

Oracle9i

ユーザー管理バックアップおよびリカバリ・ガイド

リリース 2 (9.2)

2002 年 7 月

部品番号 : J06271-01

ORACLE®

Oracle9i ユーザー管理バックアップおよびリカバリ・ガイド, リリース 2 (9.2)

部品番号 : J06271-01

原本名 : Oracle9i User-Managed Backup and Recovery Guide, Release 2 (9.2)

原本部品番号 : A96572-01

原本著者 : Lance Ashdown

原本協力者 : Tammy Bednar, Wei Hu, Vikram Joshi, Bill Lee, Yunrui Li, Gary Ngai, Ron Obermarck, Alok Pareek, Vinay Srihari, Janet Stern, Mike Stewart, Kothanda Umamageswaran, Valarie Moore

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を用途として開発されておりません。オラクル社製品を上述のようなアプリケーションに使用することについての安全確保は、顧客各位の責任と費用により行ってください。万が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	vii
ユーザー管理のバックアップとリカバリの新機能	xv
1 ユーザー管理のバックアップとリカバリの概要	
ユーザー管理のバックアップとリカバリ	1-2
ユーザー管理のバックアップとリカバリが必要な理由	1-2
ユーザー管理のバックアップとリカバリの概要	1-3
ユーザー管理バックアップ	1-3
ユーザー管理のリストアおよびリカバリ	1-6
2 ユーザー管理バックアップの作成	
V\$ ビューの問合せによるバックアップ情報の取得	2-2
バックアップ前のデータベース・ファイルの列挙	2-2
オンライン表領域バックアップのデータ・ファイルのステータスの判別	2-3
データベース全体のユーザー管理バックアップの作成	2-4
データベース全体の一貫性のあるバックアップの作成	2-5
オフラインの表領域およびデータ・ファイルのユーザー管理バックアップの作成	2-6
オンラインの表領域およびデータ・ファイルのユーザー管理バックアップの作成	2-8
オンラインの読取り / 書込み表領域のユーザー管理バックアップの作成	2-8
オンラインの読取り / 書込み表領域の複数のユーザー管理バックアップの作成	2-10
インスタンス障害または SHUTDOWN ABORT 後のバックアップの終了	2-12
読取り専用表領域のユーザー管理バックアップの作成	2-15
UNDO 表領域のユーザー管理バックアップの作成	2-17

SUSPEND モードでのユーザー管理バックアップの作成	2-17
一時停止 / 再開機能	2-17
一時停止中のデータベースのバックアップの作成	2-18
制御ファイルのユーザー管理バックアップの作成	2-20
制御ファイルのバイナリ・ファイルへのバックアップ	2-20
制御ファイルのトレース・ファイルへのバックアップ	2-21
アーカイブ REDO ログのユーザー管理バックアップの作成	2-24
RAW デバイスへのユーザー管理バックアップの作成	2-24
UNIX での RAW デバイスへのバックアップ	2-25
Windows NT での RAW デバイスへのバックアップ	2-27
ユーザー管理バックアップの検証	2-29
バックアップのリストアのテスト	2-29
DBVERIFY ユーティリティの使用	2-30
エクスポートを使用した論理バックアップの作成	2-31
エクスポートの使用	2-31
インポートの使用	2-32
その他の Oracle ファイルのユーザー管理バックアップの作成	2-33

3 ユーザー管理のリストア操作の実行

ユーザー管理のリストア操作	3-2
リストアのシナリオで使用する情報の記録	3-3
データ・ファイル、制御ファイルおよびオンライン REDO ログの場所の記録	3-4
アーカイブ REDO ログの場所の記録	3-4
バックアップ・ファイルの場所の記録	3-5
リカバリが必要なデータ・ファイルの判別	3-6
データ・ファイルのリストア	3-7
バックアップが利用できない場合のデータ・ファイルの再作成	3-8
制御ファイルのリストアおよび再作成	3-9
多重制御ファイルの一部が損害を受けた場合	3-10
すべての多重制御ファイルが損害を受けているが、バックアップが使用できる場合	3-12
現行の制御ファイルおよびバックアップ制御ファイルのすべてが損害を受けた場合	3-15
アーカイブ REDO ログのリストア	3-17

4 ユーザー管理のメディア・リカバリの実行

ユーザー管理のメディア・リカバリの実行概要	4-2
ユーザー管理のリカバリを実行する前提条件	4-2
RECOVER コマンドを使用したログの自動適用	4-3
アーカイブ・ログがデフォルトの場所にある場合のリカバリ	4-6
アーカイブ・ログがデフォルト以外の場所にある場合のリカバリ	4-7
アーカイブ・ログの出力先の再設定	4-7
アーカイブ・ログの出力先の変更	4-8
REDO ログが適用できない場合の対応	4-9
ユーザー管理の完全メディア・リカバリの実行	4-10
クローズしているデータベースのリカバリの実行	4-10
オープン状態のデータベースでのデータ・ファイルのリカバリの実行	4-14
ユーザー管理の不完全メディア・リカバリの実行	4-17
不完全リカバリの準備	4-17
不完全リカバリ実行前のデータ・ファイルのリストア	4-18
取消ベースの不完全リカバリの実行	4-20
時間ベースの不完全リカバリの実行	4-22
変更ベースの不完全リカバリの実行	4-23
NOARCHIVELOG モードでのデータベースのリカバリ	4-24
デフォルトの場所へのデータベースのリストア	4-24
新しい場所へのデータベースのリストア	4-25
パラレルのメディア・リカバリの実行	4-27
ユーザー管理のメディア・リカバリ後のデータベースのオープン	4-28
RESETLOGS 操作	4-28
オンライン REDO ログをリセットするかの判断	4-30
RESETLOGS 操作後の処理	4-32
RESETLOGS 前に作成したバックアップのリカバリ	4-33
ユーザー管理のメディア・リカバリの中断	4-36
ユーザー管理のメディア・リカバリの制限	4-37
リカバリ不能の表および索引のユーザー管理のリカバリ	4-37
現行ではない制御ファイルを使用した、読取り専用表領域のユーザー管理のリカバリ	4-38

5 ユーザー管理のメディア・リカバリのトラブルシューティング

ユーザー管理のメディア・リカバリの問題	5-2
メディア・リカバリの問題の調査: フェーズ 1	5-4
ブロックを破損させない修正の試行: フェーズ 2	5-5
リカバリによるブロックの破損を許可するかどうかの決定: フェーズ 3	5-7
リカバリによるブロックの破損の許可: フェーズ 4	5-9
試行リカバリの実行	5-9
試行リカバリ	5-9
試行リカバリの仕組み	5-10
RECOVER ... TEST 文の実行	5-11

6 ユーザー管理のメディア・リカバリのシナリオ

データ・ファイルが損害を受けた後のリカバリ: 使用例	6-2
NOARCHIVELOG モードでデータ・ファイルが損害を受けた場合	6-2
ARCHIVELOG モードでデータ・ファイルが損害を受けた場合	6-3
データ・ファイルの追加を伴うリカバリ: 使用例	6-3
トランスポータブル表領域のリカバリ: 使用例	6-5
オンライン REDO ログ・ファイルが損害を受けた後のリカバリ: 使用例	6-6
多重オンライン REDO ログ・グループの一部のメンバーが損害を受けた後のリカバリ	6-7
オンライン REDO ログ・グループのすべてのメンバーが損害を受けた後のリカバリ	6-8
アーカイブ REDO ログ・ファイルが損害を受けた後のリカバリ: 使用例	6-13
ユーザー・エラーからのリカバリ: 使用例	6-14
分散環境でのメディア・リカバリの実行: 使用例	6-15
時間ベースおよび変更ベースの分散データベースのリカバリの調整	6-15

7 ユーザー管理の TSPITR の実行

ユーザー管理の表領域の Point-in-Time リカバリ (TSPITR) の概要	7-2
TSPITR の用語	7-2
TSPITR の方法	7-3
表領域の Point-in-Time リカバリの準備 : 基本手順	7-4
手順 1: TSPITR の要件の検討	7-5
手順 2: リカバリ・セットおよび補助セットの表領域内のすべてのファイルの識別	7-5
手順 3: 失われるオブジェクトの判別	7-6
手順 4: 補助インスタンスへの接続方法の選択	7-7
手順 5: 補助インスタンスの Oracle パスワード・ファイルの作成	7-7
手順 6: 補助インスタンスの初期化パラメータ・ファイルの作成	7-7
補助データベースのリストアおよびリカバリ : 基本手順	7-9
同じホスト上での補助データベースのリストアおよびリカバリ	7-10
同じパス名を使用する別のホスト上での補助データベースのリストア	7-12
異なるパス名を使用する別のホスト上での補助データベースのリストア	7-13
トランSPORTABLE表領域を使用した TSPITR の実行	7-14
手順 1: 補助データベースからの表領域の取出し	7-14
手順 2: 表領域のプライマリ・データベースへのトランSPORT	7-15
パーティション表の部分 TSPITR の実行	7-16
手順 1: プライマリ・データベースにおける個々のパーティションをリカバリするための 表の作成	7-16
手順 2: リカバリするパーティションの索引の削除	7-17
手順 3: スタンドアロンの表とパーティションとの交換	7-17
手順 4: リカバリ・セットの表領域の削除	7-17
手順 5: 補助データベースでの表の作成	7-17
手順 6: リカバリするパーティションの索引の削除	7-17
手順 7: 補助データベースのパーティションとスタンドアロン表の交換	7-18
手順 8: リカバリ・セットの表領域のトランSPORT	7-18
手順 9: プライマリ・データベースのパーティションとスタンドアロン表の交換	7-18
手順 10: プライマリ・データベースのリカバリされた表領域のバックアップ	7-18

パーティションが削除された場合のパーティション表の部分 TSPITR の実行	7-19
手順 1: 削除されたパーティションの下限および上限の検出	7-19
手順 2: 一時的な表の作成	7-19
手順 3: パーティション表のレコードの削除	7-19
手順 4: リカバリ・セットの表領域の削除	7-20
手順 5: 補助データベースでの表の作成	7-20
手順 6: リカバリするパーティションの索引の削除	7-20
手順 7: スタンドアロンの表とパーティションとの交換	7-20
手順 8: リカバリ・セットの表領域のトランスポート	7-20
手順 9: パーティション表へのスタンドアロン表の挿入	7-21
手順 10: プライマリ・データベースのリカバリされた表領域のバックアップ	7-21
パーティションが分割されている場合のパーティション表の TSPITR の実行	7-22
手順 1: プライマリ・データベースの分割された 2 つのパーティションのうち下位にある パーティションの削除	7-22
手順 2: パーティション表の部分 TSPITR の実行と同じ手順の実行	7-22

索引

はじめに

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、次の作業を実行するデータベース管理者を対象としています。

- オペレーティング・システム・ユーティリティを使用してデータベース・ファイルのバックアップおよびリストアを実行し、SQL*Plus の RECOVER コマンドを使用してリカバリを実行する。
- バックアップおよびリカバリのソリューションとして Recovery Manager を使用するが、いくつかのファイルはユーザー管理方式でバックアップすることを選択する。

このマニュアルを使用するには、下記のマニュアルに記載されている、リレーショナル・データベースの概念およびデータベース管理の基本について理解する必要があります。

- 『Oracle9i データベース概要』
- 『Oracle9i バックアップおよびリカバリ概要』
- 『Oracle9i データベース管理者ガイド』
- また、Oracle を実行しているオペレーティング・システム環境についても習熟している必要があります。

このマニュアルの構成

このマニュアルの構成は、次のとおりです。

「ユーザー管理のバックアップとリカバリの新機能」

ユーザー管理のバックアップとリカバリの新機能および機能強化について説明します。

第1章「ユーザー管理のバックアップとリカバリの概要」

ユーザー管理のバックアップとリカバリ方式の目的および基本機能について説明します。

第2章「ユーザー管理バックアップの作成」

オペレーティング・システム・コマンドを使用して制御ファイル、データ・ファイルおよびアーカイブ REDO ログをバックアップする方法を説明します。

第3章「ユーザー管理のリストア操作の実行」

オペレーティング・システム・コマンドを使用して制御ファイル、データベース・ファイルおよびアーカイブ REDO ログをリストアする方法を説明します。

第4章「ユーザー管理のメディア・リカバリの実行」

リストアしたデータ・ファイルでメディア・リカバリを実行するための SQL*Plus の RECOVER コマンドの使用方法を説明します。

第5章「ユーザー管理のメディア・リカバリのトラブルシューティング」

ユーザー管理のメディア・リカバリで発生する可能性のある問題のトラブルシューティングの方法を説明します。

第6章「ユーザー管理のメディア・リカバリのシナリオ」

ユーザー管理のリストアおよびリカバリの基本的なシナリオについて説明します。

第7章「ユーザー管理の TSPITR の実行」

表領域を、データベースの残りの部分とは異なる時点までリカバリする方法を説明します。

関連文書

詳細は、次の Oracle マニュアルを参照してください。

- バックアップおよびリカバリの概要については、『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。
- Recovery Manager の使用方法については、『Oracle9i Recovery Manager ユーザーズ・ガイド』を参照してください。

マニュアル・セットに含まれるマニュアルの多くでは、Oracle のインストール時にデフォルトで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。これらのスキーマがどのように作成されているか、およびその使用方法については、『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、インストレーション・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/>

OTN-J のユーザー名とパスワードを取得済みであれば、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

このマニュアル・セットの本文とコード例に使用されている表記規則について説明します。

- 本文の表記規則
- コード例の表記規則
- Windows オペレーティング・システムの表記規則

本文の表記規則

本文中には、特別な用語が一目でわかるように様々な表記規則が使用されています。次の表は、本文の表記規則と使用例を示しています。

表記規則	意味	例
太字	太字は、本文中に定義されている用語または用語集に含まれている用語、あるいはその両方を示します。	この句を指定する場合は、 索引構成表 を作成します。
固定幅フォントの大文字	固定幅フォントの大文字は、システムにより指定される要素を示します。この要素には、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、パッケージとメソッドの他、システム指定の列名、データベース・オブジェクトと構造体、ユーザー名、およびロールがあります。	この句は NUMBER 列に対してのみ指定できます。 BACKUP コマンドを使用すると、データベースのバックアップを作成できます。 USER_TABLES データ・ディクショナリ・ビューの 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サンプルのユーザー指定要素を示します。この要素には、コンピュータ名とデータベース名、ネット・サービス名、接続識別子の他、ユーザー指定のデータベース・オブジェクトと構造体、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があります。 注意： 一部のプログラム要素には、大文字と小文字の両方が使用されます。この場合は記載されているとおりに入力してください。	sqlplus と入力して SQL*Plus をオープンします。 パスワードは orapwd ファイルに指定されています。 /disk1/oracle/dbs ディレクトリ内で、データ・ファイルと制御ファイルのバックアップを作成します。 department_id、department_name および location_id の各列は、hr.departments 表にあります。 初期化パラメータ QUERY_REWRITE_ENABLED を true に設定します。 oe ユーザーで接続します。 これらのメソッドは、JRepUtil クラスで実装されます。

表記規則	意味	例
固定幅フォントの 小文字の イタリック	固定幅フォントの小文字のイタリックは、プレースホルダまたは変数を示します。	<i>parallel_clause</i> を指定できます。 <i>Uold_release</i> .SQL を実行します。この場合、 <i>old_release</i> はアップグレード前にインストールしていたリリースです。

コード例の表記規則

コード例は、SQL、PL/SQL、SQL*Plus またはその他のコマンドラインを示します。次のように、固定幅フォントで、通常の本文とは区別して記載されています。

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

次の表は、コード例の記載上の表記規則と使用例を示しています。

表記規則	意味	例
[]	大カッコで囲まれている項目は、1 つ以上のオプション項目を示します。大カッコ自体は入力しないでください。	DECIMAL (<i>digits</i> [, <i>precision</i>])
{ }	中カッコで囲まれている項目は、そのうちの 1 つのみが必要であることを示します。中カッコ自体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複数の選択肢を区切るために使用します。オプションのうち 1 つを入力します。縦線自体は入力しないでください。	{ENABLE DISABLE} [COMPRESS NOCOMPRESS]
...	<p>水平の省略記号は、次のいずれかを示します。</p> <ul style="list-style-type: none"> ■ 例に直接関係のないコード部分が省略されていること。 ■ コードの一部が繰り返し可能なこと。 	<pre>CREATE TABLE ... AS subquery;</pre> <pre>SELECT col1, col2, ... , coln FROM employees;</pre>
.	垂直の省略記号は、例に直接関係のない数行のコードが省略されていることを示します。	<pre>SQL> SELECT NAME FROM V\$DATAFILE; NAME ----- /fsl/dbs/tbs_01.dbf /fsl/dbs/tbs_02.dbf . . . /fsl/dbs/tbs_09.dbf 9 rows selected.</pre>

表記規則	意味	例
その他の表記	大カッコ、中カッコ、縦線および省略記号以外の記号は、表示されているとおりに入力してください。	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
イタリック	イタリックの文字は、特定の値を指定する必要があるプレースホルダまたは変数を示します。	CONNECT SYSTEM/system_password DB_NAME = database_name
大文字	大文字は、システムにより指定される要素を示します。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記載されています。大カッコで囲まれている場合を除き、記載されているとおりの順序とスペルで入力してください。ただし、この種の用語は大 / 小文字区別がないため、小文字でも入力できます。	SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;
小文字	小文字は、ユーザー指定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を示します。 注意： 一部のプログラム要素には、大文字と小文字の両方が使用されます。この場合は記載されているとおりに入力してください。	SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjjones IDENTIFIED BY ty3MU9;

Windows オペレーティング・システムの表記規則

次の表は、Windows オペレーティング・システムの表記規則と使用例を示しています。

表記規則	意味	例
「スタート」→を選択	プログラムの起動方法。	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。
ファイル名とディレクトリ名	ファイル名とディレクトリ名には、大 / 小文字区別はありません。特殊文字のうち、左山カッコ (<)、右山カッコ (>)、コロン (:)、二重引用符 (")、スラッシュ (/)、パイプ () およびハイフン (-) は使用できません。特殊文字の円記号 (¥) は、引用符で囲まれていても要素のセパレータとして扱われます。¥¥ で始まるファイル名は、Windows では汎用命名規則を使用しているものとみなされます。	c:¥winnt"¥"system32 は C:¥WINNT¥SYSTEM32 と同じです。
C:¥>	現行のハード・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・プロンプトで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反映されます。このマニュアルでは、コマンド・プロンプトと呼んでいます。	C:¥oracle¥oradata>
特殊文字	Windows コマンド・プロンプトでは、二重引用符 (") のエスケープ文字として特殊文字である円記号 (¥) が必要な場合があります。カッコと一重引用符 (') には、エスケープ文字は不要です。エスケープ文字と特殊文字の詳細は、Windows オペレーティング・システムのマニュアルを参照してください。	C:¥>exp scott/tiger TABLES=emp QUERY=¥"WHERE job='SALESMAN' and sal<1600¥" C:¥>imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept)
HOME_NAME	Oracle ホーム名を表します。ホーム名は英数字で 16 文字以内です。ホーム名に使用できる特殊文字は、アンダースコアのみです。	C:¥> net start OracleHOME_NAME_TNSListener

