、『Oracle9i SQL リファレンス』を参照してください。
- オペレーティング・システムの要件の詳細は、ご使用のオペレーティング・システム固有の Oracle マニュアルを参照してください。

SQL*Net V2、Net8 および Oracle Net Services は、Oracle の様々なリリースで動作します。そのため、Oracle7 データベース、Oracle8 および Oracle8i データベース、Oracle9i データベースは、SQL*Net V2、Net8 および Oracle Net Services を使用して通信できます。ただし、SQL*Net V1 は、異なるネットワーク・アドレッシング・スキームを使用しているため、Oracle8 以上では使用できません。

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

プリコンパイラと OCI アプリケーションのアップグレード・パスは似ています。この項では、これらのアプリケーションのアップグレード選択肢について説明します。また、必要に応じてプリコンパイラと OCI アプリケーションの違いを注記します。

本番環境をアップグレードする前に、テスト環境を作成します。テスト環境には、アップグレードされたアプリケーションおよび新しい Oracle9i データベースが必要です。また、テスト環境では、アプリケーションの現実的なテストを行う必要があります。

参照： これらのプログラミング環境の使用の詳細は、『Pro*C/C++ Precompiler プログラマーズ・ガイド』、『Pro*COBOL Precompiler プログラマーズ・ガイド』および『Oracle Call Interface プログラマーズ・ガイド』を参照してください。

ソフトウェアのアップグレードおよびクライアント / サーバー構成の理解

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選択肢を理解するには、まず、実行するソフトウェアのアップグレード・タイプおよびクライアント / サーバー構成を理解する必要があります。

ソフトウェアのアップグレード・タイプ

クライアントおよびサーバーの Oracle ソフトウェアに対して、次の 2 つのタイプのアップグレードを行うことができます。

データベースのメジャー・リリースのアップグレード このアップグレードは、リリース番号の最初の数字を変更します。たとえば、Oracle8i から Oracle9i へのアップグレードは、データベースのメジャー・リリースのアップグレードです。

データベースのメンテナンス・リリースのアップグレード このアップグレードは、リリース番号の 2 番目の数字を変更します。たとえば、リリース 1 (9.0.1) からリリース 2 (9.2) へのアップグレードは、メンテナンス・リリースのアップグレードです。

注意： Oracle9i リリース 2 (9.2) 以上では、リリース番号の 2 番目の数字が Oracle のメンテナンス・リリースを表します。以前のリリースでは、3 番目の数字がメンテナンス・リリースを表していました。

使用可能なクライアント / サーバー構成

ご使用のプリコンパイラおよび OCI アプリケーションは、クライアント / サーバー環境のクライアントで実行されます。ここで、Oracle データベース・サーバーはサーバーにあります。ご使用の環境で、次の 1 つ以上のクライアント / サーバー構成を使用できます。

異なるコンピュータ クライアント・ソフトウェアおよびサーバー・ソフトウェアが異なるコンピュータにあり、それぞれがネットワークで接続されています。クライアント環境とサーバー環境は独立しています。

同じコンピュータ上の異なる Oracle ホーム・ディレクトリ クライアント・ソフトウェアおよびサーバー・ソフトウェアは同じコンピュータ上にありますが、異なる Oracle ホーム・ディレクトリにインストールされています。クライアント環境とサーバー環境は独立しています。

同じ Oracle ホーム クライアント・ソフトウェアおよびサーバー・ソフトウェアが同じコンピュータ上の同じ Oracle ホームにインストールされています。この場合、サーバー・ソフトウェアをアップグレードすると、クライアント・ソフトウェアもアップグレードされます。

参照： クライアント / サーバー環境の詳細は、『Oracle9i データベース概要』および『Oracle9i Heterogeneous Connectivity Administrator's Guide』を参照してください。

Oracle ソフトウェアをアップグレードする場合のアプリケーションの互換性規則

この項では、Oracle サーバー・ソフトウェアまたは Oracle クライアント・ソフトウェアをアップグレードする場合に適用される、互換性規則について説明します。この規則は、実行するソフトウェア・アップグレードのタイプおよびご使用のクライアント / サーバー構成に基づきます。

この項では、次のようなアップグレードの互換性規則について説明します。

- [Oracle サーバー・ソフトウェアのアップグレード](#)
- [Oracle クライアント・ソフトウェアのアップグレード](#)

注意： この項では、6-3 ページの「[ソフトウェアのアップグレードおよびクライアント / サーバー構成の理解](#)」で説明する用語を使用しています。

Oracle サーバー・ソフトウェアのアップグレード

次の規則は、Oracle サーバー・ソフトウェアをアップグレードする場合に適用されます。

クライアント環境を変更しない場合、再リンクは不要 ご使用のクライアントとサーバーが異なるコンピュータにあるか、または同じコンピュータの異なる Oracle ホーム・ディレクトリにあるときに、クライアント・ソフトウェアを変更しないで Oracle サーバー・ソフトウェアをアップグレードする場合は、アプリケーションをプリコンパイル、コンパイルまたは再リンクする必要はありません。このような場合、クライアント・ソフトウェアはサーバー・ソフトウェアから独立しており、サーバーに対し継続して機能します。

ただし、アプリケーションが Oracle データベース・サーバーと同じ Oracle ホームを使用している場合は、サーバー・ソフトウェアをアップグレードするとクライアント・ソフトウェアもアップグレードされます。この場合、6-6 ページの「[Oracle クライアント・ソフトウェアのアップグレード](#)」の規則に従う必要があります。

注意： Oracle サーバー・ソフトウェアと、プリコンパイラまたは OCI クライアント・ソフトウェアが同じ Oracle ホームを使用している場合、新しいプリコンパイラまたは OCI クライアント・ソフトウェアはインストールしないで、Oracle サーバー・ソフトウェアをアップグレードすることができます。この場合、クライアント・ソフトウェアはアップグレードされません。ただし、このような構成はお勧めしません。

Oracle サーバーの新/旧リリースでアプリケーションを実行できる データベース・サーバーに対してプリコンパイラまたは OCI アプリケーションを実行する場合、データベースのサーバー・ソフトウェアのリリースをクライアント・ソフトウェアのリリース以上にすることをお勧めします。ただし、これは必須ではありません。たとえば、Oracle クライアント・ソフトウェアがリリース 8.1.7 の場合、クライアントのプリコンパイラ・アプリケーションをサーバーに対して実行するには、Oracle サーバー・ソフトウェアはリリース 8.1.7 以上である必要があります。

OCI では、Oracle7 のクライアント・ソフトウェアは、リリース 8.0 以上の Oracle サーバーに対して実行できます。また、リリース 8.0 以上のクライアント・ソフトウェアは、Oracle7 サーバーに対して実行できます。リリース 8.0 以上のクライアントを Oracle7 サーバーに対して実行している場合、アプリケーションは、リリース 8.0 以上で使用可能な機能（オブジェクト機能を含む）を使用できません。

Oracle クライアント・ソフトウェアのアップグレード

ご使用のクライアント・ソフトウェアをアップグレードして、現行のサーバー・ソフトウェアと一致させることをお勧めします。たとえば、ご使用の Oracle データベース・サーバーを Oracle9i リリース 2 (9.2) にアップグレードする場合は、クライアント・ソフトウェアも Oracle9i リリース 2 (9.2) にアップグレードすることをお勧めします。サーバー・ソフトウェアとクライアント・ソフトウェアのリリース番号を同じにしておくと、アプリケーションの最大の安定性を得ることができます。さらに、最新の Oracle クライアント・ソフトウェアは、以前のリリースにはない追加機能およびパフォーマンス強化を提供する場合があります。

次の規則は、Oracle クライアント・ソフトウェアをアップグレードする場合に適用されます。

アプリケーションは、より新しいライブラリにリンクできる プリコンパイラ・アプリケーションによって生成されたコードは、サーバーのリリース以上のクライアント・ライブラリのリリースにリンクできます。また、Oracle7、リリース 8.0 以上の SQLLIB 関数コールは、同じアプリケーションおよび同じトランザクションで混在できません。

OCI アプリケーションは、そのアプリケーションが開発された OCI ライブラリのバージョン以上の OCI ランタイム・ライブラリのバージョンとリンクできます。

静的にリンクされたアプリケーションを再リンクさせる必要はない 静的にリンクされたアプリケーションに対しては、クライアント・ソフトウェアのどのタイプのアップグレードを実行しても、プリコンパイラおよび OCI アプリケーションを再リンクさせる必要はありません。ただし、パフォーマンスが向上する場合があるため、再リンクさせることをお勧めします。

動的にリンクされたアプリケーションを再リンクさせる必要はない クライアント・ソフトウェアのアップグレードを実行する場合、動的にリンクされたプリコンパイラおよび OCI アプリケーションを再リンクさせる必要はありません。ただし、パフォーマンスが向上する場合があるため、再リンクさせることをお勧めします。

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選択肢

プリコンパイラおよび OCI アプリケーションをアップグレードする場合、次の 4 つの選択肢があります。

選択肢 1: アプリケーションを変更しません。アプリケーションを再リンク、プリコンパイルまたはコンパイルしません。また、アプリケーション・コードを変更しません。アプリケーションは、Oracle9i データベースで継続して動作します。

選択肢 2: アプリケーションを新しい Oracle9i ライブラリと再リンクさせます。アプリケーションをプリコンパイルまたはコンパイルしません。また、アプリケーション・コードを変更しません。

- 選択肢 3:** 新しい Oracle9i ソフトウェアを使用して、Oracle9i アプリケーションをプリコンパイルまたはコンパイル（あるいはその両方）し、再リンクさせます。アプリケーション・コードは変更しません。
- 選択肢 4:** アプリケーション・コードを変更して、Oracle9i の新機能を使用します。コードをプリコンパイルまたはコンパイル（あるいはその両方）し、再リンクさせます。

これらの選択肢は、難易度が高く、考えられるメリットが大きくなる順序に記述されています。つまり、選択肢 1 は最も簡単ですが、考えられるメリットは最も小さくなります。選択肢 4 は最も難しい方法ですが、考えられるメリットは最も大きくなります。これらの選択肢については次の項で説明します。

選択肢 1: アプリケーションを変更しない

アプリケーションを変更しません。アプリケーションは Oracle9i データベースで継続して動作します。この選択肢の主なメリットは、単純で簡単であることです。さらに、この選択肢では、すべてのクライアント・コンピュータをアップグレードする必要がないため、管理の必要性が最も少なくなります。多くのクライアント・コンピュータがある場合は、すべてのコンピュータのアップグレードに費やす管理コストを削減することが大変重要になります。

この選択肢のデメリットは、アプリケーションが Oracle9i の新しいリリースで使用可能な機能を使用できないことです。さらに、アプリケーションは、Oracle9i の新しいリリースのパフォーマンスのメリットを得ることができません。

選択肢 2: アプリケーションを新しい Oracle9i ライブラリと再リンクさせる

コード変更および再コンパイルしないで、アプリケーションを新しい Oracle9i ライブラリに再リンクさせることができます。再リンクさせることによって、アプリケーションは、新しいライブラリのみで得られるパフォーマンスの向上を得られます。本番環境でアプリケーションを再リンクさせる前に、必ずテスト環境でアプリケーションを再リンクさせてください。

注意： 共有ライブラリをサポートしていないオペレーティング・システムでは、実行可能ファイルに新しいライブラリを追加する場合、アプリケーションを再リンクさせてください。

選択肢 3: 新しい Oracle9i ソフトウェアを使用してアプリケーションをプリコンパイルまたはコンパイルする

コードを変更しないで、新しい Oracle9i ソフトウェアを使用してアプリケーションをプリコンパイルまたはコンパイルできます。この選択肢では、各クライアント・コンピュータに新しい Oracle クライアント・ソフトウェアをインストールする必要があります。ただし、クライアントの数にかかわらず、必要なアプリケーションのプリコンパイルまたはコンパイルおよび再リンクは 1 回のみです。これは大きなメリットです。

再コンパイルすることによって、アプリケーション・コードの構文チェックを実行します。新しい Oracle ソフトウェアを使用してプリコンパイルまたはコンパイルすると、Oracle ソフトウェアの以前のリリースでは検出されなかったアプリケーション・コードのいくつかの問題が、検出される場合があります。したがって、新しいソフトウェアを使用したプリコンパイルまたはコンパイルは、以前に検出されなかった可能性があるアプリケーション・コードの問題の検出および解決に役立ちます。

また、アプリケーションは新しい Oracle ソフトウェアで動作するため、再コンパイルすることによって、アプリケーションの最大の安定性を得ることができます。さらに、使用可能な最新のツール製品および機能を使用して、ご使用の環境で新しい開発を行うことができます。さらに、再コンパイルおよび再リンクの実行後のみ、新しい Oracle ソフトウェアで得られるパフォーマンスの向上を得ることができます。

選択肢 4: アプリケーション・コードを変更して、Oracle9i の新機能を使用する

アプリケーションのコードを変更することで、新しい Oracle9i 機能のメリットを得ることができます。この選択肢は最も難しい方法ですが、考えられるメリットも最大です。この選択肢では、選択肢 3 で説明したすべてのメリットを得ることができます。さらに、アプリケーションへの変更によって、Oracle9i で得られるパフォーマンスおよびスケーラビリティのメリットも得ることができます。さらに、Oracle9i の新しいリリースでのみ使用できる新機能を、アプリケーションに追加できます。

『Oracle9i データベース新機能』を参照して、新しい Oracle9i の機能を理解してください。使用する機能の実装方法については、ご使用の開発環境の Oracle マニュアルを参照してください。プリコンパイラの詳細は、『Pro*C/C++ Precompiler プログラマーズ・ガイド』および『Pro*COBOL Precompiler プログラマーズ・ガイド』を参照してください。OCI の詳細は、『Oracle Call Interface プログラマーズ・ガイド』を参照してください。

使用する新機能を決定したら、アプリケーションのコードを変更して、これらの機能を使用できるようにします。ご使用の開発環境に基づいて、次の項の該当する指示に従ってください。

- [プリコンパイラ・アプリケーションの変更](#)
- [OCI アプリケーションの変更](#)

プリコンパイラ・アプリケーションの変更 Oracle9i の機能を使用できるようにプリコンパイラ・アプリケーションを変更するには、次の手順を実行します。

1. 既存のアプリケーションが Oracle7、Oracle8 または Oracle8i のいずれかによって、次のいずれかの処理を実行します。
 - Oracle7 アプリケーションを使用している場合、Oracle7 と Oracle9i の違いを反映させるように、既存の Oracle7 アプリケーションを変更するか、または新しいアプリケーションが記述されている必要があります。
 - Oracle8 または Oracle8i のアプリケーションを使用していて、Oracle9i の新機能のメリットを得る場合、Oracle9i の新機能のコードを既存の Oracle8 または Oracle8i のアプリケーションに取り込みます。
2. Oracle プリコンパイラを使用して、アプリケーションをプリコンパイルします。
3. アプリケーションをコンパイルします。
4. プリコンパイラに付属する Oracle9i 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させます。

OCI アプリケーションの変更 Oracle9i の機能を使用できるように OCI アプリケーションを変更するには、次の手順を実行します。

1. 次のいずれかの方法で、OCI コールを変更します。
 - アプリケーションが Oracle7 の OCI コールを使用している場合、新しい Oracle9i の OCI コールのみを使用するように、アプリケーションを変更します。
 - アプリケーションが Oracle7 の OCI コールを使用している場合、Oracle7 のコールをいくつかの操作に使用したままで、Oracle9i の OCI コールを既存のアプリケーションに取り込みます。
 - アプリケーションが Oracle8 または Oracle8i のコールのみを使用している場合、新しい Oracle9i の OCI コールを既存のアプリケーションに取り込みます。
2. アプリケーションをコンパイルします。
3. アプリケーションを Oracle9i のランタイム・ライブラリに再リンクさせます。

SQL*Plus スクリプトのアップグレード

SQL*Plus リリース 8.0 以上、リリース 8.0 以上のデータベースおよび PL/SQL リリース 8.0 以上の機能を使用するには、次の手順を実行します。

1. SQL*Plus リリース 3.x のスクリプトを次のように変更して、SQL*Plus リリース 8.0 以上と互換性のあるスクリプトに変換します。
 - a. スクリプトに `SET COMPATIBILITY VERSION 7` という行がある場合は、それを `SET COMPATIBILITY NATIVE` に変更します。または、この行を削除してデフォルト設定を使用します。Oracle9i では、デフォルトの設定は NATIVE です。
 - b. `login.sql` および `glogin.sql` ファイルを確認して、`SET COMPATIBILITY VERSION 7` という行を `SET COMPATIBILITY NATIVE` に変更します。
2. Oracle9i の新機能を使用するには、Oracle9i の新しい構文を使用するように既存の SQL スクリプトを変更します。Oracle9i の新機能を使用しない場合は、既存の SQL スクリプトに変更を加えなくても Oracle9i で実行できます。

参照：

- SQL*Plus の新機能の詳細は、『SQL*Plus ユーザーズ・ガイドおよびリファレンス』を参照してください。
- SQL スクリプトのアップグレードの詳細は、『Oracle9i SQL リファレンス』を参照してください。

注意： PL/SQL パッケージ、プロシージャまたはファンクションを変更する必要はありません。

Oracle7 Forms または Oracle Developer 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、Oracle7、Oracle8、Oracle8i および Oracle9i で同様に動作します。ただし、『Oracle9i データベース新機能』で説明されている新機能を確認して、Oracle9i の新機能がアプリケーションに有効か、またはアプリケーションに影響を与えるかを判断してください。Oracle9i の機能とどのように相互作用するかについては、Oracle Developer のドキュメント・セットを参照してください。また、オペレーティング・システムの Oracle Developer ドキュメントには、ご使用の Forms または Developer アプリケーションのアップグレード方法が記述されています。

注意： Oracle Developer の新しいリリースでは、Oracle Developer 固有の新しい予約語が導入されている場合があります。アプリケーションがこれらの予約語を使用している場合は、コードを変更する必要があります。

以前の Oracle リリースへのデータベースのダウングレード

この章では、データベースを Oracle の以前のリリースへダウングレードする処理について説明します。この章では、次の項目について説明します。

- [ダウングレードがサポートされているリリース](#)
- [全オフライン・バックアップの実行](#)
- [非互換性の削除](#)
- [データベース互換性のリセット](#)
- [特定のコンポーネントのダウングレード](#)
- [データベースのダウングレード](#)

参照： ダウングレードには、オペレーティング・システム固有の部分があります。ご使用のオペレーティング・システムでのダウングレードに関するその他の指示は、オペレーティング・システム固有の Oracle マニュアルを参照してください。

ダウングレードがサポートされているリリース

Oracle9i リリース 2 (9.2) では、次のリリースのみダウングレードが可能です。

- リリース 9.0.1.3.0
- リリース 8.1.7.3.0

以前のデータベースのリリース番号が、リリース 8.1.7 (リリース 8.1.7.3.0 より前) の場合は、ダウングレードする前に、パッチ・リリース 8.1.7.3.0 のソフトウェアをインストールしてください。

同様に、データベースのリリース番号が、リリース 1 (9.0.1) (リリース 9.0.1.3.0 より前) の場合は、ダウングレードする前に、パッチ・リリース 9.0.1.3.0 のソフトウェアをインストールしてください。

以前のデータベースを、最初にパッチ・リリース 8.1.7.3.0 または 9.0.1.3.0 へアップグレードする必要はありませんが、パッチ・リリースのソフトウェアは、リリース 2 (9.2) からダウングレードする前にインストールする必要があります。

全オフライン・バックアップの実行

ダウングレードする前に、Oracle9i リリース 2 (9.2) のデータベースに対し、全オフライン・バックアップを実行してください。

参照： 詳細は、『Oracle9i ユーザー管理バックアップおよびリカバリ・ガイド』を参照してください。

非互換性の削除

非互換性の削除のプロセスは、ダウングレード先のリリースによって異なります。まず、ご使用のデータベースの互換性レベルを調査して、ダウングレードするリリースと互換性があるかどうかを確認します。

データベースの互換性レベルの調査

ご使用のデータベースの互換性レベルがダウングレードするリリースより高い場合、ダウングレードする前に削除する必要がある非互換性が、データベースにある可能性があります。ご使用のデータベースの互換性レベルは、COMPATIBLE 初期化パラメータの設定によって決まります。次の SQL 文を発行して、COMPATIBLE 初期化パラメータの設定を調査します。

```
SQL> SELECT name, value, description FROM v$parameter  
WHERE name='compatible';
```


COMPATIBLE パラメータが、ダウングレードしようとしているリリース以前に設定されている場合、非互換性の削除は必要ありません。たとえば、リリース 1 (9.0.1) にダウングレードする場合、COMPATIBLE パラメータが 9.0.0 以下に設定されていれば、非互換性の削除は必要ありません。この場合、使用しているデータベースには、ダウングレードするリリースとの間の非互換性は存在しません。7-20 ページの「[特定のコンポーネントのダウングレード](#)」に進んでください。

ただし、COMPATIBLE パラメータがダウングレードするリリースより高く設定されている場合は、非互換性がいくつか存在している可能性があります。たとえば、リリース 8.1.7 にダウングレードする場合、COMPATIBLE が 9.0.0 以上に設定されていると、非互換性が存在する可能性があります。

非互換性の識別

ダウングレードするリリースに存在する可能性がある非互換性を識別するには、次の手順を実行します。

1. Oracle ホーム・ディレクトリの所有者としてシステムにログインします。
2. システム・プロンプトで、ORACLE_HOME/rdbms/admin ディレクトリへ移動します。
3. SQL*Plus を起動します。
4. SYSDBA 権限を持つユーザーとして、データベース・インスタンスに接続します。
5. RESTRICT モードでインスタンスを起動します。

```
SQL> STARTUP RESTRICT
```

PFILE オプションを使用して、初期化パラメータ・ファイルの場所を指定する必要がある場合があります。

6. V\$COMPATIBILITY の動的パフォーマンス・ビューに問い合わせ、非互換性を識別します。

```
SQL> SELECT * FROM v$compatibility WHERE release != '0.0.0.0.0';
```

RELEASE 列の値がダウングレードするリリースより高い場合は、非互換性が存在します。

注意： この問い合わせは、0.0.0.0.0 の互換性レベルの機能を表示しません。これらの機能は現在使用されておらず、その機能に対しての処置は必要ありません。

7. `utlincmp.sql` を実行します。

```
SQL> SPOOL utlincmp.log
SQL> @utlincmp.sql
SQL> SPOOL OFF
```

`utlincmp.sql` スクリプトは、この章のこれ以降に示すすべての問合せを実行して、非互換性を識別します。したがって、`utlincmp.sql` スクリプトを実行することによって、この章のこれ以降の部分で説明するすべての `SELECT` 文を実行できます。

`utlincmp.sql` スクリプトの実行後、`utlincmp.log` ファイルを表示し、`SELECT` 文が値を戻したインスタンスを検索します。戻された値が、ダウングレードするリリースとの非互換性で、削除が必要な場合があります。

8. すべての非互換性を削除または変更し、ダウングレードするリリースと互換性のあるデータベースを作成します。

次の項では、Oracle の以前のリリースとの非互換性の削除について説明します。ダウングレードするリリースによっては、次の項の一部またはすべての手順の実行が必要になる場合があります。

たとえば、リリース 1 (9.0.1) にダウングレードする場合は、7-4 ページの「[リリース 2 \(9.2\) の非互換性の削除](#)」の手順の実行のみが必要です。ただし、リリース 8.1.7 へダウングレードする場合は、7-4 ページの「[リリース 2 \(9.2\) の非互換性の削除](#)」および 7-8 ページの「[リリース 1 \(9.0.1\) の非互換性の削除](#)」の手順を実行する必要があります。

注意： Oracle9i Enterprise Edition から Oracle9i (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、Oracle9i Enterprise Edition の拡張機能を使用するアプリケーションを変更し、それらの拡張機能を使用できないようにしてください。Edition 間の違いに関する詳細は、『Oracle9i データベース新機能』を参照してください。

リリース 2 (9.2) の非互換性の削除

リリース 1 (9.0.1) 以下にダウングレードする場合は、次の項に示す処置を実行して非互換性を削除します。

- [リリース 2 \(9.2\) のデフォルト・パーティション](#)
- [リリース 2 \(9.2\) のパーティション化メソッド](#)
- [リリース 2 \(9.2\) の Streams](#)
- [リリース 2 \(9.2\) のサブパーティション・テンプレート](#)
- [LOB リテンション](#)
- [LOB を含む自動セグメント領域で管理される表領域](#)

リリース 2 (9.2) のデフォルト・パーティション

この項では、リリース 2 (9.2) のデフォルト・パーティションに関する非互換性の削除について説明します。

リスト・パーティション表にあるすべてのデフォルト・パーティションの削除 リリース 1 (9.0.1) 以下にダウングレードする前に、リスト・パーティション表のすべてのデフォルト・パーティションを削除します。デフォルト・パーティションのあるすべてのリスト・パーティション表を識別するには、次の SQL 文を発行します。

```
SELECT u.name AS OWNER, o.name AS TABLE_NAME, o.subname AS PARTITION_NAME
FROM sys.user$ u, sys.obj$ o, sys.tabpart$ tp
WHERE BITAND(tp.flags, 16384) = 16384
AND tp.obj# = o.obj# AND o.owner# = u.user#;
```

OWNER.TABLE_NAME 列で表される表の PARTITION_NAME 列で表される各パーティションを削除します。

```
ALTER TABLE OWNER.TABLE_NAME DROP PARTITION PARTITION_NAME;
```

リリース 2 (9.2) のパーティション化メソッド

この項では、リリース 2 (9.2) のパーティション化メソッドに関する非互換性の削除について説明します。

レンジ・リスト・メソッドを使用するすべてのパーティション表の削除 リリース 1 (9.0.1) 以下にダウングレードする前に、レンジ・リスト・メソッドを使用するすべてのパーティション表を削除します。レンジ・リスト・メソッドを使用してパーティション化された既存の表を識別するには、次の SQL 文を発行します。

```
SELECT u.name AS OWNER, o.name AS TABLE_NAME
FROM sys.user$ u, sys.obj$ o, sys.partobj$ po
WHERE po.parttype = 1 AND MOD(po.spare2, 256) = 4
AND o.obj# = po.obj# AND o.owner# = u.user#;
```

表データを保存する必要がない場合、OWNER.TABLE_NAME 列で表される各表を削除します。

```
DROP TABLE OWNER.TABLE_NAME;
```

ただし、表データを保存する必要がある場合、データを非パーティション表へコピーするか、またはデータをレンジ、ハッシュ、リストまたは他のコンポジット・メソッドによってパーティション化された表へコピーします。

リリース 2 (9.2) の Streams

この項では、リリース 2 (9.2) の Streams に関する非互換性の削除について説明します。

すべての Streams 取得プロセスの削除 リリース 1 (9.0.1) 以下にダウングレードする前に、すべての Streams 取得プロセスを削除します。既存の取得プロセスを識別するには、次の SQL 文を発行します。

```
SELECT capture_name FROM dba_capture;
```

CAPTURE_NAME 列にリストされている各取得プロセスに対して、次の SQL 文を発行します。

```
EXECUTE dbms_capture_adm.drop_capture(capture_name => 'CAPTURE_NAME');
```

すべての Streams 適用プロセスの削除 リリース 1 (9.0.1) 以下にダウングレードする前に、すべての Streams 適用プロセスを削除します。既存の適用プロセスを識別するには、次の SQL 文を発行します。

```
SELECT apply_name FROM dba_apply;
```

APPLY_NAME 列にリストされている各適用プロセスに対して、次の SQL 文を発行します。

```
EXECUTE dbms_apply_adm.drop_apply(apply_name => 'APPLY_NAME');
```

リリース 2 (9.2) のサブパーティション・テンプレート

この項では、リリース 2 (9.2) のサブパーティション・テンプレートに関する非互換性の削除について説明します。

コンポジット・パーティション表にあるすべてのサブパーティション・テンプレートの削除 リリース 1 (9.0.1) 以下にダウングレードする前に、コンポジット・パーティション表のすべてのサブパーティション・テンプレートを削除します。サブパーティション・テンプレートを含む既存のコンポジット・パーティション表を識別するには、次の SQL 文を発行します。

```
SELECT u.name AS OWNER, o.name AS TABLE_NAME
FROM sys.user$ u, sys.obj$ o
WHERE o.owner# = u.user#
      AND o.obj# in (SELECT DISTINCT bo# FROM defsubpart$)
UNION
SELECT u.name AS OWNER, o.name AS TABLE_NAME
FROM sys.user$ u, sys.obj$ o
WHERE o.owner# = u.user#
      AND o.obj# in (SELECT DISTINCT bo# from defsubpartlob$);
```

OWNER.TABLE_NAME 列で表される表のサブパーティション・テンプレートを削除します。

```
ALTER TABLE OWNER.TABLE_NAME SET SUBPARTITION TEMPLATE ();
```

LOB リテンション

この項では、LOB リテンションに関する非互換性の削除について説明します。

LOB 列に格納されたリテンションの削除 リリース 1 (9.0.1) 以下にダウングレードする前に、LOB 列に格納されたリテンションを削除します。リテンションを含む既存の LOB 列を識別するには、次の SQL 文を発行します。

```
SELECT u.name AS OWNER, o.name AS TABLE_NAME, c.name AS LOB_COL_NAME
FROM sys.user$ u, sys.obj$ o, sys.col$ c, sys.lob$ l
WHERE BITAND(l.flags, 64) = 64 AND l.obj# = o.obj#
AND c.obj# = o.obj# AND c.col# = l.col#
AND o.owner# = u.user#;
```

OWNER.TABLE_NAME 列で表される表の LOB_COL_NAME 列で表される各列のリテンションを削除します。

```
ALTER TABLE OWNER.TABLE_NAME MODIFY LOB(LOB_COL_NAME)
(REBUILD FREEPOOLS);
```

LOB を含む自動セグメント領域で管理される表領域

この項では、LOB を含む自動セグメント領域で管理される表領域に関する非互換性の削除について説明します。

自動セグメント領域で管理される表領域の LOB 列の削除 リリース 1 (9.0.1) 以下にダウングレードする前に、自動セグメント領域で管理される表領域のすべての LOB 列を削除します。LOB 列を含む自動セグメント領域を管理する表領域を識別するには、次の SQL 文を発行します。

```
SELECT u.name AS OWNER, o.name AS TABLE_NAME, c.name AS LOB_COL_NAME
FROM sys.lob$ l, sys.ts$ t, sys.user$ u, sys.obj$ o, sys.col$ c
WHERE l.ts# = t.ts# AND
(DECODE(BITAND(t.flags, 32), 32, 1, 0) = 1 AND t.online$ <> 3) AND
o.owner# = u.user# AND l.obj# = o.obj# AND
l.obj# = c.obj# AND l.col# = c.col#;
```

リストされている各 LOB セグメントについては、次のいずれかの処理を実行します。

- 自動セグメント領域で管理されていない表領域へ LOB 列を移動します。

```
ALTER TABLE OWNER.TABLE_NAME MOVE LOB(LOB_COL_NAME) STORE AS
(TABLESPACE manual segment space-managed tablespace);
```

- LOB 列を含む表を削除します。

```
DROP TABLE OWNER.TABLE_NAME;
```

リリース 1 (9.0.1) の非互換性の削除

リリース 8.1.7 以下にダウングレードする場合、次の項に示す処置を実行して非互換性を削除します。

- 表領域
- スキーマ・オブジェクト
- リリース 1 (9.0.1) のパーティション化メソッド
- ハッシュ・パーティション索引構成表
- 不変 PDML ITL
- LOB を含むパーティション索引構成表
- データ型
- ユーザー定義データ型
- SQL および PL/SQL
- 制約およびトリガー
- OLAP

表領域

この項では、リリース 1 (9.0.1) で導入された表領域に関する非互換性の削除について説明します。

自動セグメント領域で管理されるすべての表領域の削除 リリース 8.1.7 以下にダウングレードする前に、自動セグメント領域で管理されるすべての表領域を削除します。自動セグメント領域で管理される表領域を識別するには、次の SQL 文を発行します。

```
SELECT TABLESPACE_NAME FROM dba_tablespaces
       WHERE segment_space_management = 'AUTO';
```

TABLESPACE_NAME 列で表される各表領域を削除します。

```
DROP TABLESPACE TABLESPACE_NAME;
```

すべての UNDO 表領域の削除 リリース 8.1.7 以下にダウングレードする前に、すべての UNDO 表領域を削除します。既存の UNDO 表領域を識別するには、次の SQL 文を発行します。

```
SELECT name AS TABLESPACE_NAME FROM sys.ts$
       WHERE BITAND(flags, 16) = 16 AND online$ <> 3;
```

TABLESPACE_NAME 列で表される各表領域を削除します。

```
DROP TABLESPACE TABLESPACE_NAME;
```

スキーマ・オブジェクト

この項では、リリース 1 (9.0.1) で導入されたスキーマ・オブジェクトに関する非互換性の削除について説明します。

すべての外部表の削除 リリース 8.1.7 以下にダウングレードする前に、すべての外部表を削除します。既存の外部表を識別するには、次の SQL 文を発行します。

```
SELECT u.name AS OWNER, o.name AS TABLE_NAME
       FROM sys.user$ u, sys.obj$ o, sys.tab$ t
       WHERE t.obj# = o.obj# AND o.owner# = u.user# AND
              BITAND(t.property, 2147483648) != 0;
```

OWNER.TABLE_NAME 列で表される各表を削除します。

```
DROP TABLE OWNER.TABLE_NAME;
```

索引構成表上のビットマップ 2 次索引の削除 リリース 8.1.7 以下にダウングレードする前に、データベース内にある非パーティション化およびパーティション化された索引構成表のすべてのビットマップ 2 次索引を削除します。索引構成表の既存のビットマップ 2 次索引を識別するには、次の SQL 文を発行します。

```
SELECT index_name, i.owner, t.table_name
       FROM dba_indexes i, dba_tables t
       WHERE i.index_type = 'BITMAP' AND i.table_name = t.table_name
              AND t.owner = i.table_owner AND t.iot_type = 'IOT';
```

マッピング表を持たない索引構成表の再作成 リリース 8.1.7 以下にダウングレードする前で、非パーティション化およびパーティション化された索引構成表のすべてのビットマップ 2 次索引を削除した後に、マッピング表を持たない索引構成表を再構築する必要があります。

マッピング表を持つ索引構成表を識別するには、次の SQL 文を発行します。

```
SELECT owner, iot_name
       FROM dba_tables
       WHERE iot_type = 'IOT_MAPPING';
```

各表（たとえば、iot）では、次のようにマッピング表なしで再構築できます。

```
ALTER TABLE iot MOVE NOMAPPING;
```

ヒープおよび索引構成表上の UROWID データ型の B ツリー索引の削除 リリース 8.1.7 以下にダウングレードする前に、ヒープおよび索引構成表のすべての B ツリー索引を削除します。そのような B ツリー索引を識別するには、次の SQL 文を発行します。

```
SELECT index_owner, index_name
      FROM dba_ind_columns ic, dba_tab_columns tc
      WHERE tc.data_type = 'UROWID' AND tc.table_name = ic.table_name
            AND tc.column_name = ic.column_name;
```

大きいキーを持つ索引の削除 リリース 8.1.7 以下にダウングレードする前に、大きいキーを持つすべての索引を削除します。そのような索引を識別するには、次の SQL 文を発行します。

```
SELECT u.name, o.name, i.flags
      FROM sys.obj$ o, sys.user$ u, sys.ind$ i
      WHERE u.user# = o.owner#
            AND o.obj# = i.obj#
            AND BITAND(i.flags, 16384) != 0;
```

この文によって識別されたすべての索引を削除してください。

リリース 1 (9.0.1) のパーティション化メソッド

この項では、リリース 1 (9.0.1) のパーティション化メソッドに関する非互換性の削除について説明します。

リスト・メソッドを使用するすべてのパーティション表の削除 リリース 8.1.7 以下にダウングレードする前に、リスト・メソッドを使用するすべてのパーティション表を削除します。リスト・メソッドを使用してパーティション化された既存の表を識別するには、次の SQL 文を発行します。

```
SELECT u.name AS OWNER, o.name AS TABLE_NAME
      FROM sys.user$ u, sys.obj$ o, sys.partobj$ po
      WHERE po.parttype = 4
            AND o.obj# = po.obj# AND o.owner# = u.user#;
```

表データを保存する必要がない場合、OWNER.TABLE_NAME 列で表される各表を削除します。

```
DROP TABLE OWNER.TABLE_NAME;
```

ただし、表データを保存する必要がある場合、データを非パーティション表へコピーするか、またはデータをレンジ、ハッシュまたは他のコンポジット・メソッドによってパーティション化された表へコピーします。

ハッシュ・パーティション索引構成表

この項では、ハッシュ・パーティション索引構成表に関する非互換性の削除について説明します。

すべてのハッシュ・パーティション索引構成表の削除 リリース 8.1.7 以下にダウングレードする前に、ハッシュ・パーティション化されたすべての索引構成表を削除します。既存のハッシュ・パーティション化された索引構成表を識別するには、次の SQL 文を発行します。

```
SELECT t.OWNER, t.TABLE_NAME
       FROM dba_tables t, dba_part_tables p
       WHERE t.table_name = p.table_name AND t.owner = p.owner
             AND t.iot_type = 'IOT' AND t.partitioned = 'YES'
             AND p.partitioning_type = 'HASH';
```

表データを保存する必要がない場合、OWNER.TABLE_NAME 列で表される各表を削除します。

```
DROP TABLE OWNER.TABLE_NAME;
```

ただし、表データを保存する必要がある場合は、次のいずれかの処理を行います。

- CREATE TABLE ... AS SELECT 文を使用して、表データをレンジ・パーティション索引構成表または非パーティション索引構成表へ移動します。


```
CREATE range or non-partitioned index-organized table ... AS SELECT * FROM
OWNER.TABLE_NAME;
DROP TABLE OWNER.TABLE_NAME;
```
- Oracle9i エクスポート・ユーティリティを使用して表をエクスポートします。データは、リリース 8.1.7 のインポート・ユーティリティを使用して非パーティション索引構成表またはレンジ・パーティション索引構成表へロードされます。

不変 PDML ITL

リリース 8.1.7 以下にダウングレードする前に、すべての不変 PDML ITL を削除します。既存の不変 PDML ITL を識別するには、次の SQL 文を発行します。

```
SELECT COUNT(*) FROM sys.tab$
       WHERE BITAND(property, 536870912) > 0;
```

この問合せで 0 を超える結果が戻される場合、次の手順を実行します。

1. ORACLE_HOME/rdbms/admin ディレクトリへ移動します。
2. utlpitl.sql を実行します。

```
SQL> @utlpitl.sql
```

LOB を含むパーティション索引構成表

この項では、LOB を含むパーティション索引構成表に関する非互換性の削除について説明します。

パーティション索引構成表のすべての LOB 列の削除 リリース 8.1.7 以下にダウングレードする前に、パーティション索引構成表のすべての LOB 列を削除します。LOB 列のある既存のパーティション索引構成表を識別するには、次の SQL 文を発行します。

```
SELECT t.OWNER, t.TABLE_NAME, l.COLUMN_NAME
FROM dba_lobs l, dba_tables t
WHERE l.table_name = t.table_name and l.owner = t.owner
AND t.iot_type = 'IOT' AND t.partitioned = 'YES';
```

LOB 列およびそのデータを保存する必要がない場合、OWNER.TABLE_NAME 列で表される表の COLUMN_NAME 列で表される各列を削除します。

```
ALTER TABLE OWNER.TABLE_NAME DROP COLUMN COLUMN_NAME;
```

ただし、LOB 列を保存する必要がある場合は、対応する非パーティション索引構成表を作成できます。

```
CREATE non-partitioned index-organized table ... AS SELECT * FROM OWNER.TABLE_NAME;
DROP TABLE OWNER.TABLE_NAME;
```

パーティション索引構成表のすべての VARRAY 列の削除 リリース 8.1.7 以下にダウングレードする前に、パーティション索引構成表のすべての VARRAY 列を削除します。VARRAY 列のある既存のパーティション索引構成表を識別するには、次の SQL 文を発行します。

```
SELECT v.OWNER, v.PARENT_TABLE_NAME, v.PARENT_TABLE_COLUMN
FROM dba_varrays v, dba_tables t
WHERE v.parent_table_name = t.table_name and v.owner = t.owner
AND t.iot_type = 'IOT' AND t.partitioned = 'YES';
```

VARRAY 列およびそのデータを保存する必要がない場合、OWNER.PARENT_TABLE_NAME 列で表される表の PARENT_TABLE_COLUMN 列で表される各列を削除します。

```
ALTER TABLE OWNER.PARENT_TABLE_NAME DROP COLUMN PARENT_TABLE_COLUMN;
```

ただし、VARRAY 列を保存する必要がある場合は、対応する非パーティション索引構成表を作成できます。

```
CREATE non-partitioned index-organized table ... AS SELECT * FROM OWNER.PARENT_
TABLE_NAME;
DROP TABLE OWNER.PARENT_TABLE_NAME;
```

データ型

この項では、リリース 1 (9.0.1) 以上で有効なデータ型を使用禁止にする方法について説明します。

日時データ型および期間データ型の使用の中断 リリース 8.1.7 以下にダウングレードする前に、次の日時データ型および期間データ型を削除する必要があります。

- TIMESTAMP
- TIMESTAMP WITH TIME ZONE
- TIMESTAMP WITH LOCAL TIME ZONE
- INTERVAL YEAR TO MONTH
- INTERVAL DAY TO SECOND

ただし、データ型が TIMESTAMP WITH LOCAL TIME ZONE の場合は、明示的に ALTER TABLE 文を発行して、TIMESTAMP WITH LOCAL TIME ZONE 列を DATE 列に変換できます。

ALTER TABLE 文は、表のすべての行をスキャンします。TIMESTAMP WITH LOCAL TIME ZONE データに秒の小数部分がある場合は、その小数部分を四捨五入して列の行データを更新します。TIMESTAMP WITH LOCAL TIME ZONE データの分フィールドが 60 を超える場合は、その分フィールドから 60 を引いて列の行データを更新します。TIMESTAMP WITH LOCAL TIME ZONE 列を DATE 列に変更すると、秒の小数部分およびタイムゾーンの調整のための情報が失われます。

次のいずれかのオブジェクトがデータベースに存在する場合、ダウングレードは正常に実行されません。

- これらの新しいデータ型の列を含む表
- これらの新しいデータ型の引数または結果で宣言された（スタンドアロンまたはパッケージに含まれる）プロシージャまたはファンクション
- これらの新しいデータ型の属性を持つオブジェクト型、あるいはこれらの新しいデータ型の引数または結果を持つメンバー・ファンクション
- 要素型がこれらの新しいデータ型の 1 つであるコレクション型

以前のリリースにダウングレードするには、これらのオブジェクトを削除する必要があります。

TIMESTAMP 型の列を持つ表をリストするには、次の SQL 文を発行します。

```
SELECT owner, table_name, column_name
FROM dba_tab_columns
WHERE data_type LIKE 'TIMESTAMP(%)';
```

この文の結果、リストされた各表の `TIMESTAMP` データ型の列を削除するか、または表全体を削除します。

`TIMESTAMP WITH TIME ZONE` 型の列を持つ表をリストするには、次の `SQL` 文を発行します。

```
SELECT owner, table_name, column_name
FROM dba_tab_columns
WHERE data_type LIKE 'TIMESTAMP(%) WITH TIME ZONE';
```