表記規則	意味	例
<code>ORACLE_HOME</code> と <code>ORACLE_BASE</code>	<p>Oracle8 リリース 8.0 以前では、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はデフォルトで次のいずれかの名前を使用して、トップレベルの <code>ORACLE_HOME</code> ディレクトリの下に置かれていました。</p> <ul style="list-style-type: none"> ■ Windows NT の場合は <code>C:\orant</code> ■ Windows 98 の場合は <code>C:\orawin98</code> <p>このリリースは Optimal Flexible Architecture (OFA) のガイドラインに従っています。すべてのサブディレクトリがトップレベルの <code>ORACLE_HOME</code> ディレクトリの下にあるとはかぎりません。 <code>ORACLE_BASE</code> というトップレベルのディレクトリがあり、これはデフォルトでは <code>C:\oracle</code> です。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最新の Oracle リリースをインストールすると、最初の Oracle ホーム・ディレクトリのデフォルト設定は <code>C:\oracle\orann</code> となります。この場合、<code>nn</code> は最新リリース番号です。Oracle ホーム・ディレクトリは、<code>ORACLE_BASE</code> の直下にあります。</p> <p>このマニュアルに記載されているディレクトリ・パスの例は、すべて OFA の表記規則に従っています。</p>	<code>%ORACLE_HOME%\rdbms\admin</code> ディレクトリにアクセスします。

ユーザー管理のバックアップとリカバリの 新機能

この章では、Oracle9i リリース 2 (9.2) の、新しいユーザー管理のバックアップとリカバリ機能について説明します。また、関連項目の参照先について記載します。現行リリースに移行するユーザーのために、前のリリースからの新機能についての情報も提供されます。

ユーザー管理のバックアップとリカバリの新機能については、次の項で説明します。

- [Oracle9i のユーザー管理のバックアップとリカバリの新機能](#)
- [Oracle8i のユーザー管理のバックアップとリカバリの新機能](#)

Oracle9i のユーザー管理のバックアップとリカバリの新機能

この項では、次の項目について説明します。

- [Oracle9i リリース 2 \(9.2\) ユーザー管理のバックアップとリカバリの新機能](#)
- [Oracle9i リリース 1 \(9.0.1\) のユーザー管理のバックアップとリカバリの新機能](#)

Oracle9i リリース 2 (9.2) ユーザー管理のバックアップとリカバリの新機能

Oracle9i リリース 2 (9.2) では、データベースの可用性および管理能力を向上させるために、バックアップおよびリカバリに関する次の新機能を提供しています。

- **REDO ログの並列化**

この機能により、サーバー・プロセスは REDO をパラレルに作成でき、更新集中型のワークロードのスループットが向上します。パラレル REDO 機能が使用可能な場合、Oracle では REDO ログが新規のフォーマットで生成されます。Oracle9i リリース 2 (9.2) より前のリリースでは、この新規フォーマットの REDO ログは適用できません。そのため、Oracle9i リリース 2 (9.2) より前のリリースでパラレル・ログを適用する場合は、Oracle9i リリース 2 (9.2) に一時的にアップグレードし、データベースをリカバリしてから、以前のリリースにダウングレードする必要があります。

関連項目： [パラレル REDO の問題のトラブルシューティング方法](#)については、5-2 ページの「[ユーザー管理のメディア・リカバリの問題](#)」を参照してください。この機能を使用可能および使用禁止にする方法については、『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。

Oracle9i リリース 1 (9.0.1) のユーザー管理のバックアップとリカバリの新機能

Oracle9i リリース 1 (9.0.1) では、データベースの可用性および管理能力を向上させるために、バックアップおよびリカバリに関する次の新機能を提供しています。

- **バックアップ・モードのバッチ終了**

ALTER DATABASE END BACKUP 文は、現在バックアップ・モードになっているすべてのデータ・ファイルでバックアップ・モードを終了させます。この機能は、オンライン・バックアップ中に障害が発生した場合にでも、クラッシュ・リカバリ・スクリプトが中断されることなくデータベースを再起動できるようにすることを目的としています。オンライン・バックアップ中にデータベースがクラッシュした場合、以前は各表領域で個別にバックアップ・モードを終了するか、データベースに対してメディア・リカバリを実行する必要がありました。

関連項目： [2-12 ページ「インスタンス障害または SHUTDOWN ABORT 後のバックアップの終了」](#)

- **メディア・リカバリの問題発生後のデータベースのオープンが可能**

データベースのメディア・リカバリ中に問題が発生した場合、Oracle はデータベースを停止して一貫した状態に保ちます。その後、データベースを読取り専用でオープンするか、RESTLOGS オプションを指定してオープンできます。

関連項目： 5-2 ページ「ユーザー管理のメディア・リカバリの問題」

- **試行リカバリ**

SQL*Plus の RECOVER ... TEST 文は、物理データベースに影響を与えることなく、メモリー内で試行リカバリを実行できます。この機能を使用すると、ディスク上のファイルに実際に変更を適用することなく、バックアップおよびリカバリ計画をテストできます。また、メディア・リカバリの問題のトラブルシューティングを行う場合にも、試行リカバリを使用して、通常のリカバリを続行した場合に発生する可能性のある問題を予見することができます。

関連項目： 5-9 ページ「試行リカバリの実行」

- **メディア・リカバリの問題からのリカバリ**

データ・ファイルのメディア・リカバリ中に、リカバリにブロックを破損させることを許可するには、RECOVER 文で ALLOW ... CORRUPTION 句を指定します。リカバリ完了後、Recovery Manager を使用して、破損したブロックに対してブロック・メディア・リカバリを実行できます。この機能によりリカバリ時間が短縮され、データベースの可用性が向上します。

関連項目： 第 5 章「ユーザー管理のメディア・リカバリのトラブルシューティング」

- ***_FILE_NAME_CONVERT パラメータの複数の変換ペア**

DB_FILE_NAME_CONVERT および LOG_FILE_NAME_CONVERT 初期化パラメータで複数の変換ペアを指定できます。

- **LOG_ARCHIVE_DEST_n による最大 10 の場所のサポート**

LOG_ARCHIVE_DEST_n 初期化パラメータは、最大 10 の場所にアーカイブできます。

関連項目： 2-2 ページ「バックアップ前のデータベース・ファイルの列挙」および 3-2 ページ「ユーザー管理のリストア操作」

Oracle8i のユーザー管理のバックアップとリカバリの新機能

この項では、次の項目について説明します。

- [Oracle8i リリース 8.1.6 のユーザー管理のバックアップとリカバリの新機能](#)
- [Oracle8i リリース 8.1.5 のユーザー管理のバックアップとリカバリの新機能](#)

Oracle8i リリース 8.1.6 のユーザー管理のバックアップとリカバリの新機能

Oracle8i リリース 8.1.6 では、データ破損に対する保護を強化するために、多数の内部改良が行われています。

■ 論理的な破損に対する保護

論理的なデータ破損は通常はアプリケーション・エラーにより発生し、破損が REDO ログに反映されるため、修復は困難です。論理的な破損の多くは、データベースに損害を与える変更を検出してロールバックすることができる、ブロック・チェックを有効にすることにより防止できます。ブロック・チェックは次のように強化されています。

- － ロールバック・セグメント・ブロック、トランザクション表ブロック、セグメント・ヘッダー・ブロックなど、より多くのブロック・タイプがチェックされます。
- － ブロック・チェックがさらに効率的になり、システム・オーバーヘッドを増加させることなく、より多くのブロックをチェックできます。
- － SYSTEM 表領域では、DB_BLOCK_CHECKING 初期化パラメータの設定値に関係なく、常にブロック・チェックがオンになります。

■ メモリー破損に対する保護

ブロック・チェックがオンの場合には、データベース・ライター・プロセスはディスクにブロックを書き込む直前にブロック・チェックを実行します。このチェックにより、Oracle はメモリー内にある間に破損を認識し、ディスクに書き込まれる前に自動的に破損ブロックを修復できるようになります。

■ 物理的なデータ破損に対する保護

Oracle は通常、各データ・ブロックにチェックサムを格納することにより、物理的な I/O 破損を検出します。Oracle リリース 8.1.6 では、DB_BLOCK_CHECKSUM パラメータの設定値に関係なく、常に SYSTEM 表領域でチェックサム計算が実行されます。

Oracle が制御ファイルまたは REDO ログを読み込んだときに計算されたチェックサムと格納されているチェックサムが一致しない場合に、Oracle が異なるログまたは同じメンバーから再読み込みを行う状況が、以前の Oracle リリースよりも多くなりました。このため、Oracle がデータの正しいコピーを検出して物理的なデータ破損を修復できる機会が増えることとなります。

Oracle8i リリース 8.1.5 のユーザー管理のバックアップとリカバリの新機能

リリース 8.1.5 のバックアップとリカバリの新機能は次のとおりです。

■ SUSPEND/RESUME 機能を使用したバックアップ

データベースを停止することなく、データベース処理を一時停止してから再開できます。データベースの一時停止中に、分割されたミラーのオンライン・バックアップを作成できます。

関連項目： 2-17 ページ「[SUSPEND モードでのユーザー管理バックアップの作成](#)」

■ TSPITR によるトランスポートابل表領域のサポート

表領域の Point-in-Time リカバリ (TSPITR) を実行するために、トランスポートابل表領域を使用できます。

関連項目： 7-14 ページ「[トランスポートابل表領域を使用した TSPITR の実行](#)」

■ LOG_ARCHIVE_DEST_n 初期化パラメータ

LOG_ARCHIVE_DEST_n (この *n* は 1 ～ 5 の整数) 初期化パラメータは、最大 5 つの場所にアーカイブできます。

ユーザー管理のバックアップとリカバリの概要

この章では、ユーザー管理のバックアップとリカバリの基礎となる、データベースの概要について説明します。

この章で説明する項目は、次のとおりです。

- [ユーザー管理のバックアップとリカバリ](#)
- [ユーザー管理のバックアップとリカバリが必要な理由](#)
- [ユーザー管理のバックアップとリカバリの概要](#)

関連項目： バックアップおよびリカバリの重要な概念については、『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。

ユーザー管理のバックアップとリカバリ

ユーザー管理のバックアップとリカバリとは、バックアップおよびリカバリの主なツールとして **Recovery Manager (RMAN)** が使用されないすべての計画です。ユーザー管理のバックアップ計画の基本は、オペレーティング・システム・コマンドを使用して、データ・ファイルおよびアーカイブ・ログを定期的にバックアップすることです。

メディア障害からのユーザー管理のリカバリを行う基本手順は、次のとおりです。

1. オペレーティング・システム・コマンドを使用してデータベース・ファイルのバックアップをリストアします。
2. **SQL*Plus** の **RECOVER** 文を使用して、リストアしたデータ・ファイルのリカバリを行います。
3. データベースがクローズしている場合は、通常の使用のためにオープンします。データベースがオープンしている場合は、リカバリした表をオンラインに戻します。

ユーザー管理のバックアップとリカバリが必要な理由

オラクル社は、バックアップおよびリカバリ計画の基礎としては **Recovery Manager** を使用することをお勧めします。しかし、ユーザー管理の方法（つまり **Recovery Manager** を使用しない方法）も同じように有効です。ブロック・メディア・リカバリなど、**Recovery Manager** を使用した場合にしか実行できない機能もあることに注意してください。

次のような状況では、**Recovery Manager** を使用するかわりにユーザー管理の方法を選択できます。

- 古いバージョンのデータベースから現行バージョンに移行するときに、これまで使用してきたバックアップ・スクリプトを即時に更新しない場合。
- **Oracle7** 以上のデータベースを含むネットワークがある場合に、すべてのデータベースを同じ方法で処理する、単一のバックアップおよびリカバリ方法が必要な場合。
Recovery Manager がサポートするのはリリース 8.0 以上の **Oracle** データベースのみです。
- **Recovery Manager** のすべてのバックアップが消失したため、ユーザー管理バックアップをリストアし、**SQL*Plus** の **RECOVER** コマンドを使用してリカバリを実行する必要がある場合。

オラクル社は、**Recovery Manager** の有効な代替策として、ユーザー管理のバックアップとリカバリをサポートしています。

ユーザー管理のバックアップとリカバリの概要

この項では、次の項目について説明します。

- ユーザー管理バックアップ
- ユーザー管理のリストアおよびリカバリ

ユーザー管理バックアップ

ユーザー管理バックアップには、論理的なものと物理的なものがあります。表、ビューおよびストアド・プロシージャなど、論理的なオブジェクトのバックアップを作成するためにエクスポート・ユーティリティを使用し、インポート・ユーティリティを使用してこれらのオブジェクトをリストアできます。

Recovery Manager を使用しない場合には、オペレーティング・システム・ユーティリティを使用して物理バックアップを作成できます。物理バックアップとは、オペレーティング・システム上の Oracle データベース・ファイルまたはアーカイブ REDO ログのバックアップです。これらのファイルは、手動で管理されるデータベース・ファイル、または **Oracle 管理ファイル**です。Oracle Managed Files 機能を使用した場合は、Oracle がユーザーのかわりにファイルに名前を付け、ユーザーが表領域を削除したときにはユーザーのかわりにファイルを削除します。バックアップおよびリカバリの観点からは、Oracle 管理ファイルはユーザー管理のファイルと変わりません。

次の表は、物理バックアップの主なタイプと、Recovery Manager を使用せずにこれらのバックアップを実行する方法を示しています。

バックアップ・オブジェクト	バックアップ方式	例
データ・ファイル	オペレーティング・システム・ユーティリティ	% cp df3.f df3.bak
アーカイブ・ログ	オペレーティング・システム・ユーティリティ	% cp log_1_23.arc log_1_23.bak
制御ファイル	SQL 文	SQL> ALTER DATABASE BACKUP CONTROLFILE TO cf1.bak
初期化パラメータ・ファイル	SQL 文	SQL> CREATE PFILE = init.ora.bak FROM SPFILE;
ネットワークおよびパスワード・ファイル	オペレーティング・システム・ユーティリティ	% cp tnsnames.ora tnsnames.bak C:¥> copy tnsnames.ora tnsnames.bak
論理的オブジェクト (表、索引、PL/SQL ユニット)	エクスポート・ユーティリティ	% export SYSTEM/manager TABLE=hr.emp FILE=emp.dmp

関連項目：

- エクスポートおよびインポート・ユーティリティの使用方法については、『Oracle9i データベース・ユーティリティ』を参照してください。
- Oracle Managed Files 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。

バックアップの基本的な方法

データベース全体のユーザー管理バックアップを作成する基本的な方法は次のとおりです。

1. 動的パフォーマンス・ビューまたはデータ・ディクショナリ表を問い合わせ、バックアップするデータ・ファイル、制御ファイルおよびアーカイブ REDO ログを識別します（手順は 2-2 ページの「[V\\$ ビューの問合せによるバックアップ情報の取得](#)」を参照）。
2. UNIX の cp コマンドなどのオペレーティング・システム・コマンドを使用して、データ・ファイルおよびアーカイブ REDO ログをバックアップします（手順は 2-4 ページの「[データベース全体のユーザー管理バックアップの作成](#)」を参照）。
3. SQL 文を使用して制御ファイルをバックアップします（手順は 2-20 ページの「[制御ファイルのユーザー管理バックアップの作成](#)」を参照）。
4. UNIX の cp コマンドなどのオペレーティング・システム・コマンドを使用して、構成ファイルをバックアップします（手順は 2-33 ページの「[その他の Oracle ファイルのユーザー管理バックアップの作成](#)」を参照）。

注意： オンライン REDO ログはバックアップしないでください。メディア・リカバリ後にオンライン・ログをリセットし、バックアップしたログを誤ってデータベースに適用すると、データベースが破損することがあります。

ユーザー管理の一貫性バックアップと非一貫性バックアップ

Recovery Manager またはオペレーティング・システム・コマンドを使用して、**非一貫性バックアップ**または**一貫性バックアップ**を作成できます。非一貫性バックアップは、データベースのオープン中に、またはデータベースが正常に停止されなかった後で作成された、1 つ以上のデータベース・ファイルのバックアップです。一貫性バックアップは、データベースが正常に停止された後で作成された、1 つ以上のデータベース・ファイルのバックアップです。一貫性バックアップでは非一貫性バックアップとは異なり、リストアした後でリカバリを実行する必要はありません。

リカバリに必要な REDO を使用できないため、NOARCHIVELOG モードで実行されているデータベースの場合は、一貫性のあるデータベース全体のバックアップが、唯一有効なバックアップ・オプションになります。NOARCHIVELOG モードでは、Oracle は最初にアーカイブすることなく、REDO レコードに上書きします。

データベースを ARCHIVELOG モードで実行している場合には、データベースをオープンしたままデータベース・ファイルをバックアップできます。これらのバックアップは一貫性はありませんが、必要なアーカイブ REDO ログがあれば、これらのバックアップをリカバリできます。表領域をオフラインにしてデータ・ファイルのバックアップを作成するか、**オンライン・バックアップ**を実行できます。オンライン・バックアップは、表領域がオンラインの状態で行われます。オンライン・バックアップを実行するには、表領域の**バックアップ・モード**を開始および終了する SQL 文を使用して、バックアップを開始および終了する必要があります。

関連項目：

- 2-5 ページ「データベース全体の一貫性のあるバックアップの作成」
- 2-6 ページ「オフラインの表領域およびデータ・ファイルのユーザー管理バックアップの作成」
- 2-8 ページ「オンラインの表領域およびデータ・ファイルのユーザー管理バックアップの作成」

SUSPEND モードのバックアップ

サード・パーティ・ツールには、ディスクまたは論理デバイスのセットをミラー化（つまりプライマリ・データの正確な複製を別の場所に保持）し、後で**ミラーを分割**できるものがあります。ミラーの分割では、コピーが分離されるため、それぞれを別々に使用できます。

SUSPEND/RESUME の機能を使用すると、データベースに対する I/O を一時停止した後、ミラーを分割し、分割されたミラーのバックアップを作成できます。オンライン・バックアップ機能を補完するこの機能を使用することにより、I/O が新しく実行されないように、データベースを休止できます。その後、一時停止中のデータベースにアクセスし、I/O に妨害されることなくバックアップを作成できます。

関連項目： 2-17 ページ「一時停止 / 再開機能」

バックアップの検証

バックアップを検証する最良の方法は、別の場所にデータベースをリストアおよびリカバリしてテストすることです。この操作が正常に実行された場合は、バックアップは有効です。

また、DBVERIFY ユーティリティを使用して、バックアップが破損していないかをテストすることもできます。DBVERIFY は、オフラインのデータ・ファイルに対して、物理的なデータ構造の整合性チェックを実行する、外部のコマンドライン・ユーティリティです。

DBVERIFY は、バックアップのデータ・ファイルをリストアする前にバックアップ・データ・ファイルの有効性を確認する必要がある場合、またはデータ破損の問題を検出した場合の診断の補助として主に使用されます。DBVERIFY 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 Sun/Sequent システムの場合は dbv）。

関連項目： 2-29 ページ「ユーザー管理バックアップの検証」

ユーザー管理のリストアおよびリカバリ

データ・ファイルに損傷を与えるメディア障害が発生した場合には、オペレーティング・システム・コマンドを使用して、影響を受けたデータ・ファイルのバックアップをリストアした後、SQL*Plus の RECOVER コマンドを使用してリカバリを実行する必要があります。一部のデータ・ファイルのみをリストアして、リストアされたデータ・ファイルを含む表領域のリカバリを実行することも、データベース全体のリストアおよびリカバリを実行することもできます。データ・ファイルの元の場所やバックアップの場所がわかるように、バックアップの記録を残す必要があります。

データベースが ARCHIVELOG モードで実行されているときにメディア・リカバリ操作を開始するには、SQL*Plus の RECOVER コマンドを使用します。メディア・リカバリには、データベースで生成されたすべての REDO が適用される**完全リカバリ**と、既存の REDO のすべてが適用されるわけではない**不完全リカバリ**という 2 つの基本的なタイプがあります。不完全リカバリは、データベース全体のリストアおよびリカバリでのみ有効です。個々の表領域の不完全リカバリを実行する特殊な手順は、**表領域の Point-in-Time リカバリ (TSPITR)**と呼ばれます。

リストアおよびリカバリの基本的な方法

ユーザー管理のリストアおよびリカバリ計画の基本手順は次のとおりです。

1. リストアおよびリカバリが必要な対象を判別します（手順は 3-6 ページの「[リカバリが必要なデータ・ファイルの判別](#)」を参照）。
2. オペレーティング・システム・ユーティリティを使用して、メディア障害により永久に損傷したファイルのバックアップをリストアします。データ・ファイルを元の場所にリストアできないときには、リストアされたデータ・ファイルの場所を変更し、制御ファイルの中に記録されている場所を変更します（手順は 3-7 ページの「[データ・ファイルのリストア](#)」を参照）。
3. オペレーティング・システム・ユーティリティを使用して、必要なアーカイブ REDO ログ・ファイルをすべてリストアします（手順は 3-17 ページの「[アーカイブ REDO ログのリストア](#)」を参照）。
4. 4-2 ページの「[ユーザー管理のメディア・リカバリの実行概要](#)」の説明に従って、SQL*Plus の RECOVER コマンドを使用して、ファイルのリカバリをします。

メディア・リカバリにおけるアーカイブ・モードの意味

データベースのアーカイブ・モードは、ユーザーが実行できるリカバリのタイプを決定します。たとえば、データベースが NOARCHIVELOG モードのときに、メディア障害によりデータ・ファイルの一部またはすべてが破損した場合は、最新のデータベース全体の一貫性バックアップをリストアしてオープンすることが、リカバリの唯一の手段になるのが普通です。

NOARCHIVELOG モードのデメリットは、最新の全体バックアップの時点からメディア障害の時点までデータベースのリカバリを行うには、この間に実行されたすべての変更を手動で再入力する必要があることです。データベースが ARCHIVELOG モードの場合で、この期間の REDO ログがアーカイブ・ログ・ファイルまたはオンライン・ログ・ファイルとして使用できる場合は、完全リカバリまたは不完全リカバリを使用してデータベースを再構築できるため、失われる変更の数を最小化できます。

ユーザー管理の表領域の Point-in-Time リカバリ (TSPITR)

ユーザー管理の表領域の Point-in-Time リカバリ (TSPITR) を使用すると、1 つ以上の表領域 (SYSTEM 以外の表領域) を、データベースの残りの部分とは異なる時点まで迅速にリカバリできます。

ユーザー管理の TSPITR は、次の場合のリカバリに特に有用です。

- 誤った DROP TABLE または TRUNCATE TABLE 操作
- 論理的に破損した表
- データベースの一部に対してのみ影響する、不正なバッチ・ジョブまたはその他の DML 文の実行
- 1 つの物理データベースの別個の表領域に複数のスキーマが存在するときに、ある論理スキーマの物理データベースの残りの部分とは異なる時点までのリカバリ
- バックアップからデータベース全体をリストアしてロール・フォワードするよりも TSPITR が効率的な場合の、VLDB (非常に大規模なデータベース) の表領域 (判断する前に 7-4 ページの「[表領域の Point-in-Time リカバリの準備: 基本手順](#)」を参照)

関連項目： [第 7 章「ユーザー管理の TSPITR の実行」](#)

ユーザー管理バックアップの作成

Recovery Manager (RMAN) を使用しない場合には、ユーザー管理の方法を使用して、データベース・ファイルのバックアップを作成できます。

この章で説明する項目は、次のとおりです。

- V\$ ビューの問合せによるバックアップ情報の取得
- データベース全体のユーザー管理バックアップの作成
- オフラインの表領域およびデータ・ファイルのユーザー管理バックアップの作成
- オンラインの表領域およびデータ・ファイルのユーザー管理バックアップの作成
- SUSPEND モードでのユーザー管理バックアップの作成
- 制御ファイルのユーザー管理バックアップの作成
- アーカイブ REDO ログのユーザー管理バックアップの作成
- RAW デバイスへのユーザー管理バックアップの作成
- ユーザー管理バックアップの検証
- エクスポートを使用した論理バックアップの作成
- その他の Oracle ファイルのユーザー管理バックアップの作成

V\$ ビューの問合せによるバックアップ情報の取得

バックアップを作成する前に、データベース内のすべてのファイルを確認します。次に、バックアップが必要な対象を確認します。

この項では、次の項目について説明します。

- [バックアップ前のデータベース・ファイルの列挙](#)
- [オンライン表領域バックアップのデータ・ファイルのステータスの判別](#)

バックアップ前のデータベース・ファイルの列挙

バックアップを開始する前に、データベースに問合せを行い、バックアップが必要なファイルを確認します。Oracle 管理ファイルのバックアップは、ユーザーが手動で命名したデータベース・ファイルのバックアップと同じです。

データ・ファイル、オンライン REDO ログおよび制御ファイルを列挙する手順は、次のとおりです。

1. SQL*Plus を起動し、V\$DATAFILE を問い合せてデータ・ファイルのリストを取得します。たとえば、次のように入力します。

```
SQL> SELECT NAME FROM V$DATAFILE;
```

V\$TABLESPACE ビューと V\$DATAFILE ビューを結合して、データ・ファイルのリストと、関連する表領域のリストを取得することもできます。

```
SELECT t.NAME "Tablespace", f.NAME "Datafile"
  FROM V$TABLESPACE t, V$DATAFILE f
 WHERE t.TS# = f.TS#
 ORDER BY t.NAME;
```

2. V\$LOGFILE ビューを問い合せて、オンライン REDO ログ・ファイルのファイル名を取得します。たとえば、次の問合せを発行します。

```
SQL> SELECT MEMBER FROM V$LOGFILE;
```

3. V\$CONTROLFILE ビューを問い合せて、現行の制御ファイルのファイル名を取得します。たとえば、次の問合せを発行します。

```
SQL> SELECT NAME FROM V$CONTROLFILE;
```

多重制御ファイルの場合は、バックアップする必要があるのは 1 つのコピーのみです。

4. ALTER DATABASE BACKUP CONTROLFILE TO 'filename' 文を使用して制御ファイルのバックアップを作成する場合には、制御ファイルのバックアップとともに、すべてのデータ・ファイルおよびオンライン REDO ログ・ファイルのリストを保存します。現在のデータベース構造は、特定の制御ファイルのバックアップが作成された時点のデータベース構造とは異なるものである場合があるため、バックアップ制御ファイルに記録されたファイルのリストを保存しておくとしカバリする際に便利です。

オンライン表領域バックアップのデータ・ファイルのステータスの判別

データ・ファイルが現行のオンライン表領域のバックアップの一部であるかどうかを確認するには、V\$BACKUP ビューを問い合わせます。このビューはユーザー管理のオンライン表領域のバックアップの場合にのみ役に立ちます。オフライン表領域のバックアップまたは Recovery Manager バックアップでは意味がありません。

V\$BACKUP ビューは、データベースがオープンしている場合に特に有用です。また、障害時のファイルのバックアップ・ステータスも表示されるため、インスタンス障害の直後にも有用です。この情報を使用して、バックアップ・モードのまま残っている表領域があるかどうかを確認します。

現在使用されている制御ファイルが、リストアされたバックアップか、メディア障害の発生後に作成された新しい制御ファイルの場合には、V\$BACKUP は役に立ちません。リストアされた制御ファイルまたは再作成された制御ファイルには、Oracle が V\$BACKUP に正確に表示するために必要とする情報が含まれていません。また、ファイルのバックアップをリストアした場合には、V\$BACKUP の中のこのファイルの STATUS は、最新のバージョンではなく、ファイルの古いバージョンのバックアップ・ステータスを反映したものになります。このため、このビューにはリストアされたファイルに関して誤解を招くデータが表示される場合があります。

たとえば、次の問合せは、バックアップ・モードに設定された表領域に現在どのようなデータ・ファイルが含まれているかを表示します。

```
SELECT t.name AS "TB_NAME", d.file# as "DF#", d.name AS "DF_NAME", b.status
FROM V$DATAFILE d, V$TABLESPACE t, V$BACKUP b
WHERE d.TS#=t.TS#
AND b.FILE#=d.FILE#
AND b.STATUS='ACTIVE'
/
```

出力のサンプルは次のようになります。

TB_NAME	DF#	DF_NAME	STATUS
-----	-----	-----	-----
TBS_1	3	/oracle/dbs/tbs_11.f	ACTIVE
TBS_1	4	/oracle/dbs/tbs_12.f	ACTIVE

STATUS 列において、NOT ACTIVE が、ファイルが現在バックアップ・モード（つまり ALTER TABLESPACE ... BEGIN BACKUP）ではないことを意味するのに対して、ACTIVE は、ファイルが現在バックアップ・モードであることを意味します。

データベース全体のユーザー管理バックアップの作成

NORMAL、IMMEDIATE または TRANSACTIONAL オプションを使用してデータベースを停止した後で、データベースの中のすべてのファイルのデータベース全体のバックアップを作成できます。データベースのオープン中、またはインスタンス障害または SHUTDOWN ABORT の後に作成されたデータベース全体のバックアップは一貫性のないものになります。この場合のファイルは、チェックポイント SCN に対して一貫性がありません。

データベースを ARCHIVELOG モードまたは NOARCHIVELOG モードのどちらで稼働しているときにも、データベース全体のバックアップを作成できます。しかしデータベースを NOARCHIVELOG モードで実行する場合には、一貫性のあるバックアップが必要です。つまり、バックアップ前にデータベースを正しく停止する必要があります。

一貫性のあるデータベース全体のバックアップの結果としてのバックアップ・ファイルのセットは、すべてのファイルが同じ SCN にチェックポイントされているため、一貫性があります。リカバリを実行せずに、データベースの一貫性バックアップをリストアできます。データベースを ARCHIVELOG モードで稼働している場合には、バックアップ・ファイルをリストアした後で、データベースをより新しい時点まで回復するために追加のリカバリ手順を実行できます。また、データベースが ARCHIVELOG モードの場合には、一貫性のないデータベース全体のバックアップを作成することもできます。

制御ファイルは、データベースのリストアおよびリカバリに重要な役割を果たします。ARCHIVELOG モードで実行しているデータベースの場合には、ALTER DATABASE BACKUP CONTROLFILE TO '*filename*' 文を使用して、制御ファイルをバックアップしておくことをお勧めします。完結した、一貫性のあるデータベース全体のバックアップ中にオペレーティング・システム・ユーティリティを使用して制御ファイルをバックアップした場合は、当該バックアップに含めたその他のデータ・ファイルをリストアする場合以外はこの制御ファイルを使用しないでください。一貫性バックアップ中にオペレーティング・システム・ユーティリティを使用してバックアップされた制御ファイルがリカバリに使用されることもあります（ただし RECOVER 文の USING BACKUP CONTROLFILE 句を指定した場合のみ）、USING BACKUP CONTROLFILE 句の指定を忘れるとリカバリの問題が発生する可能性があるため、この方法はお勧めしません。

関連項目： 制御ファイルのバックアップの詳細は、2-20 ページの「[制御ファイルのユーザー管理バックアップの作成](#)」を参照してください。

データベース全体の一貫性のあるバックアップの作成

データベースのデータ・ファイルが一貫性のあるものになるように、データベース全体のバックアップを作成する前に、NORMAL、IMMEDIATE または TRANSACTIONAL オプションを使用してデータベースを停止します。

注意： データベースが NOARCHIVELOG モードの場合には、インスタンスの障害または異常終了後にデータベース全体のバックアップを実行しないでください。このバックアップには一貫性がなく、一貫性のあるものにするためにはリカバリが必要です。オンライン REDO ログの中に必要な REDO が存在し、これらのログがそのまま残っていないかぎり、このバックアップは使用できません。

一貫性のあるデータベース全体のバックアップを作成する手順は、次のとおりです。

1. データベースがオープンしている場合は、SQL*Plus を使用して、NORMAL、IMMEDIATE または TRANSACTIONAL オプションを指定してデータベースを停止します。たとえば、次のコマンドのいずれかを実行します。

```
SQL> SHUTDOWN NORMAL
SQL> SHUTDOWN IMMEDIATE
SQL> SHUTDOWN TRANSACTIONAL
```

インスタンスが異常終了したか、障害のために停止した場合には、データベース全体のバックアップを作成しないでください。可能であればデータベースを再オープンし、正しく停止します。

2. オペレーティング・システム・ユーティリティを使用して、すべてのデータ・ファイルと、初期化パラメータ・ファイルの CONTROL_FILES パラメータで指定されたすべての制御ファイルのバックアップを作成します。また、初期化パラメータ・ファイル、およびその他の Oracle 製品の初期化ファイルもバックアップします。これらのファイルを検索するには、Oracle ホーム・ディレクトリから開始してすべてのサブディレクトリを再帰的に検索し、*.ora を探します。

注意： リストア操作を実行する必要がある場合には、初期化パラメータ・ファイルで指定されたすべての場所に制御ファイルをリストアする必要があります。このため、多重制御ファイルのすべてのファイルのコピーを作成すれば（制御ファイルが同じであっても）、リストア時の問題を回避できます。

たとえば、次のようにして、/disk1/oracle/dbs ディレクトリの中のデータ・ファイルおよび制御ファイルを /disk2/backup にバックアップできます。

```
% cp /disk1/oracle/dbs/*.dbf /disk2/backup
% cp /disk1/oracle/dbs/*.cf /disk2/backup
% cp /disk1/oracle/network/admin/*.ora /disk2/backup
% cp /disk1/oracle/rdbms/admin/*.ora /disk2/backup
```

- 3. データベースを再起動します。たとえば、次のように入力します。

```
SQL> STARTUP
```

関連項目： データベースの起動および停止の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

オフラインの表領域およびデータ・ファイルのユーザー管理バックアップの作成

表領域をオフライン化して、個々の表領域のすべてのデータ・ファイルまたは一部のデータ・ファイルをバックアップできます。データベースの残りの表領域はすべて、オープンのままにし、システム全体で使用できます。表領域のオフライン化およびオンライン化には、DBA 権限または `MANAGE TABLESPACE` システム権限が必要です。

オフライン表領域のバックアップを行うときには、次のガイドラインに注意してください。

- `SYSTEM` 表領域またはアクティブなロールバック・セグメントを含む表領域をオフライン化することはできません。このような表領域では、次の手順は使用できません。
- 表領域 `Primary` 内に表があり、その索引が表領域 `Index` にあるとします。表領域 `Index` をオフラインにし、表領域 `Primary` をオンラインのままにしておくと、`Primary` 内にある索引付きの表に対して `DML` が発行されたときにエラーが発生する可能性があります。この問題は、オブティマイザによって選択されたアクセス方法が、`Index` 表領域内の索引へのアクセスを必要としたときのみ発生します。

オフライン表領域をバックアップする手順は、次のとおりです。

- 1. 表領域のバックアップを開始する前に、`DBA_DATA_FILES` ビューを問い合せて、表領域のデータ・ファイルを確認します。たとえば、`users` 表領域をバックアップするとします。SQL*Plus に次のように入力します。

```
SELECT TABLESPACE_NAME, FILE_NAME
FROM SYS.DBA_DATA_FILES
WHERE TABLESPACE_NAME = 'users';

TABLESPACE_NAME                                FILE_NAME
-----
users                                /oracle/dbs/users.f
```

この例の `/oracle/dbs/users.f` は、`users` 表領域の中のデータ・ファイルに対応する完全指定のファイル名です。

2. 可能であれば通常優先度で表領域をオフラインにします。表領域のリカバリを必要とすることなく、後で表領域を確実にオンラインに戻せるため、通常優先度をお薦めします。たとえば、次の文は `users` という名前の表領域を `NORMAL` 指定でオフラインにします。

```
SQL> ALTER TABLESPACE users OFFLINE NORMAL;
```

通常優先度で表領域をオフラインにすると、表領域のすべてのデータ・ファイルはクローズされます。

3. オフライン・データ・ファイルをバックアップします。たとえば、`UNIX` ユーザーは次のように入力してデータ・ファイル `users.f` をバックアップできます。

```
% cp /disk1/oracle/dbs/users.f /disk2/backup/users.backup
```

4. 表領域をオンライン化します。たとえば、次の文は表領域 `users` をオンラインに戻します。

```
ALTER TABLESPACE users ONLINE;
```

注意： 一時または即時優先度を使用して表領域をオフライン化した場合には、表領域のリカバリを実行しないかぎり、表領域をオンライン化することはできません。

表領域をオンライン化すると、表領域がオープンし、使用可能になります。

5. 表領域のバックアップのリカバリに必要な `REDO` がアーカイブされるように、アーカイブされていない `REDO` ログをアーカイブします。たとえば、次のように入力します。

```
ALTER SYSTEM ARCHIVE LOG CURRENT;
```

オンラインの表領域およびデータ・ファイルのユーザー管理バックアップの作成

データベースのオープン中は、オンライン表領域のすべてのデータ・ファイルまたは特定のデータ・ファイルをバックアップできます。オンライン表領域が読取り / 書込みか読取り専用かによって手順は異なります。

注意： 一時表領域はバックアップしないでください。

この項では、次の項目について説明します。

- [オンラインの読取り / 書込み表領域のユーザー管理バックアップの作成](#)
- [オンラインの読取り / 書込み表領域の複数のユーザー管理バックアップの作成](#)
- [インスタンス障害または SHUTDOWN ABORT 後のバックアップの終了](#)
- [読取り専用表領域のユーザー管理バックアップの作成](#)
- [UNDO 表領域のユーザー管理バックアップの作成](#)

オンラインの読取り / 書込み表領域のユーザー管理バックアップの作成

表領域がオンラインで、データベースがオープンしているときに、データ・ファイルのユーザー管理バックアップを作成するには、読取り / 書込み表領域をバックアップ・モードに設定する必要があります。ALTER TABLESPACE BEGIN BACKUP 文は表領域をバックアップ・モードに設定します。

表領域がバックアップ・モードになると、Oracle は表領域内のデータ・ファイルにチェックポイントを記録しなくなります。オペレーティング・システムのバックアップ・ユーティリティがコピーしている最中にブロックの部分更新が発生する可能性があるため、Oracle はバックアップ・モード中は変更されたデータ・ブロック全体を REDO ストリームにコピーします。ユーザーが ALTER TABLESPACE ... END BACKUP または ALTER DATABASE END BACKUP 文を使用して表領域のバックアップ・モードを終了すると、Oracle はデータ・ファイルのヘッダーを現在のデータベース・チェックポイントまで進めます。

この方法でバックアップされたデータ・ファイルをリストアすると、データ・ファイルのヘッダーには、オンライン表領域のバックアップ中ではなく、バックアップ前に発生した最新のデータ・ファイル・チェックポイントが記録されます。このため、リカバリが必要な場合は、REDO ログ・ファイルの適切なセットを適用するように Oracle から指示があります。REDO ログには、データ・ファイルをリカバリし、データ・ファイルを一貫性のある状態にするために必要なすべての変更が含まれています。

オープン状態のデータベース中のオンラインの読取り / 書き込み表領域をバックアップする手順は、次のとおりです。

1. 表領域のバックアップを開始する前に、DBA_DATA_FILES データ・ディクショナリ・ビューを使用して、表領域内のすべてのデータ・ファイルを確認します。たとえば、users 表領域をバックアップするとします。次のように入力します。

```
SELECT TABLESPACE_NAME, FILE_NAME
FROM SYS.DBA_DATA_FILES
WHERE TABLESPACE_NAME = 'users';
```