この文の結果、リストされた各表の `TIMESTAMP WITH TIME ZONE` データ型の列を削除するか、または表全体を削除します。

`TIMESTAMP WITH LOCAL TIME ZONE` 型の列を持つ表をリストするには、次の `SQL` 文を発行します。

```
SELECT owner, table_name, column_name
FROM dba_tab_columns
WHERE data_type LIKE 'TIMESTAMP(%) WITH LOCAL TIME ZONE';
```

この文の結果、リストされた各表の `TIMESTAMP WITH LOCAL TIME ZONE` データ型の列を削除するか、または表全体を削除します。

`INTERVAL YEAR TO MONTH` 型の列を持つ表をリストするには、次の `SQL` 文を発行します。

```
SELECT owner, table_name, column_name
FROM dba_tab_columns
WHERE data_type LIKE 'INTERVAL YEAR(%) TO MONTH';
```

この文の結果、リストされた各表の `INTERVAL YEAR TO MONTH` データ型の列を削除するか、または表全体を削除します。

`INTERVAL DAY TO SECOND` 型の列を持つ表をリストするには、次の `SQL` 文を発行します。

```
SELECT owner, table_name, column_name
FROM dba_tab_columns
WHERE data_type LIKE 'INTERVAL DAY(%) TO SECOND';
```

この文の結果、リストされた各表の `INTERVAL DAY TO SECOND` データ型の列を削除するか、または表全体を削除します。

`TIMESTAMP` 型の引数または結果で宣言されたプロシージャまたは関クションのリストを検索するには、次の `SQL` 文を発行します。

```
SELECT owner, object_name, package_name, argument_name
FROM all_arguments
WHERE data_type = 'TIMESTAMP';
```

TIMESTAMP WITH TIME ZONE 型の引数または結果で宣言されたプロシージャまたはファンクションのリストを検索するには、次の SQL 文を発行します。

```
SELECT owner, object_name, package_name, argument_name
       FROM all_arguments
       WHERE data_type = 'TIMESTAMP WITH TIME ZONE';
```

TIMESTAMP WITH LOCAL TIME ZONE 型の引数または結果で宣言されたプロシージャまたはファンクションのリストを検索するには、次の SQL 文を発行します。

```
SELECT owner, object_name, package_name, argument_name
       FROM all_arguments
       WHERE data_type = 'TIMESTAMP WITH LOCAL TIME ZONE';
```

INTERVAL YEAR TO MONTH 型の引数または結果で宣言されたプロシージャまたはファンクションのリストを検索するには、次の SQL 文を発行します。

```
SELECT owner, object_name, package_name, argument_name
       FROM all_arguments
       WHERE data_type = 'INTERVAL YEAR TO MONTH';
```

INTERVAL DAY TO SECOND 型の引数または結果で宣言されたプロシージャまたはファンクションのリストを検索するには、次の SQL 文を発行します。

```
SELECT owner, object_name, package_name, argument_name
       FROM all_arguments
       WHERE data_type = 'INTERVAL DAY TO SECOND';
```

TIMESTAMP 型の属性を持つオブジェクト型、あるいは TIMESTAMP 型の引数または結果を持つメンバー・ファンクションを検索するには、次の SQL 文を発行します。

```
SELECT owner, type_name, attr_name
       FROM dba_type_attrs
       WHERE attr_type_name = 'TIMESTAMP';
```

```
SELECT owner, type_name, method_name, param_name
       FROM dba_method_params
       WHERE param_type_name = 'TIMESTAMP';
```

```
SELECT owner, type_name, method_name
       FROM dba_method_results
       WHERE result_type_name = 'TIMESTAMP';
```

TIMESTAMP WITH TIME ZONE 型の属性を持つオブジェクト型、あるいは TIMESTAMP WITH TIME ZONE 型の引数または結果を持つメンバー・ファンクションを検索するには、次の SQL 文を発行します。

```
SELECT owner, type_name, attr_name
FROM dba_type_attrs
WHERE attr_type_name = 'TIMESTAMP WITH TIME ZONE';
```

```
SELECT owner, type_name, method_name, param_name
FROM dba_method_params
WHERE param_type_name = 'TIMESTAMP WITH TIME ZONE';
```

```
SELECT owner, type_name, method_name
FROM dba_method_results
WHERE result_type_name = 'TIMESTAMP WITH TIME ZONE';
```

TIMESTAMP WITH LOCAL TIME ZONE 型の属性を持つオブジェクト型、あるいは
TIMESTAMP WITH LOCAL TIME ZONE 型の引数または結果を持つメンバー・ファンクシ
ョンを検索するには、次の SQL 文を発行します。

```
SELECT owner, type_name, attr_name
FROM dba_type_attrs
WHERE attr_type_name = 'TIMESTAMP WITH LOCAL TIME ZONE';
```

```
SELECT owner, type_name, method_name, param_name
FROM dba_method_params
WHERE param_type_name = 'TIMESTAMP WITH LOCAL TIME ZONE';
```

```
SELECT owner, type_name, method_name
FROM dba_method_results
WHERE result_type_name = 'TIMESTAMP WITH LOCAL TIME ZONE';
```

INTERVAL YEAR TO MONTH 型の属性を持つオブジェクト型、あるいは INTERVAL YEAR
TO MONTH 型の引数または結果を持つメンバー・ファンクションを検索するには、次の SQL
文を発行します。

```
SELECT owner, type_name, attr_name
FROM dba_type_attrs
WHERE attr_type_name = 'INTERVAL YEAR TO MONTH';
```

```
SELECT owner, type_name, method_name, param_name
FROM dba_method_params
WHERE param_type_name = 'INTERVAL YEAR TO MONTH';
```

```
SELECT owner, type_name, method_name
FROM dba_method_results
WHERE result_type_name = 'INTERVAL YEAR TO MONTH';
```

INTERVAL DAY TO SECOND 型の属性を持つオブジェクト型、あるいは INTERVAL DAY
TO SECOND 型の引数または結果を持つメンバー・ファンクションを検索するには、次の
SQL 文を発行します。

```
SELECT owner, type_name, attr_name
FROM dba_type_attrs
WHERE attr_type_name = 'INTERVAL DAY TO SECOND';
```

```
SELECT owner, type_name, method_name, param_name
FROM dba_method_params
WHERE param_type_name = 'INTERVAL DAY TO SECOND';
```

```
SELECT owner, type_name, method_name
FROM dba_method_results
WHERE result_type_name = 'INTERVAL DAY TO SECOND';
```

TIMESTAMP 型の要素を持つコレクション型のリストを検索するには、次の SQL 文を発行します。

```
SELECT owner, type_name, coll_type
FROM dba_coll_types
WHERE elem_type_name = 'TIMESTAMP';
```

TIMESTAMP WITH TIME ZONE 型の要素を持つコレクション型のリストを検索するには、次の SQL 文を発行します。

```
SELECT owner, type_name, coll_type
FROM dba_coll_types
WHERE elem_type_name = 'TIMESTAMP WITH TIME ZONE';
```

TIMESTAMP WITH LOCAL TIME ZONE 型の要素を持つコレクション型のリストを検索するには、次の SQL 文を発行します。

```
SELECT owner, type_name, coll_type
FROM dba_coll_types
WHERE elem_type_name = 'TIMESTAMP WITH LOCAL TIME ZONE';
```

INTERVAL YEAR TO MONTH 型の要素を持つコレクション型のリストを検索するには、次の SQL 文を発行します。

```
SELECT owner, type_name, coll_type
FROM dba_coll_types
WHERE elem_type_name = 'INTERVAL YEAR TO MONTH';
```

INTERVAL DAY TO SECOND 型の要素を持つコレクション型のリストを検索するには、次の SQL 文を発行します。

```
SELECT owner, type_name, coll_type
FROM dba_coll_types
WHERE elem_type_name = 'INTERVAL DAY TO SECOND';
```

ユーザー定義データ型

この項では、リリース 1 (9.0.1) で有効なユーザー定義データ型に関連する機能を使用禁止にする方法について説明します。

ユーザー定義集計関数の削除 リリース 8.1.7 以下にダウングレードする前に、すべてのユーザー定義集計関数を削除します。既存のユーザー定義集計関数を識別するには、次の SQL 文を発行します。

```
SELECT procedure_name FROM dba_procedures
       WHERE aggregate = 'YES';
```

リストされたすべての集計関数を削除します。

すべての発展型およびそれに依存する型と表の削除 リリース 8.1.7 以下にダウングレードする前に、すべての発展型およびそれに依存する型と表を削除する必要があります。すべての発展型を識別するには、次の SQL 文を発行します。

```
SELECT UNIQUE owner, type_name
       FROM dba_types
       WHERE version_name != '$8.0';
```

発展型を参照するすべての表を識別するには、次の SQL 文を発行します。

```
SELECT UNIQUE owner, table_name
       FROM dba_tab_columns
       WHERE data_type_owner IS NOT NULL
              AND version_name != '$8.0';
```

サブタイプおよび NOT FINAL 型の使用の中断 リリース 8.1.7 以下にダウングレードする前に、表のサブタイプおよび NOT FINAL 型すべての使用を中断します。表の既存のサブタイプおよび NOT FINAL 型の使用を識別するには、次の SQL 文を発行します。

```
SELECT c.name AS COLUMN_NAME, o.name AS TABLE_NAME, u.name AS TABLE_OWNER
       FROM user$ u, sys.obj$ o, sys.col$ c, sys.coltype$ ct, sys.type$ t
       WHERE u.user# = o.owner# AND o.obj# = c.obj# AND c.obj# = ct.obj#
              AND c.intcol# = ct.intcol# and ct.toid = t.toid AND o.type# = 2
              AND BITAND(t.properties, 3153928) > 0;
```

SQL および PL/SQL

次の項では、SQL および PL/SQL のダウングレード問題について説明します。これらの項で説明する処置を行うと、SQL スクリプトおよびストアド・プロシージャでのコンパイルおよび実行中のエラーを回避できます。必須ではありませんが、オラクル社ではダウングレード前にこれらの処置を行うことをお勧めします。

パイプライン表関数の使用の中断 リリース 8.1.7 以下にダウングレードする前に、すべてのパイプライン表関数の使用を中断します。既存のパイプライン表関数を識別するには、次の SQL 文を発行します。

```
SELECT procedure_name FROM dba_procedures
WHERE pipelined = 'YES';
```

パラレル表関数の使用の中断 リリース 8.1.7 以下にダウングレードする前に、すべてのパラレル表関数の使用を中断します。既存のパラレル表関数を識別するには、次の SQL 文を発行します。

```
SELECT procedure_name FROM dba_procedures
WHERE parallel = 'YES';
```

制約およびトリガー

この項では、制約およびトリガーに関する非互換性の削除について説明します。

すべてのビュー制約の削除 リリース 8.1.7 以下にダウングレードする前に、主キー、一意キーおよび外部キー制約に関連するすべてのビューを削除します。既存のビューの制約を識別するには、次の SQL 文を発行します。

```
SELECT * FROM dba_constraints WHERE view_related = 'DEPEND_ON_VIEW';
```

OLAP

この項では、OLAP に関する非互換性の削除について説明します。

OLAP の非互換性の削除 リリース 8.1.7 以下にダウングレードする前に、次のスクリプトを実行して、OLAP に関する非互換性を削除します。

```
ORACLE_HOME/olap/admin/olapidrp.sql
```

データベース互換性のリセット

ダウングレード先のリリースとの非互換性をすべて削除した後、データベースの互換性レベルを（以前のリリースに対する互換性に）リセットします。

参照： 5-7 ページの「[COMPATIBLE 初期化パラメータ値を減らす](#)」を参照してください。

COMPATIBLE 初期化パラメータ値を減らしても非互換性がまだ存在する場合は、データベースをオープンできません。その場合、再度 COMPATIBLE 初期化パラメータを増やします。その後、非互換性を削除し、データベースの互換性を再設定します。ダウングレード先のリリースのすべての非互換性を削除してから、ダウングレード処理を続行します。

参照： 非互換性の削除の詳細は、7-2 ページの「[非互換性の削除](#)」を参照してください。

特定のコンポーネントのダウングレード

Oracle データベース・サーバーのコンポーネントによっては、前述の一般的なダウングレード手順とは別の処理（7-22 ページの「[データベースのダウングレード](#)」を参照）が必要な場合があります。表 7-1 に、コンポーネントおよびそのダウングレード状態を示します。

表 7-1 コンポーネントのダウングレード状態

インストール済のコンポーネント	自動ダウングレードの有無
Oracle9i Catalog Views	はい
Oracle9i Packages and Types	はい
Oracle JVM	はい
Oracle9i Java Packages	はい
Oracle XDK for Java	はい
Messaging Gateway	はい
Oracle Text	いいえ
Oracle9i Real Application Clusters	はい
Oracle Workspace Manager	はい
Oracle Data Mining	はい
Oracle Ultra Search	いいえ
OLAP Catalog	はい
OLAP Analytic Workspace	はい
Oracle Spatial	いいえ
Oracle <i>interMedia</i>	いいえ
Oracle Visual Information Retrieval	いいえ
Oracle Label Security	はい

次の処理を完了して、自動的にダウングレードされないコンポーネントをダウングレードします。

Oracle Spatial のダウングレード

Oracle システムに Oracle Spatial がインストールされている場合は、Oracle Spatial のダウングレード方法について、『Oracle Spatial ユーザーズ・ガイドおよびリファレンス』を参照してください。

interMedia のダウングレード

Oracle *interMedia* のダウングレード方法については、UNIX プラットフォームでは `$ORACLE_HOME/ord/im/admin/README.txt` を、Windows プラットフォームでは `%ORACLE_HOME%\ord\im\admin\README.txt` を参照してください。

Oracle Visual Information Retrieval のダウングレード

Oracle Visual Information Retrieval のダウングレード方法については、UNIX プラットフォームでは `$ORACLE_HOME/ord/im/admin/README.txt` を、Windows プラットフォームでは `%ORACLE_HOME%\ord\im\admin\README.txt` を参照してください。

Oracle Text のダウングレード

Oracle システムに Oracle Text がインストールされている場合は、次の手順を実行します。

1. リリース 2 (9.2) の Oracle ホーム・ディレクトリの所有者としてシステムにログインします。
2. システム・プロンプトで、`ORACLE_HOME/ctx/admin` ディレクトリへ移動します。
3. SQL*Plus を起動します。
4. ユーザー CTXSYS. としてデータベース・インスタンスに接続します。
5. インスタンスが実行されている場合は、`SHUTDOWN IMMEDIATE` を使用してインスタンスを停止します。

```
SQL> SHUTDOWN IMMEDIATE
```

6. RESTRICT モードでインスタンスを起動します。

```
SQL> STARTUP RESTRICT
```

PFILE オプションを使用して、初期化パラメータ・ファイルの場所を指定する必要があります。

- 適切なダウングレード・スクリプトを実行して、Oracle Text をダウングレードします。
リリース 8.1.7 にダウングレードしている場合は、次のスクリプトを実行します。

```
SQL> @d0801070.sql
```

リリース 1 (9.0.1) にダウングレードしている場合は、次のスクリプトを実行します。

```
SQL> @d0900010.sql
```

- インスタンスを停止します。

```
SQL> SHUTDOWN IMMEDIATE
```

- SQL*Plus を終了します。

Oracle Ultra Search のダウングレード

Oracle システムに Oracle Ultra Search がインストールされている場合は、次の手順を実行します。

- SQL*Plus を起動します。
- ユーザー WKSYS としてデータベース・インスタンスに接続します。
- 次の SQL 文を発行します。

```
SQL> EXECUTE dbms_registry.downgraded('WK','9.0.1.0.0');
```

- SQL*Plus を終了します。

データベースのダウングレード

この項のダウングレードの手順を行う前に、データベースがダウングレード先のリリースと互換性があることを確認してください。

参照： まだ非互換性を削除していない場合は、7-2 ページの「[非互換性の削除](#)」を参照してください。

リリース 2 (9.2) のデータベースをそれ以前のリリースの Oracle へダウングレードするには、次の手順を実行します。

1. リリース 2 (9.2) の Oracle ホーム・ディレクトリの所有者としてシステムにログインします。
2. システム・プロンプトで、`ORACLE_HOME/rdbms/admin` ディレクトリへ移動します。
3. SQL*Plus を起動します。
4. SYSDBA 権限を持つユーザーとして、データベース・インスタンスに接続します。
5. MIGRATE モードでインスタンスを起動します。

```
SQL> STARTUP MIGRATE
```

PFILE オプションを使用して、初期化パラメータ・ファイルの場所を指定する必要がある場合があります。

6. 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。

```
SQL> SPOOL downgrade.log
```

7. `old_release` がダウングレード先のリリースを参照する場合、`dold_release.sql` を実行します。適切なスクリプトの選択は、表 7-2 を参照してください。各スクリプトによって、ダウングレード先に指定したリリースに直接ダウングレードできます。

スクリプトを実行するには、次のように入力します。

```
SQL> @dold_release.sql
```

表 7-2 ダウングレード・スクリプト

ダウングレード先	実行スクリプト
9.0.1	d0900010.sql
8.1.7	d0801070.sql

注意： ダウングレード先のリリースが表 7-2 に含まれていない場合、実行する正しいダウングレード・スクリプトについては、新しいインストールの README ファイルを参照してください。

スクリプトの実行についての注意は、次のとおりです。

- リリース 2 (9.2) に付属するスクリプトのバージョンを使用してください。
- リリース 2 (9.2) の環境でスクリプトを実行する必要があります。
- ダウングレードが複数のリリースにわたる場合でも、1つのスクリプトのみを実行する必要があります。たとえば、リリース 8.1.7 にダウングレードする場合は、`d0801070.sql` のみを実行します。

このスクリプトまたは残りの手順のいずれかのスクリプトを実行中になんらかの問題が発生した場合は、問題を解決してスクリプトを再実行します。必要に応じて、この章で説明されているスクリプトのいずれかを実行できます。

8. 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。

```
SQL> SPOOL OFF
```

次に、スプール・ファイルを確認して、すべてのパッケージおよび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ます。手順 6 で、スプール・ファイル名を `downgrade.log` としました。このファイルで見つけた問題を処理し、必要に応じて適切なダウングレード・スクリプトを再実行します。

9. インスタンスを停止します。

```
SQL> SHUTDOWN IMMEDIATE
```

クラスタ・データベースをダウングレードする場合は、すべてのインスタンスを停止します。

10. SQL*Plus を終了します。

11. リリース 1 (9.0.1) にダウングレードする場合、リリース 2 (9.2) の Oracle ホームからリリース 1 (9.0.1) の Oracle ホームに次のファイルをコピーします。

コンポーネント	リリース 2 (9.2) の Oracle ホームからのコピー	以前の Oracle ホームへのコピー
Oracle JVM	<code>ORACLE_HOME/javavm/install/jvmd901.sql</code>	<code>ORACLE_HOME/javavm/install</code>
Oracle XDK for Java	<code>ORACLE_HOME/xdk/admin/xmld901.sql</code>	<code>ORACLE_HOME/xdk/admin</code>
Messaging Gateway	<code>ORACLE_HOME/mgw/admin/mgwd901.sql</code>	<code>ORACLE_HOME/mgw/admin</code>
Oracle Workspace Manager	<code>ORACLE_HOME/rdbms/admin/owmd901.plb</code>	<code>ORACLE_HOME/rdbms/admin</code>

リリース 8.1.7 にダウングレードする場合、リリース 2 (9.2) の Oracle ホームからリリース 8.1.7 の Oracle ホームに次のファイルをコピーします。

コンポーネント	リリース 2 (9.2) の Oracle ホームからのコピー	以前の Oracle ホームへのコピー
Oracle JVM	ORACLE_HOME/javavm/install/jvmd817.sql	ORACLE_HOME/javavm/install
Oracle XDK for Java	ORACLE_HOME/xdk/admin/xml817.sql	ORACLE_HOME/oracore/admin
Oracle Label Security	ORACLE_HOME/rdbms/admin/olsd817.sql	ORACLE_HOME/lbac/admin

12. ご使用のオペレーティング・システムが UNIX の場合、次の環境変数がダウングレード先のリリースのディレクトリを指定するように変更してください。
- ORACLE_HOME
 - PATH
 - ORA_NLS33
 - LD_LIBRARY_PATH

注意： クラスタ・データベースをダウングレードしている場合は、このクラスタ・データベースがインスタンスを構成しているすべてのノードでこの手順を実行してください。

参照： ご使用のオペレーティング・システムでのその他の重要な環境変数の設定は、オペレーティング・システム固有の Oracle9i インストール・ガイドを参照してください。

13. ご使用のオペレーティング・システムが Windows の場合、次の手順を実行します。
- a. リリース 2 (9.2) のデータベースのすべての Oracle サービス (OracleServiceSID など) を停止します。SID はインスタンスの名前です。たとえば、SID が ORCL の場合、コマンド・プロンプトで次のように入力します。
- ```
C:¥> NET STOP OracleServiceORCL
```

**参照：** サービスの停止方法については、『Oracle9i Database for Windows 管理者ガイド』を参照してください。

- b. コマンド・プロンプトで ORADIM コマンドを発行して、Oracle サービスを削除します。たとえば、SID が ORCL の場合、次のコマンドを入力します。

```
C:¥> ORADIM -DELETE -SID ORCL
```

- c. コマンド・プロンプトで ORADIM コマンドを使用して、ダウングレードするデータベースの Oracle サービスを作成します。

```
C:¥> ORADIM -NEW -SID SID -INTPWD PASSWORD -MAXUSERS USERS
-STARTMODE AUTO -PFILE ORACLE_HOME¥DATABASE¥INITSID.ORA
```

この構文には次の変数が含まれます。

|             |                                                                                                                        |
|-------------|------------------------------------------------------------------------------------------------------------------------|
| SID         | ダウングレードするデータベースと同じ SID 名です。                                                                                            |
| PASSWORD    | データベース・インスタンスのパスワードです。SYSDBA 権限で接続したユーザー用のパスワードです。-INTPWD オプションは必須ではありません。これを指定しない場合、オペレーティング・システムの認証が使用され、パスワードは不要です。 |
| USERS       | SYSDBA 権限および SYSOPER 権限を付与できるユーザーの最大数。                                                                                 |
| ORACLE_HOME | ダウングレード先のデータベースの Oracle ホーム・ディレクトリです。-PFILE オプションを使用して、(Oracle ホーム・ディレクトリのドライブを含む) フルパス名を指定する必要があります。                  |

たとえば、ダウングレード先がリリース 8.1.7、SID が ORCL、PASSWORD が TWxy579、USERS の最大数が 10、ORACLE\_HOME ディレクトリが C:¥ORANT の場合、次のコマンドを入力します。

```
C:¥> ORADIM -NEW -SID ORCL -INTPWD TWxy579 -MAXUSERS 10
-STARTMODE AUTO -PFILE C:¥ORANT¥DATABASE¥INITORCL.ORA
```

14. サーバー・パラメータ・ファイルを使用してインスタンスを起動する場合、または初期化パラメータ・ファイルが SPFILE (サーバー・パラメータ・ファイル) エントリを含む場合は、次の手順を実行します。

- a. サーバー・パラメータ・ファイルを従来の初期化パラメータ・ファイルへエクスポートします。

```
CREATE PFILE[=pfile-name] FROM SPFILE[=spfile-name];
```

初期化パラメータ・ファイルはテキスト・ファイルとして作成されます。Oracle9i Real Application Clusters 環境では、すべてのインスタンスのパラメータ設定が初期化パラメータ・ファイルに含まれます。



- b. SPFILE パラメータを使用してサーバー・パラメータ・ファイルを指定していた場合は、インスタンスの起動に使用する初期化パラメータ・ファイル内の SPFILE パラメータを IFILE に変更します。IFILE パラメータがサーバー・パラメータ・ファイルからエクスポートした初期化パラメータ・ファイルを指定していることを確認します。
- c. Oracle9i Real Application Clusters を使用している場合は、インスタンス固有の初期化パラメータ・ファイルを作成します。サーバー・パラメータ・ファイルからエクスポートした初期化パラメータ・ファイルから、すべてのインスタンス固有のパラメータを削除します。

インスタンス固有の各パラメータ・ファイル内の IFILE パラメータを使用して、サーバー・パラメータ・ファイルからエクスポートした初期化パラメータ・ファイルを指定できます。

15. リリース 2 (9.2) の Oracle ホーム・ディレクトリから、ダウングレードするリリースの Oracle ホームに構成ファイルをコピーします。
  - a. リリース 2 (9.2) の Oracle ホームから、ダウングレードするリリースの Oracle ホームにパラメータ・ファイルをコピーします。デフォルトでは、Oracle はパラメータ・ファイルを UNIX プラットフォームの場合は `$ORACLE_HOME/dbs` で、Windows オペレーティング・システムの場合は `%ORACLE_HOME%\database` で検索します。初期化パラメータ・ファイルは任意の場所に格納できます。ただし、リリース 2 (9.2) の Oracle ホーム内には格納できません。
  - b. 初期化パラメータ・ファイルに IFILE (インクルード・ファイル) エントリがあり、IFILE エントリで指定されたファイルがリリース 2 (9.2) の Oracle ホーム・ディレクトリに存在する場合、IFILE エントリが指定したファイルを、ダウングレードするリリースの Oracle ホームにコピーします。IFILE エントリ内に指定されたファイルには、追加の初期化パラメータがあります。このファイルをコピーした後、パラメータ・ファイルを編集して新しい場所を指定します。
  - c. リリース 2 (9.2) の Oracle ホームに存在するパスワード・ファイルがある場合は、そのパスワード・ファイルを、ダウングレード先のリリースの Oracle ホームに移動またはコピーします。パスワード・ファイルの名前と位置は、オペレーティング・システムによって異なります。UNIX プラットフォームでは、デフォルトのパスワード・ファイルは `$ORACLE_HOME/dbs/orapwsid` です。Windows オペレーティング・システムでは、デフォルトのパスワード・ファイルは `%ORACLE_HOME%\database\pwsid.ora` です。両方とも、`sid` は Oracle インスタンスの ID です。

---

**注意：** クラスタ・データベースをダウングレードしている場合は、このクラスタ・データベースがインスタンスを構成しているすべてのノードでこの手順を実行してください。

- リリース 1 (9.0.1) へダウングレードしている場合は、  
CLUSTER\_DATABASE 初期化パラメータを `false` に設定します。
- リリース 8.1.7 へダウングレードしている場合は、PARALLEL\_SERVER  
初期化パラメータを `false` に設定します。

ダウングレード後、この該当する初期化パラメータの設定を `true` に戻す必要があります。

---

16. 次の初期化パラメータをパラメータ・ファイルに追加します。

```
_SYSTEM_TRIG_ENABLED = false
JOB_QUEUE_PROCESSES = 0
AQ_TM_PROCESSES = 0
```

リリース 1 (9.0.1) にダウングレードする場合、次の初期化パラメータをパラメータ・ファイルに追加します。

```
NLS_LENGTH_SEMANTICS = BYTE
```

前述の初期化パラメータは、ダウングレード完了後にパラメータ・ファイルから削除する必要があります。

17. システム・プロンプトで、以前のリリースの `ORACLE_HOME/rdbms/admin` ディレクトリへ移動します。
18. SQL\*Plus を起動します。

---

**注意：** リリース 8.1.7 にダウングレードする場合は、Server Manager を起動してください。SQL\*Plus は起動しないでください。

---

19. SYSDBA 権限を持つユーザーとして、データベース・インスタンスに接続します。
20. インスタンスを起動します。

UNIX プラットフォームでは、次の手順に従います。

```
STARTUP RESTRICT
```

PFILE オプションを使用して、初期化パラメータ・ファイルの場所を指定する必要がある場合があります。

Windows オペレーティング・システムでは、次の手順に従います。

```
ALTER SYSTEM ENABLE RESTRICTED SESSION;
```

21. 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。

```
SPOOL old_scripts.log
```

22. utlip.sql を実行します。

```
@utlip.sql
```

utlip.sql スクリプトは一定のディクショナリ表を変更し、データベースが必要とする形式で次の再コンパイルが発生するように、既存のすべての PL/SQL モジュールを無効にします。また、PL/SQL コンパイルに必要なパッケージ STANDARD および DBMS\_STANDARD を再ロードします。

**参照：** ワード・サイズ変更の詳細は、1-10 ページの「[ワード・サイズの変更](#)」を参照してください。

23. 次のカタログ・スクリプトを実行します。

- a. catalog.sql を実行します。

```
@catalog.sql
```

- b. catproc.sql を実行します。

```
@catproc.sql
```

- c. リリース 8.1.7 にダウングレードする場合は、catrep.sql を実行します。

```
@catrep.sql
```

- d. クラスタ・データベースをリリース 1 (9.0.1) にダウングレードしている場合は、catclust.sql を実行します。

```
@catclust.sql
```

クラスタ・データベースをリリース 8.1.7 にダウングレードしている場合は、catparr.sql を実行します。

```
@catparr.sql
```

- e. 必要に応じて、その他のカタログ・スクリプトも実行します。たとえば、異機種間サービスのデータ・ディクショナリ・ビュー、データ・ディクショナリ表およびデータ・ディクショナリ・パッケージを再作成するには、`caths.sql` を実行します。

```
@caths.sql
```

インスタンスを停止し、再起動します。

```
SHUTDOWN IMMEDIATE
```

```
STARTUP
```

- 24. 以前のデータベースに Oracle JVM をインストールしている場合は、適切なダウングレード・スクリプト（手順 11 で以前の Oracle ホームへコピー済）を実行して、Oracle JVM をダウングレードします。スクリプトを実行するときは、`ORACLE_HOME` を以前の Oracle ホーム・ディレクトリのフルパスに置き換えます。

リリース 1 (9.0.1) にダウングレードしている場合は、次のスクリプトを実行します。

```
@ORACLE_HOME/javavm/install/jvmd901.sql
```

リリース 8.1.7 にダウングレードしている場合は、次のスクリプトを実行します。

```
@ORACLE_HOME/javavm/install/jvmd817.sql
```

次のスクリプトを実行して、以前のデータベースに Java クラスを再ロードします。

```
@ORACLE_HOME/rdbms/admin/initsoxx.sql
```

```
@ORACLE_HOME/rdbms/admin/initjms.sql
```

リリース 1 (9.0.1) にダウングレードしている場合は、次のスクリプトを実行します。

```
@ORACLE_HOME/rdbms/admin/initcdc.sql
```

```
@ORACLE_HOME/rdbms/admin/initqsm.sql
```

```
@ORACLE_HOME/rdbms/admin/initstjty.sql
```

```
@ORACLE_HOME/rdbms/admin/initapcx.sql
```

- 25. 以前のデータベースに Oracle XDK for Java をインストールしている場合は、適切なダウングレード・スクリプト（手順 11 で次のディレクトリにコピー済）を実行して、Oracle XDK for Java をダウングレードします。ダウングレード・スクリプトを実行するときは、`ORACLE_HOME` を、ダウングレード先のリリースの Oracle ホーム・ディレクトリのフルパスに置き換えます。

リリース 1 (9.0.1) にダウングレードしている場合は、次のスクリプトを実行します。

```
@ORACLE_HOME/xdk/admin/xml901.sql
```

リリース 8.1.7 にダウングレードしている場合は、次のスクリプトを実行します。

```
@ORACLE_HOME/oracore/admin/xmlsd817.sql
```

インスタンスを停止し、再起動します。

```
SHUTDOWN IMMEDIATE
```

```
STARTUP
```

26. 以前のデータベースに **Messaging Gateway** をインストールしている場合は、適切なダウングレード・スクリプト（手順 11 で次のディレクトリにコピー済）を実行して、**Messaging Gateway** をダウングレードします。ダウングレード・スクリプトを実行するときは、**ORACLE\_HOME** を、ダウングレード先のリリースの **Oracle** ホーム・ディレクトリのフルパスに置き換えます。

リリース 1（9.0.1）にダウングレードしている場合は、次のスクリプトを実行します。

```
@ORACLE_HOME/mgw/admin/mgwd901.sql
```

27. 以前のデータベースに **Oracle Workspace Manager** をインストールしている場合は、適切なダウングレード・スクリプト（手順 11 で次のディレクトリにコピー済）を実行して **Oracle Workspace Manager** をダウングレードします。ダウングレード・スクリプトを実行するときは、**ORACLE\_HOME** を、ダウングレード先のリリースの **Oracle** ホーム・ディレクトリのフルパスに置き換えます。

リリース 1（9.0.1）にダウングレードしている場合は、次のスクリプトを実行します。

```
@ORACLE_HOME/rdbms/admin/owmd901.plb
```

28. 以前のデータベースに **Oracle Label Security** をインストールしている場合は、適切なダウングレード・スクリプトを実行して、**Oracle Label Security** をダウングレードします。スクリプトを実行するときは、**ORACLE\_HOME** を以前の **Oracle** ホーム・ディレクトリのフルパスに置き換えます。

リリース 1（9.0.1）にダウングレードしている場合は、次のスクリプトを実行します。

```
@ORACLE_HOME/rdbms/admin/catols.sql
```

リリース 8.1.7 にダウングレードしている場合は、次の文を発行します。

```
GRANT RESTRICTED SESSION TO LBACSYS;
```

```
CONNECT LBACSYS/LBACSYS
```

```
@ORACLE_HOME/lbac/admin/olsd817.sql
```

**SYSDBA** 権限を持つユーザーとして、データベース・インスタンスに接続します。

```
REVOKE RESTRICTED SESSION FROM LBACSYS;
```

29. utlrlp.sql を実行します。この手順はオプションで、ワード・サイズに変更がある場合でも実行できます。

```
@utlrlp.sql
```

utlrlp.sql スクリプトは、以前 INVALID 状態だったパッケージ、プロシージャ、型などのすべての既存の PL/SQL モジュールを、再コンパイルします。これらの処置はオプションですが、これらの処理を行うと、再コンパイルはダウングレード後ではなくダウングレード中に行われます。

オラクル社では、このオプションの utlrlp.sql を実行することをお勧めします。

30. 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。

```
SPOOL OFF
```

次に、スプール・ファイルを確認して、すべてのパッケージおよび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ます。手順 21 で、スプール・ファイル名を old\_scripts.log としました。このファイルで見つけた問題を処理し、必要に応じて適切なスクリプトを再実行します。

31. インスタンスを停止します。

```
SHUTDOWN IMMEDIATE
```

---

---

### 注意：

- リリース 1 (9.0.1) ヘダウングレードしている場合は、CLUSTER\_DATABASE 初期化パラメータを false に設定します。
- リリース 8.1.7 ヘダウングレードしている場合は、PARALLEL\_SERVER 初期化パラメータを false に設定します。

ダウングレード処理が完了した後、true に戻すことができます。

---

---

32. 手順 18 で起動した Server Manager または SQL\*Plus を終了します。

33. 手順 16 で追加した初期化パラメータをパラメータ・ファイルから削除します。  
データベースがダウングレードされました。

---

# エクスポート / インポートによる データベースの移行

この章では、エクスポート / インポート・ユーティリティを使用して、Oracle データベースのデータをアップグレードおよびダウングレードする処理について説明します。この章では、次の項目について説明します。

- [エクスポート・ダンプ・ファイルの互換性](#)
- [ソース・データベースおよびターゲット・データベース](#)
- [エクスポート / インポートを使用したソース・データベースのアップグレード](#)

**参照：** エクスポート / インポート・ユーティリティの詳細は、『Oracle9i データベース・ユーティリティ』を参照してください。

# エクスポート・ダンプ・ファイルの互換性

エクスポート・ダンプ・ファイルは、Oracle のすべての上位リリースにインポートできます。たとえば、Oracle7 のエクスポート・ダンプ・ファイルは、リリース 8.1.7、リリース 1 (9.0.1) およびリリース 2 (9.2) のインポート・ユーティリティでインポートできます。

ただし、エクスポート・ダンプ・ファイルは、それより前の Oracle リリースのインポート・ユーティリティとは下位互換性がありません。エクスポートされたデータは、以前の Oracle リリースのインポート・ユーティリティではインポートできません。たとえば、リリース 8.1.7 のエクスポート・ダンプ・ファイルは、リリース 8.0.6 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。また、Oracle9i のエクスポート・ダンプ・ファイルは、Oracle7 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。

以前のリリースのエクスポート・ユーティリティおよびインポート・ユーティリティを使用してデータをエクスポートおよびインポートしている場合は、データベースの内容を以前の Oracle リリースにインポートすることができます。表 8-1 に、このサポートの詳細を示します。

表 8-1 エクスポート/インポートの下位互換性サポート

| データのエクスポート元    | インポート先         | 使用するエクスポート/インポート・ユーティリティ                   |
|----------------|----------------|--------------------------------------------|
| リリース 2 (9.2)   | リリース 7.3.4     | リリース 7.3.4                                 |
| リリース 1 (9.0.1) |                | 注意: エクスポートする前に、catexp7.sql スクリプトを実行してください。 |
| リリース 8.1.7     |                |                                            |
| リリース 8.0.6     |                |                                            |
| リリース 2 (9.2)   | リリース 8.0.6     | リリース 8.0.6                                 |
| リリース 1 (9.0.1) |                |                                            |
| リリース 8.1.7     |                |                                            |
| リリース 2 (9.2)   | リリース 8.1.7     | リリース 8.1.7                                 |
| リリース 1 (9.0.1) |                |                                            |
| リリース 2 (9.2)   | リリース 1 (9.0.1) | リリース 1 (9.0.1)                             |

表 8-1 に示すとおり、Oracle8 以上のデータベースから Oracle7 データベースへデータをエクスポートするには、Oracle7 のエクスポート・ユーティリティを使用してデータをエクスポートする前に、まず Oracle8 以上のデータベースで catexp7.sql スクリプトを実行する必要があります。

データを Oracle8i 以上のデータベースから Oracle8 データベースにエクスポートする場合は、catexp7.sql スクリプトを実行する必要はありません。



## 以前のリリースと互換性がないデータでのエクスポート/インポートの使用方法

データを以前のリリースにエクスポートする場合、以前のリリースと互換性のないデータはエクスポートできないか、またはいくつかの機能が失われた状態でエクスポートされます。

たとえば、パーティション表は Oracle7 のエクスポート・ユーティリティではエクスポートされません。パーティション表を Oracle8 以上のデータベースから Oracle7 データベースに移動する必要がある場合は、まずその表をパーティション化されていない表へ再編成します。また、Oracle8i 以上で実行者権限を使用するプロシージャをエクスポート/インポートする場合にもこの問題が発生します。Oracle8 のエクスポート・ユーティリティを使用する場合、これらのプロシージャはエクスポートされますが、Oracle8 では正常に機能しません。これは、Oracle8 では実行者権限がサポートされていないためです。通常は、データを以前のリリースにエクスポートする必要がある場合は、データをエクスポートする前に、以前のリリースとの非互換性をできるだけ多く削除します。

## ソース・データベースおよびターゲット・データベース

ソース・データベースは、エクスポートするデータを含むデータベースです。ターゲット・データベースは、エクスポートしたデータをインポートするデータベースです。

### エクスポート・ユーティリティ要件

データベースをアップグレードするには、ソース・データベースの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（新しい Oracle9i 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・ユーティリティでインポートする前に、新しい Oracle9i データベースが作成済みですでに操作可能な状態である必要があります。

たとえば、Oracle7 リリース 7.3.4 から Oracle9i リリース 2 (9.2) へアップグレードする場合は、Oracle7 リリース 7.3.4 のエクスポート・ユーティリティを使用します。

### インポート・ユーティリティ要件

データベースをアップグレードするには、ターゲット・データベース（新しい Oracle9i データベース）の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。たとえば、Oracle7 リリース 7.3.4 から Oracle9i リリース 2 (9.2) へアップグレードする場合は、Oracle9i リリース 2 (9.2) のインポート・ユーティリティを使用します。

## エクスポート / インポートを使用したソース・データベースのアップグレード

エクスポート / インポート・ユーティリティを使用してデータベースをアップグレードするには、次の手順を実行します。

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・データベースのデータ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の詳細は、ソース・データベースのサーバー・ユーティリティのドキュメントを参照してください。

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ある必要があります。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新しい Oracle9i 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. 新しい Oracle9i ソフトウェアをインストールします。インストールは、オペレーティング・システム固有の作業です。Oracle9i のインストール手順は、ご使用のオペレーティング・システム固有の Oracle マニュアルを参照してください。
3. 新しい Oracle9i 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 Oracle9i 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. 新しい Oracle9i ターゲット・データベースを作成します。

**参照：** Oracle9i データベースの作成の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

5. SQL\*Plus を新しい Oracle9i 環境で起動します。
6. SYSDBA 権限を持つユーザーとして、データベース・インスタンスに接続します。
7. STARTUP を使用して、Oracle9i データベース・インスタンスを起動します。
8. 記憶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、ユーザーおよび表を事前に作成しておきます。SQL\*Plus 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する必要があります。

---

**注意：** 新しい Oracle9i データベースをソース・データベースと同じコンピュータ上に作成し、ソース・データベースのデータ・ファイルを上書きしない場合は、事前に表領域を作成し、インポート時に IGNORE=Y および DESTROY=N を指定する必要があります。

---

9. 新しい Oracle9i データベースのインポート・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。

**参照：** インポート・ユーティリティの詳細は、『Oracle9i データベース・ユーティリティ』を参照してください。

10. インポート作業が完了したら、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（失敗したものがある場合）を、インポート・ログ・ファイルで確認します。

**参照：** エラー処理情報については、『Oracle9i データベース・ユーティリティ』を参照してください。

11. その他のインポート方法（『Oracle9i データベース・ユーティリティ』を参照）を使用するか、またはソース・オブジェクトを作成する 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なおします（または、完全に新規のインポートを開始します）。
12. 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、それらの変更が新しい Oracle9i データベースに反映されていることを確認します。詳細は、8-4 ページの手順 1 を参照してください。
13. 第 4 章「データベースのアップグレード後の作業」に示す手順を実行します。



---

# 初期化パラメータおよびデータ・ディクショナリ の変更

この付録では、Oracle の異なるリリース間での初期化パラメータおよびデータ・ディクショナリの変更点を示します。また、特定の初期化パラメータにおける互換性の問題についても説明します。

この付録では、次の項目について説明します。

- [初期化パラメータの変更](#)
- [初期化パラメータの互換性の問題](#)
- [静的データ・ディクショナリ・ビューの変更](#)
- [動的パフォーマンス・ビューの変更](#)

---

**注意：** この付録では、Oracle8 での初期化パラメータおよびデータ・ディクショナリの変更点は示しません。Oracle7 からアップグレードする場合は、この付録で説明する変更点の他に、D-39 ページの「[リリース 8.0 での初期化パラメータおよびデータ・ディクショナリの変更](#)」を参照してください。

---

# 初期化パラメータの変更

次の項では、Oracle の異なるリリース間での初期化パラメータの変更点を示します。

- [非推奨の初期化パラメータ](#)
- [廃止された初期化パラメータ](#)

**参照：** Oracle9i の新しい初期化パラメータのリストは、『Oracle9i データベース・リファレンス』を参照してください。

## 非推奨の初期化パラメータ

次の項では、非推奨の初期化パラメータを示します。非推奨のパラメータは通常のパラメータと同様に動作しますが、非推奨のパラメータをパラメータ・ファイルに指定した場合、インスタンスの起動時に警告メッセージが表示されます。また、非推奨のすべてのパラメータは、インスタンスの起動時にアラート・ログに記録されます。

- [リリース 2 \(9.2\) で非推奨の初期化パラメータ](#)
- [リリース 1 \(9.0.1\) で非推奨の初期化パラメータ](#)

### リリース 2 (9.2) で非推奨の初期化パラメータ

次の初期化パラメータはリリース 2 (9.2) で非推奨のものです。

| 非推奨の初期化パラメータ | 新しい初期化パラメータ     |
|--------------|-----------------|
| DRS_START    | DG_BROKER_START |