TABLESPACE_NAME	FILE_NAME
USERS	/oracle/dbs/tbs_21.f
USERS	/oracle/dbs/tbs_22.f

この例の /oracle/dbs/tbs_21.f および /oracle/dbs/tbs_22.f は、users 表領域のデータ・ファイルに対応する完全指定のファイル名です。

2. オンライン表領域のバックアップの開始をマークします。たとえば、次の文は、表領域 users のオンライン・バックアップの開始をマークします。

```
SQL> ALTER TABLESPACE users BEGIN BACKUP;
```

注意： オンライン表領域のバックアップの開始マークを忘れるか、オンライン表領域のバックアップ前に BEGIN BACKUP 文が完了していることを確認しなかった場合は、バックアップ・データ・ファイルをその後のリカバリ操作で使用できなくなります。このようなバックアップのリカバリは危険であり、一貫性のないデータを生むエラーが発生する可能性があります。たとえば、リカバリ操作を試みるとファジー・ファイルの警告が発行され、データベースの一貫性が失われ、データベースをオープンできなくなる可能性があります。

3. オペレーティング・システム・コマンドを使用して、オンライン表領域のオンライン・データ・ファイルをバックアップします。たとえば、UNIX ユーザーは次のように入力できます。

```
% cp /oracle/dbs/tbs_21.f /oracle/backup/tbs_21.backup
% cp /oracle/dbs/tbs_22.f /oracle/backup/tbs_22.backup
```

4. オンライン表領域のデータ・ファイルをバックアップした後、SQL 文 ALTER TABLESPACE で END BACKUP オプションを指定して、オンライン・バックアップの終了を指示します。たとえば、次の文は、表領域 users のオンライン・バックアップを終了します。

```
SQL> ALTER TABLESPACE users END BACKUP;
```

5. 表領域のバックアップのリカバリに必要な REDO がアーカイブされるように、アーカイブされていない REDO ログをアーカイブします。たとえば、次のように入力します。

```
SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;
```

注意： 表領域のバックアップ・モードを終了しないと、Oracle はこの表領域内のデータ・ブロックのコピー全体をオンライン REDO ログに書き込み続けるため、パフォーマンスの問題が発生する可能性があります。また、表領域がまだバックアップ・モードのときにデータベースを停止しようとする、ORA-01149 エラーが表示されます。

オンラインの読取り / 書込み表領域の複数のユーザー管理バックアップの作成

複数のオンライン表領域をバックアップする場合には、シリアルまたはパラレルにバックアップできます。必要性に応じていずれかの手順を使用してください。

オンライン表領域のパラレル・バックアップ

バックアップが必要なすべての表領域を同時にバックアップ・モードにできます。REDO には変更された各データ・ブロックのコピーを含める必要があるため、複数のユーザーがこれらの表領域を更新しているときには、オンライン REDO ログは大きくなることがあります。

オンライン表領域をパラレルにバックアップする手順は、次のとおりです。

1. 必要な ALTER TABLESPACE 文をすべて一度に発行して、すべてのオンライン表領域でバックアップの準備をします。たとえば、次のようにして表領域 ts1、ts2 および ts3 をバックアップ・モードにします。

```
SQL> ALTER TABLESPACE ts1 BEGIN BACKUP;  
SQL> ALTER TABLESPACE ts2 BEGIN BACKUP;  
SQL> ALTER TABLESPACE ts3 BEGIN BACKUP;
```

2. オンライン表領域のすべてのファイルをバックアップします。たとえば、UNIX ユーザーは次のようにして、tbs_ という接頭辞を持つデータ・ファイルをバックアップできます。

```
% cp /oracle/dbs/tbs_* /oracle/backup
```

3. 次の例のように、表領域のバックアップ・モードを終了します。

```
SQL> ALTER TABLESPACE ts1 END BACKUP;  
SQL> ALTER TABLESPACE ts2 END BACKUP;  
SQL> ALTER TABLESPACE ts3 END BACKUP;
```


4. 表領域のバックアップのリカバリに必要な REDO がアーカイブされるように、アーカイブされていない REDO ログをアーカイブします。たとえば、次のように入力します。

```
SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;
```

オンライン表領域のシリアル・バックアップ

オンライン・バックアップが必要なすべての表領域を 1 つずつバックアップ・モードにできます。ALTER TABLESPACE ... BEGIN/END BACKUP 文の間の間隔が最小化されるため、シリアル・バックアップ・オプションをお勧めします。データ・ブロック全体が REDO ログにコピーされるため、オンライン・バックアップ中は表領域に関してさらに REDO 情報が生成されます。

オンライン表領域のシリアル・バックアップの手順は、次のとおりです。

1. 表領域でオンライン・バックアップの準備をします。たとえば、表領域 tbs_1 をバックアップ・モードにするには、次のように入力します。

```
SQL> ALTER TABLESPACE tbs_1 BEGIN BACKUP;
```

2. 表領域内のデータ・ファイルをバックアップします。たとえば、次のように入力します。

```
% cp /oracle/dbs/tbs_1.f /oracle/backup/tbs_1.bak
```

3. 表領域のバックアップ・モードを終了します。たとえば、次のように入力します。

```
SQL> ALTER TABLESPACE tbs_1 END BACKUP;
```

4. 対象のすべての表領域でバックアップが完了するまで、残りの各表領域でこの手順を繰り返します。
5. 表領域のバックアップのリカバリに必要な REDO がアーカイブされるように、アーカイブされていない REDO ログをアーカイブします。たとえば、次のように入力します。

```
SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;
```

インスタンス障害または SHUTDOWN ABORT 後のバックアップの終了

この項では、次の項目について説明します。

- 表領域がバックアップ・モードの場合のインスタンス障害
- ALTER DATABASE END BACKUP 文を使用したバックアップ・モードの終了
- RECOVER コマンドを使用したバックアップ・モードの終了

表領域がバックアップ・モードの場合のインスタンス障害

次の場合には、表領域のバックアップが失敗し、不完全になることがあります。

- バックアップが完了したが、ユーザーが ALTER TABLESPACE ... END BACKUP 文を使用してオンライン表領域のバックアップ操作の終了を指示していない場合。
- バックアップが完了する前に、インスタンス障害または SHUTDOWN ABORT によりバックアップが中断された場合。

クラッシュ・リカバリが必要な場合に（すでにデータ・ファイルがオープンしているのでインスタンス・リカバリではない）、データ・ファイルをオープンしようとしたときにデータ・ファイルがバックアップ・モードになっていると、システムはこのファイルがリストアされたバックアップであると解釈します。リカバリのコマンドが発行されるか、データ・ファイルのバックアップ・モードが終了するまでは、Oracle はデータベースをオープンしません。

たとえば、ユーザーが STARTUP 文を実行すると、次のようなメッセージが表示されることがあります。

```
ORA-01113: ファイル 12 はメディア・リカバリが必要です  
ORA-01110: データ・ファイル 12: '/oracle/dbs/tbs_41.f'
```

ユーザーが表領域のオンライン・バックアップを終了しなかったために、複数の表領域のデータ・ファイルでメディア・リカバリが必要であると Oracle から指示があった場合には、データベースがマウントされていれば、ALTER DATABASE END BACKUP 文を実行して、すべてのデータ・ファイルで同時にバックアップ・モードを終了できます。

高可用性の状況や、DBA がデータベースを監視していない状況（たとえば早朝など）では、ユーザーの介入を必要とする事態は回避する必要があります。このために、次の内容のクラッシュ・リカバリ・スクリプトを作成しておくことができます。

1. データベースをマウントする。
2. ALTER DATABASE END BACKUP 文を実行する。
3. ALTER DATABASE OPEN を実行し、システムが自動的に起動できるようにする。

ALTER DATABASE END BACKUP を含む自動化されたクラッシュ・リカバリ・スクリプトは、特に次の状況で便利です。

- Oracle Real Application Clusters 構成中のすべてのノードが障害を起こした場合。
- コールド・フェイルオーバー・クラスタ（最初のノードが障害を起こした場合は 2 番目のノードをマウントしてデータベースのリカバリを行う必要がある Oracle Real Application Clusters ではないクラスタ）内の 1 つのノードが障害を起こした場合。

また、表領域がバックアップ・モードのときに発生したシステム障害後に、次の手動の方法を使用することもできます。

- データベースをリカバリし、END BACKUP 文を発行せずに済ませる。
- データベースをマウントし、まだバックアップ・モードになっている各表領域に対して ALTER TABLESPACE ... END BACKUP を実行する。

ALTER DATABASE END BACKUP 文を使用したバックアップ・モードの終了

複数の表領域がバックアップ・モードのままになっているときには、ALTER DATABASE END BACKUP 文を実行できます。このコマンドの主な目的は、DBA の介入なしに、クラッシュ・リカバリ・スクリプトで障害の発生したシステムを再起動することです。また、次の手順を手動で実行することもできます。

複数の表領域でバックアップ・モードを同時に終了する手順は、次のとおりです。

1. データベースをマウントしますが、オープンはしません。たとえば、次のように入力します。

```
SQL> STARTUP MOUNT
```

2. この手順を手動で実行する場合は（つまりクラッシュ・リカバリ・スクリプトの一部としてではなく）、V\$BACKUP ビューを問い合せて、データベースが再起動される前にバックアップされていた表領域のデータ・ファイルのリストを表示します。

```
SQL> SELECT * FROM V$BACKUP WHERE STATUS = 'ACTIVE';
```

FILE#	STATUS	CHANGE#	TIME

12	ACTIVE	20863	25-NOV-00
13	ACTIVE	20863	25-NOV-00
20	ACTIVE	20863	25-NOV-00
3 rows selected.			

3. ALTER DATABASE END BACKUP 文を発行し、現在バックアップ・モードになっているすべてのデータ・ファイルでバックアップ・モードを終了します。たとえば、次のように入力します。

```
SQL> ALTER DATABASE END BACKUP;
```

この文を使用できるのは、データベースがマウントされているが、オープンされていない場合のみです。データベースがオープンしている場合には、影響を受ける各表領域またはデータ・ファイルに対して、ALTER TABLESPACE ... END BACKUP または ALTER DATABASE DATAFILE ... END BACKUP を使用します。

注意： 影響を受けるファイルがバックアップからリストアされている場合は、ALTER DATABASE END BACKUP を使用しないでください。

RECOVER コマンドを使用したバックアップ・モードの終了

オンライン・バックアップが失敗した場合の対処方法は、ALTER DATABASE END BACKUP 文のみではありません。RECOVER コマンドを実行することもできます。他のユーザーがバックアップをリストアしているかどうか分からないときには、この方法が便利です。他のユーザーが実際にバックアップをリストアしていた場合には、RECOVER コマンドによりバックアップが最新に更新されるためです。ALTER DATABASE END BACKUP または ALTER TABLESPACE ... END BACKUP 文は、ファイルが現行のものであることが確実な場合にのみ実行してください。

注意： オンライン・バックアップが開始されてから以降に生成された REDO をスキャンする必要があるため、RECOVER コマンドを使用した方法には時間がかかります。

RECOVER コマンドを使用して表領域のバックアップ・モードを終了する手順は、次のとおりです。

1. データベースをマウントします。たとえば、次のように入力します。

```
SQL> STARTUP MOUNT
```

2. データベースを通常にリカバリします。たとえば、次のように入力します。

```
SQL> RECOVER DATABASE
```

3. V\$BACKUP ビューを使用して、アクティブなデータ・ファイルがないことを確認します。

```
SQL> SELECT * FROM V$BACKUP WHERE STATUS = 'ACTIVE';
FILE#          STATUS          CHANGE#      TIME
-----
0 rows selected.
```

関連項目： データベースのリカバリに関する情報は、[第4章「ユーザー管理のメディア・リカバリの実行」](#)を参照してください。

読取り専用表領域のユーザー管理バックアップの作成

オンラインの読取り専用表領域をバックアップする場合は、オンライン・データ・ファイルを単純にバックアップできます。システムはデータ・ファイルに対する変更を許可するため、表領域をバックアップ・モードに設定する必要はありません。

読取り専用表領域のセットが自己完結型の場合には、オペレーティング・システム・コマンドを使用して表領域をバックアップする他に、トランスポートابل表領域機能を使用して、表領域のメタデータをエクスポートすることもできます。メディア・エラーまたはユーザー・エラー（たとえば読取り専用表領域中の表を誤って削除するなど）の場合には、表領域をデータベースに移動して戻すことができます。

関連項目： 表領域の移動の方法については、『Oracle9i データベース管理者ガイド』を参照してください。

オープン状態のデータベース中のオンラインの読取り専用表領域をバックアップする手順は、次のとおりです。

1. DBA_TABLESPACES ビューを問い合せて、読取り専用の表領域を確認します。たとえば、次の問合せを実行します。

```
SELECT TABLESPACE_NAME, STATUS
FROM DBA_TABLESPACES
WHERE STATUS = 'READ ONLY';
```

2. 読取り専用表領域のバックアップを開始する前に、DBA_DATA_FILES データ・ディクショナリ・ビューを問い合せて、表領域のすべてのデータ・ファイルを確認します。たとえば、history 表領域をバックアップするとします。次のように入力します。

```
SELECT TABLESPACE_NAME, FILE_NAME
FROM SYS.DBA_DATA_FILES
WHERE TABLESPACE_NAME = 'HISTORY';
```

TABLESPACE_NAME	FILE_NAME
HISTORY	/oracle/dbs/tbs_hist1.f
HISTORY	/oracle/dbs/tbs_hist2.f

この例の /oracle/dbs/tbs_hist1.f および /oracle/dbs/tbs_hist2.f は、history 表領域のデータ・ファイルに対応する完全指定のファイル名です。

3. オペレーティング・システム・コマンドを使用して、読取り専用表領域のオンライン・データ・ファイルをバックアップします。ユーザーによる読取り専用表領域への変更は自動的に防止されるため、表領域をオフラインにしたり、表領域をバックアップ・モードにする必要はありません。たとえば、UNIX ユーザーは次のように入力できます。

```
% cp /oracle/dbs/tbs_hist*.f /backup
```

注意： 読取り専用表領域のバックアップをリストアする場合は、表領域をオフラインにし、データ・ファイルをリストアした後、表領域をオンラインにします。読取り専用表領域がバックアップ後に読取り / 書込みにされている場合にも、読取り専用表領域のバックアップは使用可能ですが、リストアされたバックアップでリカバリが必要になります。

4. 必要に応じて、読取り専用表領域のメタデータをエクスポートします。トランスポート表領域機能を使用すると、メディア障害またはユーザー・エラーの場合に、データ・ファイルを迅速にリストアし、メタデータをインポートできます。たとえば、次のように、表領域 history のメタデータをエクスポートします。

```
% exp TRANSPORT_TABLESPACE=y TABLESPACES=(history) FILE=/oracle/backup/tbs_hist.dmp
```

関連項目： DBA_DATA_FILES および DBA_TABLESPACES ビューの詳細は、『Oracle9i データベース・リファレンス』を参照してください。

UNDO 表領域のユーザー管理バックアップの作成

Oracle9i より前のリリースでは、UNDO 領域の管理は、ロールバック・セグメントに基づいて行われていました。この方式は**手動 UNDO 管理モード**と呼ばれています。Oracle9i では、データベースを**自動 UNDO 管理モード**に設定するオプションを使用できます。この設計では、静的に割り当てられたロールバック・セグメントのセットに UNDO 領域を分散するかわりに、1 つの UNDO 表領域に UNDO 領域を割り当てます。

UNDO 表領域のバックアップの手順は、他の読取り / 書込み表領域のバックアップの場合とまったく同じです。自動 UNDO 表領域はリカバリおよび読込み一貫性のために非常に重要であるため、手動 UNDO 管理モードで実行しているときに、ロールバック・セグメントを含む表領域について行うのと同じように、この表領域を頻繁にバックアップする必要があります。

データベースのオープン中に UNDO 表領域内のデータ・ファイルが失われた場合に、バックアップがないと、コミットされていない変更を含むオブジェクトを問い合わせたときにエラー・メッセージが表示されることがあります。また、インスタンス障害が発生したときには、コミットされていないトランザクションを元の値にロールバックできません。

関連項目： UNDO 領域の管理方法の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

SUSPEND モードでのユーザー管理バックアップの作成

この項で説明する項目は、次のとおりです。

- [一時停止 / 再開機能](#)
- [一時停止中のデータベースのバックアップの作成](#)

一時停止 / 再開機能

サード・パーティ・ツールには、ディスクまたは論理デバイスのセットをミラー化（つまりプライマリ・データの正確な複製を別の場所に保持）し、後でミラーを分割できるものがあります。ミラーの分割では、コピーが分離されるため、それぞれを別々に使用できます。

SUSPEND/RESUME の機能を使用すると、データベースに対する I/O を一時停止した後、ミラーを分割し、分割されたミラーのバックアップを作成できます。バックアップ・モード機能を補完するこの機能を使用することにより、I/O が新しく実行されないように、データベース I/O を一時停止できます。その後、一時停止中のデータベースにアクセスし、I/O に妨害されることなくバックアップを作成できます。

ほとんどの場合は、ミラーの分割によるバックアップを作成するために SUSPEND/RESUME を使用する必要はありませんが、使用しているシステムで、ボリュームを分割する前にデータベース・キャッシュの**使用済みバッファ**を解放しておく必要がある場合には、このコマンドが必要になります。

注意： RAID デバイスの中には、分割操作の実行中に書き込みを一時停止することによるメリットを享受できるものがあります。使用しているシステムでこの機能のメリットを享受できるかどうか、RAID のベンダーに確認してください。

ALTER SYSTEM SUSPEND 文は、データ・ファイル・ヘッダー、データ・ファイルおよび制御ファイルに対する I/O を停止して、データベースを一時停止します。データベースが一時停止されると、事前に存在していた I/O 操作はすべて完了できますが、データベースに対する新規の I/O アクセスの試みはキューに入れられます。

ALTER SYSTEM SUSPEND 文および ALTER SYSTEM RESUME 文は、インスタンスのみではなく、データベースに対して実行されます。ALTER SYSTEM SUSPEND 文が Oracle Real Application Clusters 構成中の 1 つのシステムで入力された場合には、内部のロッキング・メカニズムにより全インスタンスに停止要求が伝播され、クラスタ内のアクティブなインスタンスのすべてで I/O 操作が一時停止されます。

一時停止中のデータベースのバックアップの作成

データベースが正常に一時停止した後、ディスクにデータベースをバックアップするか、ミラー化を解除できます。データベースを一時停止しても I/O が即時に終了されるとはかぎらないため、表領域をバックアップ・モードにするために、ALTER SYSTEM SUSPEND 文の前に BEGIN BACKUP 文を使用することをお勧めします。

分割されたミラーをバックアップするには、従来からのユーザー管理のバックアップ方式を使用する必要があります。データ・ファイル・ヘッダーの読取りが必要になるため、Recovery Manager はデータベースのバックアップまたはコピーを作成できません。データベースのバックアップが終了するか、ミラーの分割が完了した後、ALTER SYSTEM RESUME 文を使用して、通常のデータベース操作を再開できます。

この間はデータベースにアクセスできないため、ミラーを分割せずに一時停止中のデータベースをバックアップすると、データベースが長期間停止する可能性があります。ミラーを分割してバックアップを実行すれば、停止は短時間で済みます。停止時間は、フラッシュするキャッシュのサイズ、データ・ファイルの数、およびミラー化の解除に必要な時間によって異なります。

SUSPEND/RESUME 機能には次の制限があります。

- Oracle Real Application Clusters 構成では、オリジナル・ノードの一時停止中に新規インスタンスを起動しないでください。
- ALTER SYSTEM SUSPEND 文または ALTER SYSTEM RESUME 文からはチェックポイントは開始されません。
- データベースの一時停止中は、IMMEDIATE、NORMAL または TRANSACTIONAL オプションを指定して SHUTDOWN コマンドを発行できません。
- すでに一時停止中のデータベースに対して SHUTDOWN ABORT を発行すると、データベースが再度アクティブになります。この操作は、メディア・リカバリまたはクラッシュ・リカバリがハングすることを防止します。

SUSPEND モードでミラーの分割によるバックアップを作成する手順は、次のとおりです。

1. データベースの表領域をバックアップ・モードにします。たとえば、表領域 users をバックアップ・モードにするには、次のように入力します。

```
ALTER TABLESPACE users BEGIN BACKUP;
```

2. ミラー化システムで、ディスクの書き込み中にミラーの分割に問題が発生する場合には、データベースを一時停止します。たとえば、次のコマンドを発行します。

```
ALTER SYSTEM SUSPEND;
```

3. V\$INSTANCE を問い合せて、データベースが一時停止されていることを確認します。たとえば、次のように入力します。

```
SELECT DATABASE_STATUS FROM V$INSTANCE;
```

```
DATABASE_STATUS
-----
SUSPENDED
```

4. オペレーティング・システムまたはハードウェア・レベルでミラーを分割します。
5. データベースの一時停止を終了します。たとえば、次の文を発行します。

```
ALTER SYSTEM RESUME;
```

6. V\$INSTANCE を問い合せて、データベースがアクティブなことを確認します。たとえば、次のように入力します。

```
SELECT DATABASE_STATUS FROM V$INSTANCE;
```

```
DATABASE_STATUS
-----
ACTIVE
```

7. 指定された表領域のバックアップ・モードを終了します。たとえば、表領域 `users` のバックアップ・モードを終了するには、次のように入力します。

```
ALTER TABLESPACE users END BACKUP;
```

8. バックアップのために、通常どおりに制御ファイルをコピーし、オンライン REDO ログをアーカイブします。

注意： 表領域をバックアップ・モードに設定する代替方法として `ALTER SYSTEM SUSPEND` 文を使用しないでください。

関連項目： `SUSPEND/RESUME` 機能の詳細は、『Oracle9i データベース管理者ガイド』を参照し、`ALTER SYSTEM` 文の詳細は、『Oracle9i SQL リファレンス』を参照してください。

制御ファイルのユーザー管理バックアップの作成

`ARCHIVELOG` モードで稼働中のデータベースの構造変更を行った後で、データベースの制御ファイルをバックアップします。データベースの制御ファイルをバックアップするには、`ALTER DATABASE` システム権限が必要です。

制御ファイルをバックアップする場合のオプションは次のとおりです。

- [制御ファイルのバイナリ・ファイルへのバックアップ](#)
- [制御ファイルのトレース・ファイルへのバックアップ](#)

制御ファイルのバイナリ・ファイルへのバックアップ

制御ファイルをバックアップする第一の手段は、SQL 文を使用してバイナリ・ファイルを作成する方法です。バイナリ・バックアップにはアーカイブ・ログ履歴、読取り専用およびオフライン表領域のオフライン範囲、およびバックアップ・セットとコピー（`Recovery Manager` を使用する場合）などの追加情報が含まれるため、トレース・ファイル・バックアップよりもバイナリ・バックアップをお勧めします。制御ファイルのバイナリ・バックアップには、一時ファイル・エントリが含まれないことに注意してください。

構造変更の後に制御ファイルをバックアップする手順は、次のとおりです。

1. データベースを変更します。たとえば、新規の表領域を作成します。

```
CREATE TABLESPACE tbs_1 DATAFILE 'file_1.f' SIZE 10M;
```

2. 出力バイナリ・ファイルのファイル名を指定して、データベースの制御ファイルをバックアップします。次の SQL 文は、/oracle/backup/cf.bak にデータベースの制御ファイルをバックアップします。

```
ALTER DATABASE BACKUP CONTROLFILE TO '/oracle/backup/cf.bak' REUSE;
```

REUSE オプションを指定すると、新規の制御ファイルで現在存在している制御ファイルを上書きできます。

制御ファイルのトレース・ファイルへのバックアップ

ALTER DATABASE BACKUP CONTROLFILE 文の TRACE オプションは、制御ファイルの管理およびリカバリの助けとなります。TRACE オプションは、バイナリ・バックアップを生成するかわりに、データベースのトレース・ファイルに SQL 文を書き込むように Oracle に指示します。トレース・ファイル中の文は、データベースの起動、制御ファイルの再作成、データベースのリカバリおよびオープンを適切に実行します。

制御ファイルをトレース・ファイルにバックアップするには、データベースをマウントまたはオープンし、次の SQL 文を発行します。

```
ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
```

SQL 文の中で RESETLOGS オプションも NORESETLOGS オプションも指定しなかった場合の出力は、CREATE CONTROLFILE ... NORESETLOGS 文を含むトレース・ファイルになります。制御ファイルのバイナリ・バックアップの場合と同じように、一時ファイル・エントリはトレース出力に含まれません。

関連項目： CREATE CONTROLFILE 文に含まれる、読取り専用ファイル、NORMAL モードでオフラインされたファイルおよび一時ファイルに関連する特殊な問題については、4-38 ページの「[再作成された制御ファイルを使用した、読取り専用ファイルのリカバリ](#)」を参照してください。

制御ファイルのトレース・ファイルへのバックアップ: 例

`sales` データベースの制御ファイルを再作成するスクリプトを生成する場合を考えてみます。データベースの特性は次のとおりです。

- 3つのスレッドが有効になっています。スレッド2はパブリックで、スレッド3はプライベートです。
- REDO ログは、それぞれが2つのメンバーを含む3つのグループに多重化されています。
- データベースには次のデータ・ファイルが含まれています。
 - `/diska/prod/sales/db/filea.dbf` (オンライン表領域の中のオフライン・データ・ファイル)
 - `/diska/prod/sales/db/database1.dbf` (SYSTEM 表領域の中のオンライン・データ・ファイル)
 - `/diska/prod/sales/db/fileb.dbf` (読取り専用表領域の中の唯一のファイル)

次の文を発行して、`CREATE CONTROLFILE ... NORESETLOGS` 文を含むトレース・ファイルを作成します。

```
ALTER DATABASE BACKUP CONTROLFILE TO TRACE NORESETLOGS;
```

次に、トレース・ファイルを編集し、トレース・ファイルの生成時点で現行のものであった制御ファイルに基づいて、`sales` データベースの新しい制御ファイルを作成するスクリプトを作成します。`NORMAL` モードでオフラインにされた表領域または読取り専用表領域をリカバリしないように、トレース・ファイルの中の `CREATE CONTROLFILE` 文の外部で編集してください。再作成された制御ファイルを使用してデータベースをオープンすると、ディスクジョナリ・チェック・コードではこれら省略されたファイルは `MISSING` としてマークされます。`ALTER DATABASE RENAME FILE` 文を実行すると、オリジナルのファイル名に名前が戻されます。

たとえば、トレース・ファイルの中の `CREATE CONTROLFILE ... NORESETLOGS` スクリプトを次のように編集して、`MISSING` というラベルが付けられたファイルの名前を変更できます。

```
# The following statements will create a new control file and use it to open the database.
# Log history and RMAN metadata will be lost. Additional logs may be required for media
# recovery of offline datafiles. Use this only if the current version of all online logs
# are available.
```

```
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE SALES NORESETLOGS ARCHIVELOG
    MAXLOGFILES 32
    MAXLOGMEMBERS 2
    MAXDATAFILES 32
    MAXINSTANCES 16
    MAXLOGHISTORY 1600
```

```
LOGFILE
  GROUP 1
    '/diska/prod/sales/db/log1t1.dbf',
    '/diskb/prod/sales/db/log1t2.dbf'
  ) SIZE 100K
GROUP 2
  '/diska/prod/sales/db/log2t1.dbf',
  '/diskb/prod/sales/db/log2t2.dbf'
) SIZE 100K,
GROUP 3
  '/diska/prod/sales/db/log3t1.dbf',
  '/diskb/prod/sales/db/log3t2.dbf'
) SIZE 100K
DATAFILE
  '/diska/prod/sales/db/database1.dbf',
  '/diskb/prod/sales/db/filea.dbf'
;

# This datafile is offline, but its tablespace is online. Take the datafile offline
# manually.
ALTER DATABASE DATAFILE '/diska/prod/sales/db/filea.dbf' OFFLINE;

# Recovery is required if any datafiles are restored backups,
# or if the most recent shutdown was not normal or immediate.
RECOVER DATABASE;

# All redo logs need archiving and a log switch is needed.
ALTER SYSTEM ARCHIVE LOG ALL;

# The database can now be opened normally.
ALTER DATABASE OPEN;

# The backup control file does not list read-only and normal offline tablespaces so that
# Oracle can avoid performing recovery on them. Oracle checks the data dictionary and
# finds information on these absent files and marks them 'MISSINGxxxx'. It then renames
# the missing files to acknowledge them without having to recover them.
ALTER DATABASE RENAME FILE 'MISSING0002'
  TO '/diska/prod/sales/db/fileb.dbf';
```

アーカイブ REDO ログのユーザー管理バックアップの作成

プライマリのアーカイブ場所のディスク領域を節約するために、アーカイブ・ログをテープまたは代替のディスクの場所にバックアップできます。複数の場所にアーカイブするときには、各ログ順序番号の1つのコピーのみバックアップします。

アーカイブ REDO ログをバックアップする手順は、次のとおりです。

1. V\$ARCHIVED_LOG を問い合せて、データベースから生成されたアーカイブ REDO ログ・ファイルを確認します。たとえば、次の問合せを実行します。

```
SELECT THREAD#,SEQUENCE#,NAME  
FROM V$ARCHIVED_LOG;
```

2. オペレーティング・システム・ユーティリティを使用して、各ログ順序番号で1つのコピーをバックアップします。この例では、プライマリのアーカイブ場所からログのバックアップ専用のディスクにすべてのログをバックアップします。

```
% cp /oracle/dbs/arc_dest/* /disk7/log_backups
```

関連項目： データ・ディクショナリ・ビューの詳細は、『Oracle9i データベース・リファレンス』を参照してください。

RAW デバイスへのユーザー管理バックアップの作成

RAW デバイスとは、ファイル・システムを持たないディスクまたはパーティションです。つまり、RAW デバイスに保管できるファイルは1つのみです。RAW デバイスにファイルをバックアップする場合は、オペレーティング・システム固有の問題が発生します。この後の項では、Oracle で最も一般的な2つのオペレーティング・システムである UNIX および Windows NT での問題のいくつかを説明します。

関連項目： Oracle Real Application Clusters に関連する、RAW デバイスの概要については、『Oracle9i Real Application Clusters セットアップおよび構成』を参照してください。

UNIX での RAW デバイスへのバックアップ

RAW デバイスとの間でバックアップを行う場合は、UNIX の `dd` コマンドが最も一般的なバックアップ・ユーティリティになります。このユーティリティの詳細は、オペレーティング・システムのマニュアルを参照してください。

`dd` を使用する場合の最も重要なことは、どのオプションを指定するかを決定することです。次の情報を確認してください。

データ	説明
ブロック・サイズ	<code>dd</code> がデータのコピーに使用するバッファのサイズを指定できます。たとえば、 <code>dd</code> で 8 KB 単位または 64 KB 単位でデータをコピーするように指定できます。 <code>dd</code> のブロック・サイズは、Oracle のブロック・サイズまたはオペレーティング・システムのブロック・サイズに対応している必要はありません。これは、 <code>dd</code> がコピーを行うときに使用するバッファのサイズにすぎません。
RAW オフセット	システムによっては、RAW デバイス上のファイルの最初の部分が、オペレーティング・システムの使用のために確保されていることがあります。この記憶領域を RAW オフセット と呼びます。これらのバイトをバックアップまたはリストアしないでください。
Oracle ブロック 0 (ゼロ) のサイズ	すべての Oracle ファイルの最初の部分には、オペレーティング・システム固有のコードによって、 ブロック 0 (ゼロ) と呼ばれる Oracle ブロックが配置されます。Oracle 汎用コードはこのブロックを認識しませんが、このブロックは、オペレーティング・システム上でファイルのサイズに含まれます。一般的にこのブロックは、ファイル中の他の Oracle ブロックと同じサイズになります。

前述の表の情報を使用して、[表 2-1](#) で指定された `dd` オプションを設定できます。

表 2-1 `dd` コマンドのオプション

オプション	指定対象
<code>if</code>	入力ファイル、つまり読み込むファイルの名前。
<code>of</code>	出力ファイル、つまり書き出すファイルの名前。
<code>bs</code>	<code>dd</code> がデータのコピーに使用するバッファ・サイズ。
<code>skip</code>	RAW オフセットが存在する場合に、入力 RAW デバイスでスキップする <code>dd</code> バッファの数。たとえば、64 KB の RAW オフセットを持つ RAW デバイス上のファイルをバックアップする場合、 <code>dd</code> バッファ・サイズが 8 KB であれば、 <code>skip=8</code> と指定すると、64 KB のオフセットからコピーを開始できます。

表 2-1 dd コマンドのオプション (続き)

オプション	指定対象
seek	RAW オフセットが存在する場合に、出力 RAW デバイスでスキップする dd バッファの数。たとえば、64 KB の RAW オフセットを持つ RAW デバイスにファイルをバックアップする場合、dd バッファ・サイズが 8 KB であれば、skip=8 と指定すると、64 KB のオフセットからコピーを開始できます。
count	dd でコピーする、入力 RAW デバイス上のブロックの数。RAW デバイスからファイル・システムにコピーするときには、コピーする正確なブロック数を指定するのが最良です。そうでない場合、Oracle データ・ファイルで使 されない、RAW ボリュームの最後の余分な領域がファイル・システムにコ ピーされることになります。 入力ファイルの合計サイズにブロック 0 (ゼロ) を必ず含めてください。た とえば、dd ブロック・サイズが 8 KB の場合に、30720 KB のデータ・フ ァイルをバックアップするときには、count=3841 と設定できます。count の値は実際は 30728KB をバックアップします。追加の 8KB は Oracle ブロ ック 0 (ゼロ) 用です。

RAW デバイスはバックアップの入力または出力デバイスに使用できるため、4 つのバックアップ・シナリオを想定できます。dd で選択できるオプションは、表 2-2 に示すように、どのシナリオを選択するかによって異なります。

表 2-2 dd バックアップのシナリオ

バックアップ元	バックアップ先	dd コマンドで指定されるオプション
RAW デバイス	RAW デバイス	if、of、bs、skip、seek、count
RAW デバイス	ファイル・システム	if、of、bs、skip、count
ファイル・システム	RAW デバイス	if、of、bs、seek
ファイル・システム	ファイル・システム	if、of、bs

UNIX での dd ユーティリティを使用したバックアップ例

これらの dd ユーティリティの使用方法的例では、次のように想定しています。

- 30720 KB のデータ・ファイルをバックアップします。
- データ・ファイルの最初には 8 KB のブロック 0 (ゼロ) があります。
- RAW オフセットは 64 KB です。
- コピーに RAW デバイスが関係するときには、dd ブロック・サイズを 8 KB に設定します。

この例では、1 つの RAW デバイスから別の RAW デバイスにバックアップします。

```
% dd if=/dev/rsd1b of=/dev/rsd2b bs=8k skip=8 seek=8 count=3841
```

この例では、RAW デバイスからファイル・システムにバックアップします。

```
% dd if=/dev/rsd1b of=/backup/df1.dbf bs=8k skip=8 count=3841
```

この例では、ファイル・システムから RAW デバイスにバックアップします。

```
% dd if=/backup/df1.dbf of=/dev/rsd2b bs=8k seek=8
```

この例では、ファイル・システムからファイル・システムにバックアップします。このため、ブロック・サイズを大きく設定して、I/O パフォーマンスを高めることができます。

```
% dd if=/oracle/dbs/df1.dbf of=/backup/df1.dbf bs=1024k
```

Windows NT での RAW デバイスへのバックアップ

Windows NT は UNIX と同じように RAW ディスク・パーティションをサポートするため、Oracle はデータ・ファイル、オンライン・ログおよび制御ファイルをこの中に格納できます。各 RAW パーティションにはドライブ名または物理ドライブ番号が割り当てられます。ファイル・システムは含まれません。UNIX の場合と同じように、NT でも各 RAW パーティションは 1 つのファイルにマップされます。

NT では、Oracle ファイルのネーミング規則は UNIX の場合と異なります。NT の場合、RAW データ・ファイル名は次のような形式になります。

```
%%.¥drive_letter:  
%%.¥PHYSICALDRIVEdrive_number
```

たとえば、次の名前は RAW ファイル名として使用できます。

```
%%.¥G:  
%%.¥PHYSICALDRIVE3
```

また、RAW ファイル名の別名も作成できます。Oracle の標準インストールでは、%%.¥PHYSICALDRIVE3 などのファイル名をポイントする、%%.¥Datafile12 などの別名を作成できる、SETLINKS ユーティリティが提供されます。

RAW データ・ファイルのユーザー管理バックアップを作成する手順は、NT から提供される copy.exe または ntbackup.exe ユーティリティを使用するかわりに Oracle の OCOPY ユーティリティを使用する必要があることを除き、NT ファイル・システムでのファイルのコピーと基本的に同じです。また、MKS ユーティリティがある場合には、dd ユーティリティを使用できます。OCOPY は 64 ビットのファイル I/O、物理 RAW ドライブおよび RAW ファイルをサポートします。OCOPY はテープに直接バックアップできません。

OCOPY のオンライン・マニュアルを表示するには、Windows NT プロンプトに OCOPY とだけ入力します。出力のサンプルは次のようになります。

```
Usage of OCOPY:
  ocopy from_file [to_file [a | size_1 [size_n]]]
  ocopy -b from_file to_drive
  ocopy -r from_drive to_dir
```

次の表は、OCOPY の重要なオプションを示しています。

オプション	指定対象
b	入力ファイルを複数の出力ファイルに分割します。このオプションは、入力ファイルよりも小さいデバイスにバックアップするときに便利です。
r	複数の入力ファイルを組み合わせて、1 つの出力ファイルに書き込みます。このオプションは、-b オプションを指定して作成したバックアップをリストアするときに便利です。

OCOPY を使用したバックアップ例

この例では、次のように想定しています。

- データ・ファイル 12 は RAW パーティション ¥¥.¥G: にマウントされています。
- C: ドライブにファイル・システムをマウントしています。
- データベースはオープンしています。

RAW パーティション ¥¥.¥G: 上のデータ・ファイルをローカル・ファイル・システムにバックアップするには、データ・ファイル 12 をバックアップ・モードにした後で、NT プロンプトで次のコマンドを実行します。

```
OCOPY "¥¥.G:" C:¥¥backup¥datafile12.bak
```

OCOPY の -b および -r オプションの指定例

この例では、次のように想定しています。

- ¥¥.¥G: はデータ・ファイル 7 を含む RAW パーティションです。
- A: ドライブは取り外し可能なディスク・ドライブです。
- データベースはオープンしています。