### リリース 1 (9.0.1) で非推奨の初期化パラメータ

次の初期化パラメータはリリース 1 (9.0.1) で非推奨のものです。

| 非推奨の初期化パラメータ        | 新しい初期化パラメータ            |
|---------------------|------------------------|
| MTS_CIRCUITS        | CIRCUITS               |
| MTS_DISPATCHERS     | DISPATCHERS            |
| MTS_MAX_DISPATCHERS | MAX_DISPATCHERS        |
| MTS_MAX_SERVERS     | MAX_SHARED_SERVERS     |
| MTS_SERVERS         | SHARED_SERVERS         |
| MTS_SESSIONS        | SHARED_SERVER_SESSIONS |

| 非推奨の初期化パラメータ              | 新しい初期化パラメータ                |
|---------------------------|----------------------------|
| PARALLEL_SERVER           | CLUSTER_DATABASE           |
| PARALLEL_SERVER_INSTANCES | CLUSTER_DATABASE_INSTANCES |

## 廃止された初期化パラメータ

次の項では、廃止された初期化パラメータを示します。

- [リリース 2 \(9.2\)](#) で廃止された初期化パラメータ
- [リリース 1 \(9.0.1\)](#) で廃止された初期化パラメータ
- [リリース 8.1](#) で廃止された初期化パラメータ

**注意：** 1つ以上の廃止された初期化パラメータを使用すると、リリース 2 (9.2) データベースを起動できますが、警告が戻され、アラート・ログに記録されます。

### リリース 2 (9.2) で廃止された初期化パラメータ

次の初期化パラメータはリリース 2 (9.2) で廃止されたものです。

|                            |                          |
|----------------------------|--------------------------|
| DISTRIBUTED_TRANSACTIONS   | MAX_TRANSACTION_BRANCHES |
| PARALLEL_BROADCAST_ENABLED | STANDBY_PRESERVES_NAMES  |

### リリース 1 (9.0.1) で廃止された初期化パラメータ

次の初期化パラメータはリリース 1 (9.0.1) で廃止されたものです。

|                            |                            |
|----------------------------|----------------------------|
| ALWAYS_ANTI_JOIN           | ALWAYS_SEMI_JOIN           |
| DB_BLOCK_LRU_LATCHES       | DB_BLOCK_MAX_DIRTY_TARGET  |
| DB_FILE_DIRECT_IO_COUNT    | GC_DEFER_TIME              |
| GC_RELEASABLE_LOCKS        | GC_ROLLBACK_LOCKS          |
| HASH_MULTIBLOCK_IO_COUNT   | INSTANCE_NODESET           |
| JOB_QUEUE_INTERVAL         | OPS_INTERCONNECTS          |
| OPTIMIZER_PERCENT_PARALLEL | SORT_MULTIBLOCK_READ_COUNT |
| TEXT_ENABLE                |                            |

## リリース 8.1 で廃止された初期化パラメータ

次の初期化パラメータはリリース 8.1 で廃止されたものです。

|                                       |                                                                 |
|---------------------------------------|-----------------------------------------------------------------|
| ALLOW_PARTIAL_SN_RESULTS              | ARCH_IO_SLAVES                                                  |
| B_TREE_BITMAP_PLANS                   | BACKUP_DISK_IO_SLAVES                                           |
| CACHE_SIZE_THRESHOLD                  | CLEANUP_ROLLBACK_ENTRIES                                        |
| CLOSE_CACHED_OPEN_CURSORS             | COMPATIBLE_NO_RECOVERY                                          |
| COMPLEX_VIEW_MERGING                  | DB_BLOCK_CHECKPOINT_BATCH                                       |
| DB_BLOCK_LRU_EXTENDED_STATISTICS      | DB_BLOCK_LRU_STATISTICS                                         |
| DB_FILE_SIMULTANEOUS_WRITES           | DELAYED_LOGGING_BLOCK_CLEANOUTS                                 |
| DISCRETE_TRANSACTIONS_ENABLED         | DISTRIBUTED_RECOVERY_CONNECTION_HOLD_TIMEFAST_FULL_SCAN_ENABLED |
| ENT_DOMAIN_NAME                       | FREEZE_DB_FOR_FAST_INSTANCE_RECOVERY                            |
| GC_LATCHES                            | GC_LCK_PROCS                                                    |
| JOB_QUEUE_KEEP_CONNECTIONS            | LARGE_POOL_MIN_ALLOC                                            |
| LGWR_IO_SLAVES                        | LM_LOCKS                                                        |
| LM_PROCS                              | LM_RESS                                                         |
| LOCK_SGA_AREAS                        | LOG_ARCHIVE_BUFFER_SIZE                                         |
| LOG_ARCHIVE_BUFFERS                   | LOG_BLOCK_CHECKSUM                                              |
| LOG_FILES                             | LOG_SIMULTANEOUS_COPIES                                         |
| LOG_SMALL_ENTRY_MAX_SIZE              | MTS_LISTENER_ADDRESS                                            |
| MTS_MULTIPLE_LISTENERS                | MTS_RATE_LOG_SIZE                                               |
| MTS_RATE_SCALE                        | MTS_SERVICE                                                     |
| OGMS_HOME                             | OPS_ADMIN_GROUP                                                 |
| OPTIMIZER_SEARCH_LIMIT                | PARALLEL_DEFAULT_MAX_INSTANCES                                  |
| PARALLEL_MIN_MESSAGE_POOL             | PARALLEL_SERVER_IDLE_TIME                                       |
| PARALLEL_TRANSACTION_RESOURCE_TIMEOUT | PUSH_JOIN_PREDICATE                                             |
| REDUCE_ALARM                          | ROW_CACHE_CURSORS                                               |



|                                |                                   |
|--------------------------------|-----------------------------------|
| SEQUENCE_CACHE_ENTRIES         | SEQUENCE_CACHE_HASH_BUCKETS       |
| SHARED_POOL_RESERVED_MIN_ALLOC | SNAPSHOT_REFRESH_KEEP_CONNECTIONS |
| SNAPSHOT_REFRESH_PROCESSES     | SORT_DIRECT_WRITES                |
| SORT_READ_FAC                  | SORT_SPACEMAP_SIZE                |
| SORT_WRITE_BUFFER_SIZE         | SORT_WRITE_BUFFERS                |
| SPIN_COUNT                     | TEMPORARY_TABLE_LOCKS             |
| USE_ISM                        |                                   |

## 初期化パラメータの互換性の問題

この付録の前半で示した非推奨および廃止された初期化パラメータの一覧では、Oracle の異なるリリース間での初期化パラメータの変更点を示しました。ただし、特定の初期化パラメータの変更は、データベースの互換性に問題が発生する可能性があるため、特に注意が必要です。この項では、これらパラメータの変更について説明します。

## DB\_BLOCK\_CHECKSUM の新規デフォルト値

リリース 1 (9.0.1) 以上では、DB\_BLOCK\_CHECKSUM 初期化パラメータは新規のデフォルト値が設定されています。以前のリリースではデフォルト値が `false` でしたが、リリース 1 (9.0.1) 以上でのデフォルト値は `true` です。

**参照：** 詳細は、『Oracle9i データベース・リファレンス』の「DB\_BLOCK\_CHECKSUM」を参照してください。

## ジョブ・キュー・プロセスの最大数

Oracle9i では、1 つのインスタンスにつき、起動されるジョブ・キュー・プロセスのインスタンスは 1000 です。以前のリリースでは、最大 36 でした。JOB\_QUEUE\_PROCESSES 初期化パラメータは、ジョブ・キュー・プロセスの数を制御します。

**参照：** 詳細は、『Oracle9i データベース・リファレンス』の「JOB\_QUEUE\_PROCESSES」を参照してください。

## ORACLE\_TRACE\_ENABLE パラメータ

リリース 8.1.7 以上では、ORACLE\_TRACE\_ENABLE 初期化パラメータは動的です。デフォルト値は false です。

サーバーの Oracle Trace のコレクション機能を使用可能にするには、ALTER SYSTEM または ALTER SESSION を使用して、ORACLE\_TRACE\_ENABLE を true に設定します。この設定のみでは Oracle Trace のコレクション機能は起動されませんが、そのサーバーで Oracle Trace が使用可能になります。

ORACLE\_TRACE\_ENABLE を true に設定すると、次のいずれかの方法で、サーバー・イベント・データの Oracle Trace のコレクション機能を実行できます。

- Oracle Trace Manager アプリケーション（Oracle Diagnostics Pack とともに提供されます）を使用します。
- Oracle Trace のコマンドライン・インタフェース（サーバーとともに提供されます）を使用します。
- ORACLE\_TRACE\_COLLECTION\_NAME 初期化パラメータにコレクション名を指定します。

**参照：** 詳細は、『Oracle9i データベース・リファレンス』および『Oracle9i データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』を参照してください。

## SERIALIZABLE パラメータ

リリース 8.1.6 以上では、SERIALIZABLE 初期化パラメータに true を設定することはできません。これは「廃止」とは異なります。V\$PARAMETER データ・ディクショナリ・ビューでは、有効なパラメータとして表示されます。

今後、SERIALIZABLE が false に設定されているようにデフォルトで動作します。SET TRANSACTION ISOLATION LEVEL SERIALIZABLE コマンドを使用すると、同様のトランザクション分離動作を実行します。

ALTER SESSION SET ISOLATION\_LEVEL=SERIALIZABLE を使用すると、すべてのセッションでこの動作を実行できます。

## SORT\_AREA\_SIZE および SORT\_DIRECT\_WRITES パラメータ

リリース 8.1 以上では、SORT\_DIRECT\_WRITES 初期化パラメータは廃止されています。以前のリリースで SORT\_DIRECT\_WRITES を false または auto に設定した場合、ソート・バッファは、バッファ・キャッシュに保存されます（可能な場合）。SORT\_DIRECT\_WRITES はリリース 8.1 以上では廃止されたため、SORT\_AREA\_SIZE 初期化パラメータを調整しなければ、ソート・バッファを直接ディスクへ送ることができます。

以前のリリースに次のいずれかの条件が該当する場合、`SORT_AREA_SIZE` の値を大きくする必要があります。

- `SORT_DIRECT_WRITES` が `false` に設定されている。
- `SORT_DIRECT_WRITES` が `auto` に設定され、`SORT_AREA_SIZE` が 640KB 以下に設定されている。

このいずれかの条件が以前のリリースに該当する場合、パフォーマンスを向上させるために `SORT_AREA_SIZE` の値を大きくします。

## LOG\_CHECKPOINT\_TIMEOUT の新規デフォルト値

リリース 8.1.5 以上では、`LOG_CHECKPOINT_TIMEOUT` 初期化パラメータには新規のデフォルト値が設定されています。以前のリリースでのデフォルト値は 0 秒でしたが、リリース 8.1.5 以上でのデフォルト値は 1800 秒です。

**参照：** 詳細は、『Oracle9i データベース・リファレンス』の「`LOG_CHECKPOINT_TIMEOUT`」を参照してください。

## 07\_DICTIONARY\_ACCESSIBILITY パラメータ

`07_DICTIONARY_ACCESSIBILITY` 初期化パラメータを使用して、Oracle7 のデータ・ディクショナリの動作を継承するかどうかを制御します。この初期化パラメータの使用は一時的な機能です。Oracle9i リリース 1 (9.0.1) 以上では、この初期化パラメータのデフォルト値は `false` です。

**参照：** 詳細は、5-28 ページの「[データ・ディクショナリの保護](#)」を参照してください。

## DML\_LOCKS パラメータ

Oracle9i システムでは、通常、DDL 操作の実行中に Oracle7 のシステムで必要とする以上の DML ロックを使用します。ただし、Oracle7 の `DML_LOCKS` パラメータのデフォルト設定は、Oracle9i のシステムおよび DML を非常に多く使用するアプリケーションに適しています。

`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、Oracle9i 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 Oracle7 より少なくなっています。したがって、`DML_LOCKS` の Oracle9i でのデフォルト値が低くなっています。ある極端な負荷条件では、`DML_LOCKS` パラメータの値を大きくする必要があります。

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。それはオペレーティング・システム固有の設定によって異なります。この変更についての情報メッセージは、データベースの起動操作中に表示されます。

## DB\_DOMAIN パラメータ

リリース 8.1 以上では、`DB_DOMAIN` 初期化パラメータが設定されていない場合、このパラメータはデフォルトで `NULL` に設定されます。Oracle の以前のリリースでは、デフォルト設定は次のように設定されていました。

`WORLD`

`DB_DOMAIN` の `NULL` 設定は、環境によってはデータベース接続の問題の原因になることがあります。リリース 8.0.6 以下からアップグレードする場合は、初期化パラメータ・ファイルの `DB_DOMAIN` 初期化パラメータを次のいずれかに設定する必要があります。

- `WORLD`
- 環境に有効なドメイン設定

既存のデータベースに `DB_DOMAIN` を設定していない場合は、アップグレードする前に、`WORLD` に設定してください。

`DB_DOMAIN` を既存のデータベースの環境に有効なドメインに設定した場合は、アップグレードする前に、その設定を初期化パラメータ・ファイルに保存してください。

## ラージ・プールから割り当てられたパラレル実行

リリース 8.1 以上では、`PARALLEL_AUTOMATIC_TUNING` を `true` に設定すると、パラレル実行のメッセージ・バッファをラージ・プールから割り当てることができます。以前のリリースでは、共有プールから割り当てられていました。リリース 8.0.6 以下からアップグレードし、`PARALLEL_AUTOMATIC_TUNING` に `true` を設定すると、次の初期化パラメータの変更によって問題を回避できます。

- `SHARED_POOL_SIZE`
- `LARGE_POOL_SIZE`

通常、問題を回避するには、`SHARED_POOL_SIZE` の設定を小さくし、`LARGE_POOL_SIZE` の設定を大きくしますが、かわりに `SHARED_POOL_SIZE` の設定を小さくし、`LARGE_POOL_SIZE` の設定を Oracle に計算させます。`PARALLEL_AUTOMATIC_TUNING` を `true` に設定し、`LARGE_POOL_SIZE` を設定しない場合のみ、Oracle がデフォルトの `LARGE_POOL_SIZE` を計算します。

`PARALLEL_AUTOMATIC_TUNING` を設定していない場合、または `false` と設定している場合、および `LARGE_POOL_SIZE` を設定していない場合、`LARGE_POOL_SIZE` のデフォルトに対する値は 0 になります。

**参照：** PARALLEL\_AUTOMATIC\_TUNING の初期化パラメータのその他の影響については、『Oracle9i データベース・リファレンス』および『Oracle9i データベース・パフォーマンス・チューニング・ガイド』およびリファレンス』を参照してください。

次に、リリース 8.1 以上へのアップグレードした際の、個々の初期化パラメータ設定の結果として発生する動作を示します。

## パラメータ設定を変更しないで保存

パラメータを以前の設定から変更しない場合：

**表 A-1 パラメータ設定を変更しないで保存**

| パラメータ                     | 設定                      |
|---------------------------|-------------------------|
| PARALLEL_AUTOMATIC_TUNING | 未設定（デフォルトは false）       |
| SHARED_POOL_SIZE          | パラレル実行に必要な領域も含めて大きい値に設定 |
| LARGE_POOL_SIZE           | 未設定（デフォルトは 0）           |

これらの設定方法が最も一般的です。この場合、共有プールでのパラレル実行に必要な領域をすでに占めています。

## PARALLEL\_AUTOMATIC\_TUNING の使用

パラメータを以前の設定から次の設定に変更した場合：

**表 A-2 PARALLEL\_AUTOMATIC\_TUNING の使用**

| パラメータ                     | 設定                                   |
|---------------------------|--------------------------------------|
| PARALLEL_AUTOMATIC_TUNING | true に設定                             |
| SHARED_POOL_SIZE          | パラレル実行を除く、全クライアントを占める小さい値に設定         |
| LARGE_POOL_SIZE           | 未設定（デフォルトは、パラレル実行に必要な領域も含めて大きい値にします） |

この場合、パラレル実行は自動計算に基づいて、ラージ・プールからバッファを割り当てます。バッファの割当てはより効果的で、割当てに失敗すると、共有プールのクライアントから分離されます。

PARALLEL\_AUTOMATIC\_TUNING の使用および LARGE\_POOL\_SIZE の設定

パラメータを以前の設定から次の設定に変更した場合：

表 A-3 PARALLEL\_AUTOMATIC\_TUNING の使用および LARGE\_POOL\_SIZE の設定

| パラメータ                     | 設定                           |
|---------------------------|------------------------------|
| PARALLEL_AUTOMATIC_TUNING | true に設定                     |
| SHARED_POOL_SIZE          | パラレル実行を除く、全クライアントを占める小さい値に設定 |
| LARGE_POOL_SIZE           | パラレル実行に必要な領域も含めて大きい値に設定      |

この場合、パラレル実行はラージ・プールからのバッファを割り当てます。  
LARGE\_POOL\_SIZE の設定を解除した状態で初期テストを終了した後、LARGE\_POOL\_SIZE に対するデフォルトの計算値がラージ・プールの要件を反映していないと判断した場合の設定です。したがって、LARGE\_POOL\_SIZE の設定は手動になります。LARGE\_POOL\_SIZE を正しく設定するとバッファの割当てはより効果的になり、割当てに失敗すると、共有プールのクライアントから分離されます。

SHARED\_POOL\_SIZE を変更しないで PARALLEL\_AUTOMATIC\_TUNING を使用

パラメータを以前の設定から次の設定に変更した場合：

表 A-4 SHARED\_POOL\_SIZE を変更しないで PARALLEL\_AUTOMATIC\_TUNING を使用

| パラメータ                     | 設定                                   |
|---------------------------|--------------------------------------|
| PARALLEL_AUTOMATIC_TUNING | true に設定                             |
| SHARED_POOL_SIZE          | パラレル実行に必要な領域も含めて大きい値に設定              |
| LARGE_POOL_SIZE           | 未設定（デフォルトは、パラレル実行に必要な領域も含めて大きい値にします） |

この場合、ラージ・プールからのバッファを割り当てますが、SHARED\_POOL\_SIZE の変更がなかったため SGA は不必要に大きくなり、パフォーマンスの妨げになる可能性があります。したがって、適切な SHARED\_POOL\_SIZE および LARGE\_POOL\_SIZE の設定をしてから、PARALLEL\_AUTOMATIC\_TUNING を true に設定してください。

## アーカイブ・ログの宛先のパラメータ

リリース 8.1 以上で、新規アーカイブ・ログの宛先のパラメータをサポートします。アップグレードの後、リリース 8.1 より前の旧パラメータ (LOG\_ARCHIVE\_DEST および LOG\_ARCHIVE\_DUPLEX\_DEST) からリリース 8.1 以上の新しいパラメータ (LOG\_ARCHIVE\_DEST\_n および LOG\_ARCHIVE\_DEST\_STATE\_n) への動的な変換が可能になります。旧パラメータへの動的な変換も可能になります。

---

---

**注意：** Oracle9i では、アーカイブ・ログ出力先の数が 5 から 10 に増えました。

---

---

### 新規アーカイブ・ログの宛先のパラメータへの変更

新規アーカイブ先、対応する状態およびオプションを決めたら、旧アーカイブ・ログの宛先のパラメータから新規アーカイブ・ログの宛先のパラメータへ変更するため、次の手順を実行してください。

1. ALTER SYSTEM を使用して、LOG\_ARCHIVE\_MIN\_SUCCEED\_DEST を 1 に設定します。
2. ALTER SYSTEM を使用して、LOG\_ARCHIVE\_DUPLEX\_DEST を NULL に設定します。
3. ALTER SYSTEM を使用して、LOG\_ARCHIVE\_DEST を NULL に設定します。
4. ALTER SYSTEM を使用して、LOG\_ARCHIVE\_DEST\_STATE\_n のパラメータを指示どおりに「defer」または「enable」に設定します。enable はデフォルトですが、オラクル社では、状態を各宛先に明確に設定することをお勧めします。
5. ALTER SYSTEM を使用して、ローカルの宛先を指定する値に、1 つ以上の LOG\_ARCHIVE\_DEST\_n パラメータを設定します。
6. ALTER SYSTEM を使用して、指定どおりに、その他の LOG\_ARCHIVE\_DEST\_n パラメータを設定します。
7. ALTER SYSTEM を使用して、LOG\_ARCHIVE\_MIN\_SUCCEED\_DEST を指定した値に設定します。

たとえば、次の 2 つの宛先があると仮定します。

- /oracle/dbs/arclog
- /backup/dbs/arclog

両方の宛先が固定されます（最小宛先件数は2）。新規の宛先は次のとおりです。

- /oracle/dbs/arclog（ローカル）
- standby1（スタンバイ・データベース）
- /backup/dbs/arclog
- /backup2/dbs/arclog

最初の宛先である standby および backup のどちらかが固定されます（最小宛先件数は3）。

これらの仮定をもとに、次の SQL 文を発行して旧アーカイブ・ログの宛先を新規に変更してください。

```
ALTER SYSTEM SET LOG_ARCHIVE_MIN_SUCCEED_DEST = 1;
```

```
ALTER SYSTEM SET LOG_ARCHIVE_DUPLEX_DEST = ' ';
```

```
ALTER SYSTEM SET LOG_ARCHIVE_DEST = ' ';
```

```
ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_1 = 'enable';
```

```
ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2 = 'enable';
```

```
ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_3 = 'enable';
```

```
ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_4 = 'enable';
```

```
ALTER SYSTEM SET LOG_ARCHIVE_DEST_1 = 'LOCATION=/oracle/dbs/arclog MANDATORY';
```

```
ALTER SYSTEM SET LOG_ARCHIVE_DEST_2 = 'SERVICE=standby1 MANDATORY';
```

```
ALTER SYSTEM SET LOG_ARCHIVE_DEST_3 = 'LOCATION=/backup/dbs/arclog OPTIONAL';
```

```
ALTER SYSTEM SET LOG_ARCHIVE_DEST_4 = 'LOCATION=/backup2/dbs/arclog OPTIONAL';
```

```
ALTER SYSTEM SET LOG_ARCHIVE_MIN_SUCCEED_DEST = 3;
```

## 旧アーカイブ・ログの宛先のパラメータへの変更

次の手順を実行して、旧アーカイブ・ログの宛先のパラメータに戻してください。

1. ALTER SYSTEM を使用して、LOG\_ARCHIVE\_MIN\_SUCCEED\_DEST を 1 に設定します。
2. ALTER SYSTEM を使用して、指定どおりに、すべての LOG\_ARCHIVE\_DEST\_n パラメータを NULL に設定します。
3. ALTER SYSTEM を使用して、ローカルの宛先を指定する値に LOG\_ARCHIVE\_DEST のパラメータを設定します。
4. ALTER SYSTEM を使用して、指定どおりに、LOG\_ARCHIVE\_MIN\_SUCCEED\_DEST に設定します。
5. ALTER SYSTEM を使用して、LOG\_ARCHIVE\_MIN\_SUCCEED\_DEST を指定した値に設定します。



たとえば、次の2つの宛先があると仮定します。

- /oracle/dbs/arclog (LOG\_ARCHIVE\_DEST\_1)
- /backup/dbs/arclog (LOG\_ARCHIVE\_DEST\_4)

宛先は両方とも固定されています。新規宛先および最小件数は同じです。

これらの仮定をもとに、次のSQL文を発行して新規アーカイブ・ログの宛先を旧アーカイブ・ログの宛先に変更してください。

```
ALTER SYSTEM SET LOG_ARCHIVE_MIN_SUCCEED_DEST = 1;
```

```
ALTER SYSTEM SET LOG_ARCHIVE_DEST_4 = ' ';
```

```
ALTER SYSTEM SET LOG_ARCHIVE_DEST_1 = ' ';
```

```
ALTER SYSTEM SET LOG_ARCHIVE_DEST = '/oracle/dbs/arclog';
```

```
ALTER SYSTEM SET LOG_ARCHIVE_DUPLEX_DEST = '/backup/dbs/arclog';
```

```
ALTER SYSTEM SET LOG_ARCHIVE_MIN_SUCCEED_DEST = 2;
```

## トランザクション・パラメータ中に発生する可能性のあるエラー

この項の前半で説明した手順に従ってアーカイブ先のパラメータを変更する場合、アーカイブが使用可能になると、ログ・ファイル中に次のエラー・メッセージが表示されることがあります。

- アラート・ログ: Archiving not possible: No available destinations
- トレース・ログ: ARCH: INCOMPLETE, no available destinations

アーカイブが使用禁止の場合、これらのエラーは発生しません。有効なアーカイブ先がない場合、作業中にエラーが発生します。ただし、パラメータの変更が完了すればエラーは発生しなくなります。これらのエラーを回避するため、変換中にアーカイブを使用禁止にしないでください。

# 静的データ・ディクショナリ・ビューの変更

次の項では、Oracle の異なるリリース間での静的データ・ディクショナリ・ビューの変更点を示します。

- [非推奨の静的データ・ディクショナリ・ビュー](#)
- [廃止された静的データ・ディクショナリ・ビュー](#)
- [列の名前が変更された静的データ・ディクショナリ・ビュー](#)
- [列が削除された静的データ・ディクショナリ・ビュー](#)
- [NULL を戻す可能性のある列を持つ静的データ・ディクショナリ・ビュー](#)

**参照：** Oracle9i の新しい静的データ・ディクショナリ・ビューのリストは、『Oracle9i データベース・リファレンス』の第 2 章「静的データ・ディクショナリ・ビュー」を参照してください。

## 非推奨の静的データ・ディクショナリ・ビュー

次の項では、非推奨の静的データ・ディクショナリ・ビューを示します。

- [リリース 2 \(9.2\) で非推奨の静的データ・ディクショナリ・ビュー](#)
- [リリース 1 \(9.0.1\) で非推奨の静的データ・ディクショナリ・ビュー](#)
- [リリース 8.1 で非推奨の静的データ・ディクショナリ・ビュー](#)

### リリース 2 (9.2) で非推奨の静的データ・ディクショナリ・ビュー

次の静的データ・ディクショナリ・ビューはリリース 2 (9.2) で非推奨のものです。

| 非推奨の静的データ・ディクショナリ・ビュー | 新しい静的データ・ディクショナリ・ビュー |
|-----------------------|----------------------|
| ALL_RULESETS          | ALL_RULE_SETS        |
| DBA_RULESETS          | DBA_RULE_SETS        |
| USER_RULESETS         | USER_RULE_SETS       |

## リリース 1 (9.0.1) で非推奨の静的データ・ディクショナリ・ビュー

次の静的データ・ディクショナリ・ビューはリリース 1 (9.0.1) で非推奨のものです。

| 非推奨の静的データ・ディクショナリ・ビュー          | 新しい静的データ・ディクショナリ・ビュー        |
|--------------------------------|-----------------------------|
| ALL_REGISTERED_SNAPSHOTS       | ALL_REGISTERED_MVIEWS       |
| ALL_SNAPSHOT_LOGS              | ALL_BASE_TABLE_MVIEWS       |
|                                | ALL_MVIEW_LOGS              |
| ALL_SNAPSHOT_REFRESH_TIMES     | ALL_MVIEW_REFRESH_TIMES     |
| DBA_REGISTERED_SNAPSHOT_GROUPS | DBA_REGISTERED_MVIEW_GROUPS |
| DBA_REGISTERED_SNAPSHOTS       | DBA_REGISTERED_MVIEWS       |
| DBA_SNAPSHOT_LOG_FILTER_COLS   | DBA_MVIEW_LOG_FILTER_COLS   |
| DBA_SNAPSHOT_LOGS              | DBA_BASE_TABLE_MVIEWS       |
|                                | DBA_MVIEW_LOGS              |
| DBA_SNAPSHOT_REFRESH_TIMES     | DBA_MVIEW_REFRESH_TIMES     |
| USER_REGISTERED_SNAPSHOTS      | USER_REGISTERED_MVIEWS      |
| USER_SNAPSHOT_LOGS             | USER_BASE_TABLE_MVIEWS      |
|                                | USER_MVIEW_LOGS             |
| USER_SNAPSHOT_REFRESH_TIMES    | USER_MVIEW_REFRESH_TIMES    |

## リリース 8.1 で非推奨の静的データ・ディクショナリ・ビュー

次の静的データ・ディクショナリ・ビューはリリース 8.1 で非推奨のものです。

| 非推奨の静的データ・ディクショナリ・ビュー     | 新しい静的データ・ディクショナリ・ビュー       |
|---------------------------|----------------------------|
| ALL_SNAPSHOTS             | ALL_MVIEWS                 |
| ALL_SUMMARIES             | ALL_MVIEW_ANALYSIS         |
| ALL_SUMMARY_AGGREGATES    | ALL_MVIEW_AGGREGATES       |
| ALL_SUMMARY_DETAIL_TABLES | ALL_MVIEW_DETAIL_RELATIONS |
| ALL_SUMMARY_JOINS         | ALL_MVIEW_JOINS            |
| ALL_SUMMARY_KEYS          | ALL_MVIEW_KEYS             |

| 非推奨の静的データ・ディクショナリ・ビュー      | 新しい静的データ・ディクショナリ・ビュー        |
|----------------------------|-----------------------------|
| DBA_SNAPSHOTS              | DBA_MVIEWS                  |
| DBA_SUMMARIES              | DBA_MVIEW_ANALYSIS          |
| DBA_SUMMARY_AGGREGATES     | DBA_MVIEW_AGGREGATES        |
| DBA_SUMMARY_DETAIL_TABLES  | DBA_MVIEW_DETAIL_RELATIONS  |
| DBA_SUMMARY_JOINS          | DBA_MVIEW_JOINS             |
| DBA_SUMMARY_KEYS           | DBA_MVIEW_KEYS              |
| USER_SNAPSHOTS             | USER_MVIEWS                 |
| USER_SUMMARIES             | USER_MVIEW_ANALYSIS         |
| USER_SUMMARY_AGGREGATES    | USER_MVIEW_AGGREGATES       |
| USER_SUMMARY_DETAIL_TABLES | USER_MVIEW_DETAIL_RELATIONS |
| USER_SUMMARY_JOINS         | USER_MVIEW_JOINS            |
| USER_SUMMARY_KEYS          | USER_MVIEW_KEYS             |

廃止された静的データ・ディクショナリ・ビュー

次の項では、廃止された静的データ・ディクショナリ・ビューを示します。

- [リリース 8.1](#) で廃止された静的データ・ディクショナリ・ビュー

リリース 8.1 で廃止された静的データ・ディクショナリ・ビュー

次の静的データ・ディクショナリ・ビューはリリース 8.1 で廃止されたものです。

ALL\_LABELS

## 列の名前が変更された静的データ・ディクショナリ・ビュー

次の項では、列の名前が変更された静的データ・ディクショナリ・ビューを示します。

- リリース 1 (9.0.1) で列の名前が変更された静的データ・ディクショナリ・ビュー

### リリース 1 (9.0.1) で列の名前が変更された静的データ・ディクショナリ・ビュー

リリース 1 (9.0.1) では、表 A-5 に示す静的データ・ディクショナリ・ビューの列の名前が変更されました。

表 A-5 リリース 1 (9.0.1) で列の名前が変更された静的データ・ディクショナリ・ビュー

| 静的データ・ディクショナリ・ビュー        | リリース 1 (9.0.1) より前の列名      | リリース 1 (9.0.1) 以上の列名 |
|--------------------------|----------------------------|----------------------|
| DBA_RSRC_PLAN_DIRECTIVES | MAX_ACTIVE_SESS_TARGET_P1  | ACTIVE_SESS_POOL_P1  |
| DBA_RSRC_PLANS           | MAX_ACTIVE_SESS_TARGET_MTH | ACTIVE_SESS_POOL_MTH |

## 列が削除された静的データ・ディクショナリ・ビュー

次の項では、列が削除された静的データ・ディクショナリ・ビューを示します。

- リリース 1 (9.0.1) で列が削除された静的データ・ディクショナリ・ビュー
- リリース 8.1 で列が削除された静的データ・ディクショナリ・ビュー

### リリース 1 (9.0.1) で列が削除された静的データ・ディクショナリ・ビュー

リリース 1 (9.0.1) では、次の静的データ・ディクショナリ・ビューの列が削除されました。

| 静的データ・ディクショナリ・ビュー        | 削除された列                     |
|--------------------------|----------------------------|
| DBA_RSRC_PLAN_DIRECTIVES | MAX_ACTIVE_SESS_TARGET_P1  |
| DBA_RSRC_PLANS           | MAX_ACTIVE_SESS_TARGET_MTH |

リリース 8.1 で列が削除された静的データ・ディクショナリ・ビュー

リリース 8.1 では、次の静的データ・ディクショナリ・ビューの列が削除されました。

| 静的データ・ディクショナリ・ビュー    | 削除された列                |
|----------------------|-----------------------|
| DBA_AUDIT_OBJECT     | OBJECT_LABEL          |
| USER_AUDIT_OBJECT    | SESSION_LABEL         |
| DBA_AUDIT_SESSION    | SESSION_LABEL         |
| USER_AUDIT_SESSION   |                       |
| DBA_AUDIT_STATEMENT  | SESSION_LABEL         |
| USER_AUDIT_STATEMENT |                       |
| DBA_AUDIT_TRAIL      | OBJECT_LABEL          |
| USER_AUDIT_TRAIL     | SESSION_LABEL         |
| DBA_CONTEXT          | ATTRIBUTE             |
| ALL_IND_COLUMNS      | COLUMN_EXPRESSION     |
| DBA_IND_COLUMNS      |                       |
| USER_IND_COLUMNS     |                       |
| ALL_JOBS             | CLEARANCE_HI          |
| DBA_JOBS             | CLEARANCE_LO          |
| USER_JOBS            | CURRENT_SESSION_LABEL |
| ALL_REFS             | HAS_REFERENTIAL_CONS  |
| DBA_REFS             | REFERENTIAL_CONS_NAME |
| USER_REFS            |                       |

## NULL を戻す可能性のある列を持つ静的データ・ディクショナリ・ビュー

リリース 8.1 以上では、表 A-6 に示す静的データ・ディクショナリ・ビューの列は、NULL を戻す可能性があります。以前のリリースでは、これらの列は NULL を戻すことはできませんでした。アプリケーションでこれらの 1 つ以上の列に対して NULL 以外の値が必要とされる場合は、それに応じてアプリケーションを変更します。

表 A-6 リリース 8.1 で NULL を戻す可能性のある列を持つ静的データ・ディクショナリ・ビュー

| 静的データ・ディクショナリ・ビュー   |                 | 説明                                                                     |
|---------------------|-----------------|------------------------------------------------------------------------|
| DBA_DATA_FILES      | AUTOEXTENSIBLE  | データ・ファイルがオフラインの場合は読み込み不可能であるため、これらの列は NULL を戻します。                      |
|                     | BLOCKS          |                                                                        |
|                     | BYTES           |                                                                        |
|                     | INCREMENT_BY    |                                                                        |
|                     | MAXBLOCKS       |                                                                        |
|                     | MAXBYTES        |                                                                        |
| ALL_IND_COLUMNS     | COLUMN_NAME     | 索引が列ではなく機能に付いている場合、この列は NULL を戻します。この場合、示すべき列がありません。                   |
| DBA_IND_COLUMNS     |                 |                                                                        |
| USER_IND_COLUMNS    |                 |                                                                        |
| ALL_IND_PARTITIONS  | INITIAL_EXTENT  | 索引がコンポジット・メソッドを使用して区分される場合や、デフォルト値が区分に対して特定されていない場合、これらの列は NULL を戻します。 |
| DBA_IND_PARTITIONS  | MAX_EXTENT      |                                                                        |
| USER_IND_PARTITIONS | MIN_EXTENT      |                                                                        |
|                     | NEXT_EXTENT     |                                                                        |
|                     | PCT_INCREASE    |                                                                        |
| ALL_OBJECT_TABLES   | TABLESPACE_NAME | オブジェクト表がパーティション化されている場合、または一時表の場合、この列は NULL を戻します。                     |
| DBA_OBJECT_TABLES   |                 |                                                                        |
| USER_OBJECT_TABLES  |                 |                                                                        |

表 A-6 リリース 8.1 で NULL を戻す可能性のある列を持つ静的データ・ディクショナリ・ビュー（続き）

| 静的データ・ディクショナリ・列<br>ビュー |                | 説明                                                                                                                                                                |
|------------------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ALL_SEGMENTS           | BLOCKS         | ファイルがオフラインであるためセグメント・ヘッダーが読み込み不可能の場合、または他に問題がある場合、BLOCKS 列、BYTES 列および EXTENTS 列は NULL を戻します。                                                                      |
| DBA_SEGMENTS           | BYTES          |                                                                                                                                                                   |
| USER_SEGMENTS          | EXTENTS        |                                                                                                                                                                   |
|                        | NEXT_EXTENT    |                                                                                                                                                                   |
|                        | PCT_INCREASE   | セグメントを格納する表領域がローカル管理され、AUTOALLOCATE オプションを使用する場合、NEXT_EXTENT や PCT_INCREASE 列は NULL を戻します。それは、システムが拡張サイズを選択し、アルゴリズムは NEXT_EXTENT や PCT_INCREASE の点において説明できないためです。 |
| ALL_TAB_PARTITIONS     | INITIAL_EXTENT | 索引がコンポジット・メソッドを使用してパーティション化される場合や、パーティション化について、デフォルト値が特定されていない場合、これらの列は NULL を戻します。                                                                               |
| DBA_TAB_PARTITIONS     | MAX_EXTENT     |                                                                                                                                                                   |
| USER_TAB_PARTITIONS    | MIN_EXTENT     |                                                                                                                                                                   |
|                        | NEXT_EXTENT    |                                                                                                                                                                   |
|                        | PCT_INCREASE   |                                                                                                                                                                   |
| ALL_TABLESPACES        | NEXT_EXTENT    | 表領域がローカル管理され、AUTOALLOCATE オプションを使用する場合、これらの列は NULL を戻します。それは、システムが拡張サイズを選択し、アルゴリズムは NEXT_EXTENT や PCT_INCREASE の点において説明できないためです。                                  |
| DBA_TABLESPACES        | PCT_INCREASE   |                                                                                                                                                                   |
| USER_TABLESPACES       |                |                                                                                                                                                                   |
| ALL_TRIGGERS           | TABLE_NAME     | トリガーがシステム・トリガーの場合、この列は NULL を戻します。この場合、トリガーのオブジェクト基本型は、TABLE や VIEW ではなく、SCHEMA や DATABASE になります。                                                                 |
| DBA_TRIGGERS           |                |                                                                                                                                                                   |
| USER_TRIGGERS          |                |                                                                                                                                                                   |



# 動的パフォーマンス・ビューの変更

次の項では、Oracle の異なるリリース間での動的パフォーマンス・ビュー (V\$ ビュー) の変更点を示します。

- [非推奨の動的パフォーマンス・ビュー](#)
- [廃止された動的パフォーマンス・ビュー](#)
- [列の名前が変更された動的パフォーマンス・ビュー](#)
- [列が削除された動的パフォーマンス・ビュー](#)

**参照：** Oracle9i の新しい動的パフォーマンス・ビューのリストは、『Oracle9i データベース・リファレンス』を参照してください。

## 非推奨の動的パフォーマンス・ビュー

次の項では、非推奨の動的パフォーマンス・ビューを示します。

- [リリース 2 \(9.2\) で非推奨の動的パフォーマンス・ビュー](#)
- [リリース 1 \(9.0.1\) で非推奨の動的パフォーマンス・ビュー](#)

### リリース 2 (9.2) で非推奨の動的パフォーマンス・ビュー

次の動的パフォーマンス・ビューはリリース 2 (9.2) で非推奨のものです。

| 非推奨の動的パフォーマンス・ビュー | 新しい動的パフォーマンス・ビュー  |
|-------------------|-------------------|
| GV\$SORT_USAGE    | GV\$TEMPSEG_USAGE |
| V\$SORT_USAGE     | V\$TEMPSEG_USAGE  |

リリース 1 (9.0.1) で非推奨の動的パフォーマンス・ビュー

次の動的パフォーマンス・ビューはリリース 1 (9.0.1) で非推奨のものです。

| 非推奨の動的パフォーマンス・ビュー              | 新しい動的パフォーマンス・ビュー                |
|--------------------------------|---------------------------------|
| GV\$BSP                        | GV\$CR_BLOCK_SERVER             |
| GV\$CLASS_PING                 | GV\$CLASS_CACHE_TRANSFER        |
| GV\$DLM_ALL_LOCKS              | GV\$GES_ENQUEUE                 |
| GV\$DLM_CONVERT_LOCAL          | GV\$GES_CONVERT_LOCAL           |
| GV\$DLM_CONVERT_REMOTE         | GV\$GES_CONVERT_REMOTE          |
| GV\$DLM_LATCH                  | GV\$GES_LATCH                   |
| GV\$DLM_LOCKS                  | GV\$GES_BLOCKING_ENQUEUE        |
| GV\$DLM_MISC                   | GV\$GES_STATISTICS              |
| GV\$DLM_RESS                   | GV\$GES_RESOURCE                |
| GV\$DLM_TRAFFIC_CONTROLLER     | GV\$GES_TRAFFIC_CONTROLLER      |
| GV\$FILE_PING                  | GV\$FILE_CACHE_TRANSFER         |
| GV\$LOCK_ELEMENT               | GV\$GC_ELEMENT                  |
| GV\$LOCKS_WITH_COLLISIONS      | GV\$GC_ELEMENTS_WITH_COLLISIONS |
| GV\$MAX_ACTIVE_SESS_TARGET_MTH | GV\$ACTIVE_SESS_POOL_MTH        |
| GV\$MTS                        | GV\$SHARED_SERVER_MONITOR       |
| GV\$PING                       | GV\$CACHE_TRANSFER              |
| GV\$TEMP_PING                  | GV\$TEMP_CACHE_TRANSFER         |
| V\$BSP                         | V\$CR_BLOCK_SERVER              |
| V\$CLASS_PING                  | V\$CLASS_CACHE_TRANSFER         |
| V\$DLM_ALL_LOCKS               | V\$GES_ENQUEUE                  |
| V\$DLM_CONVERT_LOCAL           | V\$GES_CONVERT_LOCAL            |
| V\$DLM_CONVERT_REMOTE          | V\$GES_CONVERT_REMOTE           |
| V\$DLM_LATCH                   | V\$GES_LATCH                    |
| V\$DLM_LOCKS                   | V\$GES_BLOCKING_ENQUEUE         |
| V\$DLM_MISC                    | V\$GES_STATISTICS               |
| V\$DLM_RESS                    | V\$GES_RESOURCE                 |

| 非推奨の動的パフォーマンス・ビュー             | 新しい動的パフォーマンス・ビュー               |
|-------------------------------|--------------------------------|
| V\$DLM_TRAFFIC_CONTROLLER     | V\$GES_TRAFFIC_CONTROLLER      |
| V\$FILE_PING                  | V\$FILE_CACHE_TRANSFER         |
| V\$LOCK_ELEMENT               | V\$GC_ELEMENT                  |
| V\$LOCKS_WITH_COLLISIONS      | V\$GC_ELEMENTS_WITH_COLLISIONS |
| V\$MAX_ACTIVE_SESS_TARGET_MTH | V\$ACTIVE_SESS_POOL_MTH        |
| V\$MTS                        | V\$SHARED_SERVER_MONITOR       |
| V\$PING                       | V\$CACHE_TRANSFER              |
| V\$TEMP_PING                  | V\$TEMP_CACHE_TRANSFER         |

## 廃止された動的パフォーマンス・ビュー

次の項では、廃止された動的パフォーマンス・ビューを示します。

- [リリース 2 \(9.2\)](#) で廃止された動的パフォーマンス・ビュー
- [リリース 1 \(9.0.1\)](#) で廃止された動的パフォーマンス・ビュー
- [リリース 8.1](#) で廃止された動的パフォーマンス・ビュー