データ・ファイルをドライブ A: にバックアップするには、データ・ファイル 7 をバックアップ・モードにした後で、NT プロンプトで次のコマンドを実行します。

```
# first argument is filename, second argument is drive
OCOPY -b "¥¥.¥G:" A:¥
```

ドライブ A: が満杯の場合、別のディスクを使用できます。この方法で、データ・ファイル 1 のバックアップを複数のファイルに分割できます。

同様に、バックアップをリストアするときには、データ・ファイル 7 を含む表領域をオフラインにし、次のコマンドを実行します。

```
# first argument is drive, second argument is directory
OCOPY -r A:¥ "¥¥.¥G:"
```

ユーザー管理バックアップの検証

バックアップがリカバリに使用可能なものであることを確認するために、バックアップを定期的に検証する必要があります。この項で説明する項目は、次のとおりです。

- [バックアップのリストアのテスト](#)
- [DBVERIFY ユーティリティの使用](#)

バックアップのリストアのテスト

バックアップの有効性を確認する最良の方法は、別のホストにバックアップをリストアし、必要な場合はメディア・リカバリを実行し、データベースをオープンしてみることです。このオプションでは、リストア手順のために別のホストを使用する必要があります。

関連項目：

- データ・ファイルのリストア方法については、3-7 ページの「[データ・ファイルのリストア](#)」を参照してください。
- 制御ファイルのリストア方法については、3-9 ページの「[制御ファイルのリストアおよび再作成](#)」を参照してください。
- アーカイブ REDO ログのリストア方法については、3-17 ページの「[アーカイブ REDO ログのリストア](#)」を参照してください。
- ファイルのリカバリの方法については、4-10 ページの「[ユーザー管理の完全メディア・リカバリの実行](#)」を参照してください。

DBVERIFY ユーティリティの使用

DBVERIFY プログラムは、オフライン・データ・ファイルに対して、物理的なデータ構造の整合性チェックを実行する、外部のコマンドライン・ユーティリティです。DBVERIFY は、バックアップのデータ・ファイルをリストアする前にデータ・ファイルのユーザー管理バックアップの有効性を確認する必要がある場合、またはデータ破損の問題を検出した場合の診断の補助として主に使用されます。

DBVERIFY の名前と位置はオペレーティング・システムごとに異なります。たとえば、UNIX 上でデータ・ファイル `tbs_52.f` の整合性チェックを実行するには、次のように `dbv` コマンドを実行できます。

```
% dbv file=tbs_52.f
```

`dbv` の出力のサンプルは次のようになります。

```
DBVERIFY: Release 9.2.0.0.0
```

```
(c) Copyright 2000 Oracle Corporation. All rights reserved.
```

```
DBVERIFY - Verification starting : FILE = tbs_52.f
```

```
DBVERIFY - Verification complete
```

```
Total Pages Examined          : 250
Total Pages Processed (Data)   : 4
Total Pages Failing (Data)     : 0
Total Pages Processed (Index) : 15
Total Pages Failing (Index)    : 0
Total Pages Processed (Other) : 29
Total Pages Empty              : 202
Total Pages Marked Corrupt     : 0
Total Pages Influx             : 0
```

関連項目： DBVERIFY 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。

エクスポートを使用した論理バックアップの作成

エクスポートおよびインポートは、Oracle データベースへ、あるいは Oracle データベースから Oracle データを移動するユーティリティです。エクスポートは、Oracle データベースからオペレーティング・システム・ファイルに特殊なバイナリ形式でデータを書き出します。インポートはエクスポート・ファイルを読み取り、既存のデータベース内に対応する情報をリストアします。エクスポートおよびインポートは Oracle データを移動するために設計されたものですが、物理データベースのバックアップを補うために使用することもできます。

この項では、次の項目を含む、インポートおよびエクスポート・ユーティリティについて説明します。

- [エクスポートの使用](#)
- [インポートの使用](#)

関連項目： エクスポートおよびインポート・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。

エクスポートの使用

エクスポート・ユーティリティは、データベースがオープンし、使用可能である間に、論理データベース・オブジェクトをバックアップできます。このユーティリティは、データベースのオブジェクトの読み込み一貫性ビューをオペレーティング・システム・ファイルに書き込みます。システム監査オプションはエクスポートされません。

注意： 論理バックアップを実行するためにエクスポート・ユーティリティを使用する場合は、バックアップが単一の時点を反映したものになるように、すべてのデータを論理的に一貫した方法でエクスポートする必要があります。エクスポートの実行中にデータベースが変更されないようにしてください。理想的には、一般ユーザーがデータにアクセスできないように、データのエクスポート中はデータベースを `ALTER SYSTEM QUIESCE RESTRICTED` モードで実行する必要があります。また、データのエクスポート前にデータベースを休止し、その後データベースを再起動することもできます。

表 2-3 は使用可能なエクスポート・モードを示しています。

表 2-3 エクスポート・モード

モード	説明
ユーザー（所有者）	ユーザーが所有するすべてのオブジェクトをエクスポートします。
表領域	表領域に含まれるすべてのオブジェクトをエクスポートします。
表	ユーザーが所有するすべての表または指定された表と、権限、トリガー、ビューおよび索引など、これらの表に定義されたオブジェクトをエクスポートします。
全データベース	データベースのすべてのオブジェクトをエクスポートします。

インポートの使用

インポート・ユーティリティは、以前に作成されたエクスポート・ファイルに保管されているデータベース情報をリストアできます。このユーティリティはエクスポートを補完するものです。

エクスポート・ファイルおよびインポート・ユーティリティを使用してデータベースをリカバリする手順は、次のとおりです。

- 1. すべての表領域およびユーザーを含めて、データベース構造を再作成します。再作成された構造はオブジェクトを含まないものである必要があります。
- 2. 適切なエクスポート・ファイルをインポートし、可能なかぎり最新の状態にデータベースをリストアします。エクスポート・スケジュールの実施方法によって、データベースのリストアに必要となるインポートの程度は様々です。

その他の Oracle ファイルのユーザー管理バックアップの作成

初期化パラメータ・ファイル、ネットワーク・ファイルと構成ファイル、およびパスワード・ファイルは必ずバックアップする必要があります。メディア障害によりこれらのファイルが破壊された場合は、元の環境を再作成するのは困難です。たとえば、データベース・ファイルおよびサーバー・パラメータ・ファイルをバックアップしても、ネットワーク・ファイル（たとえば `tnsnames.ora` および `listener.ora`）がバックアップされていないければ、データベースのリストアおよびリカバリは可能であっても、ネットワーク・ファイルを再作成するまでは Oracle Net によってユーザーを認証できなくなります。

一般的なルールとして、その他の Oracle ファイルは変更があるたびにバックアップする必要があります。たとえば、データベースへのアクセスに使用できるネット・サービス名を追加または変更した場合は、`tnsnames.ora` ファイルの新しいバックアップを作成します。

構成ファイルを検索する最も簡単な方法は、Oracle ホーム・ディレクトリから開始し、`.ora` 拡張子で終了するすべてのファイルを再帰的に検索することです。たとえば、UNIX の場合は次のコマンドを実行します。

```
% find $ORACLE_HOME -name "*.ora" -print
```

構成ファイルをバックアップするには、サード・パーティのユーティリティを使用する必要があります。たとえば、UNIX の `cp` コマンドを使用して、次のように `tnsnames.ora` および `listener.ora` ファイルをバックアップできます。

```
% cp $ORACLE_HOME/network/admin/tnsnames.ora /disk2/bkups/tnsnames01-22-01.ora
% cp $ORACLE_HOME/network/admin/listener.ora /disk2/bkups/listener01-22-01.ora
```

サーバー・パラメータ・ファイルをバックアップするためにオペレーティング・システム・ユーティリティを使用することもできます。データベースを起動するためにサーバー・パラメータ・ファイルの特定のバージョンが存在している必要はありませんが、ファイルに対する変更が失われないように、このファイルの比較的新しいバックアップを保持しておく必要があります。サーバー・パラメータ・ファイルを失った場合は、いつでも新しいファイルを作成するか、クライアント側の初期化パラメータ・ファイル（`PFILE`）を使用してインスタンスを起動できます。

関連項目： サーバー・パラメータ・ファイルの管理およびエクスポート
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。

ユーザー管理のリストア操作の実行

この章では、データベースのリカバリの方法について説明します。この章で説明する項目は、次のとおりです。

- ユーザー管理のリストア操作
- リストアのシナリオで使用する情報の記録
- リカバリが必要なデータ・ファイルの判別
- データ・ファイルのリストア
- バックアップが利用できない場合のデータ・ファイルの再作成
- 制御ファイルのリストアおよび再作成
- アーカイブ REDO ログのリストア

ユーザー管理のリストア操作

ファイルのリストアとは、バックアップ・ファイルでファイルを置き換えることです。一般的には、メディア障害またはユーザー・エラーにより元のファイルが破損したか削除された場合に、ファイルをリストアします。リストア操作の候補となるファイルは次のとおりです。

- データ・ファイル
- 制御ファイル
- アーカイブ REDO ログ
- サーバー・パラメータ・ファイル

いずれのファイルの場合にも、プライマリ・ファイルの損害と、バックアップのリストアは、メディア・リカバリ上は次のような意味を持ちます。

消失したもの	対応
1 つ以上のデータ・ファイル	バックアップからリストアし、メディア・リカバリを実行する必要があります。データ・ファイル・ヘッダーの中のチェックポイント SCN が、制御ファイルの中に記録されたデータ・ファイルのチェックポイント SCN と一致しないときには、リカバリが必要です。
現行の制御ファイルのすべてのコピー	バックアップ制御ファイルをリストアし、RESETLOGS オプションを指定してデータベースをオープンする必要があります。 バックアップがない場合には、制御ファイルを再作成してください。可能であれば、ALTER DATABASE BACKUP CONTROLFILE TO TRACE 出力に入っているスクリプトを使用します。制御ファイルの構造を現在のデータベース構造と一致させるために追加作業が必要になることがあります。
多重制御ファイルの 1 つのコピー	多重制御ファイルの影響を受けていないファイルの 1 つを、破損または欠落した制御ファイルの場所にコピーし、データベースをオープンします。元の場所に制御ファイルをコピーできない場合には（たとえばディスク・ドライブを救済できないため）、新しい場所を反映するように初期化パラメータ・ファイルを編集します。その上で、データベースをオープンします。

消失したもの	対応
メディア・リカバリに必要な1つ以上のアーカイブ・ログ	メディア・リカバリを続行するには、これらのアーカイブ・ログのバックアップをリストアする必要があります。デフォルトの場所またはデフォルト以外の場所にリストアできます。バックアップがない場合には、欠落している最初のログの前まで不完全リカバリを実行し、RESETLOGS でオープンする必要があります。
サーバー・パラメータ・ファイル	サーバー・パラメータ・ファイルのバックアップがある場合には、これをリストアします。また、クライアント側の初期化パラメータ・ファイルのバックアップがある場合には、このファイルのバックアップをリストアし、インスタンスを起動した後、サーバー・パラメータ・ファイルを再作成できます。

注意： Oracle 管理ファイルのリストアおよびリカバリは、ユーザー命名ファイルのリストアおよびリカバリと同じです。

リストアのシナリオで使用する情報の記録

ユーザー管理のバックアップとリカバリの最も重要な点の1つは、現在のデータベース・ファイルおよびこれらのファイルのバックアップのすべてについて、記録を残しておくことです。たとえば、次のファイルの場所を記録しておく必要があります。

- データ・ファイル
- 制御ファイル
- オンライン REDO ログ（オンライン・ログはバックアップされることはありません）
- アーカイブ REDO ログ
- 初期化パラメータ・ファイル
- パスワード・ファイル
- ネットワーク関連ファイル

データ・ファイル、制御ファイルおよびオンライン REDO ログの場所の記録

次の便利な SQL スクリプトは、データベースのすべての制御ファイル、データ・ファイルおよびオンライン REDO ログ・ファイルの場所を表示します。

```
SELECT NAME FROM V$DATAFILE
UNION ALL
SELECT MEMBER FROM V$LOGFILE
UNION ALL
SELECT NAME FROM V$CONTROLFILE;
```

出力のサンプルは次のようになります。

```
NAME
-----
/oracle/dbs/tbs_01.f
/oracle/dbs/tbs_02.f
/oracle/dbs/tbs_11.f
/oracle/dbs/tbs_12.f
/oracle/dbs/t1_log1.f
/oracle/dbs/t1_log2.f
/oracle/dbs/cf1.f
/oracle/dbs/cf2.f
```

関連項目： V\$ ビューの詳細は、『Oracle9i データベース・リファレンス』を参照してください。

アーカイブ REDO ログの場所の記録

次の SQL スクリプトを実行すると、ログのアーカイブ先のデフォルト位置を確認できます。

```
SELECT NAME, VALUE
FROM V$PARAMETER
WHERE NAME LIKE log_archive_dest%
AND VALUE IS NOT NULL
/
```

NAME	VALUE

log_archive_dest_1	LOCATION=/oracle/work/arc_dest/arc
log_archive_dest_state_1	enable

次のように SHOW を実行して、アーカイブ・ログの書式を確認します。

```
SHOW PARAMETER LOG_ARCHIVE_FORMAT
```

NAME	TYPE	VALUE

log_archive_format	string	r_%t_%s.arc

制御ファイル内に記録されているすべてのアーカイブ・ログのリストを表示するには、次の問合せを発行します。

```
SELECT NAME FROM V$ARCHIVED_LOG;

NAME
-----
/oracle/work/arc_dest/arcr_1_110.a
/oracle/work/arc_dest/arcr_1_111.a
/oracle/work/arc_dest/arcr_1_112.a
/oracle/work/arc_dest/arcr_1_113.a
```

バックアップ・ファイルの場所の記録

バックアップ・ファイルの場所を記録するだけでは十分ではありません。バックアップ・ファイルとオリジナル・ファイルを関連付ける必要があります。可能であれば、バックアップ・ファイルにはプライマリ・ファイルと同じ相対ファイル名を付けてください。どのような命名方法を使用した場合にでも、関連情報を表にして保管してください。たとえば、リストアを行う事態に備えて、データベース・ファイルの場所の記録として次のような表を保管しておくことができます。

データ・ファイル 番号	表領域	バックアップ・ファイル名
0（制御ファイル）	0（制御ファイル）	/dsk3/backup/cf.f
1	SYSTEM	/dsk3/backup/tbs_01.f
2	undo	/dsk3/backup/tbs_02.f
3	temp	/dsk3/backup/tbs_11.f
4	users	/dsk3/backup/tbs_12.f

リカバリが必要なデータ・ファイルの判別

動的パフォーマンス・ビュー V\$RECOVER_FILE を使用して、メディア・リカバリの準備としてどのファイルをリストアするかを判別できます。このビューはリカバリの必要なすべてのファイルを表示し、リカバリの必要な理由を示します。

次の問合せは、メディア・リカバリの必要なデータ・ファイルのファイル ID 番号と、リカバリの理由（判明している場合）、およびリカバリを開始する必要がある SCN および時刻を表示します。

```
SELECT * FROM V$RECOVER_FILE;
```

FILE#	ONLINE	ERROR	CHANGE#	TIME
14	ONLINE			0
15	ONLINE	FILE NOT FOUND		0
21	OFFLINE	OFFLINE NORMAL		0

注意： 現在使用されている制御ファイルがリストアされたバックアップや、メディア障害の発生後に作成された新しい制御ファイルの場合には、このビューは役に立ちません。リストアされた制御ファイルまたは再作成された制御ファイルには、Oracle が V\$RECOVER_FILE に正確に表示するために必要とする情報が含まれていません。

リカバリが必要なデータ・ファイルのファイル名および表領域名を取得するために、V\$DATAFILE および V\$TABLESPACE を問い合わせます。たとえば、次のように入力します。

```
SELECT d.NAME, t.NAME AS tablespace_name
FROM V$DATAFILE d, V$TABLESPACE t
WHERE t.TS# = d.TS#
AND d.FILE# IN (14,15,21); # use values obtained from V$RECOVER_FILE query
```

NAME	TABLESPACE_NAME
/oracle/dbs/tbs_14.f	TBS_1
/oracle/dbs/tbs_15.f	TBS_2
/oracle/dbs/tbs_21.f	TBS_3

これらの問合せを、次の SQL*Plus スクリプトのように組み合わせることができます（出力のサンプルを次の例に示します）。

```
COL df# FORMAT 999
COL df_name FORMAT a20
COL tbs#_name FORMAT a10
COL status FORMAT a7
COL error FORMAT a10
```

```
SELECT r.FILE# AS df#, d.NAME AS df_name, t.NAME AS tbspc_name,
       d.STATUS, r.ERROR, r.CHANGE#, r.TIME
FROM V$RECOVER_FILE r, V$DATAFILE d, V$TABLESPACE t
WHERE t.TS# = d.TS#
AND d.FILE# = r.FILE#
/
```

出力のサンプルは次のようになります。

DF#	DF_NAME	TBSP_NAME	STATUS	ERROR	CHANGE#	TIME
14	/oracle/dbs/tbs_14.f	TBS_1	OFFLINE	OFFLINE NORMAL	0	
15	/oracle/dbs/tbs_15.f	TBS_2	OFFLINE	OFFLINE NORMAL	0	
21	/oracle/dbs/tbs_21.f	TBS_3	OFFLINE	OFFLINE NORMAL	0	

データ・ファイルのリストア

メディア障害によりデータベースの1つ以上のデータ・ファイルが永久に損傷したときには、損傷したファイルをリカバリする前に、これらのデータ・ファイルのバックアップをリストアする必要があります。破損したデータ・ファイルを元の場所にリストアできないときには（たとえばディスクを置き換える必要があるため、ファイルを別のディスクにリストアするなど）、これらのファイルの新しい場所を制御ファイルに反映する必要があります。

Oracle ファイルを RAW ディスクまたは RAW パーティションにリストアする場合の手順は、ファイル・システム上のファイルにリストアする場合と基本的に同じです。しかし、RAW デバイス上のファイルのネーミング規則（オペレーティング・システムによって異なる）を認識し、RAW デバイスをサポートするオペレーティング・システム・ユーティリティを使用する必要があります。

関連項目： RAW デバイスでファイルをバックアップおよびリストアする場合の考慮事項は、2-24 ページの「[RAW デバイスへのユーザー管理バックアップの作成](#)」を参照してください。

デフォルトの場所にバックアップ・データ・ファイルをリストアする手順は、次のとおりです。

1. 3-6 ページの「リカバリが必要なデータ・ファイルの判別」に記載された方法を使用して、リカバリが必要なデータ・ファイルを判別します。
2. データベースがオープンしている場合は、アクセスできないデータ・ファイルを含む表領域をオフライン化します。たとえば、次のように入力します。

```
ALTER TABLESPACE users OFFLINE IMMEDIATE;
```

3. オペレーティング・システム・コマンドを使用して、破損したデータ・ファイルのバックアップをデフォルトの場所にコピーします。たとえば、UNIX 上で `tbs_24.f` をリストアするには、次のコマンドを発行します。

```
% cp /disk2/backup/tbs_24.bak /disk1/oracle/dbs/tbs_24.f
```

4. 影響を受けた表領域のリカバリを行います。たとえば、次のように入力します。

```
RECOVER TABLESPACE users
```

5. リカバリした表領域をオンライン化します。たとえば、次のように入力します。

```
ALTER TABLESPACE users ONLINE;
```

バックアップが利用できない場合のデータ・ファイルの再作成

データ・ファイルが破損し、ファイルのバックアップが利用できない場合にも、次の条件が整えば、データ・ファイルのリカバリが可能です。

- 元のデータ・ファイルの作成後に書き込まれたすべてのアーカイブ・ログ・ファイルを利用できる
- 制御ファイルに破損したファイルの名前が含まれている（制御ファイルは現行か、破損したデータ・ファイルがデータベースに追加された後で作成されたバックアップである）

リカバリのためにデータ・ファイルを再作成する手順は、次のとおりです。

1. 対応するバックアップのない、破損したデータ・ファイルを置き換えるために、新しい空のデータ・ファイルを作成します。たとえば、データ・ファイル `/disk1/users1.f` が破損し、このファイルのバックアップがないとします。次の文は、元のデータ・ファイル（同じサイズ）を `disk2` 上に再作成します。

```
ALTER DATABASE CREATE DATAFILE '/disk1/users1.f' AS '/disk2/users1.f';
```

この文は、失われたファイルと同じサイズの、空のファイルを作成します。Oracle は制御ファイルおよびデータ・ディクショナリ内の情報を検索して、サイズ情報を入手します。古いデータ・ファイルは新しいデータ・ファイルとして改名されます。

2. 空のデータ・ファイルに対してメディア・リカバリを実行します。たとえば、次のように入力します。

```
RECOVER DATAFILE '/disk2/users1.f'
```

3. リカバリ時には、元のデータ・ファイルが作成された後で書き込まれたすべてのアーカイブ REDO ログをマウントし、損害を受けたデータ・ファイルの新しい空のバージョンに再適用する必要があります。

注意： 必要な REDO データを用意できないため、ALTER DATABASE 文の CREATE DATAFILE 句を使用して SYSTEM 表領域のデータ・ファイルを再作成することはできません。

制御ファイルのリストアおよび再作成

メディア障害がデータベースの制御ファイルに影響を与えたときには（制御ファイルが多重であってもなくても）、データベースは、Oracle バックグラウンド・プロセスが制御ファイルへのアクセスを必要とするまで稼働し続けます。この時点で、データベースおよびインスタンスは自動的に停止します。

メディア障害が一時的なものであり、データベースがまだ停止していない場合には、メディア障害を即時に解決して、データベースが自動的に停止されないようにします。しかし、一時的なメディア障害を解決する前にデータベースが停止したときには、問題を解決し、制御ファイルへのアクセスをリストアした後で、データベースを再起動できます。

データベースの制御ファイルへのアクセスを永続的に阻害するメディア障害の場合の適切なリカバリ手順は、制御ファイルを多重化しているかどうかで異なります。この後の項では、適切な手順について説明します。

- 多重制御ファイルの一部が損害を受けた場合
- すべての多重制御ファイルが損害を受けているが、バックアップが使用できる場合
- 現行の制御ファイルおよびバックアップ制御ファイルのすべてが損害を受けた場合

多重制御ファイルの一部が損害を受けた場合

永続的なメディア障害によりデータベースの 1 つ以上の制御ファイルが破損したが、メディア障害による損害を受けていない制御ファイルが少なくとも 1 つある場合には、データベースのリカバリに次の手順を使用します。

多重制御ファイルのデフォルトの場所へのコピー

損害を受けた制御ファイルを含むディスクおよびファイル・システムが影響を受けていないと仮定すると、影響を受けていない制御ファイルの 1 つを、欠落した制御ファイルの場所に単純にコピーできます。この場合は、CONTROL_FILES 初期化パラメータの設定を変更する必要はありません。

多重制御ファイルをコピーして破損した制御ファイルを置き換える手順は、次のとおりです。

1. インスタンスがまだ実行されている場合は、シャットダウンします。

```
SHUTDOWN ABORT
```

2. メディア障害の原因であるハードウェアの問題を解決します。ハードウェアの問題を迅速に解決できない場合は、3-11 ページの「[多重制御ファイルのデフォルト以外の場所へのコピー](#)」の説明に従って、破損した制御ファイルを代替の記憶デバイスにリストアしてデータベースのリカバリを進めることができます。
3. データベースの現行の制御ファイルの多重コピーの中の、影響を受けていないものを使用して、破損した制御ファイルの上にコピーします。たとえば、bad_cf.f を good_cf.f で置き換えるには、次のように入力します。

```
% cp /oracle/good_cf.f /oracle/dbs/bad_cf.f
```

4. 新しいインスタンスを起動し、データベースをマウントしてオープンします。たとえば、次のように入力します。

```
STARTUP
```

多重制御ファイルのデフォルト以外の場所へのコピー

損害を受けた制御ファイルを含むディスクおよびファイル・システムが影響を受けている場合には、正常な制御ファイルの1つを、欠落した制御ファイルの場所に単純にコピーすることはできません。この場合には、欠落した制御ファイルの新しい場所を示すように CONTROL_FILES 初期化パラメータを変更する必要があります。

デフォルト以外の場所に制御ファイルをリストアする手順は、次のとおりです。

1. インスタンスがまだ実行されている場合は、シャットダウンします。

```
SHUTDOWN ABORT
```

2. メディア障害の原因であるハードウェアの問題を解決できない場合は、影響を受けていない制御ファイルを代替の場所にコピーします。たとえば、good_cf.f を new_cf.f にコピーするには、次のコマンドを発行します。

```
% cp /oracle/dbs/good_cf.f /oracle/dbs/new_cf.f
```

3. CONTROL_FILES パラメータですべての制御ファイルの現在の場所が反映され、リストアされていないすべての制御ファイルが除外されるように、データベースのパラメータ・ファイルを編集します。たとえば、初期化パラメータ・ファイルの内容が次のようになっているとします。

```
CONTROL_FILES = '/oracle/dbs/good_cf.f', '/oracle/dbs/bad_cf.f'
```

次のように編集します。

```
CONTROL_FILES = '/oracle/dbs/good_cf.f', '/oracle/dbs/new_cf.f'
```

4. 新しいインスタンスを起動し、データベースをマウントしてオープンします。たとえば、SQL*Plus に次のように入力します。

```
STARTUP
```

すべての多重制御ファイルが損害を受けているが、バックアップが使用できる場合

永続的なメディア障害によりデータベースのすべての制御ファイルが損害を受けたが、制御ファイルのバックアップがある場合には、次の手順を使用して、バックアップ制御ファイルをリストアします。制御ファイルにアクセスできない場合、インスタンスは起動できますが、データベースをマウントできません。制御ファイルを使用できないときにデータベースをマウントしようとする、次のエラー・メッセージが表示されます。

ORA-00205: 制御ファイル識別エラー。詳細はアラート・ログをチェックしてください。

制御ファイルを再度アクセス可能にするまでは、データベースをマウントしてオープンすることはできません。バックアップ制御ファイルをリストアした後、RESETLOGS オプションを指定してデータベースをオープンする必要があります。

表 3-1 に示すように、制御ファイルのリストアの手順は、オンライン REDO ログを利用できるかどうかによって異なります。

表 3-1 制御ファイルが損害を受けた場合のシナリオ

オンライン・ログのステータス	データ・ファイルのステータス	対策
使用可能	現行	リカバリに必要な REDO がオンライン・ログに含まれている場合には、バックアップ制御ファイルをリストアし、リカバリ時にログを適用します。データベースをオープンするためには、変更を含むオンライン・ログのファイル名を指定する必要があります。リカバリ後に RESETLOGS でオープンします。
使用不可能	現行	リカバリに必要な REDO がオンライン・ログに含まれている場合には、制御ファイルを再作成する必要があります。ログにはアクセスできないため、RESETLOGS でオープンします。
使用可能	バックアップ	バックアップ制御ファイルをリストアし、完全リカバリを実行し、RESETLOGS でオープンします。
使用不可能	バックアップ	バックアップ制御ファイルをリストアし、不完全リカバリを実行し、RESETLOGS でオープンします。

デフォルトの場所へのバックアップ制御ファイルのリストア

可能であれば、元の場所に制御ファイルをリストアします。この方法では、初期化パラメータ・ファイルで制御ファイルの新しい場所を指定する必要はありません。

デフォルトの場所にバックアップ制御ファイルをリストアする手順は、次のとおりです。

1. インスタンスがまだ実行されている場合は、シャットダウンします。

```
SHUTDOWN ABORT
```

2. メディア障害の原因であるハードウェアの問題を解決します。

3. CONTROL FILES 初期化パラメータで指定されたすべての場所にバックアップ制御ファイルをリストアします。たとえば、サーバー・パラメータ・ファイルに記述された制御ファイルの場所が /dsk1/oracle/dbs/cf1.f および /dsk2/cf2.f の場合は、オペレーティング・システム・ユーティリティを使用して、これらの場所にバックアップ制御ファイルをリストアします。

```
% cp /backup/cf.bak /dsk1/oracle/dbs/cf1.f
% cp /backup/cf.bak /dsk2/cf2.f
```

4. 新しいインスタンスを起動し、データベースをマウントします。たとえば、次のように入力します。

```
STARTUP MOUNT
```

5. USING BACKUP CONTROLFILE 句を指定して RECOVER コマンドを実行し、リカバリを開始します。不完全リカバリを実行する場合は、UNTIL CANCEL を指定します。たとえば、次のように入力します。

```
RECOVER DATABASE USING BACKUP CONTROLFILE UNTIL CANCEL
```

6. プロンプトで指示されたアーカイブ・ログを適用します。必要なアーカイブ・ログがないことを告げる別のメッセージが表示された場合には、必要な REDO レコードがオンライン REDO ログに入っていることが考えられます。インスタンスがクラッシュしたときに、アーカイブされていない変更がオンライン・ログに入っていた場合には、このような状況が発生することがあります。

たとえば、次のメッセージが表示されたとします。

```
ORA-00279: 変更 55636 (06/08/2000 16:59:47 で生成) にはスレッド番号 1 が必要です
ORA-00289: 検討すべきログ・ファイル :/oracle/work/arc_dest/arcr_1_111.arc
ORA-00280: 変更 55636 (スレッド 1) は順序番号 111 に存在します。
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
```

オンライン REDO ログの名前を指定し、[Enter] を押します（正しいログが見つかるまで何度か繰り返すことになる場合があります）。

```
/oracle/dbs/t1_log1.f
Log applied.
Media recovery complete.
```

なんらかの理由によりオンライン・ログにアクセスできない場合には、オンライン・ログを適用せずにリカバリを取り消すことができます。すべてのデータ・ファイルが現行のものである場合に、リカバリに必要な REDO がオンライン・ログの中に含まれているときには、オンライン・ログを適用せずにデータベースをオープンすることはできません。オンライン・ログにアクセスできない場合は、制御ファイルを再作成する必要があります（3-15 ページの「[現行の制御ファイルおよびバックアップ制御ファイルのすべてが損害を受けた場合](#)」を参照）。

7. リカバリが終了した後、RESETLOGS オプションを指定してデータベースをオープンします。

```
ALTER DATABASE OPEN RESETLOGS;
```

8. 2-4 ページの「[データベース全体のユーザー管理バックアップの作成](#)」で説明しているように、事前の用心のためにデータベースを即時にバックアップします。

バックアップ制御ファイルのデフォルト以外の場所へのリストア

メディアの損傷が重大なため、元の場所に制御ファイルをリストアできない場合は、サーバー・パラメータ・ファイルで制御ファイルの新しい場所を指定する必要があります。

CONTROL_FILES 初期化パラメータで指定されたすべての場所に有効な制御ファイルが存在する必要があります。そうでない場合は、データベースをマウントできません。

デフォルト以外の場所に制御ファイルをリストアする手順は、次のとおりです。

3-13 ページの「[デフォルトの場所へのバックアップ制御ファイルのリストア](#)」の手順に従いますが、手順 2 の後に、次の手順を追加します。

制御ファイルの新しい場所を反映するように、CONTROL_FILES 初期化パラメータで指定されたすべての場所を編集します。たとえば、サーバー・パラメータ・ファイルに制御ファイルの場所が次のように列挙されているとします。

```
CONTROL_FILES = '/dsk1/oracle/dbs/cf1.f', '/dsk2/cf2.f'
```

この初期化パラメータを次のように変更できます。

```
CONTROL_FILES = '/dsk3/tmp/cf1.f', 'dsk3/tmp/cf2.f'
```

現行の制御ファイルおよびバックアップ制御ファイルのすべてが損害を受けた場合

永続的なメディア障害によりすべての制御ファイルが失われたか損傷した場合には、すべてのオンライン REDO ログ・ファイルが影響を受けていなければ、新しい制御ファイルを作成した後でデータベースをリカバリできます。この手順では、RESETLOGS オプションを指定してデータベースをオープンする必要はありません。

制御ファイルのバックアップが存在しているかどうか、そして制御ファイルの新しさによっては、CREATE CONTROLFILE 文のテキストを生成するときに表 3-2 のオプションを使用できます。データベースに対する変更は alert_SID.log に記録されるため、どのオプションを選択するかを決定するときには、このログを調べてください。

表 3-2 制御ファイル作成のオプション

状況	対応
データベースの最後の構造変更の後で ALTER DATABASE BACKUP CONTROLFILE TO TRACE NORESETLOGS を実行し、SQL コマンド・トレース出力を保存してある場合	トレース出力の CREATE CONTROLFILE 文をそのまま使用します。
データベースの構造変更を行う前に、最後の ALTER DATABASE BACKUP CONTROLFILE TO TRACE を実行した場合	変更を反映するように ALTER DATABASE BACKUP CONTROLFILE TO TRACE の出力を編集します。たとえば、最近データベースにデータ・ファイルを追加した場合には、このデータ・ファイルを CREATE CONTROLFILE 文の DATAFILE 句に追加します。
ALTER DATABASE BACKUP CONTROLFILE TO filename 文を使用して制御ファイルをバックアップした場合（TO TRACE オプションではなく）	制御ファイルのコピーを使用して SQL 出力を取得します。バックアップ制御ファイルをコピーし、ALTER DATABASE BACKUP CONTROLFILE TO TRACE NORESETLOGS の前に STARTUP MOUNT を実行します。制御ファイルのコピーが最新の構造変更の前の日付になっている場合には、構造変更を反映するようにトレース出力を編集します。
TO TRACE 形式でも TO filename 形式でも制御ファイルのバックアップがない場合	CREATE CONTROLFILE 文を手動で作成します（『Oracle9i SQL リファレンス』を参照）。

注意： デフォルトの US7ASCII 以外のキャラクタ・セットを使用している場合には、CREATE CONTROLFILE 文の引数としてキャラクタ・セットを指定する必要があります。データベース・キャラクタ・セットは、起動時にアラート・ログに書き込まれます。また、キャラクタ・セット情報は BACKUP CONTROLFILE TO TRACE の出力にも記録されます。

新しい制御ファイルを作成する手順は、次のとおりです。

1. データベースを NOMOUNT モードで起動します。たとえば、次のように入力します。

```
STARTUP NOMOUNT
```

2. NORESETLOGS オプションを指定して、CREATE CONTROLFILE 文により制御ファイルを作成します（オプションについては表 3-2 を参照）。次の例では、キャラクタ・セットがデフォルトの US7ASCII であるとしています。

```
CREATE CONTROLFILE REUSE DATABASE SALES NORESETLOGS ARCHIVELOG
    MAXLOGFILES 32
    MAXLOGMEMBERS 2
    MAXDATAFILES 32
    MAXINSTANCES 16
    MAXLOGHISTORY 1600
LOGFILE
    GROUP 1 (
        '/diska/prod/sales/db/log1t1.dbf',
        '/diskb/prod/sales/db/log1t2.dbf'
    ) SIZE 100K
    GROUP 2 (
        '/diska/prod/sales/db/log2t1.dbf',
        '/diskb/prod/sales/db/log2t2.dbf'
    ) SIZE 100K,
DATAFILE
    '/diska/prod/sales/db/database1.dbf',
    '/diskb/prod/sales/db/filea.dbf';
```

制御ファイルを作成した後、Oracle はデータベースをマウントします。

3. データベースを通常にリカバリします（USING BACKUP CONTROLFILE 句を指定しない）。

```
RECOVER DATABASE
```

4. メディア・リカバリが完了した後、データベースをオープンします。

```
ALTER DATABASE OPEN;
```

RESETLOGS は必要ありません。

5. 即時に制御ファイルをバックアップします。次の SQL 文は、?/dbs/cf.bak にデータベースの制御ファイルをバックアップします。

```
ALTER DATABASE BACKUP CONTROLFILE TO '?/dbs/cf.bak' REUSE;
```

関連項目： 2-21 ページ「[制御ファイルのトレース・ファイルへのバックアップ](#)」

アーカイブ REDO ログのリストア

メディア・リカバリのためには、リストアされたバックアップが作成された時点からリカバリの目標時点までの間に生成されたすべてのアーカイブ REDO ログ・ファイルが必要です。リカバリ時に Oracle が使用できるように、アーカイブ・ログは最終的にはディスクに置いてある必要があります。

必要なアーカイブ REDO ログをリストアする手順は、次のとおりです。

1. 必要なアーカイブ REDO ログ・ファイルを確認するために、V\$ARCHIVED_LOG および V\$RECOVERY_LOG を問い合わせます。データ・ファイルのリカバリが必要な場合に、データ・ファイルのバックアップが存在しないときには、データベースにデータ・ファイルが追加されてから以降に生成されたすべての REDO が必要になります。

ビュー	説明
V\$ARCHIVED_LOG	すべてのアーカイブ・ログのすべてのファイル名を列挙します。
V\$RECOVERY_LOG	Oracle がメディア・リカバリを実行する必要があるアーカイブ REDO ログのみを列挙します。LOG_ARCHIVE_FORMAT を使用して、考えられるファイルの名前も表示されます。 注意： このビューは、データ・ファイルのリカバリが必要な場合にのみ移入されます。このため、ユーザー・エラーなど、計画されたリカバリの場合にはこのビューは役に立ちません。

2. 使用可能な領域がある場合には、LOG_ARCHIVE_DEST_1 で指定された場所に、必要なアーカイブ REDO ログ・ファイルをリストアします。Oracle はメディア・リカバリ中に必要になると、自動的に正しいログを検索します。たとえば、次のように入力します。

```
% cp /disk2/arc_backup/*.arc /disk1/oracle/dbs/arc_dest
```

3. アーカイブ先の初期化パラメータで指定された場所に十分な使用可能領域がない場合には、必要なアーカイブ REDO ログ・ファイルの一部またはすべてを、別の場所にリストアします。SQL*Plus で SET 文の LOGSOURCE パラメータを使用するか、SQL の ALTER DATABASE 文の RECOVER ... FROM パラメータを使用して、メディア・リカバリ前またはメディア・リカバリ中に場所を指定します。たとえば、次のように入力します。

```
SET LOGSOURCE /disk2/temp # set location using SET statement
DATABASE RECOVER FROM '/disk2/temp'; # set location in RECOVER statement itself
```

4. アーカイブ・ログを適用し、オフライン記憶域に各アーカイブ・ログ・グループのコピーがまだ存在していることを確認した後、アーカイブ REDO ログ・ファイルのリストアされたコピーを削除してディスク領域を解放します。たとえば、ログ・ディレクトリを作業ディレクトリとした後、次のように入力します。

```
% rm *.arc
```

関連項目： データ・ディクショナリ・ビューの詳細は、『Oracle9i データベース・リファレンス』を参照し、メディア・リカバリ時のログの適用の概要は、4-2 ページの「[ユーザー管理のメディア・リカバリの実行概要](#)」を参照してください。