### リリース 2 (9.2) で廃止された動的パフォーマンス・ビュー

次の動的パフォーマンス・ビューはリリース 2 (9.2) で廃止されたものです。

|               |              |
|---------------|--------------|
| GV\$LOADCSTAT | V\$LOADCSTAT |
| GV\$LOADTSTAT | V\$LOADTSTAT |

### リリース 1 (9.0.1) で廃止された動的パフォーマンス・ビュー

次の動的パフォーマンス・ビューはリリース 1 (9.0.1) で廃止されたものです。

|               |              |
|---------------|--------------|
| GV\$TARGETRBA | V\$TARGETRBA |
|---------------|--------------|

### リリース 8.1 で廃止された動的パフォーマンス・ビュー

次の動的パフォーマンス・ビューはリリース 8.1 で廃止されたものです。

|                    |                   |
|--------------------|-------------------|
| GV\$CURRENT_BUCKET | V\$CURRENT_BUCKET |
| GV\$RECENT_BUCKET  | V\$RECENT_BUCKET  |

列の名前が変更された動的パフォーマンス・ビュー

次の項では、列の名前が変更された動的パフォーマンス・ビューを示します。

- リリース 2 (9.2) で列の名前が変更された動的パフォーマンス・ビュー
- リリース 1 (9.0.1) で列の名前が変更された動的パフォーマンス・ビュー
- リリース 8.1 で列の名前が変更された動的パフォーマンス・ビュー

リリース 2 (9.2) で列の名前が変更された動的パフォーマンス・ビュー

リリース 2 (9.2) では、表 A-7 に示す動的パフォーマンス・ビューの列の名前が変更されました。

表 A-7 リリース 2 (9.2) で列の名前が変更された動的パフォーマンス・ビュー

| 動的パフォーマンス・ビュー                                  | リリース 2 (9.2) より前の列名 | リリース 2 (9.2) 以上の列名 |
|------------------------------------------------|---------------------|--------------------|
| GV\$ARCHIVE_DEST および<br>V\$ARCHIVE_DEST        | MANIFEST            | REGISTER           |
|                                                | REGISTER            | REMOTE_TEMPLATE    |
| GV\$DATABASE および<br>V\$DATABASE                | STANDBY_MODE        | PROTECTION_MODE    |
|                                                |                     |                    |
| GV\$LOGMNR_CALLBACK およ<br>び V\$LOGMNR_CALLBACK | CALLBACK_STATE      | STATE              |
|                                                | CALLBACK_TYPE       | TYPE               |
|                                                | CALLBACK_CAPABILITY | CAPABILITY         |
| GV\$LOGMNR_REGION および<br>V\$LOGMNR_REGION      | ID                  | MEMSTATE           |
|                                                | CURRENT_STATE       | STATE              |

リリース 1 (9.0.1) で列の名前が変更された動的パフォーマンス・ビュー

リリース 1 (9.0.1) では、表 A-8 に示す動的パフォーマンス・ビューの列の名前が変更されました。

表 A-8 リリース 1 (9.0.1) で列の名前が変更された動的パフォーマンス・ビュー

| 動的パフォーマンス・ビュー                                             | リリース 1 (9.0.1) より前の<br>列名 | リリース 1 (9.0.1) 以上の列名 |
|-----------------------------------------------------------|---------------------------|----------------------|
| GV\$RSRC_CONSUMER_GROUP<br>および V\$RSRC_CONSUMER_<br>GROUP | SESSIONS_QUEUED           | QUEUE_LENGTH         |

## リリース 8.1 で列の名前が変更された動的パフォーマンス・ビュー

リリース 8.1 では、表 A-9 に示す動的パフォーマンス・ビューの列の名前が変更されました。

表 A-9 リリース 8.1 で列の名前が変更された動的パフォーマンス・ビュー

| 動的パフォーマンス・ビュー                                    | リリース 8.1 より前の列名         | リリース 8.1 以上の列名  |
|--------------------------------------------------|-------------------------|-----------------|
| GV\$DISPATCHER_RATE<br>および<br>V\$DISPATCHER_RATE | NUM_LOOPS_TRACKED       | TTL_LOOPS       |
|                                                  | NUM_MSG_TRACKED         | TTL_MSG         |
|                                                  | NUM_SVR_BUF_TRACKED     | TTL_SVR_BUF     |
|                                                  | NUM_CLT_BUF_TRACKED     | TTL_CLT_BUF     |
|                                                  | NUM_BUF_TRACKED         | TTL_BUF         |
|                                                  | NUM_IN_CONNECT_TRACKED  | TTL_IN_CONNECT  |
|                                                  | NUM_OUT_CONNECT_TRACKED | TTL_OUT_CONNECT |
|                                                  | NUM_RECONNECT_TRACKED   | TTL_RECONNECT   |

## 列が削除された動的パフォーマンス・ビュー

次の項では、列が削除された動的パフォーマンス・ビューを示します。アプリケーションでこれらの 1 つ以上の列が必要とされる場合は、それに応じてアプリケーションを変更します。

- リリース 2 (9.2) で列が削除された動的パフォーマンス・ビュー
- リリース 1 (9.0.1) で列が削除された動的パフォーマンス・ビュー
- リリース 8.1 で列が削除された動的パフォーマンス・ビュー

リリース 2（9.2）で列が削除された動的パフォーマンス・ビュー

リリース 2（9.2）では、次の動的パフォーマンス・ビューの列は削除されました。

| 動的パフォーマンス・ビュー                                 | 削除された列                                                                                                                       |
|-----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| GV\$DATABASE および V\$DATABASE                  | STANDBY_MODE                                                                                                                 |
| GV\$LOGMNR_CALLBACK および<br>V\$LOGMNR_CALLBACK | FUNC_NAME<br>CALLBACK_ID<br>CALLBACK_RESULT_SIZE<br>CALLBACK_STATE<br>CALLBACK_TYPE<br>CALLBACK_CAPABILITY<br>NUMBER_INVOKED |
| GV\$LOGMNR_REGION および<br>V\$LOGMNR_REGION     | ID<br>CURRENT_STATE                                                                                                          |

リリース 1（9.0.1）で列が削除された動的パフォーマンス・ビュー

リリース 1（9.0.1）では、次の動的パフォーマンス・ビューの列は削除されました。

| 動的パフォーマンス・ビュー                                 | 削除された列                                                                                                                                                                           |
|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GV\$LOGMNR_CONTENTS および<br>V\$LOGMNR_CONTENTS | PH1_NAME<br>PH1_REDO<br>PH1_UNDO<br>PH2_NAME<br>PH2_REDO<br>PH2_UNDO<br>PH3_NAME<br>PH3_REDO<br>PH3_UNDO<br>PH4_NAME<br>PH4_REDO<br>PH4_UNDO<br>PH5_NAME<br>PH5_REDO<br>PH5_UNDO |

| 動的パフォーマンス・ビュー                                         | 削除された列          |
|-------------------------------------------------------|-----------------|
| GV\$RSRC_CONSUMER_GROUP および<br>V\$RSRC_CONSUMER_GROUP | SESSIONS_QUEUED |

リリース 8.1 で列が削除された動的パフォーマンス・ビュー

リリース 8.1 では、次の動的パフォーマンス・ビューの列は削除されました。

| 動的パフォーマンス・ビュー      | 削除された列                                                                                                                                                                |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| V\$ARCHIVE_DEST    | ARCMODE                                                                                                                                                               |
| V\$DLM_LATCH       | IMM_GETS<br>LATCH_TYPE<br>TTL_GETS                                                                                                                                    |
| V\$DLM_LOCKS       | RESOURCE_NAME                                                                                                                                                         |
| V\$SESSION_LONGOPS | APPLICATION_DATA_1<br>APPLICATION_DATA_2<br>APPLICATION_DATA_3<br>COMPNAM<br>CURRENT_TIME<br>MSG<br>OBJID<br>OPID<br>STEPID<br>STEPSOFAR<br>STEPTOTAL<br>UPDATE_COUNT |





---

## Oracle Net Services のアップグレードに関する考慮点

この付録では、Oracle Net Services の共存およびアップグレードについて説明します。この付録では、次の項目について説明します。

- サポートされない Oracle Net Services の機能の概要
- サポートされないパラメータおよび制御ユーティリティ・コマンド
- クライアントおよびデータベースの共存問題
- Oracle Net Manager を使用した互換性の問題の処理
- Oracle Net Services へのアップグレード
- Oracle Names バージョン 9 の使用

## サポートされない Oracle Net Services の機能の概要

インターネット用の構成における決定事項を簡略化するために、次の項では、今後はサポートされない機能および構成ファイルについて説明します。

- [Identix および SecurID 認証方式](#)
- [NDS 外部ネーミングおよび NDS 認証](#)
- [Net8 OPEN](#)
- [protocol.ora ファイル](#)
- [事前生成済専用サーバー](#)
- [プロトコル](#)

### Identix および SecurID 認証方式

Oracle Advanced Security が提供する Identix または SecurID 認証を使用している場合は、次のいずれかの認証方式へアップグレードすることをお勧めします。

- CyberSafe
- RADIUS
- Kerberos
- Secure Sockets Layer (SSL)

**参照：**『Oracle Advanced Security 管理者ガイド』を参照してください。

### NDS 外部ネーミングおよび NDS 認証

認証方式および外部ネーミング・メソッドとしての Novell Directory Services (NDS) は、今後、サポートされません。外部ネーミング・メソッドとして NDS を使用している場合、かわりにディレクトリ・ネーミングを使用することをお勧めします。

### Net8 OPEN

プログラマがデータベースおよびデータベース以外のアプリケーションを開発できる Application Program Interface (API) を提供した Net8 OPEN は、今後、サポートされません。

## protocol.ora ファイル

protocol.ora ファイル内のパラメータは、sqlnet.ora ファイルにマージされました。Transmission Control Protocol/Internet Protocol (TCP/IP) バッファ・フラッシングにおける遅延と同様に、これらのパラメータを使用すると、データベースへのアクセス制御を構成できます。これらのパラメータには、次のものが含まれます。

- TCP.NODELAY
- TCP.EXCLUDED\_NODES
- TCP.INVITED\_NODES
- TCP.VALIDNODE\_CHECKING

**参照：** これらのパラメータの詳細は、『Oracle9i Net Services リファレンス・ガイド』を参照してください。

UNIX では \$ORACLE\_HOME/network/admin、Windows では %ORACLE\_HOME%\network\admin に protocol.ora ファイルがある場合、Net Manager の最初の起動時に、自動的にパラメータが sqlnet.ora ファイルにマージされます。

protocol.ora には、ノード固有でオペレーティング・システム固有のパラメータがあります。このため、これらのパラメータをマージまたは追加した後に、他のノードで sqlnet.ora を共有しないことをお勧めします。

## 事前生成済専用サーバー

事前生成済専用サーバー・プロセスはサポートされません。かわりに、スケーラビリティおよびシステム・リソース使用率を改善する共有サーバー（以前はマルチスレッド・サーバー）を構成します。

## プロトコル

SPX プロトコルまたは LU6.2 プロトコルを使用するプロトコル・アドレスは置換する必要があります。Oracle Net では、次のネットワーク・プロトコルをサポートしています。

- TCP/IP
- SSL 付き TCP/IP
- Named Pipes

**参照：** プロトコル・パラメータの詳細は、『Oracle9i Net Services リファレンス・ガイド』を参照してください。

## サポートされないパラメータおよび制御ユーティリティ・コマンド

**参照：** サポートされない構成パラメータおよび制御ユーティリティ・コマンドの詳細は、『Oracle9i Net Services リファレンス・ガイド』を参照してください。

## クライアントおよびデータベースの共存問題

クライアントおよびデータベース・サーバーには、Oracle Net Services または Net8 の互換性のあるリリースが必要です。たとえば、Oracle9i クライアントには Oracle Net Services をインストールする必要がある、Oracle9i データベースには Oracle Net Listener とともに Oracle Net Services をインストールする必要があります。

ご使用の環境でアップグレードが適切かどうかを判断する前に、次のようなクライアント / データベース接続の問題を考慮してください。

- [Oracle9i データベースへの接続](#)
- [Oracle8 または Oracle7 データベースへの接続](#)
- [Oracle Names](#)

## Oracle9i データベースへの接続

Oracle9i または Oracle8i データベースへ接続するために作成された接続記述子は、データベースを SERVICE\_NAME パラメータで指定するサービス名によって識別します。

Oracle9i または Oracle8i データベースへの接続記述子は、次の例のようにパラメータ SERVICE\_NAME を使用します。

```
sales=
(DESCRIPTION=
 (ADDRESS= (PROTOCOL=tcp) (HOST=sales-server) (PORT=1521))
 (CONNECT_DATA=
 (SERVICE_NAME=sales.us.acme.com)))
```

現在、SID パラメータで構成されている接続記述子は、残しておくことができます。ただし、新機能（クライアント・ロード・バランスや接続時フェイルオーバーなど）のメリットを得るために、SID を SERVICE\_NAME に置き換えることをお勧めします。

SERVICE\_NAME を使用するように接続記述子を変更するには、B-7 ページの「[Oracle Net Manager を使用した互換性の問題の処理](#)」に示すとおり、Oracle Net Manager の互換性モードを使用します。

**参照：** SID ではなく SERVICE\_NAME でデータベースを識別する方法は、『Oracle9i Net Services 管理者ガイド』を参照してください。

Oracle8 クライアントが Oracle9i データベースに接続している環境について、次のことを考慮してください。

- 使用しているサード・パーティ製アプリケーションは、Oracle Net Services の機能を使用できますか？

できません。Oracle Net ライブラリで動作するように、アプリケーションを再構築またはアップグレードする必要があります。

- クライアントがリモートの Oracle9i データベースに接続するには、Oracle Net が必要ですか？

必要ありません。クライアントがリモートの Oracle9i データベースに接続する必要がある場合は、Net8 Client リリース 8.0 のみをクライアントに設定してください。ただし、Oracle Net Services の新機能は、これらのクライアントでは使用できません。

- クライアントがローカルの Oracle9i データベースに接続するには、Oracle Net が必要ですか？

必要ありません。クライアントには Oracle ホームに Net8 Client リリース 8.0 をインストールする必要があり、Oracle9i には Oracle ホームに Oracle Net および Oracle Net Listener をインストールする必要があります。

## Oracle8 または Oracle7 データベースへの接続

Oracle8 または Oracle7 データベースへの接続記述子は、次の例のように SID を使用します。

```
sales=
(DESCRIPTION=
 (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521))
 (CONNECT_DATA=
 (SID=sales)))
```

さらに、データベース・サーバーの listener.ora ファイルは、Oracle8 データベースの SID の記述で構成されている必要があります。次の例では、リスナーは、sales という SID を持つデータベース・サービス sales.us.acme.com のリスナーで構成されます。

```
SID_LIST_listener=
(SID_LIST=
 (SID_DESC=
 (GLOBAL_DBNAME=sales.us.acme.com)
 (SID_NAME=sales)))
```

**参照：** SID によるデータベースの識別方法については、『Oracle9i Net Services 管理者ガイド』を参照してください。

Oracle9i クライアントが Oracle8 データベースに接続している環境について、次のことを考慮してください。

- クライアントがリモートの Oracle8 データベースに接続するには、Net8 Client リリース 8.0 が必要ですか？

必要ありません。クライアントがリモートの Oracle8 データベースに接続する必要がある場合は、互換性のあるリリースの Net8 Client のみをクライアントに設定してください。ただし、Oracle Net Services で使用できる新機能は、この接続では使用できません。

- クライアントがローカルの Oracle8 データベースに接続するには、Net8 Client リリース 8.0 が必要ですか？

必要です。クライアントには Oracle ホームに Oracle Net をインストールする必要があり、Oracle8 データベースには Oracle ホームに Net8 Server をインストールする必要があります。

## Oracle Names

ご使用のネットワークのすべてまたは一部を Oracle9i へアップグレードする場合は、リージョンのすべての Oracle Names Server をバージョン 9 へアップグレードする必要があります。

- Oracle8 クライアントは、Oracle Names バージョン 9 を使用してサービス名を解決できますか？

できます。

- Oracle8 クライアントは、Oracle Names バージョン 9 から戻された接続記述子を使用して、Oracle8 および Oracle8i データベースへ接続できますか？

接続記述子が Oracle Names に入力されたときに正しく指定されている場合は、接続できます。

---

---

**注意：** 将来のリリースでは、Oracle Names は中心的なネーミング・メソッドとしてはサポートされません。Oracle Names には新しい拡張機能が追加されないで、『Oracle9i Net Services 管理者ガイド』に従って、ディレクトリ・ネーミングを使用するか、または既存の Oracle Names 構成のディレクトリ・ネーミングへアップグレードすることをお勧めします。

---

---

# Oracle Net Manager を使用した互換性の問題の処理

いくつかのパラメータは Oracle9i および Oracle8i のみで使用可能であるため、Oracle Net Manager は、特定のリリースのデータベースに接続するクライアントに対して、tnsnames.ora ファイルに適切なパラメータを設定するための 2 つのオプションを提供します。表 B-1 に、これらのオプションを示します。

表 B-1 Oracle Net Manager で使用可能な互換性のあるオプション

| Oracle Net Manager オプション        | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Net8 Client 8.0 と互換性のあるオプションを使用 | <p>複数アドレス・パラメータをクライアントに構成できます。</p> <p>選択すると、Oracle Connection Manager への接続が必要な Oracle8i より前のクライアントに、SOURCE_ROUTE パラメータを使用できます。</p> <p>使用しないと、SOURCE_ROUTE、LOAD_BALANCE および FAILOVER パラメータを Oracle9i および Oracle8i クライアントに使用できます。</p> <p><b>参照：</b>アドレス・リスト・パラメータの構成の詳細は、『Oracle9i Net Services 管理者ガイド』を参照してください。</p>                                                                                                                                                           |
| Oracle8i 互換識別子を使用               | <p>接続記述子の CONNECT_DATA セクションにあるデータベース・リリース固有のパラメータを構成できます。</p> <p>選択すると、Oracle8 または Oracle7 データベースの SID を入力できます。</p> <p>選択しないと、Oracle9i または Oracle8i データベースのサービス名 (SERVICE_NAME) を入力できます。</p> <p><b>注意：</b>「Service Identification」グループにある「Advanced」ボタンが選択されているときに表示される「Advanced Service Options」ダイアログ・ボックスは、このオプションの選択の有無に影響されます。いくつかの設定は、Oracle9i または Oracle8i データベース・サービスへの接続のみで使用可能です。</p> <p><b>参照：</b>拡張接続データ・パラメータの構成の詳細は、『Oracle9i Net Services 管理者ガイド』を参照してください。</p> |

# Oracle Net Services へのアップグレード

SQL\*Net V2 から Oracle Net Services へアップグレード、または Net8 リリース 8.0 またはリリース 8.1 からアップグレードするには、次の作業を行います。

- 手順 1: サービス名およびインスタンス名の検証
- 手順 2: データベース・サーバーのソフトウェアのアップグレードの実行
- 手順 3: クライアントのソフトウェアのアップグレードの実行
- 手順 4: 機能のアップグレードの実行

## 手順 1: サービス名およびインスタンス名の検証

tnsnames.ora ファイルのサービスおよびインスタンスを識別するには、SERVICE\_NAMES および INSTANCE\_NAMES 初期化パラメータが初期化パラメータ・ファイルに設定されていることを確認します。

表 B-2 Oracle Net Services の初期化パラメータ

| パラメータ         | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SERVICE_NAMES | <p>このインスタンスが接続するデータベース・サービスに、1 つ以上の名前を指定します。複数のサービス名を指定して、同じデータベースの異なる使用を区別することができます。次にその例を示します。</p> <pre>SERVICE_NAMES = sales.us.acme.com, widgetsales.us.acme.com</pre> <p>このパラメータにある名前をドメインで修飾しない場合、Oracle はこれらの名前を DB_DOMAIN パラメータの値で修飾します。DB_DOMAIN が指定されない場合、Oracle は、データ・ディクショナリに現在あるローカル・データベースのドメインを使用します。</p> <p><b>注意：</b>データベースの実行中、SQL の ALTER SYSTEM を使用して動的に SERVICE_NAMES パラメータの値を変更できます。このパラメータについては、『Oracle9i データベース・リファレンス』を参照してください。</p> |
| INSTANCE_NAME | <p>このインスタンスの一意の名前を指定します。インスタンス名を Oracle システム識別子 (SID) の値に設定します。</p>                                                                                                                                                                                                                                                                                                                                                                                           |



## 手順 2: データベース・サーバーのソフトウェアのアップグレードの実行

データベース・サーバーのソフトウェアをアップグレードするには、Oracle Universal Installer から Oracle Net および Oracle Net Listener の最新リリースをインストールして、最新の実行可能ファイル入手します。

Oracle Universal Installer によって、システムにリリース 2 (9.2) より前のデータベースが検出された場合は、Database Upgrade Assistant を使用してデータベースをアップグレードするよう求められます。インストール・プロセス中にアップグレードしない場合は、後でこの Assistant をインストールし、使用することもできます。

Oracle Universal Installer は自動的に次の作業を実行します。

- 古いリスナーの停止
- リリース 2 (9.2) リスナーの起動

## 手順 3: クライアントのソフトウェアのアップグレードの実行

クライアントのソフトウェアのアップグレードを実行するには、Oracle Universal Installer から Oracle Net Services の最新リリースをインストールして、最新の実行可能ファイル入手します。

## 手順 4: 機能のアップグレードの実行

ソフトウェアがアップグレードされた後は、Oracle9i の機能を使用しないかぎり、構成ファイルをアップグレードする必要はありません。新機能のメリットを得るには、次の構成ファイルを確認します。

- sqlnet.ora
- tnsnames.ora
- listener.ora
- protocol.ora

廃止されたパラメータまたは名前が変更されたパラメータを置き換えます。

**参照：** サポートされない構成パラメータの詳細は、『Oracle9i Net Services リファレンス・ガイド』を参照してください。

## tnsnames.ora

次の例のように、リリース 8.1 以上のサービスに接続するために SID パラメータを SERVICE\_NAME パラメータに置き換えます。

```
sales=
(DESCRIPTION=
 (ADDRESS= (PROTOCOL=tcp) (HOST=sales-server) (PORT=1521))
 (CONNECT_DATA=
 (SERVICE_NAME=sales.us.acme.com)))
```

複数のアドレスがある場合、次の例のように、クライアント・ロード・バランスおよび接続時フェイルオーバー機能を構成できます。

```
sales=
(DESCRIPTION=
 (ADDRESS_LIST=
 (FAILOVER=on)
 (LOAD_BALANCE=on)
 (ADDRESS= (PROTOCOL=tcp) (HOST=sales1-server) (PORT=1521))
 (ADDRESS= (PROTOCOL=tcp) (HOST=sales2-server) (PORT=1521))
)
 (CONNECT_DATA=
 (SERVICE_NAME=sales.us.acme.com)))
```

### 参照：

- サービス名および複数のアドレス機能の構成については、B-7 ページの「[Oracle Net Manager を使用した互換性の問題の処理](#)」を参照してください。
- 複数アドレスの詳細は、『Oracle9i Net Services 管理者ガイド』を参照してください。

listener.ora

インスタンス情報は Oracle9i のリスナーに登録されるため、インスタンス情報を listener.ora ファイルの SID\_LIST\_listener\_name セクションに入れる必要はありません。

ただし、Oracle Enterprise Manager では、listener.ora ファイルに静的情報が必要です。Oracle Enterprise Manager を使用してデータベース・オブジェクトを管理している場合は、次の方法で、listener.ora ファイルをデータベースに関する情報で構成する必要があります。

```
SID_LIST_listener_name=
 (SID_LIST=
 (SID_DESC=
 (GLOBAL_DBNAME=global_database_name)
 (ORACLE_HOME=oracle_home)
 (SID_NAME=sid)))
```

表 B-3 listener.ora のサービス設定

| パラメータ         | 説明                                                                                                                                       |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------|
| SID_NAME      | Oracle SID は、インスタンスを識別します。SID 値は、データベース初期化ファイルの INSTANCE_NAME パラメータから取得できます。                                                             |
| GLOBAL_DBNAME | グローバル・データベース名は、データベース名およびデータベース・ドメイン名で構成されます。GLOBAL_DBNAME 値は、初期化ファイルの SERVICE_NAMES パラメータ、または DB_NAME パラメータおよび DB_DOMAIN パラメータから取得できます。 |
| ORACLE_HOME   | 指定しているデータベースの Oracle ホームの場所を識別します。<br><b>注意：</b> この設定は UNIX で必要です。                                                                       |

**重要：** Oracle9i Real Application Clusters 環境などで、接続時フェイルオーバーまたは透過的アプリケーション・フェイルオーバー（TAF）を使用している場合は、GLOBAL\_DBNAME パラメータを設定しないことをお薦めします。

**参照：** サービス情報、接続時フェイルオーバーおよび透過的アプリケーション・フェイルオーバー（TAF）の構成については、『Oracle9i Net Services 管理者ガイド』を参照してください。

## Oracle Names バージョン 9 の使用

---

**注意：** 将来のリリースでは、Oracle Names は中心的なネーミング・メソッドとしてはサポートされません。Oracle Names には新しい拡張機能が追加されないで、『Oracle9i Net Services 管理者ガイド』に従って、ディレクトリ・ネーミングの使用または既存の Oracle Names 構成のディレクトリ・ネーミングへのアップグレードをお勧めします。ここで示す情報は、現在ご使用の Oracle Names 環境を維持する場合に参照してください。

---

Oracle Names バージョン 9 は、Oracle Names バージョン 2 およびバージョン 8 と下位互換性があります。Oracle Names バージョン 9 で提供される新機能のメリットを得るには、Oracle Names バージョン 9 をすべての既存の Oracle Names のサーバー・ノードにインストールし、リージョンにあるすべての既存の Oracle Names Server をバージョン 9 へアップグレードする必要があります。

次に、注意が必要なアップグレード問題について説明します。

- データベースを使用した Oracle Names バージョン 2 からのアップグレード
- 動的検出オプションを使用した Oracle Names バージョン 2 からのアップグレード
- ROSFILES からのアップグレード
- ドメインおよびトポロジ・チェックポイント・ファイルへのリージョン・チェックポイント・ファイルのアップグレード
- アップグレード・チェックリストの確認

### データベースを使用した Oracle Names バージョン 2 からのアップグレード

既存の Oracle Names Server バージョン 2 データベースから、バージョン 9 のリージョン・データベースへデータをアップグレードおよび転送するには、Oracle Network Manager がネットワーク定義を格納しているノードで、UNIX プラットフォームでは `$ORACLE_HOME/network/admin`、Windows プラットフォームでは `%ORACLE_HOME%\network\admin` にある `namesupg.sql` スクリプトを実行します。

namesupg.sql スクリプトを実行するには、NAMES\_DOM および NAMES\_DID の 2 つの表を作成し、既存の names.ora ファイルの値を使用して移入する必要があります。

- NAMES\_DOM 表には、names.ora ファイルの NAMES.DOMAINS パラメータに指定されているドメインごとに、1 行の DOMAIN 列が必要です。
- NAMES\_DID 表には、NMO\_INFORMATION 表の NAME\_P 列に定義されている ID が必要です。NAME\_P 列は、names.ora ファイルの NAMES.ADMIN\_REGION パラメータに指定されている DOCNAME と同じです。

データをアップグレードするには、次の手順を行います。

1. 次のように、NAMES\_DOM 表を作成します。

```
SQL> CONNECT user/password
SQL> CREATE TABLE NAMES_DOM (domain varchar(256));
```

2. names.ora ファイルの NAMES.DOMAINS パラメータに指定されているドメイン名で表を移入します。たとえば、次のような NAMES.DOMAIN パラメータ設定があるとします。

```
NAMES.DOMAINS=
(DOMAIN_LIST=
(DOMAIN=
(NAME=)
(MIN_TTL=86400))
(DOMAIN=
(NAME=com)
(MIN_TTL=86400))
(DOMAIN=
(NAME=acme.com)
(MIN_TTL=86400))
```

この例では、次のように、root ドメイン、acme サブドメインおよび com ドメインに対する 3 行を作成する必要があります。

```
SQL> INSERT into NAMES_DOM values ('(root)');
SQL> INSERT into NAMES_DOM values ('acme');
SQL> INSERT into NAMES_DOM values ('acme.com');
```

3. 次のように、NAMES\_DID 表を作成します。

```
SQL> CREATE TABLE NAMES_DID (did number(10));
```

- names.ora ファイル内の NAMES.ADMIN\_REGION パラメータの DOCNAME 値を検索します。DOCNAME は、リージョンに対応付けられた名前を示します。次の例では、DOCNAME は sbx です。

```
NAMES.ADMIN_REGION= (REGION=
 (NAME=local_region.world)
 (TYPE=rosdb)
 (USERID=names)
 (PASSWORD=names)
 (description=
 (ADDRESS_LIST=
 (ADDRESS=
 (PROTOCOL=tcp)
 (HOST=nineva)
 (PORT=1387)))
 (CONNECT_DATA=(SID=em)))
 (DOCNAME=sbx)
 (VERSION=34619392) # 2.1.4
 (RETRY=60))
```

- DOCNAME に対応付けられた ID を NMO\_INFORMATION 表に問い合わせ、この ID を NAMES.DOM 表に挿入します。

```
SQL> SELECT ID from NMO_INFORMATION where name_P=docname;
SQL> INSERT into NAMES_DID
 select DID from NMO_INFORMATION
 where NAME_p='docname';
```

- namesupg.sql スクリプトを実行します。

```
SQL> CONNECT user/password
SQL> @ORACLE_HOME/network/admin/namesupg.sql;
```

## 動的検出オプションを使用した Oracle Names バージョン 2 からのアップグレード

動的検出オプションを使用して Oracle Names バージョン 2 をアップグレードする手順は、Oracle Names バージョン 8 を使用して、リージョン・データベースに情報を格納するかどうかによって異なります。

- 非リージョン・データベースのアップグレード
- リージョン・データベースのアップグレード

## 非リージョン・データベースのアップグレード

動的検出オプションを使用して Oracle Names バージョン 2 から Oracle Names バージョン 8 へアップグレードする場合、新しい Oracle Names Server は、古いチェックポイント・ファイルから登録されたデータを取得する必要があります。データが登録されていない場合は、『Oracle9i Net Services 管理者ガイド』で説明している次の手順を行うことによって、オブジェクトを登録できます。

## リージョン・データベースのアップグレード

動的検出オプションを使用して Oracle Names バージョン 2 を実行済で、リージョン・データベースを Oracle Names 情報のリポジトリとして構成する場合は、次の手順を行う必要があります。

1. Oracle Names バージョン 2 のローカル管理リージョンに格納されている情報を Oracle Network Manager から `tnsnames.ora` ファイルに書き込むか、またはバージョン 8 の Oracle Names 制御ユーティリティを使用してコマンドラインから次のコマンドを実行します。

```
NAMESCTL
NAMESCTL> DUMP_TNSNAMES
```

2. データベースが常駐しているコンピュータで、UNIX プラットフォームでは `$ORACLE_HOME/network/admin`、Windows プラットフォームでは `ORACLE_HOME\network\admin` にある `namesini.sql` スクリプトを実行します。

```
SQL> CONNECT user/password
SQL> @ORACLE_HOME/network/admin/namesini.sql;
```

3. Oracle Net Manager を使用して、すべての Oracle Names Server 構成ファイル (`names.ora`) に `NAMES.ADMIN_REGION` パラメータを構成します。

**参照：** Oracle Names Server の作成の詳細は、『Oracle9i Net Services 管理者ガイド』を参照してください。

4. Oracle Net Manager または Oracle Names 制御ユーティリティを使用して、`tnsnames.ora` ファイルを Oracle Names Server バージョン 9 にロードします。

| Oracle Net Manager を使用する場合                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Oracle Names 制御ユーティリティを使用する場合                                                            |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| <div>1. Oracle Net Manager を起動します。<br/>-UNIX では、\$ORACLE_HOME/bin で netmgr を実行します。<br/>-Windows プラットフォームでは、「スタート」&gt;「プログラム」&gt;「Oracle - HOME_NAME」&gt;「Configuration and Migration Tools」&gt;「Net Manager」を選択します。</div> <div>2. ナビゲータ・ペインで、Oracle Names Servers を拡張します。</div> <div>3. 「Oracle Names Server」を選択します。</div> <div>4. 右のペインのリストから、「Manage Data」を選択します。</div> <div>5. 「Net Service Names」タブを選択します。</div> <div>6. 「Load」を選択します。</div> <div>7. 手順 1 で作成された「File」フィールドに、Oracle Network Manager によって生成された tnsnames.ora ファイルのパス名およびファイル名を入力します。</div> <div>8. 「Execute」を選択します。</div> <div>9. 「File」&gt;「Save Network Configuration」を選択します。</div> | <div>コマンドラインから、次のコマンドを入力します。<br/>namesctl<br/>NAMESCTL&gt; LOAD_TNSNAMES file_name</div> |



## ROFILES からのアップグレード

Oracle Names バージョン 8 以上では、Resource Object Store (ROS) ファイル (ROFILES) を使用する古い構成はサポートしません。ROFILES は、Oracle Names データベース表へ直接アップグレードするか、または最初に tnsnames.ora ファイルへアップグレードしてから Oracle Names へアップグレードする必要があります。次の項では両方の手順を説明します。

- データベース表への ROFILES のアップグレード
- tnsnames.ora ファイルへの ROFILES のアップグレード

### データベース表への ROFILES のアップグレード

ROFILES をデータベース表へアップグレードするには、次の操作を行います。

1. Oracle Network Manager のデータベース・ユーザー・アカウントを作成します。

```
SQL> CONNECT system/password
SQL> CREATE USER user
 IDENTIFIED BY password
 DEFAULT TABLESPACE users
 TEMPORARY TABLESPACE temp;
```

2. 必要な表を構築するには、次に説明するスクリプトをサーバーに対して実行する必要があります。通常、これらのスクリプトは Oracle Network Manager ノードで実行されます。

```
SQL> CONNECT username/password
SQL> @ORACLE_HOME/dbs/rosbild.sql;
SQL> @ORACLE_HOME/dbs/nmcbild.sql;
SQL> @ORACLE_HOME/dbs/rosgrnt.sql;
SQL> @ORACLE_HOME/dbs/nmcgrnt.sql;
```

| スクリプト       | 説明                                                                                              |
|-------------|-------------------------------------------------------------------------------------------------|
| rosbild.sql | ROS が使用する表を作成します。                                                                               |
| nmcbild.sql | Oracle Network Manager Objects (NMO) コンポーネントが使用する表を作成します。                                       |
| rosgrnt.sql | 共通の表にアクセス権限を付与します。ユーザー名を入力するように求められます。Oracle Network Manager のアカウントを設定したときに使用したのと同じユーザー名を使用します。 |
| nmcgrnt.sql | Oracle Network Manager 表にアクセスするユーザーにアクセス権限を付与します。                                               |

3. Oracle Network Manager から、ROSFILES をデータベースに保存します。
  - a. 「File」 > 「Save As」を選択します。
  - b. 「Save Network Definition」 ダイアログ・ボックスで「Database」を選択し、「OK」を選択します。
  - c. 「Connect」 ダイアログ・ボックスに、手順 1 で作成したデータベースのユーザー名とパスワード、およびデータベースのネット・サービス名を入力します。
  - d. 「OK」を選択します。
  - e. 「Save Network Definition」 ダイアログ・ボックスで、保存するネットワーク名を選択または入力します。
  - f. 「File」 > 「Generate」を選択してネットワーク定義を保存し、保存された定義から Oracle Names 表を作成します。
  - g. 「File」 > 「Exit」を選択して、Oracle Network Manager を終了します。
4. B-12 ページの「データベースを使用した Oracle Names バージョン 2 からのアップグレード」で説明するとおり、サーバーで NAMES\_DID および NAMES\_DOM 表を作成し、namesupg.sql スクリプトを実行します。

### tnsnames.ora ファイルへの ROSFILES のアップグレード

ROSFILES を tnsnames.ora ファイルへアップグレードし、tnsnames.ora ファイルを Oracle Names へインポートするには、次の操作を行います。

1. ROSFILES から tnsnames.ora ファイルを作成します。
  - a. Oracle Network Manager から、「Special」 > 「Preferences」を選択します。
  - b. 「Preferences」ダイアログ・ボックスで、Oracle Names が選択されていないことを確認します。
  - c. 「File」 > 「Generate」を選択してネットワーク定義を更新し、tnsnames.ora ファイルを作成します。
  - d. 「File」 > 「Exit」を選択して、Oracle Network Manager を終了します。
2. Oracle Net Manager または Oracle Names 制御ユーティリティを使用して、tnsnames.ora ファイルを Oracle Names Server にロードします。

| Oracle Net Manager を使用する場合                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Oracle Names 制御ユーティリティを使用する場合                                                               |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| <p>1. Oracle Net Manager を起動します。</p> <p>-UNIX では、\$ORACLE_HOME/bin で netmgr を実行します。</p> <p>-Windows プラットフォームでは、「スタート」&gt;「プログラム」&gt;「Oracle - HOME_NAME」&gt;「Configuration and Migration Tools」&gt;「Net Manager」を選択します。</p> <p>2. ナビゲータ・ペインで、Oracle Names Servers を拡張します。</p> <p>3. 「Oracle Names Server」を選択します。</p> <p>4. 右のペインのリストから、「Manage Data」を選択します。</p> <p>5. 「Net Service Names」タブを選択します。</p> <p>6. 「Load」を選択します。</p> <p>7. 手順 1 で作成された「File」フィールドに、Oracle Network Manager によって生成された tnsnames.ora ファイルのパス名およびファイル名を入力します。</p> <p>8. 「Execute」を選択します。</p> <p>9. 「File」&gt;「Save Network Configuration」を選択します。</p> | <p>コマンドラインから、次のコマンドを入力します。</p> <pre>namesctl<br/>NAMESCTL&gt; LOAD_TNSNAMES file_name</pre> |

参照：

- 『Oracle Network Manager 管理者ガイド』を参照してください。
- 『Oracle Names 管理者ガイド』を参照してください。

## ドメインおよびトポロジ・チェックポイント・ファイルへのリージョン・チェックポイント・ファイルのアップグレード

Oracle8i では、リージョン・チェックポイント・ファイル ckpreg.ora には、トポロジおよびドメイン認可データが含まれていました。Oracle9i では、このデータは 2 つのファイルに分割されています。トポロジ・チェックポイント・ファイル ckptop.ora は、管理リージョンのドメインおよび各ドメイン用の Oracle Names Server を定義します。ドメイン・チェックポイント・ファイル ckpdom.ora には、各ドメインの認可データが含まれます。

リージョン・データベースを使用する場合、これらのファイルは自動的に生成されます。リージョン・データベースを使用せずに、チェックポイント・ファイルのデータを使用する場合は、チェックポイント・ファイルに関係なく、リージョンで実行している Oracle Names Server に依存したり、またはデータを ckpreg.ora ファイルから ckptop.ora ファイルへ移動することができます。

その他の Oracle Names Server のデータを使用するには、次の手順を行います。

1. Oracle Names Server を Oracle9i リリース 2 (9.2) にアップグレードします。
2. それぞれの Oracle Names Server には、UNIX オペレーティング・システムの場合は .sdns.ora ファイルが \$ORACLE\_HOME/network/names に、Windows オペレーティング・システムの場合は sdns.ora ファイルが %ORACLE\_HOME%\network\names にあることを確認します。

このファイルには、最初の Oracle Names Server の名前およびアドレスが含まれます。存在しない場合、「Oracle Net Manager's Command」>「Discover Oracle Names Servers command」を選択するか、または Oracle Names 制御ユーティリティの REORDER\_NS コマンドを使用して、他の Oracle Names Server を検出します。

3. Oracle Names Server を起動します。

Oracle Names Server が起動すると、他の Oracle Names Server を検索し、そこからトポロジおよびドメイン・データ情報をダウンロードします。

データを ckpreg.ora ファイルから ckptop.ora ファイルへコピーまたは移動するには、次の手順を行います。

1. Oracle Names Server を Oracle9i リリース 2 (9.2) にアップグレードします。
2. ckpreg.ora ファイルを ckptop.ora ファイルへ移動します。次にその例を示します。

```
cd network/names
mv ckpreg.ora ckptop.ora
```

3. Oracle Names Server を起動します。

Oracle Names Server が起動すると、自動的に ckpdom.ora ファイルが生成されます。

**参照：** ckptop.ora ファイルの例は、『Oracle9i Net Services 管理者ガイド』を参照してください。

## アップグレード・チェックリストの確認

次のチェックリストは、Oracle Names バージョン 9 への適切なアップグレードを確認するために提供されています。

- ❑ 各リージョンにあるすべての Oracle Names Server を、バージョン 8 の同じリリースへアップグレードします。
- ❑ Oracle Names バージョン 2 を実行済で、データベースを Oracle Names 情報のリポジトリとして更新する場合は、ネットワーク定義が格納されているノードで `namesupg.sql` スクリプトを実行します。
- ❑ 動的検出オプションを使用して Oracle Names バージョン 2 を実行済で、リージョン・データベースを Oracle Names 情報のリポジトリとして構成する場合は、次の操作を行う必要があります。
  1. データベースをインストールするノードで、`namesini.sql` スクリプトを実行します。
  2. Oracle Net Manager を使用して、すべての `names.ora` に `NAMES.ADMIN_REGION` パラメータを構成します。`NAMES.ADMIN_REGION` パラメータの詳細は、『Oracle9i Net Services リファレンス・ガイド』を参照してください。
- ❑ 各リージョンに 2 つ以上の Oracle Names Server を設定して、フォルト・トレランスを提供します。



---

## Server Manager から SQL\*Plus への移行

この付録では、Server Manager のラインモード・スクリプトを SQL\*Plus で動作させるための変更処理について説明します。Server Manager は、Oracle9i ではサポートされていません。Server Manager ラインモードを使用して SQL スクリプトを実行している場合は、SQL\*Plus と互換性を持たせるために、これらのスクリプトを変更し、SQL\*Plus を使用して実行する必要があります。

この付録では、次の項目について説明します。

- 起動の違い
- コマンド
- 構文の違い

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

---

**注意：** この付録では、Server Manager ラインモードを Server Manager と表記します。

---

## 起動の違い

Server Manager と SQL\*Plus を起動する方法は異なります。SQL スクリプトを変更し、SQL\*Plus を正しく起動する必要があります。次の項では起動の違いを説明し、SQL\*Plus を起動するためのオプションについて説明します。

### Server Manager の起動

Server Manager を起動するには、システム・プロンプトで Server Manager のプログラム名を入力します。このプログラム名はオペレーティング・システム固有です。Server Manager 起動後、次の例のように CONNECT コマンドを使用して接続します。

```
CONNECT hr/hr
```

### SQL\*Plus の起動

次の項では、SQL\*Plus を起動する様々な方法について説明します。

#### NOLOG オプションを使用した SQL\*Plus の起動

SQL\*Plus を Server Manager と同様に動作させるには、SQL\*Plus 起動時に次のように NOLOG オプションを使用します。

```
sqlplus /nolog
```

SQL\*Plus が起動し、CONNECT コマンドを使用してユーザーとして接続できます。

#### 接続情報を使用した SQL\*Plus の起動

SQL\*Plus を起動する他の方法として、プログラム起動時に接続情報を入力する方法があります。たとえば、SQL\*Plus を起動し、パスワード hr を持つユーザー hr として接続するには、次のとおり入力します。

```
sqlplus hr/hr
```

SQL\*Plus が起動し、ユーザー hr として接続します。

#### オプションまたは接続情報なしの SQL\*Plus の起動

オプションまたは接続情報なしで SQL\*Plus を起動するには、次のように入力します。

```
sqlplus
```

ユーザー名およびパスワードを入力するプロンプトが表示されます。有効なユーザー名およびパスワードを入力すると、SQL\*Plus が起動し、プロンプトで指定したユーザーとして接続します。ただし、SQL スクリプトで、ユーザー名およびパスワードの入力を省略することもできます。



# コマンド

Server Manager と SQL\*Plus は、両方のプログラムで同じ処理を行う特定のコマンドを共有しています。ただし、Server Manager での処理と SQL\*Plus での処理が異なるコマンドもあります。Server Manager から SQL\*Plus への移行を正常に行うには、このような違いおよび類似点を理解する必要があります。次の項では、SQL\*Plus によって正しく解釈されるコマンドを使用するための SQL スクリプトの変更について説明します。

## SQL\*Plus リリース 8.1 で導入されたコマンド

表 C-1 に、SQL\*Plus リリース 8.1 以上で使用可能な Server Manager コマンドを示します。SQL\*Plus で実行している SQL スクリプトで、これらのコマンドを使用できます。

**注意：** Oracle7 または Oracle8 でこれらのコマンドを含む SQL スクリプトを実行する場合、Server Manager を使用する必要があります。リリース 8.1 より前の SQL\*Plus では、これらのコマンドを含むスクリプトを実行しません。

表 C-1 SQL\*Plus リリース 8.1 で導入されたコマンド

| コマンド              | 説明                                                                                                                                         |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| ARCHIVE LOG       | オンライン REDO ログ・ファイルの自動アーカイブを開始または停止するか、特定の REDO ログ・ファイルを手動で（明示的に）アーカイブするか、またはアーカイブについての情報を表示します。                                            |
| RECOVER           | 1 つ以上の表領域、1 つ以上のデータ・ファイルまたはデータベース全体で、メディア・リカバリを実行します。                                                                                      |
| SET AUTORECOVERY  | ON にすると、リカバリ中に必要なアーカイブ REDO ログ・ファイルのデフォルトのファイル名を、RECOVER コマンドに自動的に適用します。AUTORECOVERY が ON に設定されている場合、必要なファイルが所定の名前で所定の位置に配置されているため対話は不要です。 |
| SET INSTANCE      | セッションに対するデフォルト・インスタンスを指定のインスタンス・パスに変更します。データベースへは接続しません。デフォルトのインスタンスは、インスタンスが指定されていない場合にコマンドに対して使用されます。                                    |
| SET LOGSOURCE     | リカバリ中に、アーカイブ・ログが取り出される場所を指定します。デフォルト値は、LOG_ARCHIVE_DEST 初期化パラメータによって設定されます。パス名なしで SET LOGSOURCE コマンドを発行すると、デフォルトの位置をリストアします。               |
| SHOW AUTORECOVERY | 自動リカバリが可能かどうかを示します。                                                                                                                        |

表 C-1 SQL\*Plus リリース 8.1 で導入されたコマンド（続き）

| コマンド            | 説明                                                                                                                                              |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| SHOW INSTANCE   | デフォルトのインスタンスに対する接続文字列を示します。SHOW INSTANCE を使用していない場合、または SET INSTANCE コマンドの LOCAL オプションを使用している場合は、SET INSTANCE は LOCAL 値を戻します。                   |
| SHOW LOGSOURCE  | アーカイブ・ログの位置についての現行の設定を示します。LOG_ARCHIVE_DEST 初期化パラメータによって指定されたように、デフォルトの設定が有効な場合、DEFAULT を表示します。                                                 |
| SHOW PARAMETERS | 1 つ以上の初期化パラメータについての現行の値を表示します。コマンドの後に文字列を指定しなくても、SHOW PARAMETERS コマンドはすべての初期化パラメータを表示します。                                                       |
| SHOW SGA        | 現行のインスタンスのシステム・グローバル領域についての情報を表示します。                                                                                                            |
| SHUTDOWN        | 現在実行中の Oracle インスタンスを停止し、任意でデータベースをクローズしてディスマウントします。<br><b>注意：</b> SQL*Plus リリース 8.1 の STARTUP および SHUTDOWN コマンドは、Oracle7 Server ではサポートされていません。 |
| STARTUP         | データベースのマウントやオープンなどのいくつかのオプションで、Oracle インスタンスを起動します。<br><b>注意：</b> SQL*Plus リリース 8.1 の STARTUP および SHUTDOWN コマンドは、Oracle7 Server ではサポートされていません。  |

Server Manager と SQL\*Plus に共通のコマンド

表 C-2 に示すコマンドは、Server Manager と SQL\*Plus の両方において使用可能であり、Oracle の以前のリリースの両方のプログラムでも使用可能です。SQL\*Plus を使用するために、SQL スクリプトのこれらのコマンドを変更する必要はありません。

**注意：** 2つのプログラムのこれらのコマンドに対する出力で、書式の違いが多少ある場合があります。

表 C-2 既存の SQL\*Plus コマンドに対応する Server Manager コマンド

| コマンド              | 説明                                                                                                                                                                                                                                                                                                     |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CONNECT           | 指定されたユーザー名を使用して、データベースに接続します。                                                                                                                                                                                                                                                                          |
| DESCRIBE          | ファンクション、パッケージ、パッケージ本体、プロシージャ、表、ビューまたはオブジェクト型について説明します。たとえば、表の場合、表の各列の定義を表示します。                                                                                                                                                                                                                         |
| REMARK            | 通常、SQL スクリプト・ファイルにコメントを入力します。                                                                                                                                                                                                                                                                          |
| SET COMPATIBILITY | 互換性モードを V7、V8 または NATIVE に設定します。互換性モードの設定は、文字列、整合性制約およびロールバック・セグメントの記憶域パラメータの仕様に影響します。NATIVE はデータベースのバージョンと一致します。                                                                                                                                                                                      |
| SET ECHO          | コマンド・ファイルの各コマンドの実行時に、START コマンドがそのコマンドをリストするかどうかを制御します。ON はコマンドをリストし、OFF はリストを抑止します。                                                                                                                                                                                                                   |
| SET NUMWIDTH      | 数を表示するデフォルトの幅を設定します。                                                                                                                                                                                                                                                                                   |
| SET SERVEROUTPUT  | ストアド・プロシージャの出力 (DBMS_OUTPUT.PUT_LINE) または SQL*Plus の PL/SQL ブロックを表示するかどうかを制御します。OFF は DBMS_OUTPUT.PUT_LINE の出力を停止し、ON は出力を表示します。                                                                                                                                                                       |
| SET TERMOUT       | コマンド・ファイルから実行されるコマンドによって生成される出力の表示を制御します。OFF は、コマンド・ファイルからの出力をスプールできますが、画面への表示を停止します。ON は出力を表示します。                                                                                                                                                                                                     |
| SHOW ALL          | ERRORS、PARAMETERS および SGA を除き、SET コマンドで設定されたすべてのシステム変数をアルファベット順で表示します。                                                                                                                                                                                                                                 |
| SHOW ERRORS       | プロシージャ、パッケージまたはファンクションの前のコンパイルで生成されたエラーがあれば、それを表示します。                                                                                                                                                                                                                                                  |
| SPOOL             | 問合せ結果をオペレーティング・システム・ファイル（任意で SQL*Plus）に格納し、そのファイルをプリンタに送ります。<br><br><b>注意：</b> スプール・ファイルの拡張機能は、SQL*Plus と Server Manager で異なります。拡張機能を確認するには、SPOOL コマンドを発行するときに指定してください。また、SQL*Plus は、空白を繰り返すかわりにタブ文字を使用して、端末出力の空白をフォーマットします。SQL*Plus の SET TAB OFF を使用して、この置換を回避してください。Server Manager では、タブ文字は出力されません。 |

Server Manager コマンドと同等の SQL\*Plus コマンド

表 C-3 に、Server Manager コマンドと名前は異なりますが、対応している SQL\*Plus コマンドを示します。SQL スクリプトで次のいずれかの Server Manager コマンドを使用する場合、スクリプトを変更して、かわりに SQL\*Plus コマンドを使用します。

表 C-3 Server Manager コマンドと同等の SQL\*Plus コマンド

| Server Manager コマンド | SQL*Plus コマンド | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SET CHARWIDTH       | COLUMN FORMAT | SQL*Plus で COLUMN FORMAT コマンドを使用して、キャラクタ列、日付列および数値列の列幅を設定できます。SQL スクリプトで、Server Manager の SET CHARWIDTH、SET DATEWIDTH および SET LONGWIDTH を SQL*Plus の COLUMN FORMAT コマンドに置き換えます。<br><br>COLUMN FORMAT を使用して、すべてのキャラクタ列を変更します。1 つのコマンドですべてのキャラクタ列を変更する同等のコマンドはありません。<br><br>たとえば、SQL スクリプトに次のエントリがあるとします。<br><br>SET CHARWIDTH 5<br><br>このコマンドは、Server Manager のすべてのキャラクタ列幅を 5 に設定します。<br><br>first_name のような特定の列を 5 文字幅で表示するには、次の SQL*Plus コマンドを入力します。<br><br>COLUMN first_name FORMAT A5<br><br>COLUMN FORMAT を使用して、すべてのキャラクタ列を変更します。1 つのコマンドですべてのキャラクタ列を変更する同等のコマンドはありません。<br><br>COLUMN FORMAT を使用して、すべての日付列を変更します。1 つのコマンドですべての日付列を変更する同等のコマンドはありません。<br><br>SET LONG を使用して、フェッチおよび表示する LONG 列の数を指定します。 |
| SET DATEWIDTH       |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| SET LONGWIDTH       |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

表 C-3 Server Manager コマンドと同等の SQL\*Plus コマンド (続き)

| Server Manager コマンド | SQL*Plus コマンド                         | 説明                                                                                                                                                                                                                                                                                                                                  |
|---------------------|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SET STOPONERROR     | WHENEVER SQLError<br>WHENEVER OSERROR | WHENEVER SQLError および WHENEVER OSERROR コマンドを使用して、SQL エラーまたはオペレーティング・システム・エラーが発生したときに、終了または継続するよう SQL*Plus に指示します。Server Manager の SET STOPONERROR コマンドのかわりに、SQL スクリプトでこれらのコマンドを使用します。<br><br>WHENEVER SQLError および WHENEVER OSERROR のいずれの場合でも、EXIT 句が SQL*Plus に終了の指示を出し、CONTINUE 句が SQL*Plus に継続の指示を出します。他の用語および句もこれらのコマンドに使用可能です。 |

SET TIMING コマンドの違い

SET TIMING コマンドは Server Manager と SQL\*Plus の両方で使用できますが、オペレーティング・システムによっては 2 つのプログラムで異なる機能を持つ場合もあります。詳細は、ご使用のオペレーティング・システム固有の Oracle マニュアルを参照してください。ご使用のオペレーティング・システムの 2 つのプログラムで、SET TIMING コマンドが異なる機能を持つ場合、このコマンドが SQL\*Plus で正しく機能するように、SQL スクリプトを変更します。

SQL\*Plus で使用できない Server Manager コマンド

次の Server Manager コマンドは、SQL\*Plus リリース 8.1 以上では使用できません。

- SET MAXDATA
- SET RETRIES

SQL スクリプトからこれらのコマンドを削除してください。

構文の違い

次の項では、Server Manager と SQL\*Plus の構文の違いについて説明します。SQL\*Plus を使用してスクリプトを実行する前に、SQL スクリプトを変更し、SQL\*Plus 構文の規則に適合させます。

## コメント

SQL\*Plus は、次のタイプのコメントを認識します。

- SQL\*Plus の REMARK コマンド (または REM)
- SQL コメント・デリミタ (/\*...\*/)
- ANSI/ISO コメント (--)

SQL\*Plus スクリプトでこれらのコメントを使用するための詳細は、『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。

Server Manager はこれらのタイプのコメントをサポートしますが、中には動作が異なるものもあります。また、Server Manager では使用できますが、SQL\*Plus では使用できない特定のタイプのコメントもあります。次の項では、Server Manager と SQL\*Plus の各コメントおよび構文の違いについて説明します。

### REMARK コマンド (または REM)

一般に、REMARK コマンドは Server Manager と SQL\*Plus で同じ動作をします。使用している SQL スクリプトに REMARK コマンドがあっても、変更する必要はありません。ただし、違いが 1 つあります。SQL\*Plus は、Server Manager とは異なり、REMARK コマンドを終了するハイフンを解釈します。この違いについては、C-11 ページの「[分割行として使用するハイフン](#)」を参照してください。

### SQL コメント・デリミタ (/\* ... \*/)

Server Manager では、SQL コメント・デリミタをセミコロン (;) の後に置くことができますが、SQL\*Plus では SQL コメント・デリミタをセミコロンの後に置くことはできません。この 1 つの違いを除き、SQL コメント・デリミタは、Server Manager と SQL\*Plus で同様に動作します。

SQL スクリプトにセミコロンの後に置かれた SQL コメント・デリミタが含まれる場合、コメントをその行に移動するか、セミコロンを削除して次の行にスラッシュ (/) を置き、SQL 文を終了します。

たとえば、SQL スクリプトの 1 つに、次の Server Manager コードがあるとします。

```
SELECT * FROM hr.employees
 WHERE job_id LIKE '%CLERK'; /* Includes only clerks. */
```

SQL\*Plus では、このコードを次のエントリのいずれかに置き換えます。

```
SELECT * FROM hr.employees
 WHERE job_id LIKE '%CLERK';
/* Includes only clerks. */
```

```
SELECT * FROM hr.employees
 WHERE job_id LIKE '%CLERK' /* Includes only clerks. */
/
```

## ANSI/ISO コメント (--)

Server Manager では ANSI/ISO コメントをセミコロン (;) の後に置くことができますが、SQL\*Plus では ANSI/ISO コメントをセミコロンの後に置くことはできません。この 1 つの違いを除き、ANSI/ISO コメントは、Server Manager と SQL\*Plus で同様に動作します。

SQL スクリプトにセミコロンの後に置かれた ANSI/ISO コメントが含まれる場合、コメントをその行に移動するか、セミコロンを削除して次の行にスラッシュ (/) を置き、SQL 文を終了します。

たとえば、SQL スクリプトの 1 つに、次の Server Manager コードがあるとします。

```
SELECT * FROM hr.employees
 WHERE job_id LIKE '%CLERK'; -- Includes only clerks.
```

SQL\*Plus では、このコードを次のエントリのいずれかに置き換えます。

```
SELECT * FROM hr.employees
 WHERE job_id LIKE '%CLERK';
-- Includes only clerks.

SELECT * FROM hr.employees
 WHERE job_id LIKE '%CLERK' -- Includes only clerks.
/
```

## Server Manager のシャープ (#)・コメント

Server Manager は、コメント行を示すシャープ符号 (#) の使用をサポートします。スクリプトにこれらのコメントが含まれる場合、「#」を「--」に変更し、SQL\*Plus を使用してスクリプトを実行します。

たとえば、SQL スクリプトの 1 つに、次の Server Manager コードがあるとします。

```
This statement returns only clerks.
SELECT * FROM hr.employees
 WHERE job_id LIKE '%CLERK';
```

SQL\*Plus では、このコードを次のエントリに置き換えます。

```
-- This statement returns only clerks.
SELECT * FROM hr.employees
 WHERE job_id LIKE '%CLERK';
```

## 空白行

Server Manager は SQL 文にある空白行を無視しますが、SQL\*Plus が空白行を検出すると、デフォルトの動作では文の記録を停止し、プロンプトに戻ります。

両方の製品とも、異なる SQL 文の間では空白行を許可します。この項の説明は、SQL 文の句の間にある空白行にのみ適用されます。

SQL\*Plus では、SET SQLBLANKLINES コマンドが空白行の処理方法を変更します。SQLBLANKLINES が OFF (デフォルトの設定) に設定され、空白行を含む SQL 文がある場合、SQL\*Plus は空白行に文をバッファし、文を実行せずにプロンプトに戻ります。この動作によって、対話ユーザーは不要な SQL コマンドを終了またはバッファするか、またはこのバッファされた SQL コマンドを実行または編集する前に他の SQL\*Plus コマンドを実行できます。

SQL スクリプトのいずれかが SQL 文内に空白行を含む場合、SQLBLANKLINES を ON に設定するか、または SQL\*Plus を使用してこれらのスクリプトを実行する前に空白行を削除します。

たとえば、SQL スクリプトの 1 つに、次の SQL 文があるとしてします。

```
SELECT employee_id, first_name, last_name, salary, commission_pct
 FROM hr.employees
 WHERE job_id LIKE '%MAN';
```

SQLBLANKLINES を ON にするか、または空白行を削除します。

```
SELECT employee_id, first_name, last_name, salary, commission_pct
 FROM hr.employees
 WHERE job_id LIKE '%MAN';
```

空白行を削除しないか、または SQLBLANKLINES を ON に設定しない場合は、SQL\*Plus はコードの各空白行をコマンド終了記号として扱います。

SQLBLANKLINES の値は、PL/SQL ブロックの空白行に影響しません。これらはブロックの一部として扱われるため、SQL\*Plus プロンプトに戻りません。

対話ユーザーは、SQLBLANKLINES の値にかかわらず、行にピリオドを入力することによって SQL または PL/SQL 文を終了できます。



## ハイフン継続文字

SQL\*Plus は、長い SQL 文または SQL\*Plus コマンドに対する継続文字として、ハイフン (-) の使用をサポートします。たとえば、次のように継続文字を使用できます。

```
SELECT employee_id, first_name, last_name FROM hr.employees -
WHERE job_id LIKE '%MAN';
```

Server Manager は継続文字としてのハイフンの使用をサポートしませんが、ハイフンを SQL スクリプトで他の用途に使用できます。ハイフンを SQL スクリプトで使用すると、SQL\*Plus はハイフンを継続文字として解釈し、予期しない出力が行われることがあります。

次の項では、ハイフンを別の用途で使用しているにもかかわらず、SQL\*Plus が SQL スクリプトのハイフンの使用を継続文字として解釈する場合について説明します。SQL スクリプトがハイフンを使用しているかどうかを確認し、次の記述と同様の例を避けるために、SQL スクリプトを変更します。

### 分割行として使用するハイフン

SQL スクリプトが、コードの分割行として REMARK コマンドの後に長いハイフン行を使用することがあります。SQL スクリプトからの次のサンプル行について考えてみます。

```
Rem -----
SELECT employee_id, first_name, last_name, job_id
 FROM hr.employees;
```

この文で、SQL\*Plus は、SELECT 文の最初の行を前の行の継続として、つまり REMARK コメントとして解釈します。このため、FROM 行は SQL 文の最初の行として解釈され、SQL\*Plus は次のエラーを戻します。

```
unknown command beginning "FROM hr..." - rest of line ignored.
```

SQL スクリプトの分割行としてハイフンを使用する場合、SQL\*Plus を使用してスクリプトを実行する前に、ハイフンの前の REM コマンドを削除します。

### マイナス符号として使用するハイフン

ハイフンはマイナス符号と同じキーボード文字であるため、行の最後にハイフンを付けることができます。SQL スクリプトからの次のサンプル行について考えてみます。

```
CREATE TABLE xx (
 a int,
 b int,
 c int);

INSERT INTO xx VALUES (10, 20, 30);
```

```
SELECT a + b -
 c FROM xx;
```

マイナス記号は継続文字として解釈されるため、SQL\*Plus は「c」を別名として解釈します。

```
SELECT a + b c FROM xx;
```

このため、SQL\*Plus は次の予期しない出力を戻します。

```
 C

 30
```

ただし、Server Manager はこのコードを次のように解釈します。

```
SELECT a + b - c FROM xx;
```

このため、Server Manager は次の期待どおりの出力を戻します。

```
A+B-C

 0
```

SQL スクリプトの行の最後にマイナス符号がないことを確認してください。

## アンパサンド

SQL\*Plus はアンパサンド (&) を置換変数として解釈し、Server Manager はアンパサンドを標準の文字列として解釈します。アンパサンドに続くテキストに定義された値がない場合、アンパサンドがコメントで囲まれても、SQL\*Plus は未定義値として解釈し、ユーザーに入力を求めるプロンプトを表示します。このため、アンパサンドは SQL\*Plus において予期しない出力を行う場合があります。

アンパサンドを標準のテキスト文字列として使用する SQL スクリプトがある場合、次の 2 つの回避策があります。

- SET ESCAPE コマンドを使用して、各アンパサンドの前にエスケープ文字を置きます。
- SET DEFINE OFF コマンドを使用して、置換変数の認識を使用禁止にします。

---

---

**注意：** 他に有効な置換変数がある場合、SET DEFINE OFF コマンドを使用しないでください。使用すると、他の変数が認識されなくなります。

---

---

たとえば、次の SQL 文はユーザーに SQL\*Plus での入力を求めるプロンプトを表示します。

```
CREATE TABLE "Employees & Managers" (
 Employees varchar(16),
 Managers varchar(16));
```

Enter value for managers:

## SET ESCAPE コマンドの使用

ユーザー・プロンプトを避けるには、SET ESCAPE コマンドを使用してエスケープ文字を設定できます。その後、アンパサンドの前にエスケープ文字を置きます。円記号 (¥) はエスケープ文字として使用されます。

前述の例のプロンプトを避けるには、SET ESCAPE コマンドを使用して、次のようにエントリを変更します。

```
SET ESCAPE ¥
```

```
CREATE TABLE "Employees ¥& Managers" (
 Employees varchar(16),
 Managers varchar(16));
```

## SET DEFINE OFF コマンドの使用

前述の例のプロンプトを避けるには、SET DEFINE OFF コマンドを使用して、次のようにエントリを変更します。

```
SET DEFINE OFF
```

```
CREATE TABLE "Employees & Managers" (
 Employees varchar(16),
 Managers varchar(16));
```

## CREATE TYPE および CREATE LIBRARY コマンド

SQL\*Plus は、CREATE TYPE および CREATE LIBRARY コマンドを PL/SQL ブロックとして扱います。このため、SQL\*Plus では、別の行にスラッシュ (/) を使用してこれらのコマンドを終了する必要があります。Server Manager では、セミコロン (;) でこれらのコマンドを終了することができます。

SQL スクリプトにおいて、セミコロンで CREATE TYPE または CREATE LIBRARY コマンドを終了する場合、セミコロンを削除し、次の行にスラッシュ (/) を置きます。たとえば、次の SQL 文は SQL\*Plus では認識されません。

```
CREATE OR REPLACE TYPE sys.dummy AS OBJECT (data CHAR(1));
CREATE OR REPLACE LIBRARY DEMS_SPACE_ADMIN_LIB TRUSTED AS STATIC;
```

SQL\*Plus で実行する前に、次のようにこれらの文を編集します。

```
CREATE OR REPLACE TYPE sys.aq$_dummy_t AS OBJECT (data CHAR(1))
/
CREATE OR REPLACE LIBRARY DBMS_SPACE_ADMIN_LIB TRUSTED AS STATIC
/
```

## COMMIT コマンド

SQL\*Plus では COMMIT コマンドはセミコロン (;) またはスラッシュ (/) で終了する必要がありますが、**Server Manager** では COMMIT コマンドに終了記号は必要ありません。このため、終了記号なしで SQL スクリプトの COMMIT コマンドを使用する場合、これらのスクリプトを編集して終了記号を組み込みます。

たとえば、SQL スクリプトに次の COMMIT コマンドがあるとします。

```
commit
```

次の例のいずれかのように、コマンドに終了記号を組み込みます。

```
commit;
```

```
commit
/
```

---

# 移行ユーティリティによる Oracle7 データベースのアップグレード

この付録では、移行ユーティリティを使用して Oracle7 データベースを Oracle9i の新しいリリースへ手動でアップグレードする方法について説明します。この付録では、次の項目について説明します。

- 移行ユーティリティの概要
- 移行ユーティリティ使用時のシステムの考慮点と要件
- アップグレードする Oracle7 データベースの準備
- 移行ユーティリティ・コマンドライン・オプションの検討
- 移行ユーティリティの実行
- 移行ユーティリティ・メッセージ
- 移行ユーティリティ・エラーのトラブルシューティング
- Oracle7 のアップグレードの中止
- 物理 ROWID の移行について
- リリース 8.0 での初期化パラメータおよびデータ・ディクショナリの変更

## 移行ユーティリティの概要

移行ユーティリティを使用すると、Oracle7 データベースのデータ・ディクショナリまたは構造が Oracle9i 形式に変換されます。データベースをアップグレードするには、まず Oracle9i ソフトウェアをインストールし、Oracle7 データベースで移行ユーティリティを実行します。そして、新しい Oracle9i データベースに対して一連の ALTER DATABASE 文を実行し、u0703040.sql アップグレード・スクリプトを実行します。

これらの処理の結果として、次のような Oracle7 の構造が、Oracle9i で使用できる構造に変換されます。

- データ・ファイル（ファイル・ヘッダーのみ）
- 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

## 移行ユーティリティを使用したアップグレード処理の概要

次の項では、移行ユーティリティを使用したアップグレード処理の概要について説明します。

### Oracle7 環境でのアップグレード処理

- Oracle9i 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、Oracle7 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 Oracle7 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：** Oracle9i でデータベースをオープンしなくても、Oracle9i の移行ユーティリティを複数回実行でき、Oracle7 データベースへ戻ることもできます。ただし、移行ユーティリティを実行すると、Oracle7 データベースのカatalog・ビューが自動的に削除されます（D-32 ページの「[Oracle7 のアップグレードの中止](#)」を参照）。

---

---

## Oracle9i 環境でのアップグレード処理

- 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 Oracle9i 形式に変換し、Oracle9i データベースをマウントします。

オフライン・データ・ファイルおよび読み専用表領域のファイル・ヘッダーが、アップグレード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、読み専用表領域のファイル・ヘッダーは、アップグレード後に読み / 書き込みになったときに変換されます。ただし、必ずしも読み / 書き込みになるとはかぎりません。

- ALTER DATABASE OPEN RESETLOGS MIGRATE 文を実行します。この文は、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 Oracle9i 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 Oracle9i 形式に変換します。

Oracle9i 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 Oracle9i データベース形式にすぐには変換されません。そのかわり、表領域が Oracle9i で最初にオンラインになるときに変換されます。

- u0703040.sql アップグレード・スクリプトを実行します。このスクリプトは、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 および catproc.sql を実行します。

## 移行ユーティリティ使用時のシステムの考慮点と要件

次の項では、移行ユーティリティを使用する際のシステムの考慮点と要件について説明します。これらの要件は、3-9 ページの「[システムの考慮点および要件](#)」で説明する一般的なアップグレード要件を補足するものです。

### 領域要件

Oracle9i の実行可能ファイルは、Oracle7 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。これによって、アップグレード処理中にディスク領域がなくなることがあります。すでに Oracle7 がインストールされているコンピュータ・システムに Oracle9i をインストールする場合、両方のデータベースに対して十分なハードディスク領域および RAM があることを確認してください。Oracle9i 用のシステム要件と Oracle7 用のシステム要件の両方を考慮して、全体のシステム要件を決定する必要があります。

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 Oracle7 のデータ・ディクショナリと同時に新しい Oracle9i のデータ・ディクショナリを保持するための SYSTEM 表領域のみです。

Oracle データ・ディクショナリを保持するため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 Oracle9i のデータ・ディクショナリは、Oracle7 のデータ・ディクショナリの 2 倍の領域が必要です。必要に応じて、SYSTEM 表領域にディスク領域を追加してください。

また、u0703040.sql アップグレード・スクリプトなどのスクリプトを実行するには、SYSTEM 表領域またはロールバック・セグメントにより多くの領域が必要になることがあります。領域が不足していると、スクリプトの実行時に「unable to extend」という警告が表示されます。スクリプトの実行に必要な領域は、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。変換スクリプトの実行時に「unable to extend」という警告が表示された場合、SYSTEM 表領域またはロールバック・セグメントを拡張した後、スクリプトを再実行します。

**参照：** システム要件の詳細は、ご使用のオペレーティング・システム固有のインストール・ガイドを参照してください。

## ブロック・サイズの考慮点

Oracle7 データベースおよびアップグレードした Oracle9i データベース内の DB\_BLOCK\_SIZE 初期化パラメータの値は同じである必要があります。Oracle9i 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、オペレーティング・システムの物理ブロック・サイズの整数倍にします。ただし、2KB の倍数、特に累乗（2KB、4KB、8KB、16KB）が最も操作に適しています。

Oracle9i 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。

- Oracle7 の設定と一致している。
- 2048 バイト（2KB）以上である。Oracle7 のブロック・サイズが 2KB より小さい場合は、移行ユーティリティによってエラー・メッセージが表示される。
- オペレーティング・システムの物理ブロック・サイズの整数倍である。できれば 2KB の倍数にする。

## SQL\*Net に関する考慮点

データベースを Oracle9i の新しいリリースへアップグレードする際に考慮する必要がある SQL\*Net 関連の問題が数多くあります。Oracle Net Services へ移行するかどうかの決定もこの問題の 1 つです。

**参照：** これらの問題および SQL\*Net から Oracle Net Services への移行の詳細は、付録 B「Oracle Net Services のアップグレードに関する考慮点」を参照してください。



## レプリケーション環境に関する考慮点

Oracle7 のレプリケーション環境は、Oracle9i へアップグレードできます。Oracle7 のサイトは、レプリケーション環境内で Oracle8、Oracle8i および Oracle9i のサイトと共存し、正常に実行できます。ただし、各システムに実装されている様々なレプリケーション機能に適合するように、特に注意してください。

**参照：** レプリケーション機能によるシステムのアップグレードの詳細は、付録 E「レプリケーション環境のデータベース移行および互換性」を参照してください。

## ConText から Oracle Text への移行に関する考慮点

ConText から Oracle Text への移行の詳細は、『Oracle Text アプリケーション開発者ガイド』を参照してください。

## 分散データベースの考慮点

分散データベース構成を Oracle7 からアップグレードする場合は、データベースをアップグレードする前に、DBA\_2PC\_PENDING データ・ディクショナリ・ビューでペンディング・トランザクションがないことを確認してください。これは、アップグレード後に ALTER DATABASE RESETLOGS 文を使用してデータベースをオープンしたときに、トランザクションがペンディングになっているとエラーが発生するためです。

ペンディング・トランザクションが存在する場合は、SQL コマンドの COMMIT FORCE または ROLLBACK FORCE を使用して移行の前に解決します。

## アップグレードする Oracle7 データベースの準備

Oracle7 データベースを Oracle9i の新しいリリースへアップグレードする前に、追加の準備処理を実行する必要があります。次の手順を実行します。

1. アップグレードする Oracle7 データベースの Oracle ホーム・ディレクトリの所有者として、システムにログインします。
2. Server Manager を起動します。
3. SYSDBA 権限を持つユーザーとして、データベース・インスタンスに接続します。
4. プロシージャ・オプションがインストールされていない場合、Oracle7 のインストール・メディアを使用してインストールします。詳細は、ご使用のオペレーティング・システム固有の Oracle マニュアルを参照してください。

プロシージャ・オプションがインストールされているかどうかを判断できない場合は、Server Manager を起動して確認できます。

次に、Server Manager を起動したときに表示されるメッセージの例を示します。

```
Oracle Server Manager Release 2.3.3.0.0 - Production
```

```
Copyright (c) Oracle Corporation 1994, 1995. All rights reserved.
```

```
Oracle7 Server Release 7.3.4.0.0 - Production
```

```
With the distributed, replication, parallel query, Parallel Server
and Spatial Data options
```

```
PL/SQL Release 2.3.4.0.0 - Production
```

表示されるメッセージは、インストールしたオプションおよびリリース番号によって少し異なる場合があります。前述の列の最終行にあるように、メッセージ中に「PL/SQL」が表示される場合は、プロシージャ・オプションがインストールされています。「PL/SQL」が表示されない場合は、インストールされていません。

5. すべてのデータ・ファイルおよび表領域がオンラインまたは正常にオフラインされていることを確認します。

リカバリが必要なデータ・ファイルがあるかどうかを判断するには、次の SQL 文を発行します。

```
SELECT * FROM v$recover_file;
```

「0 rows selected」というメッセージが表示されます。これは、すべてのデータ・ファイルがオンラインまたは NORMAL モードでオフラインのいずれかであることを示しています。データ・ファイルが 1 つでも表示される場合は、データベースをアップグレードする前にデータ・ファイルをリストアする必要があります。V\$DATAFILE 動的パフォーマンス・ビューを使用して、データ・ファイル番号に応じたデータ・ファイル名を検索できます。メディア・リカバリを必要とするデータ・ファイルが存在する場合、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。

確実にオフラインにできなかった表領域は、アップグレード前に削除するかオンラインにする必要があります。そうしないと、アップグレード後に Oracle9i で表領域を使用できません。通常、ALTER TABLESPACE OFFLINE IMMEDIATE または ALTER TABLESPACE OFFLINE TEMPORARY 文を使用してオフラインにする表領域には、メディア・リカバリが必要です。

アップグレード後、Oracle9i 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、Oracle7 データベースのファイル形式のまま残ります。オフラインの表領域は、アップグレード後ならいつでもオンラインにでき、ファイル・ヘッダーはそのときに Oracle9i 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リストアを回避するには、SYSTEM と ROLLBACK を除くすべての表領域をオフラインにし、別のアップグレード処理を実行する必要がある場合は、後で SYSTEM と ROLLBACK のデータ・ファイルのみをリストアできます。

6. MIGRATE という名前のユーザーまたはロールが存在しないことを確認します。これは、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ためです。

MIGRATE という名前のユーザーを確認するには、次の SQL 文を発行します。

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

MIGRATE という名前のユーザーが存在しない場合、どの行も選択されません。

MIGRATE という名前のロールを確認するには、次の SQL 文を発行します。

```
SELECT role FROM dba_roles WHERE role = 'MIGRATE';
```

MIGRATE という名前のロールが存在しない場合、どの行も選択されません。

7. SYSTEM ロールバック・セグメントに OPTIMAL 設定がないことを確認します。アップグレード中、OPTIMAL 設定によってエラーが発生する場合があります。

SYSTEM ロールバック・セグメントの OPTIMAL 設定を確認するには、次の SQL 文を発行します。

```
SELECT a.usn, a.name, b.optsize
 FROM v$rollname a, v$rollstat b
 WHERE a.usn = b.usn AND name = 'SYSTEM';
```

次のように表示されます。

```
USN NAME OPTSIZE

 0 SYSTEM
1 row selected.
```

OPTSIZE 列に値がある場合は、NULL に設定するために次の SQL 文を発行します。

```
ALTER ROLLBACK SEGMENT SYSTEM STORAGE (OPTIMAL NULL);
```

アップグレードが完了した後、OPTIMAL をリセットできます。

**参照：** D-26 ページの「[SYSTEM ロールバック・セグメントの OPTIMAL 設定](#)」にあるトラブルシューティング情報を参照してください。

8. ALTER ROLLBACK SEGMENT 文の STORAGE 句にある MAXEXTENTS パラメータを変更して、SYSTEM ロールバック・セグメントの最大エクステンツ数を増やします（オプション）。

次に、ALTER ROLLBACK SEGMENT 文の例を示します。

```
ALTER ROLLBACK SEGMENT system
 STORAGE (NEXT 500K MAXEXTENTS 121);
```

正常にアップグレードするには、SYSTEM ロールバック・セグメントにより多くの領域が必要な場合があります。SYSTEM ロールバック・セグメントに十分な領域がない場合は、移行ユーティリティ実行中にエラーが発生する場合があります。

9. 環境変数 NLS\_LANG が、ご使用のデータベースで使用するキャラクタ・セットに設定されていることを確認します。

キャラクタ・セットを確認するには、次の SQL 文を発行します。

```
SELECT * FROM v$nls_parameters
 WHERE parameter = 'NLS_LANGUAGE'
 OR parameter = 'NLS_TERRITORY'
 OR parameter = 'NLS_CHARACTERSET';
```

前述の間合せによって返された 3 つすべての値を使用して、NLS\_LANG を設定します。たとえば、前述の間合せによって次のように出力されたとします。

| PARAMETER        | VALUE    |
|------------------|----------|
| NLS_LANGUAGE     | AMERICAN |
| NLS_TERRITORY    | AMERICA  |
| NLS_CHARACTERSET | US7ASCII |

この場合、コマンド・プロンプトで NLS\_LANG を次のように設定します。

```
AMERICAN_AMERICA.US7ASCII
```

**参照：** NLS\_LANG の詳細は、『Oracle9i Database グローバリゼーション・サポート・ガイド』を参照してください。

10. 移行ユーティリティを実行するために必要な DBA 権限を持っていることを確認します。

DBA 権限を持っていることを確認するには、DBA\_ROLE\_PRIVS 静的データ・ディクショナリ・ビューを問い合わせます。たとえば、ユーザー SYSTEM として接続している場合、次の SQL 文を入力します。

```
SELECT * FROM dba_role_privs WHERE grantee = 'SYSTEM';
```

DBA がユーザーの GRANTED\_ROLE 列にリストされている場合、DBA 権限を持っています。DBA 権限がない場合、DBA 権限を持つユーザーとして接続します。

11. RESTRICTED SESSION 権限を持つ他の DBA 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。また、通常のユーザーはアップグレード処理中にデータベースへ接続することはできません。

12. SHUTDOWN NORMAL または SHUTDOWN IMMEDIATE コマンドを使用して、Oracle7 データベースを完全に停止します。SHUTDOWN ABORT は使用しないでください。Oracle7 データベースは、正しく停止する必要があります。これによって、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。

SHUTDOWN IMMEDIATE

Oracle Parallel Server を使用している場合は、すべてのインスタンスを停止します。

---

---

**注意：** アップグレードを開始する前に Oracle7 データベースを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す。

---

---

## 移行ユーティリティ・コマンドライン・オプションの検討

アップグレード処理の次の作業は、Oracle9i 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ライン・オプションを検討してください。移行ユーティリティの実行時にその一部を指定する必要がある場合があるためです。移行ユーティリティのコマンドライン・オプションの詳細が、オペレーティング・システム固有の Oracle マニュアルにも記載されている場合があります。

|            |                                                                                                                                                                                                                                                                                   |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CHECK_ONLY | true にすると、移行ユーティリティは、アップグレード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アップグレードの両方を実行します。このコマンドライン・オプションは、NO_SPACE_CHECK と相互に排他的です。                                                                                                                                           |
| DBNAME     | アップグレードするデータベース名（初期化パラメータ・ファイル内の DB_NAME）を指定します。                                                                                                                                                                                                                                  |
| MULTIPLIER | Oracle9i の i_file#_block# 索引の初期サイズを、Oracle7 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 コマンドライン・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、Oracle9i 用に作成される i_file#_block# 索引は、Oracle7 の i_file#_block# 索引より 1.5 倍大きくなります。 |
| NEW_DBNAME | アップグレードしたデータベースに新しい名前を指定します。デフォルト名の「DEFAULT」は使用しないでください。意味のある名前を選択してください。                                                                                                                                                                                                         |

|                |                                                                                                                                                                                                            |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NO_SPACE_CHECK | true にすると、移行ユーティリティはアップグレード前に領域使用率の調査を行いません。false にすると、移行ユーティリティはアップグレード前に領域使用率の調査を行います。このコマンドライン・オプションは、CHECK_ONLY と相互に排他的です。                                                                             |
| PFILE          | <p>初期化パラメータ・ファイルの名前を指定します。PFILE コマンドライン・オプションを指定しない場合、移行ユーティリティはデフォルトの初期化パラメータ・ファイルを使用します。</p> <p><b>注意：</b>UNIX 上では、次の例のように ¥ 記号でエスケープされた二重引用符でパス名を囲む必要があります。</p> <pre>mig PFILE=¥"/tmp/mig/pfile¥"</pre> |
| SPOOL          | <p>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。</p> <p><b>注意：</b>UNIX 上では、次の例のように ¥ 記号でエスケープされた二重引用符でパス名を囲む必要があります。</p> <pre>mig SPOOL=¥"/tmp/mig/spool¥"</pre>                                                              |

## 移行ユーティリティの実行

UNIX オペレーティング・システムで移行ユーティリティを実行するための手順は、Windows プラットフォームで移行ユーティリティを実行するための手順とは異なります。該当する項の手順を実行してください。

- [UNIX オペレーティング・システムでの移行ユーティリティの実行](#)
- [Windows プラットフォームでの移行ユーティリティの実行](#)

## UNIX オペレーティング・システムでの移行ユーティリティの実行

UNIX オペレーティング・システムで移行ユーティリティを実行するには、次の手順を実行します。

1. コマンド・プロンプトで、Oracle9i リリース 2 (9.2) のインストールされている \$ORACLE\_HOME/bin ディレクトリへ移動します。
2. MIGPREP ユーティリティを実行します。

MIGPREP ユーティリティは、必要なファイルを Oracle9i Oracle ホームから Oracle7 Oracle ホームへコピーして、アップグレードのために Oracle7 の環境を整えます。次のように入力して MIGPREP を実行します。

```
migprep new_oracle_home old_oracle_home
```

`new_oracle_home` は、新しい Oracle9i の Oracle ホーム・ディレクトリのフルパスです。`old_oracle_home` は、古い Oracle7 の Oracle ホーム・ディレクトリのフルパスです。

たとえば、新しい Oracle9i の Oracle ホームが `/oracle/product/9.2` で、古い Oracle7 の Oracle ホームが `/oracle/product/7.3` である場合、次のとおり入力します。

```
migprep /oracle/product/9.2 /oracle/product/7.3
```

3. 次の環境変数を、Oracle7 のディレクトリを指すように変更します。

- ORACLE\_HOME
- PATH
- LD\_LIBRARY\_PATH

`LD_LIBRARY_PATH` が、Oracle9i ライブラリ・パスのホーム・ディレクトリも指していることを確認します。

- ORA\_NLS32

---

**注意：** Oracle Parallel Server に対して、すべてのノードでこの手順を実行します。

---

4. 環境変数 `ORA_NLS33` を、Oracle7 環境の次のディレクトリに設定します。

```
$ORACLE_HOME/migrate/nls/admin/data
```

5. SYSTEM 表領域に十分な領域があることを確認します（オプション）。

アップグレードの一般的な問題は、アップグレード中に SYSTEM 表領域の領域を使い果たしてしまうことです。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アップグレード処理を完了しません。アップグレードを成功させるために必要なディスク領域を調べるために、`CHECK_ONLY` コマンドライン・オプションを `true` に設定して移行ユーティリティを実行します。そのためには、システム・プロンプトで次のように入力します。

```
mig CHECK_ONLY = true
```

`CHECK_ONLY` コマンドライン・オプションを指定すると、移行ユーティリティは、アップグレード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調べて、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` コマンドライン・オプションを `true` に設定すると、移行ユーティリティは、Oracle9i のデータ・ディクショナリの作成またはその他のアップグレード処理の実行を行いません。

CHECK\_ONLY コマンドライン・オプションが SYSTEM 表領域に領域を追加する場合、CHECK\_ONLY オプションで示された領域に 25MB を足した領域を追加する必要があります。追加の 25MB は、アップグレード処理で後ほど実行するアップグレード・スクリプトに必要なおおよその領域分です。

6. 次のとおり入力して移行ユーティリティを実行します。

```
mig
```

オペレーティング・システム固有の Oracle マニュアルで他のコマンドが指示されていないかぎり、コマンドは mig です。mig のみ入力して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 つ以上の選択オプションを続けて入力します。

**参照：** コマンドライン・オプションの詳細は、D-9 ページの「[移行ユーティリティ・コマンドライン・オプションの検討](#)」を参照してください。  
オラクル社では、SPOOL オプションを使用することをお勧めします。  
SPOOL オプションを使用すると、アップグレードが完了したときに結果の確認が簡単です。

## Windows プラットフォームでの移行ユーティリティの実行

Windows プラットフォームで移行ユーティリティを実行するには、次の手順を実行します。

1. 新しい Oracle9i の Oracle ホームで、PFILE オプションを含む移行ユーティリティ・コマンドをコマンド・プロンプトで実行することによって移行ユーティリティを実行します。

```
C:¥> mig PFILE=ORACLE7_HOME¥DATABASE¥INIT_PARAM_FILE
```

ORACLE7\_HOME 変数を Oracle7 の Oracle ホーム・ディレクトリへのフルパスで置き換えます。また、INIT\_PARAM\_FILE 変数を、Oracle7 データベース用の初期化パラメータ・ファイルの完全な名前で置き換えます。

たとえば、ORACLE7\_HOME が C:¥ORANT¥ で、INIT\_PARAM\_FILE が INITORCL.ORA の場合は、次のように入力します。

```
C:¥> mig PFILE=C:¥ORANT¥DATABASE¥INITORCL.ORA
```

デフォルト設定のオプションで実行する場合、PFILE オプションのみを入力します。mig を入力した後に 1 つ以上の選択オプションを続けて入力することもできます。

**参照：** コマンドライン・オプションの詳細は、D-9 ページの「[移行ユーティリティ・コマンドライン・オプションの検討](#)」を参照してください。  
オラクル社では、SPOOL オプションを使用することをお勧めします。  
SPOOL オプションを使用すると、アップグレードが完了したときに結果の確認が簡単です。



2. 移行ユーティリティを実行した際に「Oracle7 Password」が表示された場合、Oracle7 データベースへ SYSDBA 権限でログインしたユーザー用のパスワードを入力します。このプロンプトは、DBA\_AUTHORIZATION レジストリ・パラメータが正常に設定されていないか、設定されていない場合に表示されます。

## 移行ユーティリティの実行結果の確認

移行ユーティリティの実行後に結果を確認します。移行ユーティリティで `migrate.bsq` スクリプトを実行すると、情報メッセージが生成され、その処理過程がエコーされます。移行ユーティリティの終了時に ORA- エラーが発生した場合は、そのエラーおよび問題解決のための処置について、D-25 ページの「[移行ユーティリティ・エラーのトラブルシューティング](#)」を参照してください。

移行ユーティリティは、Oracle7 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。アップグレード処理の後半で、この変換ファイルは Oracle9i の新しい制御ファイルを作成するときに `ALTER DATABASE CONVERT` によって使用されます。

変換ファイルの名前と位置は、オペレーティング・システムによって異なります。たとえば、UNIX オペレーティング・システムでは、デフォルト位置は Oracle7 環境の `$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、Oracle7 のインスタンス ID です。Windows プラットフォームでは、デフォルト位置は Oracle9i 環境の `%ORACLE_HOME%\rdbms` であり、このディレクトリ内のデフォルト・ファイル名は `convert.ora` です。

---

**注意：** Oracle9i の移行ユーティリティによって停止された Oracle7 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 Oracle9i へ変換するときに、ディクショナリ内のシステム変更番号 (SCN)、変換ファイルおよびファイル・ヘッダーのすべてに一貫性がある必要があります。移行ユーティリティの実行後に Oracle7 データベースをオープンすると、データベースを Oracle9i へ変換するとき SCN の判別に失敗し、次のエラーが表示されます。

ORA-01211: Oracle7 データ・ファイルは Oracle8 に移行していません。

したがって、Oracle7 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する必要があります。

---

## Oracle7 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、Oracle7 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、次の目的で使用します。

- Oracle9i で ALTER DATABASE CONVERT 文を実行した後、Oracle7 データベースに戻る場合は、バックアップをリストアして Oracle7 データベースを起動し、D-32 ページの「Oracle7 のアップグレードの中止」の手順を実行します。
- Oracle9i をリカバリするための最初の Oracle9i バックアップとして使用します。
- Oracle9i データベースの変換時 (ALTER DATABASE CONVERT または ALTER DATABASE OPEN RESETLOGS MIGRATE) にエラーが発生した場合は、このバックアップをリストアし、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リストアする場合は、移行ユーティリティを再実行する必要があります。

**参照：** Oracle7 データベースでのバックアップの実行およびリカバリ操作の詳細は、『Oracle7 Server 管理者ガイド』を参照してください。

さらに、Oracle7 のホーム・ディレクトリを含む、Oracle7 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 (Oracle7 データベースのデータ・ファイルがなくなるか読み込めない場合に備えて)。ただし、これらのファイルには未処理の REDO 情報は含まれません。
- パラメータ・ファイル
- 変換ファイル
- Oracle7 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リストアできるスクリプト

## 移行ユーティリティ・メッセージ

移行ユーティリティは、アップグレード処理中にエラー・メッセージおよび情報メッセージを戻すことがあります。この項では、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、考えられる原因と処置を示します。情報メッセージも示していますが、処置は必要ありません。

Database Upgrade Assistant を使用している場合、移行ユーティリティのメッセージは、ログ・ファイルに記録されます。ログ・ファイルへのアクセス方法については、Database Upgrade Assistant のオンライン・ヘルプを参照してください。Database Upgrade Assistant を使用している場合、推奨処置に移行ユーティリティの再実行を含むメッセージが含まれていても、Database Upgrade Assistant を再実行してください。

次に、メッセージ、およびその原因と処置を示します (*string* には文字列が入ります)。

### 移行中に DBA のファイル番号ビットは削減できません

**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削減しようとした。

**処置:** オラクル社カスタマ・サポート・センター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。Oracle7 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:** Oracle7 の制御ファイルが壊れていないかを確認し、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 カタログ領域要件のみを見積る (デフォルト =FALSE)

**原因:** これは、CHECK\_ONLY コマンドライン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、相互に排他的なオプションです

**原因:** これら 2 つの相互に排他的なコマンドライン・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 つのみを使用して、移行ユーティリティを再実行してください。

### クライアント側 *nls\_characterset* がサーバー側 *nls\_characterset* と一致しません - NLS\_LANG 環境変数が設定されていることをチェックしてください

**原因:** NLS\_LANG キャラクタ・セットが、PROPS\$ のキャラクタ・セットと一致しません。

**処置:** PROPS\$ のデータベース・キャラクタ・セットおよび環境変数 NLS\_LANG が一致していることを確認してください。

**コマンドラインの引数は TRUE または FALSE にする必要があります (string)**

**原因:** コマンドライン引数に true または false 以外の値が入力されました。

**処置:** コマンドライン引数の構文を確認し、文を修正して操作を再試行してください。

**コマンドラインの引数は <keyword>=<value> の形式で指定する必要があります (string)**

**原因:** コマンドライン引数の指定方法が誤っています。

**処置:** コマンドライン引数の構文を確認し、文を修正して操作を再試行してください。

**コマンドラインの引数:**

**原因:** この情報メッセージは、コマンドライン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、migrate.bsq、Oracle9i ソフトウェアのバージョンの互換性、および migrate.bsq が破損していないこと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。

**コマンドが CMD (ARG1,ARG2, ...) の形式になっていません**

**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、migrate.bsq、Oracle9i ソフトウェアのバージョンの互換性、および migrate.bsq が破損していないこと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。

**COPYLONG コマンドは COPYLONG (U1,T1,C1,U2,T2,C2,K1<K2>) の形式にする必要があります**

**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、migrate.bsq、Oracle9i ソフトウェアのバージョンの互換性、および migrate.bsq が破損していないこと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。

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

**原因:** SYSTEM 表領域に連続した領域が十分ありません。

**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 表領域に連続した領域が十分ありません。

**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が見つかりません。**

**原因：**SYSTEM 表領域に連続した領域が十分ありません。

**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が見つかりません。**

**原因：**SYSTEM 表領域に連続した領域が十分ありません。

**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 *string* を変換できません。**

**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: *string* -- データベースはバージョン 7.1 以上である必要があります**

**原因：**現行のデータベースが、リリース 7.1 より前のリリースです。

**処置：**現行のデータベースを、ご使用のオペレーティング・システムで移行ユーティリティがサポートするリリースに移行してください。その後、移行ユーティリティを再実行してください。ご使用のオペレーティング・システムでの移行ユーティリティがサポートしているリリースについては、オペレーティング・システム固有の Oracle マニュアルを参照してください。

**列 *string* に対するデータ型は LONG 型にする必要があります**

**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、migrate.bsq、Oracle9i ソフトウェアのバージョンの互換性、および migrate.bsq が破損していないこと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。

**不整合な状態でデータ・ファイルが見つかりました。(内部エラー) --*string***

**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・サポート・センターに連絡してください。

**データ・ファイルはオフラインですが表領域がオンラインです - メディア・リカバリを適用してデータ・ファイルを移行前にオンラインにしてください -- *string***

**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、移行ユーティリティを実行できません。

**処置:** 移行ユーティリティを再実行する前に、メディア・リカバリを行ってデータ・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)**

**原因:** これは、DBNAME コマンドライン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。-*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、migrate.bsq、Oracle9i ソフトウェアのバージョンの互換性、および migrate.bsq が破損していないこと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。

**system.def\$\_call, def\$\_calldest または def\$\_error にエン트리があります - 移行前にすべての遅延トランザクションを送信してください**

**原因:** DEF\$\_CALL、DEF\$\_CALLDEST または DEF\$\_ERROR に、エントリが存在します。

**処置:** エントリが DEF\$\_CALL の中にある場合は、DEF\$\_CALL が空になるまですべての遅延トランザクションを送信してください。エントリが DEF\$\_ERROR の中にある場合は、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 のコール中にエラーが発生しました。**

**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、migrate.bsq、Oracle9i ソフトウェアのバージョンの互換性、および migrate.bsq が破損していないこと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 '*string*' の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。

**string の見積り領域要件は 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の見積り領域要件を表示します。

**処置:** ユーザーの処置は不要です。

**ファイル number は 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、指定したファイルが大きすぎます。

**処置:** オラクル社カスタマ・サポート・センター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を確認し、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を確認し、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 文が見つかりました。**

**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、migrate.bsq、Oracle9i ソフトウェアのバージョンの互換性、および migrate.bsq が破損していないこと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。

**SYSTEM 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: number**

**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。

**書込みが不完全です。**

**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。

**新規ディクショナリ用の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 の値が指定されました。**

**原因:** コマンドラインに指定した NLS\_NCHAR 値が無効です。

**処置:** コマンドラインに指定した 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。

**移行を進められません - データベースのブロック・サイズ number が 2K (Oracle8 の最小ブロック・サイズ) 未満です**

**原因:** 既存のデータベースのブロック・サイズが 2KB に達していません。

**処置:** Oracle7 データベースのブロック・サイズが最小でも 2KB あることを確認してください。Oracle7 データベースの再構築を検討することもできます。その後、移行ユーティリティを再実行してください。

**データ・ファイルがオンラインで表領域がオフラインのため移行を進められません -- string**

**原因:** 表領域はオフラインですが、表領域内のデータ・ファイルがオンラインです。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。

**アクティブ・トランザクションまたは未処理の取消しがあるオフライン表領域のために移行を進められません**

**原因:** 移行ユーティリティがデータベースをアップグレードしようとしたときに、1 つ以上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:** D-6 ページの手順 5 に戻り、オフラインのすべての表領域が完全にオフラインであることを確認してください。その後、移行ユーティリティを再実行してください。



**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 Oracle7 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\$ クラスタ索引サイズの増加ファクタ (デフォルト =15)**

**原因:** これは、MULTIPLIER コマンドライン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 の値は、最小で 2 です。**

**原因:** コマンドラインの中で Oracle9i i\_file#\_block# の初期サイズを指定する MULTIPLIER 値が、2 未満です。

**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 string が長すぎます - 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 文字を超えています。

**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 - 新規データベース名 (最大 8 文字)**

**原因:** この情報メッセージは、NEW\_DBNAME コマンドライン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**

**原因:** この情報メッセージは、NLS\_NCHAR コマンドライン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 - 領域チェックを実行しない (デフォルト =FALSE)**

**原因:** これは、NO\_SPACE\_CHECK コマンドライン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を確認してください。

**変換ファイルの作成中に処理された string の番号が正しくありません。**

**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・サポート・センター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 Oracle7 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。

**環境変数 ORA\_NLS33 が未設定か、正しく設定されていません。**

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

**処置:** 環境変数 ORA\_NLS33 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 (『Oracle9i データベース・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を削除して、初期化パラメータ・ファイルのサイズを小さくしてください。その後、移行ユーティリティを再実行してください。

**パラメータ・ファイルが *number* バイトを超えています。**

**原因:** Oracle7 データベース用の初期化パラメータ・ファイルが所定の最大サイズを超えています。

**処置:** 可能的場合は、廃止されたパラメータを削除して初期化パラメータ・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・サポート・センターに連絡してください。

**PFILE - 代替 init.ora を使用する。**

**原因:** これは、PFILE コマンドライン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 *string* でシーク・エラーが発生しました。**

**原因:** 指定したファイルの読み込み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を確認してください。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。

***number* バイト要求されましたが *number* バイトのみ読み込まれました**

**原因:** 制御ファイルの読み込みに問題があります。

**処置:** 制御ファイルが壊れていないかを確認し、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停止しています ...**

**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アクションを行ってください。

**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 Oracle7 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 - ファイルへの出力をスプールする。**

**原因:** これは、SPOOL コマンドライン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 Oracle7 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。最大長は *number* です。**

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

**処置:** コマンドライン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 NORMAL モードでオフラインされていません。表領域を移行前にオンラインまたは NORMAL モードでオフラインにするか、削除してください -- *string***

**原因:** 表領域が IMMEDIATE または TEMPORARY を使用し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 モードでオフラインにするか、または削除してください。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。(最大 *number*)**

**原因:** コマンドライン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、コマンドラインに適切な数のオプションを指定してください。

**LONG をコピーするためにバッファ領域を割当てできません。**

**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:** 使用できる十分なコンピュータ・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 *string* をオープンできません。**

**原因:** 内部エラーが発生しました。または、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み込みの権限が 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 *string* を読み込めません。**

**原因:** 内部エラーが発生しました。または、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み込みの権限が 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 *string* 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: *number***

**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 表領域に必要な空き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、追加の空き領域がどの程度あるかを確認してください。

## 移行ユーティリティ・エラーのトラブルシューティング

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、または省略されたことが考えられます。

- 誤った順序でアップグレード手順を実行する
- アップグレードの前提条件を満たしていない
- 変換の例外を検出する

## 移行ユーティリティの使用に関する問題

移行ユーティリティを実行する際に発生する可能性がある一般的なアップグレード問題の原因は、ユーザーのデータベース・システムの構成にあります。移行ユーティリティが Oracle7 データベースのアップグレードに必要な処理を実行中に、ユーザーの Oracle ソフトウェアによってエラーが発生します。通常、エラーが発生した場合には、移行ユーティリティは動作を中断し、1 つ以上のエラー・メッセージを表示します。

移行ユーティリティの実行中に次のような問題が発生した場合には、提示された処置を実行し、移行ユーティリティを再実行してください。

### SYSTEM 表領域の領域不足

この問題は、次のようなエラー・メッセージを戻すことがあります。

ORA-00604: 再帰 SQL レベル 1 でエラーが発生しました。

ORA-01653: 表 SYS を拡張できません (473 分、表領域 SYSTEM)。

アップグレード処理を正常に行うため、SYSTEM 表領域に新規データ・ファイルを追加し、新規データ・ファイルに十分な領域を割り当てます。

アップグレード処理中に、一時表領域に領域がなくなることもあります。アップグレード処理を正常に行うために、一時表領域に新規データ・ファイルを追加し、新規データ・ファイルに十分な領域を割り当ててください。

**参照：** SYSTEM 表領域の領域要件および使用可能な領域を増やすための新規データ・ファイルの追加については、D-3 ページの「[領域要件](#)」および D-11 ページの手順 5 を参照してください。

## 不適切な AUDIT\_TRAIL パラメータ設定

この問題は、次のようなエラー・メッセージを戻すことがあります (*string* には文字列が入ります)。

ORA-00604: 再帰 SQL レベル *string* でエラーが発生しました。

ORA-01552: SYSTEM 表領域でない表領域 '*string*' にシステム・ロールバック・セグメントは使用できません

ORA-02002: 監査証跡への書き込み中にエラーが発生しました。

これらのエラーは、次の場合に発生します。

- 初期化パラメータ AUDIT\_TRAIL が、DB または TRUE に設定されている場合
- SYS.AUD\$ 表が SYSTEM 以外の表領域にある場合

この問題を解決するには、次の手順を実行します。

1. データベースを実行している場合は、停止します。
2. 次の方法で、初期化パラメータ・ファイルの AUDIT\_TRAIL 初期化パラメータを設定します。

```
AUDIT_TRAIL = NONE
```

3. 移行ユーティリティを再実行します。

## SYSTEM ロールバック・セグメントの OPTIMAL 設定

この問題は、次のようなエラー・メッセージを戻すことがあります。

ORA-01562: ロールバック・セグメント番号 0 の拡張に失敗しました

ORA-01628: 最大エクステント数 (n) に達しました (ロールバック・セグメント SYSTEM)

このメッセージは、SYSTEM ロールバック・セグメントが小さすぎてアップグレードを完了できないことを示しています。アップグレードが正常に完了するために十分な大きさの SYSTEM ロールバック・セグメントがあることを確認してください。

移行ユーティリティおよび Database Upgrade Assistant は、SYSTEM ロールバック・セグメント以外をオフラインにし、SYSTEM ロールバック・セグメントの現在割り当てられたエクステント数に MAXEXTENTS を設定し、サイズを固定します。この処置には、移行ユーティリティまたは Database Upgrade Assistant のスペース・マネジメント表の処理中に、エクステント割当てなどの領域操作を防ぐ役割があります。

SYSTEM ロールバック・セグメントに OPTIMAL 設定がある場合、エクステントは、そのデータがアクティブ・トランザクションに不要の場合、動的に割り当てられません。動的割当ては、SYSTEM ロールバック・セグメントが固定された場合、現在割り当てられたエクステントを小さくすることがあります。したがって、アップグレード処理中にスペース・マネジメント表のあるトランザクションを処理するには、SYSTEM ロールバック・セグメントは小さすぎる場合があります。

この問題を解決するには、次の設定を変更します。

1. ロールバック・セグメントの OPTIMAL 設定をオフにします。
2. SYSTEM ロールバック・セグメントの NEXT EXTENT を 2 倍にします。
3. MULTIPLIER 値を 2 倍にします。
4. システム表領域に領域を追加して、ロールバック・セグメントの処理に十分な領域 (50MB 以上) があることを確認します。

**参照：** 移行ユーティリティを使用している場合、OPTIMAL 設定および必要時のリセットの確認方法については、D-7 ページの手順 7 を参照してください。

## 小さい MULTIPLIER オプション設定

この問題は、次のようなエラー・メッセージを戻すことがあります (*string* には文字列が入ります)。

ORA-01632: 最大エクステント数 (*string*) に達しました (索引 *string.string*)

移行ユーティリティは、15 という低いデフォルト値を MULTIPLIER オプションに使用します。問題を回避するには、MULTIPLIER オプションの値を大きくします。

移行ユーティリティをコマンドラインから実行する場合には、次のとおり入力して MULTIPLIER オプションを 30 まで大きくします。

```
mig MULTIPLIER=30
```

ただし、Database Upgrade Assistant を使用してバックグラウンドで移行ユーティリティを実行する場合には、アップグレードするデータベースのバックアップをリストアし、Database Upgrade Assistant を再実行してください。Database Upgrade Assistant のカスタム移行オプションを選択します。MULTIPLIER 値を入力するプロンプトが表示されたら、デフォルト値である 15 以上の値を入力します。

**参照：** MULTIPLIER オプションの詳細は、D-9 ページの「[移行ユーティリティ・コマンドライン・オプションの検討](#)」を参照してください。

## ALTER DATABASE CONVERT 文における問題

移行ユーティリティの実行後、アップグレード処理中に ALTER DATABASE CONVERT 文を発行すると、この項で説明する問題のいずれかが発生する可能性があります。通常、変換は中断し、1 つまたは複数のエラー・メッセージが表示されます。ALTER DATABASE CONVERT 文の発行時に、次のいずれかの問題が発生した場合、提示された処置を実行して問題を解決してください。

## Oracle7 制御ファイルの存在

この問題は、次のエラー・メッセージを戻すことがあります (*string* には文字列が入ります)。

ORA-00200: 制御ファイル *name* を作成できませんでした。

ORA-00202: 制御ファイル: '*string*'

ORA-27038: skgfrcre: ファイルが存在しています。

古い Oracle7 制御ファイルは、ALTER DATABASE CONVERT 文を発行する前に、名前を変更または削除する必要があります。

**参照:** 3-14 ページの手順 b を参照してください。

## NOMOUNT 以外のモードで起動するデータベース

この問題は、次のエラー・メッセージを戻すことがあります (*string* には文字列が入ります)。

ORA-00227: 制御ファイルで破損ブロックが検出されました: (ブロック *string*、番号 *string*)

ORA-00202: 制御ファイル: '*string*'

古い Oracle7 制御ファイルは、ALTER DATABASE CONVERT 文を発行する前に、名前を変更または削除する必要があります。また、ALTER DATABASE CONVERT 文を発行する場合、データベースを NOMOUNT モードで起動してください。このエラーは、データベースが NOMOUNT モード以外のモードで起動されていることを示しています。

**参照:** 3-14 ページの手順 b および 3-21 ページの手順 12 を参照してください。

## 変換ファイルが見付からない

この問題は、次のエラー・メッセージを戻すことがあります (*string* には文字列が入ります)。

ORA-00404: 変換ファイルが見つかりません: '*string*'

ORA-27037: ファイル・ステータスを取得できません。

移行ユーティリティが生成した変換ファイル (UNIX の convsid.dbf および Windows プラットフォームの convert.ora) が、所定の位置にありませんでした。所定の位置とは、UNIX においては Oracle9i 環境の \$ORACLE\_HOME/dbs ディレクトリ、Windows プラットフォームにおいては Oracle9i 環境の %ORACLE\_HOME%\rdbms ディレクトリのことです。ALTER DATABASE CONVERT 文を発行する前に、変換ファイルをこの所定の位置に移動させる必要があります。

**参照:** 3-16 ページの手順 3 を参照してください。



## REMOTE\_LOGIN\_PASSWORDFILE から EXCLUSIVE への初期化パラメータの設定

この問題は、次のエラー・メッセージを戻すことがあります。

ORA-00600: 内部エラー・コード、引数: [kzsrsdn: 1], [32]

これらのエラーは、次の場合に発生します。

- データベースがパスワード・ファイルを使用し、パスワード・ファイルが正しいディレクトリに移動されていない場合。正しいディレクトリとは、UNIX においては Oracle9i 環境の \$ORACLE\_HOME/dbs、Windows プラットフォームにおいては Oracle9i 環境の %ORACLE\_HOME%\database のことです。
- REMOTE\_LOGIN\_PASSWORDFILE 初期化パラメータが、初期化パラメータ・ファイルの EXCLUSIVE に設定されている場合。

アップグレードを続行するため、次の手順を実行します。

1. データベースを停止します。
2. 初期化パラメータ・ファイルの REMOTE\_LOGIN\_PASSWORDFILE を、NONE に設定します。

```
REMOTE_LOGIN_PASSWORDFILE = NONE
```

3. 次の SQL 文を入力して、データベースのマウントを起動します。

```
SQL> STARTUP MOUNT
```

PFILE オプションを使用して、初期化パラメータ・ファイルの場所を指定する必要がある場合があります。

4. ALTER DATABASE OPEN RESETLOGS MIGRATE 文を発行します。

```
SQL> ALTER DATABASE OPEN RESETLOGS MIGRATE;
```

5. 3-22 ページの手順 13 から、アップグレード処理を続行します。

既存のパスワードはすでに有効ではないため、使用することはできません。Oracle9i でパスワード・ファイルを使用する場合は、パスワード・ファイルを再作成し、それをユーザーの間で定着させます。必ず REMOTE\_LOGIN\_PASSWORDFILE を正しく設定してください。

## データベース名の不一致

この問題は、次のエラー・メッセージを戻すことがあります (*string* には文字列が入ります)。

ORA-01103: 制御ファイルのデータベース名 '*string*' は '*string*' ではありません

データベース名に不一致があります。次のような1つまたは複数の不一致があります。

- 初期化パラメータ・ファイルの DB\_NAME 初期化パラメータに指定したデータベース名が、convsid.dbf ファイル名のデータベース名と一致しません。
- 環境変数 ORACLE\_SID で設定した Oracle9i インスタンス ID が、convsid.dbf のファイル名のデータベース名と一致しません。

---

**注意：** この問題は、UNIX オペレーティング・システムでのみ発生します。Windows プラットフォームでは発生しません

---

問題を解決するには、正しいデータベース名が次の各項目で指定されているか確認してください。

- 環境変数 ORACLE\_SID
- 初期化パラメータ・ファイルの DB\_NAME 初期化パラメータ
- convsid.dbf ファイル名の *sid* の部分

たとえば、初期化パラメータ・ファイルの環境変数 ORACLE\_SID および DB\_NAME パラメータの両方を DB1 に設定すると、convsid.dbf ファイル名は次のようになります。

convDB1.dbf

## ALTER DATABASE CONVERT 文の再実行

この問題は、次のエラー・メッセージを戻すことがあります (*string* には文字列が入ります)。

ORA-01122: データベース・ファイル *string* の照合検査でエラーが発生しました。

ORA-01110: データ・ファイル *string*: '*string*'

ORA-01202: このファイルのインカネーションが無効です - 作成日時が一致しません

これらのエラーは、以前に ALTER DATABASE CONVERT 文を発行したが、失敗したことを意味します。これらのエラーが発生した場合は、ALTER DATABASE OPEN RESETLOGS MIGRATE 文を発行することによって、アップグレード処理の次の手順に進むことができます。ただし、問題が発生した場合、アップグレード処理を始める前に作成したバックアップをリストアし、それを使用してアップグレードを初めから再開します。第3章の最初から開始します。ただし、第2章およびこの付録で説明するアップグレード前の処置を必ず実行してください。

## データ・ファイルのバージョン整合性の問題

この問題は、次のエラー・メッセージを戻すことがあります (*string* には文字列が入ります)。

ORA-01122: データベース・ファイル *string* の照合検査でエラーが発生しました。

ORA-01110: データ・ファイル *string*: '*string*'

ORA-01211: Oracle7 データ・ファイルは Oracle8 に移行していません

移行ユーティリティは、Oracle7 の環境のデータベースに最後にアクセスするユーティリティである必要があります。エラー・メッセージの中で指摘されたデータ・ファイルは、移行ユーティリティを実行する前にバックアップしたもの、または移行ユーティリティを実行した後に Oracle7 でオープンされたデータベースのいずれかです。移行ユーティリティの実行の際のデータ・ファイルのみ、Oracle9i を使用してアクセスできます。

データ・ファイルのバージョンの整合性を保証するには、データベースを Oracle9i へ変換するときに、データ・ディクショナリ内の SCN、変換ファイルおよびファイル・ヘッダーのすべてが一貫している必要があります。移行ユーティリティを実行した後で Oracle7 でデータベースをオープンすると、SCN のチェックが、ALTER DATABASE CONVERT 文を発行するときに失敗します。

この問題を回避するには、次の手順を実行します。

1. データベースを停止します。
2. ALTER DATABASE CONVERT 文を使用して作成された制御ファイルを別の名前に変更します。
3. STARTUP NOMOUNT 文を発行する直前に保存した Oracle7 の制御ファイルのコピーをリストアします。

保存した Oracle7 の制御ファイルがない場合、移行処理の起動前に作成したバックアップをリストアします。

4. 移行処理を初めから開始し、移行ユーティリティの実行後、Oracle7 環境でデータベースを起動しないでください。第 3 章の初めから開始します。

## Oracle7 のアップグレードの中止

移行ユーティリティを実行する前に Oracle7 データベースのバックアップを行った場合、アップグレードを中止する最も簡単な方法は、バックアップをリストアすることです。ただし、バックアップを取っていない場合、または移行ユーティリティ実行後にバックアップを行った場合は、アップグレードを中止するためにこの項で説明する手順を実行する必要があります。

Oracle9i の移行ユーティリティは複数回実行でき、後で Oracle7 データベースへ戻ることもできます。ただし、移行ユーティリティを実行すると、Oracle7 データベースのカatalog・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 Oracle7 データベースに戻るには、Oracle7 の catalog.sql スクリプトを実行して、Oracle7 データベースのカatalog・ビューをリストアする必要があります。

---

---

**注意：**すでに ALTER DATABASE CONVERT 文を実行している場合は、次の手順を実行してアップグレードを中止することはできません。この文を実行した後に Oracle7 へ戻るには、[第 8 章「エクスポート / インポートによるデータベースの移行」](#)のダウングレード手順を実行してください。

---

---

アップグレードを中止するには、一般的に Oracle7 環境下で次の手順を実行して Oracle7 データベースをリストアする必要があります。

1. Server Manager を使用して Oracle7 データベースを起動します。
2. MIGRATE ユーザーを削除します。

```
DROP USER MIGRATE CASCADE;
```
3. catalog.sql および catproc.sql を再実行します。

```
@catalog.sql
@catproc.sql
```
4. catsvrmg.sql を実行します。

```
@catsvrmg.sql
```
5. Oracle 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。

```
@catparr.sql
```
6. Oracle Replication がインストールされている場合は、catrep.sql を実行します。

```
@catrep.sql
```

## 物理 ROWID の移行について

Oracle8 では、パーティション化およびグローバル索引を含む Oracle8 以上の新機能を使用可能にする、物理 ROWID 用の新しい内部フォーマットおよび外部フォーマットが導入されました。

**参照：** 詳細は、『Oracle9i アプリケーション開発者ガイド - 基礎編』および『Oracle9i データベース概要』を参照してください。

この項では、次の項目について説明します。

- [アプリケーションのアップグレードおよびデータの移行](#)
- [DBMS\\_ROWID パッケージ](#)
- [スナップショット・リフレッシュ](#)
- [Oracle7 クライアントの互換性の問題](#)
- [ROWID に関する移行および互換性の問題](#)

---

**注意：** この項の新しい ROWID についての説明は、Oracle8 で導入された ROWID 機能に適用されます。また、「ROWID」という単語は「物理 ROWID」のことです。この付録では、UROWID（ユニバーサル ROWID）データ型については説明していません。UROWID データ型に関する互換性の問題については、[第 5 章「互換性および相互運用性」](#)を参照してください。

---

## アプリケーションのアップグレードおよびデータの移行

ROWID は、ROWID データ型およびキャラクタ・タイプの列に格納できます。格納されていた Oracle7 の ROWID は、Oracle9i へアップグレードした後は無効になります。したがって、格納されていた Oracle7 の ROWID は、新しい形式に変換する必要があります。

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、変更または再コンパイルの必要がありません。新しい ROWID がホスト変数についての現行の記憶域要件を満たしているためです。

ROWID の内容を作成または分析するアプリケーションでは、DBMS\_ROWID パッケージを使用して、新しい ROWID の形式および内容を操作する必要があります。このパッケージには、Oracle7 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。

ROWID 値（ROWID データ型または文字形式）を含む列は、値が参照する表を Oracle9i へアップグレードする場合に、一緒に移行する必要があります。そうしないと、格納された値を使用して行を取り出すことができなくなります。一方、アップグレードされた表に格納された ROWID 値が Oracle7 の表を参照している場合、列の移行は必要ありません。

列の移行は、定義の移行とデータの移行の 2 段階で行われます。列の定義は、Oracle9i へのアップグレード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 Oracle9i でオープンされた後でないと実行できません。ユーザーは、異なる表を別々にアップグレードするか、または複数の表を並行して一度にアップグレードすることができます。ただし、アップグレードは、Oracle7 データベースの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ことを避けるためです。

インストールで使用できる既存の ROWID リフレッシュ手順または DBMS\_ROWID 機能を使用して、格納されている ROWID を Oracle7 形式から新しい形式に移行できます。

移行ユーティリティまたは Database Upgrade Assistant を使用したデータの移行は、ユーザー定義列に格納された ROWID にのみ適用されます。システムに格納されたすべての ROWID（索引内の ROWID など）は、アップグレード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。また、Oracle9i へアップグレード中は、索引は制限付き ROWID データ型形式を使用できるため、索引が無効になることはありません。

---

---

**注意：** ROWID を含む列をインポートすると、ROWID の妥当性を再確認する特別な注意が必要になることを示す警告メッセージが表示されます。そのような特別な注意は、インポートするすべての ROWID について必要です。したがって、エクスポート / インポートを使用したデータベースの移行には、ユーザー定義列のみでなく ROWID が含まれるすべての列について注意してください。

---

---

## DBMS\_ROWID パッケージ

DBMS\_ROWID パッケージには、次の機能が含まれます。

- Oracle7 形式と新しい形式の両方での ROWID の作成および解釈
- Oracle7 の ROWID と新しい ROWID 間での変換

格納されている ROWID の移行は、次の項で説明するように変換ファンクションを使用して実行できます。

## ROWID 変換型

変換する ROWID の型を指定する必要があります。ROWID が ROWID データ型のユーザー列に格納されているか、CHAR データ型または VARCHAR2 データ型のユーザー列に格納されているかによって、ROWID 変換ファンクションによる変換の実行方法が異なるためです。

ROWID データ型の列の場合、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;
```

CHAR または VARCHAR2 データ型の列の場合、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 変換ファンクション

次のファンクションは、ROWID の変換を行います。

- **ROWID\_TO\_EXTENDED** は、ROWID を Oracle7（制限付き）形式から新しい（拡張）形式に変換します。
- **ROWID\_TO\_RESTRICTED** は、ROWID を新しい（拡張）形式から Oracle7（制限付き）形式に変換します。
- **ROWID\_VERIFY** は、指定された ROWID を Oracle7 形式から新しい形式に変換できるかどうかを調査します。

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用します。

- **Rowid:** 外部文字フォーマットで変換する ROWID を指定します。
- **Schema Name:** ROWID を拡張形式に変換する行を含む表のスキーマ名を指定します。
- **Table Name:** ROWID を拡張形式に変換する行を含む表の名前を指定します。
- **Conversion Type:** 変換する ROWID の型を指定します。

**参照：** 詳細は、D-35 ページの「[ROWID 変換型](#)」を参照してください。

ROWID\_TO\_EXTENDED は外部文字フォーマットの新しい (拡張) ROWID を戻し、そのパラメータを次の方法で解釈します。

- 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 (NULL) 場合、ROWID\_TO\_EXTENDED は、変換する ROWID によって指定されたページをフェッチしようとしています。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いますが、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ります。フェッチ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ます。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、最終手段としてのみ使用することをお勧めします。変換された値を使用する場合は、正しい表名を知っておく必要があります。
- スキーマ名および表名が指定されている場合 (適切な方法)、ROWID\_TO\_EXTENDED は表に対する SELECT 認可レベルを検証し、この表のデータ・オブジェクト番号を使用して ROWID を拡張形式に変換します。変換された ROWID がこの表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ません。
- ROWID に NULL 値が指定されている場合、プロシージャは表の指定を無視し、NULL 値を戻します。
- ROWID に 0 または <n>0.<m>0.<p>0 などのより一般的な値が指定されている場合、表名は無視され、00000000.0000.0000 という形式の制限付き ROWID が戻されます。
- 新しい ROWID が指定されている場合は、ROWID 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) に対して検証されます。これらの 2 つの数字が一致しない場合は、INVALID ROWID エラーが発生します。一致した場合は、元の ROWID が戻されます。

**ROWID\_VERIFY** ROWID\_VERIFY という ROWID 検証プロシージャが提供されています。ROWID を拡張形式に変換できた場合、このプロシージャは ROWID\_TO\_EXTENDED と同じパラメータを使用して 0 を戻します。そうでない場合は 1 を戻します。

ただし、ユーザーが基礎となる表の SELECT 認可レベル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「object not found」というエラーを戻します。ROWID\_VERIFY は、ROWID\_TO\_EXTENDED プロシージャで移行をする前に、不正 ROWID を識別します。

### 変換プロシージャの例

次に、ROWID の変換プロシージャの例を示します。

**例 1** 表 scott.t に、ROWID データ型形式の列 c があるとしています。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 は別の表を参照することがありますが、表名は同じ行にある他の列の値に基づいて検索されます。たとえば、表 `t` の `tname` 列に、列 `c` の ROWID が参照する表の名前があるとします。

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 ファンクションを CREATE...AS SELECT 文の中で使用することもできます。変換によって ROWID データ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ため、CREATE...AS SELECT を使用した方がよい場合があります。

この場合、UPDATE よりも CREATE...AS SELECT を選択します。

```
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、変換は次の文を使用して実行できます。

```
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

新しい ROWID データ型形式では、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 Oracle9i へアップグレードするときに、すべての ROWID スナップショットが完全リフレッシュを実行する必要があります。

**参照：** レプリケーションの互換性の詳細は、[付録 E「レプリケーション環境のデータベース移行および互換性」](#)を参照してください。

## Oracle7 クライアントの互換性の問題

Oracle7 クライアントは Oracle8 以上のデータベースにアクセスでき、Oracle8 以上のクライアントは Oracle7 データベースにアクセスできます。Oracle7 データベースから Oracle8 以上のデータベースへ戻される ROWID 疑似列および ROWID データ型の列のバイナリ値および文字値は、必ず制限付き形式となります。これは、Oracle7 では拡張形式の ROWID を認識できないためです。

DBMS\_ROWID パッケージは、Oracle7 の ROWID の内容解釈および Oracle7 形式での ROWID の作成に使用できます。

Oracle7 クライアントが Oracle8 以上のデータベースにアクセスすると、ROWID を新しい（拡張）形式で受け取ります。したがって、クライアントは、新しい ROWID の内容を解釈できません。

## ROWID に関する移行および互換性の問題

下位互換用に、ROWID の制限付き形式がサポートされています。大部分の Oracle7 のデータに制限付き ROWID が存在し、拡張 ROWID 形式はパーティション表のグローバル索引に使用されているのみです。新規の表では、常に拡張 ROWID が使用されます。

**参照：**『Oracle9i データベース管理者ガイド』を参照してください。

Oracle7 クライアントから Oracle8 以上のデータベースへのアクセスは可能です。同様に、Oracle8 以上のクライアントも Oracle7 Server へアクセスできます。この場合のクライアントには、サーバーにアクセスする 3GL または 4GL のクライアント・アプリケーションと同様に、データベース・リンクを使用してサーバーにアクセスするリモート・データベースが含まれます。

ROWID\_TO\_EXTENDED ファンクションの詳細は、『Oracle9i PL/SQL パッケージ・プロシージャおよびタイプ・リファレンス』を参照してください。

### Oracle8 以上のクライアントからの Oracle7 データベースへのアクセス

戻される ROWID の値は、常に制限付き ROWID 形式です。また、Oracle7 サーバーに戻される ROWID の値も、常に制限付き ROWID 形式です。

Oracle7 サーバーにアクセスする場合は、次の ROWID 機能を使用できます。

- ROWID の選択、および取得した値の WHERE 句での使用
- WHERE CURRENT OF カーソル操作
- ROWID 型または CHAR 型のユーザー列への ROWID の格納
- 16 進エンコーディングを使用した ROWID の解釈（この方法ではなく、DBMS\_ROWID ファンクションの使用をお勧めします）

## Oracle7 クライアントからの Oracle8 以上のデータベースへのアクセス

Oracle8 以上は、ROWID を拡張形式で戻します。そのため、行えるのは次のことのみです。

- ROWID の選択、および取得した値の WHERE 句での使用
- WHERE CURRENT OF カーソル操作
- CHAR(18) データ型のユーザー列への ROWID の格納

## エクスポート/インポート

ROWID 列（擬似列ではない）のある Oracle8 以上の表は、その表の中に拡張 ROWID 値を持つ行が存在する場合は、Oracle7 クライアントにインポートできません。

# リリース 8.0 での初期化パラメータおよびデータ・ディクショナリの変更

この項では、リリース 8.0 での初期化パラメータおよびデータ・ディクショナリの変更点を示します。

## リリース 8.0 での初期化パラメータの変更

次の項では、リリース 8.0 での初期化パラメータの変更点を示します。

### リリース 8.0 で名前が変更された初期化パラメータ

リリース 8.0 では、[表 D-1](#) に示す初期化パラメータは名前が変更されました。

**表 D-1 リリース 8.0 で名前が変更された初期化パラメータ**

| リリース 8.0 より前の名前               | リリース 8.0 以上の名前            |
|-------------------------------|---------------------------|
| ASYNC_READ                    | DISK_ASYNC_IO             |
| ASYNC_WRITE                   | DISK_ASYNC_IO             |
| CCF_IO_SIZE *                 | DB_FILE_DIRECT_IO_COUNT * |
| 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        |

\* CCF\_IO\_SIZE（バイト）と DB\_FILE\_DIRECT\_IO\_COUNT（データベース・ブロック）の単位は異なります。

### リリース 8.0 で廃止された初期化パラメータ

次の初期化パラメータはリリース 8.0 で廃止されたものです。

|                             |                             |
|-----------------------------|-----------------------------|
| CHECKPOINT_PROCESS          | FAST_CACHE_FLUSH            |
| GC_DB_LOCKS                 | GC_FREELIST_GROUPS          |
| GC_ROLLBACK_SEGMENTS        | GC_SAVE_ROLLBACK_LOCKS      |
| GC_SEGMENTS                 | GC_TABLESPACES              |
| INIT_SQL_FILES              | IO_TIMEOUT                  |
| IPQ_ADDRESS                 | IPQ_NET                     |
| LM_DOMAINS                  | LM_NON_FAULT_TOLERANT       |
| MLS_LABEL_FORMAT            | OPTIMIZER_PARALLEL_PASS     |
| PARALLEL_DEFAULT_MAX_SCANS  | PARALLEL_DEFAULT_SCAN_SIZE  |
| POST_WAIT_DEVICE            | SEQUENCE_CACHE_HASH_BUCKETS |
| UNLIMITED_ROLLBACK_SEGMENTS | USE_IPQ                     |
| USE_POST_WAIT_DRIVER        | USE_READV                   |
| USE_SIGIO                   | V733_PLANS_ENABLED          |

---

---

**注意：** 1 つ以上の廃止された初期化パラメータを使用してリリース 2 (9.2) のデータベースを起動するとエラーになり、データベースを起動できません。

---

---

### リリース 8.0 での静的データ・ディクショナリ・ビューの変更

次の項では、リリース 8.0 での静的データ・ディクショナリ・ビューの変更点を示します。

#### リリース 8.0 で廃止された静的データ・ディクショナリ・ビュー

次の静的データ・ディクショナリ・ビューはリリース 8.0 で廃止されたものです。

|                |                 |
|----------------|-----------------|
| ALL_HISTOGRAMS | DBA_HISTOGRAMS  |
| DEFCALL        | USER_HISTOGRAMS |

---

# レプリケーション環境のデータベース移行 および互換性

---

この付録では、Oracle7 から Oracle9i へのレプリケーション環境のアップグレードに必要な手順について説明します。この付録では、次の項目について説明します。

- レプリケーションのデータベース移行の概要
- すべてのサイトの同時アップグレード
- 段階的アップグレード
- 主キー・マテリアライズド・ビューへのアップグレード
- Oracle の上位リリースへのアップグレードを必要とする機能
- 廃止されたプロシージャ

---

**注意：** この付録では、Oracle7 から Oracle9i へのアップグレードの必要性について説明します。Oracle8 以上から Oracle9i へのアップグレードについては説明しません。Oracle8 以上から Oracle9i へのアップグレードの詳細は、第 3 章「Oracle9i の新しいリリースへのデータベースのアップグレード」を参照してください。

---

## レプリケーションのデータベース移行の概要

ご使用の環境、特にご使用のレプリケーション環境のマルチマスター・コンポーネントを 1 ステップでアップグレードすることが最も簡単な場合があります。通常、このようなアップグレードは、小規模な構成のみで可能です。大規模な構成の場合、既存の Oracle7 レプリケーション環境を Oracle9i へ段階的にアップグレードすることを考慮する場合があります。レプリケーションおよび管理操作は、Oracle7、Oracle8、Oracle8i および Oracle9i が混在したレプリケーション環境で正常に実行されます。

ただし、相互運用を正常に行うには、次の制限事項に従う必要があります。

- Oracle9i マテリアライズド・ビュー・サイトは、Oracle7 リリース 7.3.3 以上のマスター・サイトとのみ相互運用できます。
- Oracle9i マスター・サイトは、Oracle7 リリース 7.3.4 以上のマテリアライズド・ビュー・サイトとのみ相互運用できます。
- Oracle9i マスター・サイトは、Oracle7 リリース 7.3.3 以上のマスター・サイトとのみ相互運用できます。

---

---

**注意：** Oracle の以前のリリースでは、「マテリアライズド・ビュー」は「スナップショット」と呼ばれていました。これらの用語は同義です。この付録では、以前のリリースを説明する場合でも「マテリアライズド・ビュー」を使用します。

---

---

次のアップグレード方法が、レプリケーション環境でサポートされます。

- 手動アップグレード
- Database Upgrade Assistant
- Oracle7 からの全データベース・エクスポートおよび Oracle9i への全データベース・インポート

マスター・サイトを Oracle9i へアップグレードした後、すべての関連マテリアライズド・ビュー・サイトの完全リフレッシュを実行してください。Oracle9i から Oracle7 へのレプリケーション環境のダウングレードは、サポートされていません。

特定の Oracle9i レプリケーション機能では、この機能を使用する前に、すべてのサイトを Oracle8 以上に正常にアップグレードする必要があります。たとえば、主キー・マテリアライズド・ビューを使用する前に、そのマテリアライズド・ビュー・サイトおよび関連マスター・サイトを Oracle8 以上にアップグレードする必要があります。副問合せ機能のある単純マテリアライズド・ビューおよびマスター表の再編成プロセスでは、まず ROWID マテリアライズド・ビューから主キー・マテリアライズド・ビューへアップグレードする必要があります。

同様に、特定の Oracle9i レプリケーション機能では、機能が使用できるようになる前にすべてのサイトが Oracle8i 以上に正常にアップグレードされる必要があります。特定の Oracle9i レプリケーション機能では、機能が使用できるようになる前にすべてのサイトが Oracle9i に正常にアップグレードされる必要があります。たとえば、ユーザー定義型に基づくオブジェクトをレプリケートするには、すべてのサイトが Oracle9i である必要があります。これらの機能については、E-17 ページの「[Oracle の上位リリースへのアップグレードを必要とする機能](#)」を参照してください。

**参照：** Oracle Replication の詳細は、次のドキュメントを参照してください。

- Oracle Replication の概念については、『Oracle9i アドバンスド・レプリケーション』を参照してください。このマニュアルは、Oracle8 から Oracle9i までの主なリリースの新機能についても記載しています。
- Oracle Enterprise Manager のレプリケーション管理ツールを使用して、この付録で説明する手順を完了する方法については、レプリケーション管理ツールのオンライン・ヘルプを参照してください。
- レプリケーション管理 API を使用して、この付録で説明する手順を完了する方法については、『Oracle9i レプリケーション・マネージメント API リファレンス』を参照してください。

## すべてのサイトの同時アップグレード

この項では、マルチマスター環境のすべてのマスター・サイトを、同時に Oracle9i へアップグレードする手順について説明します。Oracle9i へアップグレードしないマテリアライズド・ビュー・サイトも Oracle7 リリース 7.3.4 以上にアップグレードする必要があります。ご使用のサイトを段階的にアップグレードする場合は、E-6 ページの「[段階的アップグレード](#)」を参照してください。

すべてのマスター・サイトおよび（オプションで）マテリアライズド・ビュー・サイトを同時にアップグレードするには、次の手順を実行します。

1. マテリアライズド・ビューからアップグレードするすべてのマスターへのすべての伝播およびリフレッシュを停止します。たとえば、マテリアライズド・ビュー・サイトで自動伝播および自動リフレッシュを制御するジョブ・キューのエントリを一時的に停止または中断することによって停止できます。DBMS\_JOB.BROKEN プロシージャを使用して、ジョブを中断できます。

マテリアライズド・ビュー・サイトのすべての遅延トランザクションは、マスター・サイトのアップグレード開始前に送信される必要があります。

**参照：** 『Oracle7 Server 分散システム Vol.2: レプリケート・データ』を参照してください。

2. それぞれのマスター・サイトのローカル・エラー・キューが空になるまで、ローカル・エラー・キューにあるエラーを解決して再実行します。

**参照：**『Oracle7 Server 分散システム Vol.2: レプリケート・データ』を参照してください。

3. すべてのマスター・レプリケーション・グループのマスター定義サイトで DBMS\_REPCAT パッケージの SUSPEND\_MASTER\_ACTIVITY プロシージャを実行し、レプリケーション環境を停止します。

**参照：**『Oracle7 Server 分散システム Vol.2: レプリケート・データ』を参照してください。

4. [第3章](#)で説明するいずれかのアップグレード方法を使用して、すべてのマスター・サイトをアップグレードします。

かわりに、エクスポート / インポートを使用することもできます。Oracle7 リリース 7.3.3 以上からデータベース全体をエクスポートして Oracle9i ヘインポートするには、次の手順を実行します。

- a. SYSTEM スキーマで FULL=y を設定した状態でリリース 7.3 のエクスポート・ユーティリティを使用し、Oracle7 リリース 7.3.3 以上のデータベースをダンプ・ファイルにエクスポートします。
- b. SYSTEM スキーマで FULL=y を設定した状態で Oracle9i のインポート・ユーティリティを使用し、ダンプ・ファイルを Oracle9i データベースにインポートします。

また、個別の Oracle7 表からデータをエクスポートし、そのデータを Oracle9i 表へインポートした後、標準レプリケーション・プロシージャを使用して、マスターとしてそれらの表を Oracle9i レプリケーション環境に構成することもできます。

エクスポート / インポートを使用する場合、マスター表に基づくマテリアライズド・ビューを削除および再作成する必要がある場合があります。

**参照：**

- [第8章「エクスポート / インポートによるデータベースの移行」](#)を参照してください。
- エクスポート / インポート実行の詳細は、『Oracle9i データベース・ユーティリティ』を参照してください。



5. レプリケーション管理ツールのセットアップ・ウィザードまたはセットアップ・スクリプトを使用して、マルチマスター・レプリケーション環境を設定します。
  - a. プライマリ・マスター・レプリケーションの管理者アカウントを作成し、このユーザーをすべてのマスター・サイトでのレプリケーション管理者、プロパゲータおよび受信者として登録します。
  - b. すべてのサイトに接続する適切なデータベース・リンクを設定します。
6. レプリケーション管理ツールまたはレプリケーション管理 API を使用して、各レプリケーション・ベース・オブジェクトに対するレプリケーション・サポートを再生成します。レプリケーション管理 API を使用する場合、DBMS\_REPCAT パッケージの GENERATE\_REPLICATION\_SUPPORT プロシージャを実行します。その他のアクティビティでは、レプリケーション・サポートの作成によって、登録されたプロパゲータが生成されたオブジェクトの所有者として確立されます。
7. レプリケーション管理ツールまたはレプリケーション管理 API を使用して、レプリケーション環境のレプリケーション・アクティビティを再開します。レプリケーション管理 API を使用する場合、DBMS\_REPCAT パッケージの RESUME\_MASTER\_ACTIVITY プロシージャを実行します。
8. すべての関連マテリアライズド・ビュー・サイトを Oracle7 リリース 7.3.4 にアップグレードするか、またはこれらのサイトを Oracle9i にアップグレードする必要があります。これらのマテリアライズド・ビュー・サイトを Oracle9i へアップグレードすることをお勧めします。

**参照：**

- ご使用のマテリアライズド・ビュー・サイトをリリース 7.3.4 へアップグレードする方法は、Oracle7 のマニュアルを参照してください。
  - ご使用のマテリアライズド・ビュー・サイトを Oracle9i へアップグレードする手順は、E-8 ページの「[マテリアライズド・ビュー・サイトの段階的アップグレード](#)」を参照してください。
9. マスター・サイトを Oracle9i へアップグレードしたすべてのマテリアライズド・ビュー・サイトですべてのマテリアライズド・ビューの完全リフレッシュを行います。リフレッシュ前に、DBMS\_JOB.BROKEN プロシージャをコールして、マテリアライズド・ビュー・サイトのアップグレード中に中断した可能性のあるジョブが中断されていないことを確認してください。

マテリアライズド・ビューが REFRESH FORCE オプションで定義されている場合、次のリフレッシュは自動的に完全リフレッシュになります。REFRESH FAST オプションで定義されたマテリアライズド・ビューは、DBMS\_REFRESH.REFRESH プロシージャまたはその他のリフレッシュ・プロシージャを使用して、手動でリフレッシュする必要があります。

マテリアライズド・ビュー・サイトで開始するプロシージャ・レプリケーションをマスター・サイトで使用している場合、プロシージャ・レプリケーションが使用するすべてのパッケージおよびパッケージ本体にも、マテリアライズド・ビュー・サポートを再作成します。

**注意：** マスター・サイトを Oracle9i にアップグレードするときに、マスターのすべてのマテリアライズド・ビュー・サイトを Oracle9i へアップグレードできる場合（マテリアライズド・ビュー・サイトを段階的アップグレードする必要がない場合）は、マスターのマテリアライズド・ビュー・ログを削除し、主キーのマテリアライズド・ビュー・ログとして再作成することもできます。各マテリアライズド・ビュー・サイトでマテリアライズド・ビューを変更して、主キーのマテリアライズド・ビューに変換してください。各主キーのマテリアライズド・ビューに対して完全リフレッシュを行えるようになります。詳細は、E-14 ページの「[主キー・マテリアライズド・ビューへのアップグレード](#)」を参照してください。

10. Oracle7 マルチマスター・レプリケーション環境のメンテナンスに使用していたすべての管理アカウントおよびデータベース・リンクを削除します。これらは、Oracle9i 環境では必要ありません。不要な権限が取り消される場合もあります。Oracle7 マテリアライズド・ビュー・サイトのメンテナンスに必要なアカウントを削除しないよう注意してください。

## 段階的アップグレード

レプリケーション環境では、段階的アップグレードを行うことができます。ただし、サイト間の相互依存性を慎重に分析し、アップグレード中の相互運用を確実に継続させる必要があります。表 E-1 に、Oracle7 と Oracle9i のレプリケーション・サイトが相互運用するための条件を示します。

表 E-1 レプリケーション環境での相互運用性

| 環境                    | 処置                                                 | 条件                                                   |
|-----------------------|----------------------------------------------------|------------------------------------------------------|
| マルチマスター               | マスター・サイトを Oracle7 から Oracle9i へアップグレードします。         | その他のすべてのマスター・サイトが Oracle7 リリース 7.3.3 以上である必要があります。   |
| 依存マテリアライズド・ビューのあるマスター | マスター・サイトを Oracle7 から Oracle9i へアップグレードします。         | すべての依存マテリアライズド・ビューが Oracle7 リリース 7.3.4 以上である必要があります。 |
| 依存マテリアライズド・ビューのあるマスター | マテリアライズド・ビュー・サイトを Oracle7 から Oracle9i へアップグレードします。 | 関連マスター・サイトが Oracle7 リリース 7.3.3 以上である必要があります。         |

レプリケーション環境内での相互運用性の問題を避けるために、段階的アップグレードを実行する必要がある場合は、次の順序で実行することをお勧めします。

1. すべてのマスター・サイトを Oracle7 リリース 7.3.3 以上にアップグレードし、E-7 ページの「[段階的アップグレードのための Oracle7 マスター・サイトの準備](#)」の手順を実行して、段階的アップグレードの準備を行います。
2. E-8 ページの「[マテリアライズド・ビュー・サイトの段階的アップグレード](#)」の手順を実行して、すべてのマテリアライズド・ビュー・サイトを Oracle9i へ段階的アップグレードします。
3. E-10 ページの「[マスター・サイトの段階的アップグレード](#)」の手順を実行して、すべてのマスター・サイトを Oracle9i へ段階的アップグレードします。

**参照：** ご使用の Oracle7 サイトのリリース 7.3.4 以上へのアップグレードについては、Oracle7 のマニュアルを参照してください。

## 段階的アップグレードのための Oracle7 マスター・サイトの準備

Oracle7 マスター・サイトまたはマテリアライズド・ビュー・サイトの段階的アップグレードを開始する前に、Oracle7 リリース 7.3.3 以上のマスター・サイトを構成し、各サイトのシングル・ユーザーのセキュリティ・コンテキスト内で、すべてのレプリケーション管理および伝播が行われるようにする必要があります。また、プライマリ・マスター・レプリケーション管理者は、すべての Oracle7 および Oracle9i サイトで同じユーザー名およびパスワードを持つ必要があります。

Oracle7 マスター・サイトは、このように構成済である場合があります。そうでない場合は、次の手順を実行します。

1. ご使用のレプリケーション環境に対するプライマリ・マスター・レプリケーション管理者を選択します。現行のレプリケーション管理者を選択するか、または新しいユーザーを作成します。
2. 各マスター・サイトで、DBMS\_REPCAT\_ADMIN.GRANT\_ADMIN\_ANY\_REPGROUP および DBMS\_REPCAT\_AUTH.GRANT\_SURROGATE\_REPCAT を使用して、プライマリ・マスター・レプリケーション管理者に必要な権限を付与します。
3. 次を示すリンクがまだ存在していない場合、マルチマスター環境の各マスター・サイトから他のすべてのマスター・サイトへのデータベース・リンクを作成する必要があります。
  - SYS として作成されたパブリック・データベース・リンク。有効なグローバル・データベース名、および有効な SQL\*Net 2.3 TNS 別名が指定された USING 句が含まれます。
  - SYS として作成されたプライベート・データベース・リンク。有効なグローバル・データベース名、およびプライマリ・マスター・レプリケーション管理者のユーザー名およびパスワードが指定された CONNECT TO 句が含まれます。

- プライマリ・マスター・レプリケーション管理者が作成したプライベート・データベース・リンク。有効なグローバル・データベース名、およびプライマリ・マスター・レプリケーション管理者のユーザー名およびパスワードが指定された CONNECT TO 句が含まれます。

## マテリアライズド・ビュー・サイトの段階的アップグレード

マテリアライズド・ビュー・サイトを Oracle9i へアップグレードする前に、関連マスター・サイトを Oracle7 リリース 7.3.3 以上へアップグレードし、マスター・サイトで段階的アップグレードの準備を完全に整える必要があります。

### 参照：

- ご使用の Oracle7 サイトのリリース 7.3.4 以上へのアップグレードについては、Oracle7 のマニュアルを参照してください。
- マスター・サイトの準備については、E-7 ページの「[段階的アップグレードのための Oracle7 マスター・サイトの準備](#)」を参照してください。

Oracle7 のマテリアライズド・ビュー・サイトを Oracle9i へ段階的アップグレードするには、それぞれのマテリアライズド・ビュー・サイトで次の手順を実行します。

1. 次の手順でマテリアライズド・ビュー・サイトをレプリケーション環境から切り離します。
  - a. マテリアライズド・ビュー・サイトで更新可能なマテリアライズド・ビューへのすべてのローカルの更新を停止します。
  - b. 別のセッションで、それ以降のトランザクションを回避するために、各マテリアライズド・ビューの実表をロックします。
  - c. キューをマテリアライズド・ビュー・サイトのマスターへ送信して、ローカルの遅延トランザクション・キューを空にします。『Oracle7 Server 分散システム Vol.2: レプリケート・データ』を参照してください。
  - d. マテリアライズド・ビュー・サイトでマテリアライズド・ビューの自動伝播および自動リフレッシュを制御するジョブ・キューのエントリを一時的に停止または中断し、マテリアライズド・ビューからマスターへのすべての伝播を停止します。DBMS\_JOB.BROKEN プロシージャを使用して、ジョブを中断できます。『Oracle7 Server 分散システム Vol.2: レプリケート・データ』を参照してください。
2. [第3章](#)で説明するいずれかのアップグレード方法を実行します。

かわりに、エクスポート / インポートを使用することもできます。Oracle7 リリース 7.3.3 以上からデータベース全体をエクスポートして Oracle9i へインポートするには、次の手順を実行します。

- a. SYSTEM スキーマで FULL=y を設定した状態でリリース 7.3 のエクスポート・ユーティリティを使用し、Oracle7 リリース 7.3.3 以上のデータベースをダンプ・ファイルにエクスポートします。
- b. SYSTEM スキーマで FULL=y を設定した状態で Oracle9i のインポート・ユーティリティを使用し、ダンプ・ファイルを Oracle9i データベースにインポートします。

また、個別の Oracle7 表からデータをエクスポートし、そのデータを Oracle9i 表へインポートした後、標準レプリケーション・プロシージャを使用して、マスターとしてそれらの表を Oracle9i レプリケーション環境に構成することもできます。

#### 参照：

- [第 8 章「エクスポート / インポートによるデータベースの移行」](#) を参照してください。
  - エクスポート / インポート実行の詳細は、『Oracle9i データベース・ユーティリティ』を参照してください。
3. レプリケーション管理ツールのセットアップ・ウィザードまたは適切なレプリケーション管理 API コールを使用して、次の操作を実行します。
    - マテリアライズド・ビュー・サイトのレプリケーション管理者およびプロパゲータとしてマテリアライズド・ビューのプライマリ・レプリケーション管理者を登録します。レプリケーション管理 API を使用する場合、DBMS\_DEFER\_SYS パッケージの REGISTER\_PROPAGATOR プロシージャを使用します。
    - 関連マスター・サイトで受信者アカウントを登録します。Oracle7 マスター・サイトのマテリアライズド・ビュー・サイトでは、マスター・サイトの受信者は、前述の項で準備したプライマリ・マスター・レプリケーション管理者である必要があります。レプリケーション管理ツールのセットアップ・ウィザードを使用する場合、Customize オプションを選択してこの受信者を指定します。レプリケーション管理 API を使用する場合、DBMS\_REPCAT\_ADMIN パッケージの REGISTER\_USER\_REPGROUP プロシージャを使用します。
  4. マテリアライズド・ビュー・サイトからマスター・サイトへの適切なデータベース・リンクを作成します。

特に、マテリアライズド・ビュー・サイトからマスター・サイトへの PUBLIC データベース・リンクを作成する必要があります。これによって、それぞれのリンクに USING 句を含める必要がないため、プライベート・データベース・リンクの定義が簡単になります。マスター・サイトでのマテリアライズド・ビュー管理者からプロキシ管理者へのプライベート・データベース・リンクおよびマスター・サイトでのプロパゲータから受信者へのプライベート・データベース・リンクも必要です。

5. レプリケーション管理ツールまたは適切なレプリケーション管理 API のコールを使用して、マテリアライズド・ビュー・レプリケーション・サポートを再生成します。レプリケーション管理 API を使用する場合、DBMS\_REPCAT パッケージの `GENERATE_MVIEW_SUPPORT` プロシージャを実行します。その他のアクティビティでは、レプリケーション・サポートの作成によって、登録されたプロパゲータが生成されたオブジェクトの所有者として確立されます。
6. レプリケーション管理ツールまたは適切なレプリケーション管理 API のコールを使用して、マスターで伝播またはリフレッシュ（あるいはその両方）の間隔をスケジューリングしなおし、ローカルの更新が適切に行われるようにします。レプリケーション管理 API を使用する場合、DBMS\_DEFER\_SYS パッケージの `SCHEDULE_PUSH` プロシージャを実行して伝播スケジュールを設定し、DBMS\_REFRESH パッケージの `MAKE` プロシージャを実行してリフレッシュ・グループのリフレッシュ間隔を設定します。
7. DBMS\_JOB.BROKEN プロシージャを使用して手順 1 でマスター・サイトを切り離れた場合、中断しているジョブを元に戻してマテリアライズド・ビュー・サイトからレプリケーション・アクティビティを再開する必要があります。
8. Oracle7 レプリケーション環境のメンテナンスに使用し、Oracle9i 環境では不要な管理アカウントおよびリンクを削除します。不要な権限が取り消される場合もあります。
9. スケジュールに従って、アップグレードしていない他のマテリアライズド・ビュー・サイトについても、この手順をすべて実行します。

## マスター・サイトの段階的アップグレード

マスター・サイトを Oracle7 から Oracle9i へアップグレードする前に、次の条件を満たす必要があります。

- マルチマスター環境内のその他すべてのマスター・サイトが Oracle7 リリース 7.3.3 以上を実行している。
- E-7 ページの「[段階的アップグレードのための Oracle7 マスター・サイトの準備](#)」の手順を完了している。
- すべての依存マテリアライズド・ビューが Oracle7 リリース 7.3.4 以上を実行している必要があります。

**参照：** Oracle7 サイトのアップグレードの詳細は、Oracle7 のマニュアルを参照してください。

Oracle7 マスター・サイトを Oracle9i へ段階的アップグレードするには、次の手順を実行します。

1. アップグレードするマスター・サイトを選択します。まず、マスター定義サイトをアップグレードする必要があります。
2. プロシージャ・レプリケーションを使用している場合は、既存のプロシージャ・ラッパーの構成情報および場所を記録します。この情報は後で使用します。

3. マスター・サイトをレプリケーション環境から切り離します。切り離すには、次の手順を実行します。

- a. マスター定義サイトで、すべてのマスター・レプリケーション・グループに対して `DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY` をコールするか、またはすべてのリモート・マスター・サイトおよび依存マテリアライズド・ビュー・サイトで、(Oracle7 サイトに対しては) `DBMS_DEFER_SYS.UNSCHEDULE_EXECUTION` または (Oracle8 以上のサイトに対しては) `DBMS_DEFER_SYS.UNSCHEDULE_PUSH` をコールし、マスター・サイトへの更新を停止します。また、アップグレードされているマスター・サイトに影響する可能性があるマスター定義サイトで、管理操作を行わないでください。

**参照：**『Oracle7 Server 分散システム Vol.2: レプリケート・データ』を参照してください。

- b. アップグレードされるマスター・サイトで DML アクティビティを回避します。

**参照：**『Oracle7 Server 分散システム Vol.2: レプリケート・データ』を参照してください。

- c. キューをすべてのサイトへ手動で送信して、ローカルの遅延トランザクション・キューを空にします。

**参照：**『Oracle7 Server 分散システム Vol.2: レプリケート・データ』を参照してください。

- d. ローカル・エラー・キューが空になるまで、ローカル・エラー・キューにあるエラーを解決して再実行します。

**参照：**『Oracle7 Server 分散システム Vol.2: レプリケート・データ』を参照してください。

- e. マテリアライズド・ビュー・サイトで自動伝播および自動リフレッシュを制御する各マテリアライズド・ビュー・サイトのジョブ・キューのエントリを中断し、マテリアライズド・ビュー・サイトのリフレッシュが行われないようにします。`DBMS_JOB.BROKEN` プロシージャを使用して、ジョブを中断できます。

**参照：**『Oracle7 Server 分散システム Vol.2: レプリケート・データ』を参照してください。

4. [第3章](#)で説明するいずれかのアップグレード方法を使用して、マスター・サイトをアップグレードします。

かわりに、エクスポート / インポートを使用することもできます。Oracle7 リリース 7.3.3 以上からデータベース全体をエクスポートして Oracle9i へインポートするには、次の手順を実行します。

- a. SYSTEM スキーマで FULL=y を設定した状態でリリース 7.3 のエクスポート・ユーティリティを使用し、Oracle7 リリース 7.3.3 以上のデータベースをダンプ・ファイルにエクスポートします。
- b. SYSTEM スキーマで FULL=y を設定した状態で Oracle9i のインポート・ユーティリティを使用し、ダンプ・ファイルを Oracle9i データベースにインポートします。

また、個別の Oracle7 表からデータをエクスポートし、そのデータを Oracle9i 表へインポートした後、標準レプリケーション・プロシージャを使用して、マスターとしてそれらの表を Oracle9i レプリケーション環境に構成することもできます。

エクスポート / インポートを使用する場合、マスター表に基づくマテリアライズド・ビューを削除および再作成する必要がある場合があります。

### 参照：

- [第8章「エクスポート / インポートによるデータベースの移行」](#)を参照してください。
- エクスポート / インポート実行の詳細は、『Oracle9i データベース・ユーティリティ』を参照してください。

5. レプリケーション管理ツールのセットアップ・ウィザードまたはレプリケーション管理 API を使用して、マスター・サイトに対するレプリケーション管理者、プロパゲータおよび受信者としてプライマリ・マスター・レプリケーション管理者を登録します。

E-7 ページの「[段階的アップグレードのための Oracle7 マスター・サイトの準備](#)」の手順に従って、ご使用の Oracle7 マスター・サイトを Oracle9i との互換用に準備できている場合は、プライマリ・マスター・レプリケーション管理者からプライマリ・マスター・レプリケーション管理者へのデータベース・リンクが、その他のすべての Oracle7 および Oracle9i マスター・サイトに存在しています。

6. まだ停止状態でない場合、レプリケーション管理ツールまたはレプリケーション管理 API を使用して、すべてのマスター・グループに対してすべてのレプリケーション・アクティビティを一時的に停止します。レプリケーション管理 API を使用する場合、すべてのマスター・グループのマスター定義サイトで DBMS\_REPCAT パッケージの SUSPEND\_MASTER\_ACTIVITY プロシージャを実行します。



7. レプリケーション管理ツールまたはレプリケーション管理 API を使用して、各レプリケーション・オブジェクトに対するレプリケーション・サポートを再作成します。

レプリケーション環境のサイトが Oracle7 を実行している場合、レプリケーション・サポートを生成時に `min_communication` パラメータを `false` に設定する必要があります。すべてのサイトを Oracle9i（または、Oracle8 以上のサイトが混在した環境）へアップグレードした後、`min_communication` パラメータは `true`（デフォルト）に設定する必要があります。レプリケーション管理 API を使用する場合、DBMS\_REPCAT パッケージの `GENERATE_REPLICATION_SUPPORT` プロシージャを実行します。その他のアクティビティでは、レプリケーション・サポートの作成によって、登録されたプロパゲータが生成されたオブジェクトの所有者として確立されます。

**参照：** `min_communication` の詳細は、『Oracle9i アドバンスド・レプリケーション』を参照してください。

8. プロシージャ・レプリケーションを使用している場合、残りの Oracle7 マスター・サイトを確認してラッパーが移動されたかどうかを判断します（手順 2 で作成されたラッパーのリスト）。移動されている場合、プライマリ・レプリケーション管理者のスキーマにある新しい場所を指定するように、シノニムを以前の場所（サイトがシステム・ベース・モデルかユーザー・ベース・モデルかによって、レプリケーション管理者または表所有者のいずれかのスキーマ内にあります）に作成します。新しい所有者および場所へアクセスするために必要なオブジェクト権限が付与されているかどうかを確認してください。マテリアライズド・ビュー・サイトで開始するプロシージャ・レプリケーションを使用している場合、プロシージャ・レプリケーションが使用するすべてのパッケージおよびパッケージ本体にも、マテリアライズド・ビュー・サポートを再作成します。
9. 他のマスターへの伝播および他のマスターからの伝播をスケジューリングしないでマスターを切り離した場合、すべてのマスター・サイトに対して（Oracle7 サイトに対して）`DBMS_DEFER_SYS.SCHEDULE_EXECUTION`、または（Oracle8 以上のサイトに対して）`DBMS_DEFER_SYS.SCHEDULE_PUSH` を実行して、伝播をスケジューリングしなおします。
10. レプリケーション管理ツールまたはレプリケーション管理 API を使用して、各マスター・グループに対するレプリケーション・アクティビティを再開します。レプリケーション管理 API を使用する場合、DBMS\_REPCAT パッケージの `RESUME_MASTER_ACTIVITY` プロシージャを実行します。
11. マスター・サイトを Oracle9i へアップグレードした後にすべてのマテリアライズド・ビュー・サイトですべてのマテリアライズド・ビューの完全リフレッシュを行います。Oracle8 で導入された新しい ROWID 書式のために、マスター表のすべてのマテリアライズド・ビュー・ログはアップグレード中に切り捨てられます。
12. `DBMS_JOB.BROKEN` プロシージャを使用して手順 3 でマスター・サイトを切り離した場合、中断しているジョブを元に戻してマテリアライズド・ビュー・サイトからレプリケーション・アクティビティを再開します。

マテリアライズド・ビューが REFRESH FORCE オプションで定義されている場合、次のリフレッシュは自動的に完全リフレッシュになります。REFRESH FAST オプションで定義されたマテリアライズド・ビューは、DBMS\_REFRESH.REFRESH プロシージャまたはその他のリフレッシュ・プロシージャを使用して、手動でリフレッシュする必要があります。

---

**注意：** マスター・サイトを Oracle9i にアップグレードするときに、マスターのすべてのマテリアライズド・ビュー・サイトを Oracle9i へアップグレードできる場合（マテリアライズド・ビュー・サイトを段階的アップグレードする必要がない場合）は、マスターのマテリアライズド・ビュー・ログを削除し、主キーのマテリアライズド・ビュー・ログとして再作成することもできます。各マテリアライズド・ビュー・サイトでマテリアライズド・ビューを変更して、主キーのマテリアライズド・ビューに変換してください。各主キーのマテリアライズド・ビューに対して完全リフレッシュを行えるようになります。詳細は、E-14 ページの「[主キー・マテリアライズド・ビューへのアップグレード](#)」を参照してください。

---

13. Oracle7 マルチマスター・レプリケーション環境のメンテナンスに使用していたすべての管理アカウントおよびリンクを削除します。これらは、Oracle9i 環境では必要ありません。不要な権限が取り消される場合もあります。Oracle7 マテリアライズド・ビュー・サイトまたはマスター・サイトのメンテナンスに必要なアカウントを削除しないよう注意してください。
14. スケジュールに従って、アップグレードしていない他のマスター・サイトのこのプロシージャのすべての手順を実行します。

## 主キー・マテリアライズド・ビューへのアップグレード

マテリアライズド・ビュー・サイトおよびそのマスターが Oracle9i へアップグレードされると、ROWID マテリアライズド・ビューを Oracle9i の主キー・マテリアライズド・ビューへアップグレードできます。アップグレードするには、まず各マスター表に対するマテリアライズド・ビュー・ログを変更し、DML がマスターで実行されたときに主キー情報および ROWID 情報のログが取られるようにします。マスター・サイトでこれが完了したら、マテリアライズド・ビューが主キー・マテリアライズド・ビューに変換されるように変更して、Oracle9i マテリアライズド・ビュー・サイトの段階的な変換を行うことができます。主キーおよび ROWID の情報のログを取るように変更された Oracle9i マスターは、Oracle7 ROWID マテリアライズド・ビュー、および Oracle9i の主キー・マテリアライズド・ビューと ROWID マテリアライズド・ビューの同時段階的アップグレードをサポートします。

---

**注意：** 主キー・マテリアライズド・ビューは ROWID マテリアライズド・ビューへ変換またはダウングレードできません。

---

## マスター・サイトでの主キー・マテリアライズド・ビューの変換

主キー・マテリアライズド・ビューをサポートするには、Oracle9i マスター・サイトで次の手順を実行します。

1. 主キー制約が使用可能になっていない各マスター表で、主キー制約を定義し、使用可能にします。
2. ALTER MATERIALIZED VIEW LOG 文を使用して、高速リフレッシュをサポートする各マスター表のマテリアライズド・ビュー・ログを変更して、主キー情報を含むようにします。

たとえば、次の文では、主キー情報を記録するように既存の ROWID マテリアライズド・ビュー・ログを変更します。

```
ALTER MATERIALIZED VIEW LOG ON hr.employees
ADD PRIMARY KEY;
```

**参照：** 詳細は、『Oracle9i SQL リファレンス』の「ALTER MATERIALIZED VIEW LOG」を参照してください。

---

**注意：** 手順 1 および 2 を実行しない場合、マテリアライズド・ビュー・サイトで主キー・マテリアライズド・ビューに変換するために ALTER MATERIALIZED VIEW 文を実行するとエラーが発生します。

---

## マテリアライズド・ビュー・サイトでの主キー・マテリアライズド・ビューの変換

主キー・マテリアライズド・ビューをサポートするために Oracle9i マスター・サイトを構成した後、Oracle9i マテリアライズド・ビュー・サイトで次の手順を実行します。

1. 次の手順でマテリアライズド・ビュー・サイトをレプリケーション環境から切り離します。
  - a. マテリアライズド・ビュー・サイトで更新可能なマテリアライズド・ビューへのすべてのローカルの更新を停止します。
  - b. キューをマテリアライズド・ビュー・サイトのマスターへ送信して、ローカルの遅延トランザクション・キューを空にします。DBMS\_DEFER\_SYS.PUSH プロシージャを使用して、遅延トランザクションを送信します。詳細は、『Oracle9i レプリケーション・マネージメント API リファレンス』を参照してください。

- c. マテリアライズド・ビュー・サイトでマテリアライズド・ビューの自動伝播および自動リフレッシュを制御するジョブ・キューのエントリを一時的に停止または中断し、マテリアライズド・ビューからマスターへのすべての伝播を停止します。  
DBMS\_JOB.BROKEN プロシージャを使用して、ジョブを中断できます。詳細は、『Oracle9i PL/SQL パッケージ・プロシージャおよびタイプ・リファレンス』を参照してください。
2. 読み込み専用の ROWID マテリアライズド・ビューを主キー・マテリアライズド・ビューへ変換して、これらの ROWID マテリアライズド・ビューが主キーの列すべてを含まない場合、主キー列で読み込み専用のマテリアライズド・ビューを削除および再生成します。

**参照：** ROWID マテリアライズド・ビューの詳細は、『Oracle9i アドバンスト・レプリケーション』を参照してください。

3. 高速リフレッシュできるすべてのマテリアライズド・ビューの高速リフレッシュを実行し、マスター・マテリアライズド・ビュー・ログに残っている ROWID 参照への必要性を削除します。
4. ALTER MATERIALIZED VIEW 文を使用して、ROWID マテリアライズド・ビューを主キー・マテリアライズド・ビューに変換します。

たとえば、次の文では、ROWID マテリアライズド・ビューを主キー・マテリアライズド・ビューに変更します。

```
ALTER MATERIALIZED VIEW hr.employees_mv
 REFRESH WITH PRIMARY KEY;
```

**参照：** ALTER MATERIALIZED VIEW の完全な構文については、『Oracle9i SQL リファレンス』を参照してください。

5. マスターでの伝播またはマテリアライズド・ビュー（あるいはその両方）のリフレッシュをスケジューリングしなおし、レプリケーションを再開させて、ローカルの更新が適切に行われるようにします。レプリケーション管理 API を使用する場合、DBMS\_DEFER\_SYS パッケージの SCHEDULE\_PUSH プロシージャを実行して伝播スケジュールを設定し、DBMS\_REFRESH パッケージの MAKE プロシージャを実行してリフレッシュ・グループのリフレッシュ間隔を設定します。
6. DBMS\_JOB.BROKEN プロシージャを使用して手順 1 でマスター・サイトを切り離した場合、中断しているジョブを元に戻してマテリアライズド・ビュー・サイトからレプリケーション・アクティビティを再開する必要があります。

## Oracle の上位リリースへのアップグレードを必要とする機能

Oracle データベース・サーバーの各メジャー・リリースには、新しい機能が追加されています。次の項では、データベースを Oracle の上位リリースにアップグレードする場合に使用できる機能を示します。

**参照：** これらの新しいレプリケーション機能の詳細は、『Oracle9i アドバンスド・レプリケーション』の「新機能」を参照してください。

### Oracle9i を必要とする機能

関連するすべてのレプリケーション・サイトが Oracle9i である場合、次の機能を使用できます。

- マスター・グループを停止しない新しいマスター・サイトの追加
- マスター・グループを停止しないマスター表への新しい列の追加
- マスター・グループを停止しない単一マスター環境における安全な変更でのマスター表の変更
- 基となるユーザー定義型およびオブジェクトのレプリケーション
- 複数層の更新可能なマテリアライズド・ビュー
- パラレル伝播の行レベル依存性の追跡
- CHAR 列長セマンティクスまたは Unicode を使用する表のレプリケーション
- マテリアライズド・ビューの次のタイプ的高速リフレッシュ
  - 1 対多の副問合せによるマテリアライズド・ビュー
  - 多対多の副問合せによるマテリアライズド・ビュー
  - UNION によるマテリアライズド・ビュー

### Oracle8i 以上を必要とする機能

マスター・サイトが Oracle8i リリース 8.1.7 以上である場合、次の機能を使用できます。

- 単一マスター・レプリケーション環境の拡張可能性この機能は、単一マスター・レプリケーション環境のマスター・グループを停止する必要がある管理オペレーションの数を削減します。これらのオペレーションの完全なリストは、『Oracle9i アドバンスド・レプリケーション』の「新機能」を参照してください。

すべてのレプリケーション・サイトで、次の機能を使用するには、Oracle8i リリース 8.1.5 以上を実行する必要があります。

- 配置テンプレートを使用したマテリアライズド・ビュー・サイトのインスタンス化
- パラメータ化したマテリアライズド・ビューの配置テンプレート
- 更新可能なマテリアライズド・ビューの列のサブセット化

## Oracle8 以上を必要とする機能

すべてのレプリケーション・サイトで、次の機能を使用するには、Oracle8 以上を実行する必要があります。

- 遅延トランザクションの平行伝播
- 削減されたデータ伝播
  - DBMS\_REPCAT パッケージおよび DBMS\_OFFLINE\_SNAPSHOT パッケージの様々なプロシージャの min\_communication パラメータの使用
  - DBMS\_REPCAT パッケージの SEND\_OLD\_VALUES および COMPARE\_OLD\_VALUES プロシージャの使用
- 副問合せを使用する単一マテリアライズド・ビュー作成のデータのサブセット化
- LOB データ型のレプリケーション
- 主キー・マテリアライズド・ビュー
- グローバル認証および権限付きデータベース・リンク
- DBMS\_REPCAT パッケージの VALIDATE 機能の使用
- 高速リフレッシュの機能を使用した表の再編成
- パーティション表および索引のレプリケーション

## Oracle7 以上で使用できる機能

次の機能は、一部のサイトが Oracle7 を実行し、他のサイトが Oracle8 以上を実行している環境で自動的に動作します。ただし、これらの機能は Oracle8 以上のサイトにのみ適用します。

- ファイングレイン停止
- マスター・サイトでのマテリアライズド・ビュー登録

---

---

**注意：** Oracle7 サイトでマスター・グループが停止された場合、このサイトのすべてのマスター・グループが停止されます。

---

---

---

---

**注意：** Oracle7 マテリアライズド・ビューは、Oracle9i サイトでは自動的に登録されませんが、マスター・サイトで DBMS\_MVIEW.REGISTER\_MVIEW プロシージャを使用して手動で登録することはできます。このプロシージャの使用の詳細は、『Oracle9i レプリケーション・マネージメント API リファレンス』を参照してください。

---

---

## 廃止されたプロシージャ

次のレプリケーション管理 API は、Oracle8 以上のリリースで廃止されました。

- DBMS\_REPCAT.GENERATE\_REPLICATION\_PACKAGE
- DBMS\_REPCAT.GENERATE\_REPLICATION\_TRIGGER
- DBMS\_REPCAT\_ADMIN.GRANT\_ADMIN\_REPGROUP
- DBMS\_REPCAT\_ADMIN.GRANT\_ADMIN\_ANY\_REPGROUP
- DBMS\_REPCAT\_ADMIN.REVOKE\_ADMIN\_REPGROUP
- DBMS\_REPCAT\_ADMIN.REVOKE\_ADMIN\_ANY\_REPGROUP
- DBMS\_REPCAT\_AUTH.GRANT\_SURROGATE\_REPCAT
- DBMS\_REPCAT\_AUTH.REVOKE\_SURROGATE\_REPCAT
- DBMS\_DEFER\_SYS.EXECUTE
- DBMS\_DEFER\_SYS.SCHEDULE\_EXECUTION





## 数字

- 2GB を超えるファイル
  - オペレーティング・システム依存性, 5-48
- 32 ビットから 64 ビットへの変換
  - 「ワード・サイズ」を参照

## A

- ADA
  - 「SQL\*Module for ADA」を参照
- AL24UTFFSS キャラクタ・セット
  - Oracle9i での非サポート, 5-31
- ALTER DATABASE CONVERT 文, 3-21, D-3
  - 再実行, D-30
- ALTER DATABASE OPEN RESETLOGS MIGRATE 文, 3-21, D-3
- ALTER DATABASE RESET COMPATIBILITY 文, 5-8
- ALTER DATABASE RESETLOGS 文, D-5
- ALTER TABLE 文
  - ビットマップ索引の無効性, 5-29
- ANALYZE TABLE VALIDATE STRUCTURE 文
  - リリース 8.1 での変更, 5-19
- AUDIT\_TRAIL 初期化パラメータ
  - アップグレード用, D-26

## C

- CATALOG5.SQL スクリプト
  - 廃止, 5-28
- CATALOG6.SQL スクリプト
  - 廃止, 5-28
- CATALOG.SQL スクリプト, 3-23, 7-29, D-3
- CATCLUST.SQL スクリプト, 7-29
- CATEXP7.SQL スクリプト, 8-2

- CATPARR.SQL スクリプト, 7-29
- CATPROC.SQL スクリプト, 3-23, 7-29, D-3
- CATREP.SQL スクリプト, 7-29
- CHAR
  - 最大サイズ
    - 変更, 5-17
- CHARACTER キーワード
  - 動作の違い, 5-25
- CHECK\_ONLY
  - 移行ユーティリティ・オプション, D-9
- ckpdom.ora ファイル, B-20
- ckpreg.ora ファイル, B-20
- ckptop.ora ファイル, B-20
- CLOB
  - 互換性, 5-29
- COMMIT FORCE コマンド, D-5
- COMMIT キーワード
  - 動作の違い, 5-25
- COMMIT コマンド
  - Server Manager と SQL\*Plus との違い, C-14
- COMPATIBLE 初期化パラメータ, 5-2
  - 確認, 5-5
  - 設定, 5-6
    - 必要とする機能, 5-9
  - 設定のタイミング, 5-5
  - データベース構造, 5-3
- Connect-Time Failover
  - listener.ora の GLOBAL\_DBNAME ネットワーク・パラメータ, B-11
- ConText
  - Oracle Text への移行, D-5
- CREATE LIBRARY コマンド
  - Server Manager と SQL\*Plus との違い, C-13

## CREATE TABLE

AS 副問合せ

動作変更, 5-35

動作変更, 5-35

## CREATE TYPE コマンド

Server Manager と SQL\*Plus との違い, C-13

## D

---

### Database Upgrade Assistant

実行, 3-4, 3-5

メリット, 2-4

### DB\_BLOCK\_CHECKSUM

新規デフォルト値, A-5

### DB\_BLOCK\_CHECKSUM 初期化パラメータ

互換性, A-5

### DB\_BLOCK\_SIZE 初期化パラメータ

アップグレード用, D-4

### DB\_DOMAIN 初期化パラメータ, B-11

互換性, A-8

### DB\_NAME 初期化パラメータ, B-11

### DBMS

プリコンパイラ・コマンドライン・オプション,  
5-21

### DBMS\_APPLICATION\_INFO パッケージ

SET\_SESSION\_LONGOPS プロシージャ  
構文変更, 5-36

### DBMS\_ROWID パッケージ, D-34

### DBMS\_STATS パッケージ

統計表のアップグレード, 4-18

### DBNAME

移行ユーティリティ・オプション, D-9

### DBUA

「Database Upgrade Assistant」を参照

### DEC キーワード

動作の違い, 5-25

### DEGREE キーワード

PARALLEL 句, 5-37

### DML\_LOCKS 初期化パラメータ

互換性, A-7

### DUMP\_TNSNAMES コマンド, B-15

## E

---

### EBU

バックアップ管理, 5-41

## F

---

### FAILOVER ネットワーク・パラメータ, B-7

### FALSE キーワード

動作の違い, 5-25

### Forms

Oracle Forms アプリケーションのアップグレード,  
6-10

## G

---

### GLOBAL\_DBNAME ネットワーク・パラメータ, B-11

### GREATEST\_LB ファンクション

非サポート, 5-34

## I

---

### Identix 認証, B-2

### INIT.ORA パラメータ

「初期化パラメータ」を参照

### INSTANCE\_NAME 初期化パラメータ, B-8

### INSTANCES キーワード

PARALLEL 句からの削除, 5-37

### INT キーワード

動作の違い, 5-25

## J

---

### Java

ダウングレード, 7-30, 7-31

### JOB\_QUEUE\_PROCESSES

ジョブ・キュー・プロセスの最大数, A-5

### JOB\_QUEUE\_PROCESSES 初期化パラメータ

互換性, A-5

### JVMD817.SQL スクリプト, 7-30

### JVMD901.SQL スクリプト, 7-30

## L

---

### LARGE\_POOL\_SIZE 初期化パラメータ

動作の変更点, 4-13

パラレル実行の割当て, A-8

### LEAST\_UB ファンクション

非サポート, 5-34

### listener.ora ファイル

アップグレード, B-11

アップグレード後に変更, 4-8

移行, B-11  
パラメータ  
    GLOBAL\_DBNAME, B-11  
    ORACLE\_HOME, B-11  
    SID\_NAME, B-11  
LOAD\_BALANCE ネットワーク・パラメータ, B-7  
LOAD\_TNSNAMES コマンド, B-16, B-19  
LOB  
    互換性, 5-29  
LOB 索引句  
    互換性, 5-30  
LOB リテンション  
    ダウングレード, 7-7  
LOB を含む自動セグメント領域で管理される表領域  
    ダウングレード, 7-7  
LOB を含むパーティション索引構成表  
    ダウングレード, 7-12  
LOG\_CHECKPOINT\_TIMEOUT  
    新規デフォルト値, A-7  
LOG\_CHECKPOINT\_TIMEOUT 初期化パラメータ  
    互換性, A-7  
LogMiner  
    互換性, 5-44  
LU6.2 プロトコル, B-3

## M

---

Messaging Gateway  
    ダウングレード, 7-31  
MGWD901.SQL スクリプト, 7-31  
MIGRATE.BSQ スクリプト, D-13  
MIGRATE ユーザー  
    回避, D-7  
MULTIPLIER  
    移行ユーティリティ・オプション, D-9, D-27

## N

---

NAMES\_DID 表, B-13  
NAMES.ADMIN\_REGION ネットワーク・パラメータ, B-14, B-15  
NAMES.DOMAINS ネットワーク・パラメータ, B-13  
NCHAR  
    Oracle8i での使用, 5-31  
    移行, 5-32  
NCHAR 列  
    アップグレード, 4-11

NCLOB  
    互換性, 5-29  
Net8  
    Oracle Names リリース 1 (9.0.1) へのアップグレード, B-12  
    Oracle Net Services へのアップグレード, B-8, B-11  
        listener.ora ファイル, B-11  
        tnsnames.ora ファイル, B-10  
        クライアントのソフトウェア, B-9  
        構成ファイル, B-9  
        サーバーのソフトウェア, B-9  
    SID ネットワーク・パラメータ, B-5  
    共存問題, B-4 ~ B-6  
        Oracle9i データベース, B-4  
        SERVICE\_NAME ネットワーク・パラメータの使用, B-4  
        SID ネットワーク・パラメータの使用, B-5  
    サービス・ネーミング, 5-47  
    接続時ロード・バランシング, 5-47  
Net8 Client 8.0 と互換性のあるオプションを使用する  
    オプション, B-7  
Net8 OPEN, B-2  
NEW\_DBNAME  
    移行ユーティリティ・オプション, D-9  
NLS  
    移行, 5-32  
nmcbild.sql スクリプト, B-17  
nmcrgrnt.sql スクリプト, B-17  
NO\_SPACE\_CHECK  
    移行ユーティリティ・オプション, D-10  
Novell Directory Services (NDS)  
    外部ネーミング, B-2  
    認証, B-2  
NUMERIC キーワード  
    動作の違い, 5-25

## O

---

O7\_DICTIONARY\_ACCESSIBILITY 初期化パラメータ  
    互換性, 5-28, A-7  
OCI  
    OCIChangePassword コール, 5-39  
    OCISessionBegin コール, 5-39  
    アプリケーション  
        Oracle9i を使用するための変更, 6-9  
    アプリケーションの Oracle9i へのアップグレード, 6-3

- 互換性, 5-19
  - LISTEN コールおよび AQ, 5-20
  - クライアント通知, 5-20
  - スレッド・セーフティ, 5-19
  - バッチ・エラー・モード, 5-20
  - リンク・ライン, 5-20
- 相互運用性, 5-19
  - Oracle7 クライアント, 5-20
  - 共有構造, 5-19
- OCI アプリケーション
  - アップグレード選択肢, 6-6
- OFA, 1-10
- OLON コール
  - 廃止, 5-19
- Optimal Flexible Architecture
  - 「OFA」を参照
- Oracle Call Interface
  - 「OCI」を参照
- Oracle Developer アプリケーション
  - アップグレード, 6-10
- Oracle Dynamic Services
  - 使用不可, 5-15
- Oracle Enterprise Manager
  - listener.ora ファイルのスタティック・サービス情報, B-11
- Oracle *interMedia*
  - アップグレード, 3-26
- Oracle Managed Files
  - 互換性, 5-14
  - 相互運用性, 5-14
  - ファイル名の移行, 4-3
- Oracle Media Management API
  - 互換性
    - プロキシ・コピー要件, 5-45
- Oracle Names
  - NAMES\_DOM 表, B-13
  - namesini.sql スクリプト, B-21
  - names.ora ファイルの NAMES.ADMIN\_REGION パラメータ, B-14, B-15
  - names.ora ファイルの NAMES.DOMAINS パラメータ, B-13
  - namesupg.sql スクリプト, B-12, B-15, B-18, B-21
  - NMO\_INFORMATION 表, B-14
  - Oracle Names 制御ユーティリティ・コマンド
    - DUMP\_TNSNAMES コマンド, B-15
    - LOAD\_TNSNAMES, B-16, B-19
    - REORDER\_NS, B-20

- Oracle Names の NAMES\_DID 表, B-13
- Oracle Net Manager
  - Oracle Names Server の検出, B-20
  - tnsnames.ora ファイルの Oracle Names へのロード, B-16, B-19
- 移行
  - ckreg.ora ファイルを cktop.ora ファイルへ, B-20
  - ROSFILES, B-17
  - ROSFILES の tnsnames.ora ファイルへの移行, B-18
  - ROSFILES のデータベースの Oracle Names 表への移行, B-17
  - データベースを使用する Oracle Names バージョン 2, B-12
- 共存問題, B-6
- 動的検出オプションのある Oracle Names バージョン 2, B-14
- リリース 1 (9.0.1) へのアップグレード, B-12
- Oracle Names 制御ユーティリティ
  - コマンド
    - LOAD\_TNSNAMES, B-16, B-19
    - REORDER\_NS, B-20
- Oracle Names の NAMES\_DOM 表, B-13
- Oracle Names の namesini.sql スクリプト, B-21
- Oracle Names の namesupg.sql スクリプト, B-12, B-15, B-18, B-21
- Oracle Names の NMO\_INFORMATION 表, B-14
- Oracle Names の動的検出オプション
  - 移行, B-14
- Oracle Net
  - 移行またはアップグレード, 5-46
- Oracle Net Services
  - FAILOVER ネットワーク・パラメータ, B-7
  - LOAD\_BALANCE ネットワーク・パラメータ, B-7
  - Oracle Enterprise Manager のある listener.ora ファイル, B-11
- Oracle Net Manager
  - Net8 Client 8.0 と互換性のあるオプションを使用するオプション, B-7
  - Oracle8i 互換識別子オプションの使用, B-7
  - SERVICE\_NAME パラメータ, B-4
  - SOURCE\_ROUTE パラメータ, B-7
  - アップグレード, 4-20
  - 共存問題, B-4, B-6
    - Oracle Net Manager を使用, B-7
    - Oracle8 クライアント, B-5

- Oracle8 データベース, B-6
- SERVICE\_NAME ネットワーク・パラメータの使用, B-4
- SID ネットワーク・パラメータの使用, B-4
- サード・パーティ製アプリケーション, B-5
- サポートされていない機能
  - Identix 認証, B-2
  - LU6.2, B-3
  - Net8 OPEN, B-2
  - Novell Directory Services (NDS) 外部ネーミング, B-2
  - Novell Directory Services (NDS) 認証, B-2
  - protocol.ora ファイル, B-3
  - SecurID 認証, B-2
  - SPX, B-3
  - 事前生成済専用サーバー, B-3
- Oracle OLAP
  - 互換性, 5-15
  - 相互運用性, 5-15
- Oracle Replication
  - ダウングレード, 7-29
- Oracle Spatial
  - アップグレード, 3-26
- Oracle Syndication Server
  - 使用不可, 5-16
- Oracle Text
  - ConText からの移行, D-5
  - アップグレード, 3-26
- Oracle Ultra Search
  - アップグレード, 3-28
- Oracle Universal Installer, 2-4
- Oracle Visual Information Retrieval
  - アップグレード, 3-26
- Oracle Workspace Manager
  - ダウングレード, 7-31
- ORACLE\_HOME 初期化パラメータ, B-11
- ORACLE\_TRACE\_ENABLE 初期化パラメータ
  - 互換性, A-6
- Oracle8i 互換識別子オプションの使用, B-7
- Oracle9i
  - 初期化パラメータへの変更, A-2
  - 新機能
    - アップグレード後の追加, 4-9
  - 静的データ・ディクショナリ・ビューの変更, A-14
  - 動的パフォーマンス・ビューの変更, A-21
- Oracle9i Real Application Clusters
  - アップグレード, 3-9

- 互換性の要件, 5-37
- ダウングレード, 7-29
- Oracle システム識別子、リスナーの構成, B-11
- Oracle プリコンパイラ
  - 「プリコンパイラ」を参照
- Oracle ホーム
  - 複数, 1-9
- ORADIM
  - アップグレード, 3-15
  - ダウングレード, 7-26
- ORLON コール
  - 廃止, 5-19
- OUTLN ユーザー
  - 回避, 3-12
- OWMD901.PLB スクリプト, 7-31

## P

---

- PARALLEL 句
  - DEGREE キーワード, 5-37
  - INSTANCES キーワードの削除, 5-37
- PDML ITL 不変
  - ダウングレード, 7-11
- PFILE 移行ユーティリティ・オプション, D-10
- PL/SQL
  - PLSQL\_V2\_COMPATIBILITY 初期化パラメータ, 5-23
  - 下位互換性, 5-21
  - 互換性, 5-22
  - 相互運用性, 5-22
  - ダウングレードのための非互換性の削除, 7-18
  - 統合された SQL 分析, 5-22
  - ファンクション
    - 非サポート, 5-34
  - 変数
    - NCHAR および NLS, 5-31
- Pro\*ADA
  - SQL\*Module for ADA へのアップグレード, 5-21
- Pro\*C/C++
  - SYSDBA 権限との接続, 5-21
- Pro\*COBOL
  - SYSDBA 権限との接続, 5-21
- PROPS\$ ビュー
  - NCHAR および NLS, 5-31
- protocol.ora ファイル, B-3

## R

---

### RAW

- 最大サイズ
- 変更, 5-17

### REAL キーワード

- 動作の違い, 5-25

### Recovery Manager

- カタログの正規化, 4-13
- 互換性, 5-41
- コマンド
- 互換性, 5-41
- バックアップ管理, 5-41

### REMOTE\_LOGIN\_PASSWORDFILE 初期化パラメータ, 3-19

- アップグレード, D-29

### REORDER\_NS コマンド, B-20

### ROLLBACK FORCE コマンド, D-5

### rosbild.sql スクリプト, B-17

### ROSFILES

- nmbild.sql スクリプト, B-17
- nmcgrnt.sql スクリプト, B-17
- rosbild.sql スクリプト, B-17
- rosgmnt.sql スクリプト, B-17

### rosgmnt.sql スクリプト, B-17

### ROWID

- DBMS\_ROWID 互換性パッケージ, D-34
- Oracle7 形式からの変換, D-34
- 例, D-36
- 移行, D-33
- 拡張, D-38
- 互換性, 5-31
- クライアント・アクセス, D-38
- スナップショット・リフレッシュ, D-37
- 制限付き, D-38

## S

---

### SAVEPOINT キーワード

- 動作の違い, 5-25

### SecurID 認証, B-2

### SERIALIZABLE 初期化パラメータ

- 互換性, A-6

### Server Manager

- Oracle9i での非サポート, 4-12, C-1
- SQL\*Plus との違い
- COMMIT コマンド, C-14

### CREATE LIBRARY コマンド, C-13

### CREATE TYPE コマンド, C-13

### アンバサンド, C-12

### 起動, C-2

### 空白行, C-10

### 構文, C-7

### コマンド, C-3

### コメント, C-8

### ハイフン継続文字, C-11

### スクリプトの SQL\*Plus への移行, C-1

### SERVICE\_NAMES 初期化パラメータ, B-8

### SERVICE\_NAME パラメータ, B-4

### SET COMPATIBILITY コマンド

### SQL\*Plus スクリプト, 6-10

### SET\_SESSION\_LONGOPS プロシージャ

### 構文変更, 5-36

### SHARED\_POOL\_SIZE 初期化パラメータ

### 動作の変更点, 4-13

### パラレル実行の割当て, A-8

### SID\_NAME パラメータ, B-11

### SID ネットワーク・パラメータ, B-5

### SID、リスナーの構成, B-11

### SORT\_AREA\_SIZE 初期化パラメータ

### 互換性, A-6

### SORT\_DIRECT\_WRITES 初期化パラメータ

### 互換性, A-6

### SOURCE\_ROUTE パラメータ, B-7

### SPOOL 移行ユーティリティ・オプション, D-10

### SPX プロトコル, B-3

### SQL

### ダウングレードのための非互換性の削除, 7-18

### SQL\*Module

### for ADA, 5-21

### SQL\*Net

### Oracle Names リリース 1 (9.0.1) へのアップグレード, B-12

### Oracle Net Services へのアップグレード, 4-20

### Oracle Net Services への移行, B-8 ~ B-11

### listener.ora ファイル, B-11

### tnsnames.ora ファイル, B-10

### クライアントのソフトウェア, B-9

### 構成ファイル, B-9

### サーバーのソフトウェア, B-9

### サービス名およびインスタンス名の検証, B-8

### Oracle Net への移行, 5-46

### Oracle9i での使用, 6-3

### SID ネットワーク・パラメータ, B-5

- V1 から V2 へのアップグレード, 5-47
- 共存問題, B-4, B-6
  - Oracle9i データベース, B-4
  - SERVICE\_NAME ネットワーク・パラメータの使用, B-4
  - SID ネットワーク・パラメータの使用, B-5
- 再リンク, 6-3
- SQL\*Net での再リンク, 6-3
- SQL\*Plus
  - Server Manager からのスクリプトの移行, C-1
  - Server Manager との違い
    - COMMIT コマンド, C-14
    - CREATE LIBRARY コマンド, C-13
    - CREATE TYPE コマンド, C-13
    - アンバサンド, C-12
    - 起動, C-2
    - 空白行, C-10
    - 構文, C-7
    - コマンド, C-3
    - コメント, C-8
    - ハイフン継続文字, C-11
  - スクリプト
    - アップグレード, 6-10
- SQL コマンド
  - COMMIT FORCE, D-5
  - ROLLBACK FORCE, D-5
- STARTUP
  - 互換性, 5-25
- SUBSTR 演算子, 5-17
- SYSDBA
  - Pro\*C/C++ で接続, 5-21
  - Pro\*COBOL で接続, 5-21
- SYSTEM 表領域
  - 移行ユーティリティ, D-3
  - 領域
    - アップグレード用の領域不足, D-25
- SYSTEM ロールバック・セグメントの OPTIMAL 設定
  - アップグレード用, D-26
- SYS スキーマ
  - ユーザー作成オブジェクト, 5-28

## T

---

- tnsnames.ora ファイル
  - アップグレード, B-10
  - 移行, B-10

- パラメータ
  - FAILOVER, B-7
  - LOAD\_BALANCE, B-7
  - SERVICE\_NAME, B-4
  - SID, B-5
  - SOURCE\_ROUTE, B-7
- TO\_LABEL ファンクション
  - 非サポート, 5-34
- TRUE キーワード
  - 動作の違い, 5-25

## U

---

- UNDO 表領域
  - 削除, 7-8
- UROWID
  - 相互運用性, 5-31
- UTLCHN1.SQL スクリプト, 5-34
- UTLCONST.SQL スクリプト, 4-19
- UTLEXPT1.SQL スクリプト, 5-35
- UTLINCMP.SQL スクリプト, 7-4

## V

---

- VALUE\$ ビュー
  - NCHAR および NLS, 5-31
- VARCHAR2
  - 最大サイズ
  - 変更, 5-17
- VARRAY
  - LOB として格納
  - 互換性, 5-33

## X

---

- XMLD817.SQL スクリプト, 7-31
- XMLD901.SQL スクリプト, 7-30

## あ

---

- アーカイブ
  - エラー検出動作, 5-44
- アーカイブ REDO ログ
  - 互換性, 5-43
  - 再アーカイブ, 5-43
  - 分析
    - 他のデータベース, 5-44

- アーカイブ・ログの宛先のパラメータ
  - リリース 8.1 の新規, A-11
- アップグレード
  - ALTER DATABASE CONVERT 文, 3-21
  - AUDIT\_TRAIL 初期化パラメータ, D-26
  - Database Upgrade Assistant の使用, 3-5
  - listener.ora ファイル, 4-8
  - MIGRATE.BSQ スクリプト, D-13
  - MIGRATE ユーザー、回避, D-7
  - NCHAR 列, 4-11
  - Oracle Forms アプリケーション, 6-10
  - Oracle *interMedia*, 3-26
  - Oracle Spatial, 3-26
  - Oracle Text, 3-26
  - Oracle Ultra Search, 3-28
  - Oracle Visual Information Retrieval, 3-26
  - Oracle9i Real Application Clusters, 3-9
  - ORADIM, 3-15
  - OUTLN ユーザー、回避, 3-12
  - SQL\*Plus スクリプト, 6-10
  - SYSTEM 表領域, D-25
  - SYSTEM ロールバック・セグメントの OPTIMAL
    - 設定, D-26
  - 新しい管理手順, 4-10
  - アップグレード後, 4-1
  - アップグレード後の処理, 4-1
  - アプリケーション, 6-1
    - オプション, 6-6
    - 互換性規則, 6-4
    - 再リンク, 6-4
  - 一時表領域, D-25
  - オフライン・データ・ファイル, D-3
  - オフラインの表領域, D-6
  - キャラクタ・セット, 3-10
  - キュー表, 4-16
  - 後のチューニング, 4-21
  - 主キー・スナップショットへのアップグレード,
    - E-14
  - 初期化パラメータ, 3-17
  - スクリプト, 3-22
    - CATALOG.SQL, 3-23, D-3
    - CATPROC.SQL, 3-23, D-3
  - スタンバイ・データベース, 4-8
  - スナップショット・サイト, E-8
  - 制御ファイル, 3-14
  - 段階的, E-6
  - 中止, D-32
  - テスト, 2-9
  - テストの結果, 4-21
  - 統計表, 4-18
  - 特定のコンポーネント, 3-26
  - トラブルシューティング
    - ALTER DATABASE CONVERT 文, D-27
    - AUDIT\_TRAIL 初期化パラメータ, D-26
    - MULTIPLIER オプション, D-27
    - NOMOUNT データベース・スタート・モード,
      - D-28
    - OPTIMAL 設定, D-26
    - Oracle7 制御ファイル, D-28
    - SYSTEM 表領域, D-25
    - 移行ユーティリティのエラー・メッセージ,
      - D-15
    - 移行ユーティリティの実行, D-25
    - 一時表領域, D-25
    - データベース名の不一致, D-30
    - パスワード・ファイル, D-29
    - 不明変換ファイル, D-28
  - 排他パスワード・ファイル, D-29
  - バックアップ計画, 2-8
  - パラレル実行, 4-13
  - 必要とする機能, E-17
  - マスター・サイト, E-3
  - マテリアライズド・ビュー, 4-16
  - 読み込み専用表領域, D-3
  - リカバリ・カタログ, 4-17
  - レプリケーション, 3-10, D-5
  - ローリング・アップグレード, 1-11
  - アップグレードの準備, 2-2
  - アップグレード方法
    - Database Upgrade Assistant, 2-4
    - エクスポート / インポート, 2-6
    - 手動アップグレード, 2-6
    - 選択, 2-4
    - データのコピー, 2-8
  - アドバンスド・キューイング
    - 互換性, 5-35
    - 権限, 5-35
    - ルール・ベースのサブスクリプション, 5-36
  - 相互運用性, 5-35
    - 送信者の ID 列, 5-36
  - アプリケーション
    - OCI
      - 互換性, 5-19
      - 相互運用性, 5-19



## PL/SQL

- 互換性, 5-22
- 相互運用性, 5-22
- アップグレード, 6-1
  - オプション, 6-6
  - 互換性規則, 6-4
  - 再リンク規則, 6-4
- 開発
  - アップグレードにおける役割, 1-7
- クライアント / サーバー構成
  - アップグレード, 6-3
- 互換性, 5-17, 6-2
- 索引構成表
  - 互換性, 5-18
- 相互運用性, 5-17
- 物理 ROWID および UROWID, 5-18
- プリコンパイラ
  - 互換性, 5-21
  - 相互運用性, 5-21
- 古いサーバーでの実行, 6-5
- より新しいライブラリへのリンク, 6-6

## い

---

### 異機種間サービス

- エージェント
  - 互換性, 5-46
  - 相互運用性, 5-46
  - マルチスレッド, 5-46

### 移行

- ALTER DATABASE CONVERT 文
  - 再実行, D-30
- ALTER DATABASE OPEN RESETLOGS MIGRATE 文, 3-21
- NCHAR および NLS, 5-32
- Oracle Managed Files のファイル名, 4-3
- ROWID, D-33
- ROWID の互換性, D-38
- 異なるオペレーティング・システム, 3-9
- 制御ファイル, 3-16
- トラブルシューティング
  - データ・ファイルのバージョン整合性, D-31
- パラレル実行, 4-13
- 移行ユーティリティ
  - MULTIPLIER オプション, D-27

## SYSTEM 表領域に必要な領域, D-3

- エラー・メッセージ, D-15
- オプション
  - CHECK\_ONLY, D-9
  - DBNAME, D-9
  - MULTIPLIER, D-9
  - NEW\_DBNAME, D-9
  - NO\_SPACE\_CHECK, D-10
  - PFILE, D-10
  - SPOOL, D-10
- 概要, D-2
- コマンドライン・オプション, D-9
- 実行, D-10
  - UNIX オペレーティング・システム上, D-10
  - Windows プラットフォーム上, D-12
- メモリー要件, D-3
- 領域要件, D-3
- 一時表領域
  - 領域
    - アップグレード用の領域不足, D-25
- 一時ファイル
  - データ・ディクショナリ情報, 5-27
- インストール
  - Oracle9i ソフトウェア, 3-2, 8-4
- インポート・ユーティリティ
  - アップグレード要件, 8-3
  - データベースの移行, 8-1

## え

---

### エクスポート / インポート

- アップグレードされたデータベースへの影響, 2-7
- 互換性, 8-2
- 時間要件, 2-7
- 使用したアップグレード手順, 8-4
- スクリプト
  - CATEXP7.SQL, 8-2
- 非互換データ, 8-3
- メリット, 2-7
- メリットおよびデメリット, 2-6
- エクスポート・ユーティリティ
  - アップグレード要件, 8-3
  - データベースの移行, 8-1
- エンタープライズ・ユーザー管理
  - 相互運用性, 5-40

## お

---

大きいキーを持つ索引

削除, 7-10

大きいファイル

オペレーティング・システム依存性, 5-48

オブジェクト型

互換性, 5-14

相互運用性, 5-14

オプション

移行ユーティリティ用, D-9

削除, 1-11

オフライン・データ・ファイル

アップグレード, D-3

オペレーティング・システム

異なるオペレーティング・システムへの移行, 3-9

オラクル社が提供するアカウント

パスワードの変更, 4-2

## か

---

下位互換性

ROWID, D-38

外部表

削除, 7-9

各国語キャラクタ・セット

Oracle8i, 5-31

拡張 ROWID, D-38

型

NOT FINAL

ダウングレード, 7-18

発展

削除, 7-18

型進化

相互運用性, 5-32

環境変数

ORA\_NLS32, 5-32

ORA\_NLS33, 5-32, D-11

アップグレードに必要, 3-20

互換性

NCHAR および NLS, 5-32

環境変数 NCHAR および NLS

互換性, 5-32

環境変数 NLS\_LANG

互換性, 5-32

環境変数 NLS および NCHAR

互換性, 5-32

環境変数 ORA\_NLS32

互換性, 5-32

環境変数 ORA\_NLS33, D-11

互換性, 5-32

## き

---

キーワード

動作の違い, 5-25

機能

アップグレードが必要, E-17

新機能, 5-9

機能の互換性, 5-8

キャラクタ・セット

可変幅

CLOB および NCLOB, 5-29

データベースのアップグレード, 3-10

キュー表

アップグレード, 4-16

共有構造

相互運用性, 5-19

共有サーバー

実行要件, 5-46

## く

---

クライアント / サーバー構成, 1-8

グローバル・データベース名, B-11

## こ

---

互換性, 5-1

ALTER DATABASE RESET COMPATIBILITY 文,  
5-8

ANALYZE VALIDATE STRUCTURE 文, 5-19

COMPATIBLE 初期化パラメータ, 5-2

CREATE TABLE の動作変更, 5-35

LOB, 5-29

CLOB および NCLOB, 5-29

LOB 索引句, 5-30

LogMiner, 5-44

OCI, 5-19

スレッド・セーフティ, 5-19

リンク・ライン, 5-20

Oracle Managed Files, 5-14

Oracle OLAP, 5-15

PL/SQL, 5-22

- PLSQL\_V2\_COMPATIBILITY 初期化パラメータ, 5-23
- 統合された SQL 分析, 5-22
- ROWID, 5-31
- ROWID の問題, D-38
- STARTUP, 5-25
- VARRAY
  - LOB として格納, 5-33
- アーカイブ REDO ログ, 5-43
- アドバンスド・キューイング, 5-35
- アプリケーション, 5-17, 6-2
  - 索引構成表, 5-18
  - 物理 ROWID および UROWID, 5-18
- 異機種間サービス・エージェント, 5-46
- エクスポート / インポート, 8-2
- オブジェクト型, 5-14
- 環境変数 NLS および NCHAR, 5-32
- 機能のレベルの確認, 5-8
- 互換性レベル, 5-5
  - 必要とする機能, 5-9
- 最大サイズの変更
  - CHAR, 5-17
  - RAW, 5-17
  - VARCHAR2, 5-17
- 最適化, 5-36
- 自動セグメント領域で管理される表領域, 5-14
- 初期化パラメータ, A-5
- スキーマ・オブジェクト, 5-29
- スクリプト
  - UTLCHN1.SQL, 5-34
  - UTLEXPT1.SQL, 5-35
- スタンバイ・データベース, 5-42
- ダウングレード, 5-5
- ディクショナリ管理表領域, 5-14
- データ型, 5-29
- データ・ディクショナリ, 5-28
- データ・ファイル, 5-26
- データベースのリセット, 7-19
- 動的パフォーマンス・ビューの日付列, 5-30
- ネストした表, 5-33
- バックアップ, 5-41
- 非互換性の削除, 7-2
- 表領域, 5-26
- プリコンパイラ, 5-21
- マテリアライズド・ビュー, 5-45
- ユーザー定義データ型, 5-32
- リカバリ, 5-41

- レプリケーション, 5-45
- 互換性レベル
  - 調査, 7-2
  - リセット, 5-8
- コマンド
  - COMMIT FORCE, D-5
  - ROLLBACK FORCE, D-5
- コマンドライン
  - コマンドライン・オプション
  - 移行ユーティリティ, D-9
- コメント
  - Server Manager と SQL\*Plus との違い, C-8
- コレクション
  - コレクション列
  - ユーザー指定の記憶域, 5-33
  - コレクション・ロケータ
  - 互換性, 5-33

## さ

---

- サーバー・パラメータ・ファイル
  - 移行, 4-7
- サービス・ネーミング
  - Net8, 5-47
- 最適化
  - 互換性, 5-36
- 索引
  - ビットマップ, 4-19
  - ファンクション, 4-15
- 削除, 1-11
- サブタイプ
  - ダウングレード, 7-18

## し

---

- 事前生成済専用サーバー, B-3
- 自動セグメント領域で管理される表領域
  - 互換性のレベルの変更, 5-14
- 主キー
  - スナップショットのアップグレード, E-14
- 手動アップグレード
  - データベースの準備, 3-11, D-5
  - メリット, 2-6
- 使用不可にされた機能
  - Oracle Dynamic Services, 5-15
  - Oracle Syndication Server, 5-16
  - ディクショナリ管理表領域, 5-14

## 初期化パラメータ

- COMPATIBLE, 5-2
- LARGE\_POOL\_SIZE
  - パラレル実行の割当て, A-8
- Oracle9i での調整, 3-17, 4-10
- Oracle9i での変更, A-2
- REMOTE\_LOGIN\_PASSWORDFILE, 3-19
- SHARED\_POOL\_SIZE
  - パラレル実行の割当て, A-8
- アーカイブ・ログの宛先
  - 新規への切替え, A-11
- 互換性, A-5
  - DB\_BLOCK\_CHECKSUM, A-5
  - DB\_DOMAIN, A-8
  - DML\_LOCKS, A-7
  - JOB\_QUEUE\_PROCESSES, A-5
  - LOG\_CHECKPOINT\_TIMEOUT, A-7
  - O7\_DICTIONARY\_ACCESSIBILITY, A-7
  - ORACLE\_TRACE\_ENABLE, A-6
  - SERIALIZABLE, A-6
  - SORT\_AREA\_SIZE, A-6
  - SORT\_DIRECT\_WRITES, A-6
- 名前の変更, D-39
- 廃止, A-3
- 非推奨, A-2
- 変更, D-39

## 初期化パラメータへの変更, D-39

## 新機能

- アップグレード後の追加, 4-9
- 互換性のレベルを必要とする, 5-9

# す

---

## スキーマ・オブジェクト

- 互換性, 5-29

## スクリプト

- アップグレード, 3-22
- 再実行, 7-24
- ダウングレード, 7-23

## スタンバイ・データベース

- アップグレード, 4-8
- 互換性, 5-42

## スナップショット

- 主キーへのアップグレード, E-14
- リフレッシュ
  - 物理 ROWID, D-37

## スナップショット・サイト

- アップグレード, E-8
- スレッド・セーフティ
  - 互換性, 5-19

# せ

---

## 制御ファイル

- アップグレードのための名前の変更または削除, 3-14
- 移行のための名前の変更または削除, 3-16

## 制限付き ROWID, D-38

## 生成

- レプリケーション・サポート, E-5, E-13
- 静的データ・ディクショナリ・ビュー

- NULL を戻す可能性のある列, A-19

- Oracle9i での変更, A-14

- 廃止, 5-28, A-16

- 非推奨, A-14

- 変更, D-40

- 列の削除, A-17

- 列の名前の変更, A-17

## 静的データ・ディクショナリ・ビューの変更, D-40

- 接続
- Oracle Net でのロード・バランシング, 5-47

# そ

---

## 相互運用性, 5-1, 5-12

- OCI, 5-19

- Oracle7 クライアント, 5-20

- 共有構造, 5-19

- Oracle Managed Files, 5-14

- Oracle OLAP, 5-15

- PL/SQL, 5-22

- UROWID, 5-31

- アドバンスド・キューイング, 5-35

- アプリケーション, 5-17

- 異機種間サービス・エージェント, 5-46

- オブジェクト型, 5-14

- 型進化, 5-32

- ディクショナリ管理表領域, 5-14

- ネイティブ動的 SQL, 5-34

- プリコンパイラ, 5-21

- ユーザー定義データ型, 5-33

## た

---

### ダウングレード

- CATALOG.SQL, 7-29
- CATPARR.SQL, 7-29
- CATPROC.SQL, 7-29
- Java, 7-30, 7-31
- Messaging Gateway, 7-31
- Oracle Replication, 7-29
- Oracle Workspace Manager, 7-31
- Oracle9i Real Application Clusters, 7-29
- ORADIM, 7-26
- スクリプト, 7-23
  - JVMD817.SQL, 7-30
  - JVMD901.SQL, 7-30
  - MGWD901.SQL, 7-31
  - OWMD901.PLB, 7-31
  - XMLD817.SQL, 7-31
  - XMLD901.SQL, 7-30
- 再実行, 7-24
- データベース互換性のリセット, 7-19
- 手順, 7-22
- 非互換性の削除, 7-2
  - LOB リテンション, 7-7
  - LOB を含む自動セグメント領域で管理される表領域, 7-7
  - LOB を含むパーティション索引構成表, 7-12
  - NOT FINAL 型, 7-18
  - PDML ITL 不変, 7-11
  - SQL および PL/SQL, 7-18
  - UNDO 表領域, 7-8
  - 大きいキーを持つ索引, 7-10
  - 外部表, 7-9
  - 索引構成表のビットマップ 2 次索引, 7-9
  - サブタイプ, 7-18
  - 自動セグメント領域で管理される表領域, 7-8
  - パイプライン表関数, 7-19
  - ハッシュ・パーティション索引構成表, 7-11
  - 発展型, 7-18
  - パラレル表関数, 7-19
  - ビュー制約, 7-19
  - ユーザー定義集計関数, 7-18
- リリース 1 (9.0.1) のパーティション化メソッド, 7-10
- リリース 2 (9.2) の Streams, 7-5
- リリース 2 (9.2) のサブパーティション・テンプレート, 7-6

リリース 2 (9.2) のデフォルト・パーティション, 7-5

リリース 2 (9.2) のパーティション化メソッド, 7-5

ビュー制約, 7-19

段階的アップグレード, E-6

## ち

---

### チューニング

アップグレード後, 4-21

## て

---

### 定義

「用語」を参照

### ディクショナリ管理表領域

互換性, 5-14

使用不可, 5-14

相互運用性, 5-14

### データ型

互換性, 5-29

### データ・ディクショナリ

互換性, 5-28

保護, 5-28

### データのコピー, 2-8

### データ・ファイル

アップグレードにおけるオフライン, D-3

互換性, 5-26

### データベース

アップグレード結果のテスト, 4-21

アップグレード後のチューニング, 4-21

アップグレードのためのバックアップ, D-14

ダウングレード, 7-22

非互換性の削除後オープンできない, 5-8

### データベース移行

概要, 1-3

定義, 1-3

用語, 1-2

レプリケーション概要, E-2

### データベース管理者

アップグレードにおける役割, 1-7

### データベースのアップグレード

ALTER DATABASE CONVERT 文, D-3

ALTER DATABASE OPEN RESETLOGS MIGRATE 文, D-3

Database Upgrade Assistant の使用, 2-4

- アップグレード方法の選択, 2-4
- アプリケーション開発者の役割, 1-7
- 移行ユーティリティの概要, D-2
- エクスポート / インポートの使用, 8-4
- 手動, 3-9
  - データベースの準備, 3-11, D-5
- 手動アップグレードの実行, 2-4
- 準備, 2-2
- データベース管理者の役割, 1-7
- 手順の概要, 1-4
- ブロック・サイズ最小値, D-4
- 分散データベースの考慮点, D-5
- ロールバック・セグメント, D-3
- データベースの移行
  - エクスポート / インポートの使用, 8-1
- テスト
  - INTO 句, 2-11
  - アップグレード結果, 4-21
  - アップグレード処理, 2-12
  - アップグレード済のテスト・データベース, 2-12
  - アップグレード前後, 2-11
  - アップグレードに対するアプリケーション, 2-12
  - アップグレードの機能, 2-9
  - アップグレードの最小限度, 2-9
  - アップグレードの統合, 2-10
  - アップグレードのパフォーマンス, 2-10
  - アップグレードのボリューム / ロード・ストレス, 2-11
  - 計画の作成, 2-9
  - 実行計画, 2-11

## と

---

- 透過的アプリケーション・フェイルオーバー (TAF)
  - listener.ora の GLOBAL\_DBNAME ネットワーク・パラメータ, B-11
- 統計表
  - アップグレード, 4-18
- 動的パフォーマンス・ビュー
  - Oracle9i での変更, A-21
  - 廃止, A-23
  - 非推奨, A-21
  - 日付列
    - 互換性, 5-30
  - 列の削除, A-25
  - 列の名前の変更, A-24

- トラブルシューティング
  - アップグレード
    - ALTER DATABASE CONVERT 文, D-27
    - AUDIT\_TRAIL 初期化パラメータ, D-26
    - MULTIPLIER オプション, D-27
    - NOMOUNT データベース・スタート・モード, D-28
    - OPTIMAL 設定, D-26
    - Oracle7 制御ファイル, D-28
    - SYSTEM 表領域, D-25
    - 移行ユーティリティのエラー・メッセージ, D-15
    - 移行ユーティリティの実行, D-25
    - 一時表領域, D-25
    - データベース名の不一致, D-30
    - パスワード・ファイル, D-29
    - 不明変換ファイル, D-28
  - 移行
    - データ・ファイルのバージョン整合性, D-31
    - 非互換性の削除後データベースをオープンできない, 5-8
  - トランザクション
    - ペンディング, D-5

## な

---

- 名前が変更された初期化パラメータ, D-39

## ね

---

- ネイティブ動的 SQL
  - 相互運用性, 5-34
- ネストした表
  - 互換性, 5-33

## は

---

- パーティション・ビュー
  - パーティション表への移行, 4-19
- 廃止
  - レプリケーション・プロシージャ, E-19
  - 廃止された初期化パラメータ, A-3
  - 廃止された静的データ・ディクショナリ・ビュー, A-16
  - 廃止された動的パフォーマンス・ビュー, A-23
  - 廃止されたレプリケーション・プロシージャ, E-19

パイプライン表関数

削除, 7-19

パスワード管理

Oracle8iに必要なアプリケーションの変更, 5-39

相互運用性, 5-40

パスワードの期限切れ, 5-39

パスワードの変更

オラクル社が提供するアカウント, 4-2

パスワード・ファイル

アップグレード

排他設定, D-29

バックアップ

EBU, 5-41

Recovery Manager および EBU, 5-41

アップグレード後, 4-2

アップグレード前, D-14

計画の準備, 2-8

互換性, 5-41

ハッシュ・パーティション索引構成表

ダウングレード, 7-11

パラメータ

移行ユーティリティ用

「コマンドライン・オプション」を参照

パラレル実行

問題の回避, 4-13

ラージ・プールからの割当て, A-8

パラレル表関数

削除, 7-19

## ひ

---

非互換性

削除, 7-2

非互換性の削除後データベースをオープンできない,  
5-8

非互換性の識別

UTLINCMP.SQL, 7-4

非推奨の初期化パラメータ, A-2

非推奨の静的データ・ディクショナリ・ビュー, A-14

非推奨の動的パフォーマンス・ビュー, A-21

日付の制約

無効チェック, 4-19

ビットマップ 2 次索引

索引構成表からの削除, 7-9

ビットマップ索引

無効性, 5-29

アップグレード時, 4-19

表領域

オフライン表領域のアップグレード, D-6

互換性, 5-26

自動セグメント領域の管理

削除, 7-8

## ふ

---

ファイル名

正規化, 4-13

ファスト・スタート・パラレル・リカバリ

互換性, 5-42

ファスト・スタート・ロールバック

互換性, 5-42

ファンクション索引

無効性

アップグレード時, 4-15

複数バージョン, 1-8

プリコンパイラ

PL/SQL 下位互換性, 5-21

アプリケーション

Oracle9i を使用するための変更, 6-9

アップグレード選択肢, 6-6

アプリケーションの Oracle9i へのアップグレード,  
6-3

互換性, 5-21

相互運用性, 5-21

プロキシ・コピー

要件, 5-45

プロシージャ・オプション

アップグレードに必要, D-5

ブロック

ブロック・サイズ

DB\_BLOCK\_SIZE 初期化パラメータ, D-4

アップグレード最小値, D-4

文

ALTER DATABASE RESET COMPATIBILITY, 5-8

ALTER DATABASE RESETLOGS, D-5

分散データベース

アップグレードの準備, D-5

## へ

---

ペンディング・トランザクション

アップグレード, D-5

## ま

---

マスター・サイト  
    アップグレード, E-3  
    段階的アップグレード, E-10  
マテリアライズド・ビュー  
    アップグレード, 4-16  
    互換性, 5-45

## め

---

メモリー要件  
    移行ユーティリティ用, D-3

## ゆ

---

ユーザー作成オブジェクト  
    SYS スキーマ, 5-28  
ユーザー定義集計関数  
    削除, 7-18  
ユーザー定義データ型  
    新しい書式, 5-33  
    互換性, 5-32  
    相互運用性, 5-33

## よ

---

要件  
    インポート・ユーティリティ, 8-3  
    エクスポート・ユーティリティ, 8-3  
用語  
    データベース移行, 1-2  
用語集  
    「用語」を参照  
読み込み専用表領域  
    アップグレード, D-3

## り

---

リカバリ  
    互換性, 5-41  
リカバリ・カタログ  
    Recovery Manager との互換性, 5-41  
    アップグレード, 4-17  
リスナー  
    Oracle Enterprise Manager の構成, B-11  
    Oracle システム識別子, B-11

SID, B-11  
    グローバル・データベース名, B-11

### 領域要件

    移行ユーティリティ用, D-3

### リリース

    定義, 1-2  
    複数, 1-8

リリース 1 (9.0.1) のパーティション化メソッド  
    ダウングレード, 7-10

リリース 2 (9.2) の Streams  
    ダウングレード, 7-5

リリース 2 (9.2) のサブパーティション・テンプレート  
    ダウングレード, 7-6

リリース 2 (9.2) のデフォルト・パーティション  
    ダウングレード, 7-5

リリース 2 (9.2) のパーティション化メソッド  
    ダウングレード, 7-5

## れ

---

### レプリケーション

    アップグレード, 3-10, D-5  
    互換性, 5-45  
    データベース移行概要, E-2  
    廃止されたプロシージャ, E-19

## ろ

---

### ロード・バランシング

    Net8, 5-47

### ロールバック・セグメント

    アップグレード, D-3

### ロック

    DML ロック制限、DML\_LOCKS, A-7

## わ

---

### ワード・サイズ

    変更, 1-10, 4-21