ユーザー管理のメディア・リカバリの実行

この章では、データベースのリカバリの方法について説明します。この章で説明する項目は、次のとおりです。

- [ユーザー管理のメディア・リカバリの実行概要](#)
- [ユーザー管理の完全メディア・リカバリの実行](#)
- [ユーザー管理の不完全メディア・リカバリの実行](#)
- [NOARCHIVELOG モードでのデータベースのリカバリ](#)
- [パラレルのメディア・リカバリの実行](#)
- [ユーザー管理のメディア・リカバリ後のデータベースのオープン](#)
- [ユーザー管理のメディア・リカバリの中断](#)
- [ユーザー管理のメディア・リカバリの制限](#)

ユーザー管理のメディア・リカバリの実行概要

完全または不完全のメディア・リカバリ時には、メディア・リカバリのロールフォワード・フェーズ中に Oracle によってデータ・ファイルに REDO ログ・ファイルが適用されます。UNDO セグメントに対する変更はオンライン REDO ログに記録されているため、ロールフォワードにより、対応する UNDO セグメントが再生成されます。ロールフォワードでは、データベースを必要な時点まで進めるために必要なだけの REDO ログ・ファイルが処理されます。

リカバリの実行に Recovery Manager (RMAN) を使用しない場合には、SQL*Plus の RECOVER コマンドを使用する必要があります。SQL 文の ALTER DATABASE RECOVER を使用することもできますが、SQL*Plus の RECOVER コマンドを使用することをお勧めします。

この項では、次の項目について説明します。

- ユーザー管理のリカバリを実行する前提条件
- RECOVER コマンドを使用したログの自動適用
- アーカイブ・ログがデフォルトの場所にある場合のリカバリ
- アーカイブ・ログがデフォルト以外の場所にある場合のリカバリ
- アーカイブ・ログの出力先の再設定
- アーカイブ・ログの出力先の変更
- REDO ログが適用できない場合の対応

ユーザー管理のリカバリを実行する前提条件

どのような種類のメディア・リカバリを開始する場合にでも、次の制約を守る必要があります。

- 管理者権限が必要です。
- すべてのリカバリ・セッションが互換である必要があります。
- 別のセッションが不完全メディア・リカバリを実行しているときに、もう 1 つのセッションで完全メディア・リカバリを開始することはできません。
- 共有サーバー・プロセスによりデータベースに接続しているときにメディア・リカバリを開始することはできません。

RECOVER コマンドを使用したログの自動適用

オラクル社は、メディア・リカバリの実行には ALTER DATABASE RECOVER 文ではなく、SQL*Plus の RECOVER コマンドを使用することをお勧めします。ほとんどの場合、SQL*Plus を使用した方が簡単です。

SQL*Plus を使用してメディア・リカバリを実行する場合、最も簡単な方法は、自動リカバリを実行することです。自動リカバリでは、個々のアーカイブ・ログの適用について SQL*Plus の手動プロンプトなしに、リカバリが開始されます。

SQL*Plus を使用する場合には、リカバリ時に必要となるアーカイブ REDO ログのデフォルト・ファイル名の適用を自動化するために、2 つのオプションを使用できます。

- RECOVER コマンドを発行する前に SET AUTORECOVERY ON を発行する。
- RECOVER コマンドのオプションとして AUTOMATIC キーワードを指定する。

いずれの場合にも、必要なファイルが正しい場所に正しい名前が存在しているかぎり RECOVER コマンドを発行したときに介入は必要ありません。

自動リカバリを使用した場合に使用されるファイル名は、LOG_ARCHIVE_FORMAT の値と LOG_ARCHIVE_DEST_ *n* の値を連結して導出されます。この *n* は、有効なすべてのローカル出力先の中で最大の値です。

たとえば、データベース・インスタンスで初期化パラメータの次のような設定が有効になっているとします。

```
LOG_ARCHIVE_DEST_1 = "LOCATION=/arc_dest/loc1/"
LOG_ARCHIVE_DEST_2 = "LOCATION=/arc_dest/loc2/"
LOG_ARCHIVE_DEST_STATE_1 = DEFER
LOG_ARCHIVE_DEST_STATE_2 = ENABLE
LOG_ARCHIVE_FORMAT = arch_%t_%s.arc
```

この場合、SQL*Plus は自動的に /arc_dest/loc2/arch_%t_%s.arc というファイル名を提示します（この %t はスレッドで、%s は順序です）。

デフォルト・オプションである SET AUTORECOVERY OFF を実行した場合には、ファイル名を手動で入力するか、[Enter] キーを押して、提示されたデフォルトのファイル名を受け入れる必要があります。

自動リカバリのための SET AUTORECOVERY の使用方法

SET AUTORECOVERY ON コマンドを実行して自動リカバリを有効にします。

SET AUTORECOVERY を使用してリカバリを自動化する手順は、次のとおりです。

1. オフライン・データ・ファイルのバックアップをリストアします。次の例は、オペレーティング・システム・ユーティリティを使用して、すべてのデータ・ファイルの非一貫性バックアップをリストアします。

```
% cp /fs2/BACKUP/tbs* /oracle/dbs
```

2. データベースがマウントされていることを確認します。たとえば、データベースが停止している場合には、次のコマンドを実行します。

```
STARTUP MOUNT
```

3. 自動リカバリを有効にします。たとえば、SQL*Plus で次のコマンドを実行します。

```
SET AUTORECOVERY ON
```

4. 目的のデータ・ファイルのリカバリを行います。この例は、データベース全体のリカバリを行います。

```
RECOVER DATABASE
```

Oracle は、出力サンプルに示すように、必要なアーカイブ・ログを自動的に提示して適用します。

```
ORA-00279: 変更 53577 (01/26/00 19:20:58 で生成) にはスレッド 1 が必要です。
```

```
ORA-00289: 検討すべきログ・ファイル:/oracle/work/arc_dest/arcr_1_802.arc
```

```
ORA-00280: 変更 53577 (スレッド 1) は順序番号 802 に存在します。
```

```
Log applied.
```

```
ORA-00279: 変更 53584 (01/26/00 19:24:05 で生成) にはスレッド 1 が必要です。
```

```
ORA-00289: 検討すべきログ・ファイル:/oracle/work/arc_dest/arcr_1_803.arc
```

```
ORA-00280: 変更 53584 (スレッド 1) は順序番号 803 に存在します。
```

```
ORA-00278: ログ・ファイル"/oracle/work/arc_dest/arcr_1_802.arc" はこのリカバリでは必要なくなりました。
```

```
Log applied.
```

```
Media recovery complete.
```

5. データベースをオープンします。たとえば、次のように入力します。

```
ALTER DATABASE OPEN;
```

注意： ALTER DATABASE RECOVER 文を発行した後で、リカバリで考慮されたすべてのファイルを、V\$RECOVERY_FILE_STATUS ビューで表示できます。V\$RECOVERY_STATUS ビューで、各ファイルのステータス情報にアクセスできます。これらのビューは、リカバリ・セッションの終了後はアクセスできません。

自動リカバリのための RECOVERY AUTOMATIC の使用方法

SET AUTORECOVERY を使用して自動リカバリをオンにする他にも、RECOVER コマンドで単純に AUTOMATIC キーワードを指定することもできます。

RECOVER AUTOMATIC コマンドを使用してリカバリを自動化する手順は、次のとおりです。

1. オフライン・データ・ファイルのバックアップをリストアします。この例は、すべてのデータ・ファイルのバックアップをリストアします。

```
% cp /oracle/work/BACKUP/tbs* /oracle/dbs
```

2. データベースがマウントされていることを確認します。たとえば、データベースが停止している場合には、次のコマンドを実行します。

```
STARTUP MOUNT
```

3. AUTOMATIC キーワードを指定して、目的のデータ・ファイルのリカバリを行います。この例は、データベース全体の自動リカバリを実行します。

```
RECOVER AUTOMATIC DATABASE
```

4. Oracle は、次の出力に示すように、必要なアーカイブ・ログを自動的に提示して適用します。

```
ORA-00279: 変更 53577 (01/26/00 19:20:58 で生成) にはスレッド 1 が必要です。
ORA-00289: 検討すべきログ・ファイル :/oracle/work/arc_dest/arcr_1_802.arc
ORA-00280: 変更 53577 (スレッド 1) は順序番号 802 に存在します。
Log applied.
ORA-00279: 変更 53584 (01/26/00 19:24:05 で生成) にはスレッド 1 が必要です。
ORA-00289: 検討すべきログ・ファイル :/oracle/work/arc_dest/arcr_1_803.arc
ORA-00280: 変更 53584 (スレッド 1) は順序番号 803 に存在します。
ORA-00278: ログ・ファイル "/oracle/work/arc_dest/arcr_1_802.arc" はこのリカバリでは必要
なくなりました。
Log applied.
Media recovery complete.
```

5. データベースをオープンします。たとえば、次のように入力します。

```
ALTER DATABASE OPEN;
```

Oracle Real Application Clusters 構成を使用した場合、ユーザーが不完全リカバリを実行するか、バックアップ制御ファイルを使用したときには、Oracle は最初の REDO スレッドから最初のアーカイブ REDO ログ・ファイルの名前を算出することのみが可能です。その他の REDO スレッドからは、最初のログ・ファイルを手動で適用する必要があります。特定のスレッド内の最初のログ・ファイルが提供されれば、Oracle はこのスレッド内の後続のログの名前を提示できます。

関連項目： ログ・ファイルの適用の例は、使用しているオペレーティング・システム用のオラクル社のマニュアルを参照してください。

アーカイブ・ログがデフォルトの場所にある場合のリカバリ

アーカイブ・ログがデフォルトの場所にある場合のリカバリは、最も簡単なケースです。ログが必要になると、Oracle からファイル名が提示されます。SQL*Plus を使用して、自動ではないメディア・リカバリを実行する場合には、出力は次の形式で表示されます。

```
ORA-00279: 変更 ##### (DD/MM/YY HH:MM:SS で生成) にはスレッド # が必要です。
ORA-00289: 検討すべきログ・ファイル :logfile
ORA-00280: 変更 ##### (スレッド #) は順序番号 # に存在します。
Specify log: [<RET> for suggested | AUTO | FROM logsource | CANCEL ]
```

たとえば、SQL*Plus は次のような出力を表示します。

```
ORA-00279: 変更 53577 (01/26/00 19:20:58 で生成) にはスレッド 1 が必要です。
ORA-00289: 検討すべきログ・ファイル :/oracle/arc_dest/arcr_1_802.arc
ORA-00280: 変更 53577 (スレッド 1) は順序番号 802 に存在します。
Specify log: [<RET> for suggested | AUTO | FROM logsource | CANCEL ]
```

ALTER DATABASE ... RECOVER 文を使用した場合にも同様のメッセージが表示されます。しかし、プロンプトは表示されません。

Oracle は、初期化パラメータ LOG_ARCHIVE_DEST_ *n* (この *n* は有効なすべてのローカル出力先の中の最大の値) および LOG_ARCHIVE_FORMAT の現在の値を連結し、制御ファイルのログ履歴情報を使用して、アーカイブ REDO ログのファイル名を提示します。たとえば、次のようなアーカイブ REDO ログの設定が可能です。


```
LOG_ARCHIVE_DEST_1 = 'LOCATION = /oracle/arc_dest/'
LOG_ARCHIVE_FORMAT = arcr_%t_%s.arc
```

```
SELECT NAME FROM V$ARCHIVED_LOG;
```

```
NAME
```

```
-----
/oracle/arc_dest/arcr_1_467.arc
/oracle/arc_dest/arcr_1_468.arc
/oracle/arc_dest/arcr_1_469.arc
```

このため、必要なアーカイブ・ログ・ファイルがすべて LOG_ARCHIVE_DEST_1 という出力先にマウントされ、LOG_ARCHIVE_FORMAT の値が変更されていない場合には、Oracle はログ・ファイルを提示して適用し、メディア・リカバリを自動的に実行できます。

アーカイブ・ログがデフォルト以外の場所にある場合のリカバリ

アーカイブ・ログがデフォルトの場所でない場合にメディア・リカバリを実行するには、リカバリ手順に追加の処理が必要です。次のどちらか一方を選択します。

- アーカイブ REDO ログの場所を指定する LOG_ARCHIVE_DEST_n パラメータを編集し、通常どおりにリカバリします。
- リカバリ前に SQL*Plus の SET 文を使用してデフォルト以外のログの場所を指定するか、RECOVER コマンドの LOGFILE パラメータを使用します。

アーカイブ・ログの出力先の再設定

初期化パラメータ・ファイルを編集するか、ALTER SYSTEM 文を発行して、アーカイブ REDO ログのデフォルトの場所を変更できます。

リカバリ前にアーカイブ・ログのデフォルトの場所を変更する手順は、次のとおりです。

1. オペレーティング・システム・ユーティリティを使用して、アーカイブ・ログをデフォルト以外の場所にリストアします。たとえば、次のように入力します。

```
% cp /disk3/arc_bak/* /disk2/tmp
```

2. アーカイブ・ログ・パラメータの値を、目的のデフォルト以外の場所に変更します。インスタンスが起動されているときに ALTER SYSTEM 文を発行するか、初期化パラメータ・ファイルを編集した後にデータベース・インスタンスを起動します。たとえば、インスタンスの停止中にパラメータ・ファイルを次のように編集します。

```
LOG_ARCHIVE_DEST_1 = 'LOCATION=/disk2/tmp/arc'
LOG_ARCHIVE_FORMAT = r_%t_%s.arc
```

3. SQL*Plus を使用して、編集済みの初期化パラメータ・ファイルを指定して新しいインスタンスを起動した後、データベースをマウントします。たとえば、次のように入力します。

```
STARTUP MOUNT
```

4. 通常のようにメディア・リカバリを開始します。たとえば、次のように入力します。

```
RECOVER DATABASE
```

アーカイブ・ログの出力先の変更

場合によっては、REDO ログ・ファイルの入力元としてのアーカイブの出力先パラメータの現在の設定を変更することが必要な場合があります。たとえば、データベースがオープンしているときに、オフライン表領域のリカバリが必要になった場合、アーカイブの出力先のパラメータで指定された場所に、必要な REDO ログ・ファイルをマウントするために必要な十分な使用可能領域がないと想定してください。この場合には、次の手順の 1 つを使用します。

SET LOGSOURCE を使用して、デフォルト以外の場所のログを使用してリカバリする手順は、次のとおりです。

1. オペレーティング・システム・ユーティリティを使用して、アーカイブ REDO ログを代替の場所に移動します。たとえば、次のように入力します。

```
% cp /disk1/oracle/arc_dest/* /disk2/temp
```

2. SQL*Plus 内でリカバリ操作のための代替場所を指定します。SET 文の LOGSOURCE パラメータを使用するか、ALTER DATABASE 文の RECOVER ... FROM 句を使用します。たとえば、SQL*Plus を起動して次のコマンドを実行します。

```
SET LOGSOURCE "/disk2/temp"
```

3. オフライン表領域をリカバリします。

```
RECOVER AUTOMATIC TABLESPACE offline_tbsp
```

4. また、SET LOGSOURCE を実行せずに、単純に次のコマンドを実行することもできます。

```
RECOVER AUTOMATIC TABLESPACE offline_tbsp FROM "/disk2/temp"
```

注意： REDO ログのソースを変更しても、アーカイブされているオンライン REDO ログ・グループのアーカイブ REDO ログの出力先は影響を受けません。

REDO ログが適用できない場合の対応

SQL*Plus のリカバリ・オプション（SQL 文ではない）を使用している場合には、Oracle が REDO ログ・ファイルを正常に適用するたびに、次のメッセージが戻されます。

Log applied.

Oracle は次に、次の順序のログをプロンプトするか、最後に適用されたログが必要なログとして最後のものであった場合は、リカバリを終了します。

提示されたファイルが正しくないか、ユーザーが正しくないファイル名を指定した場合には、Oracle はエラー・メッセージを戻します。たとえば、次のようなメッセージが表示されます。

ORA-00308: アーカイブ・ログ "/oracle/work/arc_dest/arcr_1_811.arc" をオープンできません。

ORA-27037: ファイル・ステータスを取得できません。

SVR4 Error: 2: No such file or directory

追加情報: 3

必要な REDO ログ・ファイルが適用されるまでは、リカバリを続行できません。REDO ログ・ファイルの名前を指定した後で Oracle からエラー・メッセージが戻された場合には、次のように対応できます。

エラー	考えられる原因	解決方法
ORA-27037: ファイル・ステータスを取得できません。	正しくないファイル名が入力されました。 ログが欠落しています。	正しいファイル名を再入力します。 バックアップのアーカイブ REDO ログをリストアします。
ORA-27047: ファイルのヘッダー・ブロックを読み込めません。	ログが部分的に書き込まれているか、破損しています。	破損していない、または完全なログ・コピーを見つけることができたなら、そのコピーを適用してリカバリを続行します。 ログのコピーが存在しない場合、最後の有効な REDO エントリの時刻がわかっているときには、不完全リカバリを使用する必要があります。バックアップのリストアを含めて、リカバリを再起動します。

ユーザー管理の完全メディア・リカバリの実行

完全リカバリを実行すると、バックアップが現在の SCN までリカバリされます。データベース全体で一度にリカバリを実行することも、個々の表領域またデータ・ファイルでリカバリを実行することもできます。完全リカバリの場合は不完全リカバリ後のように、RESETLOGS オプションを使用してデータベースをオープンする必要はないため、最初に一部のデータ・ファイルのリカバリを行ってから、後で残りのデータ・ファイルのリカバリを行うこともできます。

この項では、完全メディア・リカバリ操作に必要な手順について説明します。この項で説明する項目は、次のとおりです。

- [クローズしているデータベースのリカバリの実行](#)
- [オープン状態のデータベースでのデータ・ファイルのリカバリの実行](#)

関連項目： リカバリの概念およびリカバリ計画の基本的情報を理解するには、『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。

クローズしているデータベースのリカバリの実行

この項では、データベースをオープンしていないときに完全リカバリを実行する手順について説明します。破損したすべてのデータ・ファイルに対して一度にリカバリ操作を実行することも、破損した個々のデータ・ファイルに対して個別にリカバリ操作を実行することもできます。

メディア・リカバリは次の段階に従って実行します。

1. 4-11 ページの「[クローズしているデータベースのリカバリの準備](#)」の説明に従って、クローズ状態のデータベースのリカバリの準備をします。
2. 4-11 ページの「[破損または欠落したファイルのバックアップのリストア](#)」の説明に従って、必要なファイルをリストアします。
3. 4-12 ページの「[データベースのリカバリ](#)」の説明に従って、リストアされたデータ・ファイルのリカバリを行います。

クローズしているデータベースのリカバリの準備

この段階では、インスタンスを停止し、問題の原因となっているメディア・デバイスを検査します。

クローズしているデータベースのリカバリを準備する手順は、次のとおりです。

1. データベースがオープンしている場合は、ABORT オプションを使用してデータベースを停止します。

```
SHUTDOWN ABORT
```

2. メディア・エラーからのリカバリの場合には、可能であればエラーを訂正します。メディア障害の原因であるハードウェアの問題が一時的なものである場合は、データが破損していなければ（たとえば、ディスクまたはコントローラの停電）、メディア・リカバリは必要ありません。データベースを単純に起動して、通常の操作を再開します。問題を解決できない場合は、次の手順に進みます。

破損または欠落したファイルのバックアップのリストア

この段階では、必要なすべてのバックアップをリストアします。

必要なファイルをリストアする手順は、次のとおりです。

1. 3-6 ページの「**リカバリが必要なデータ・ファイルの判別**」に記載された方法を使用して、リカバリが必要なデータ・ファイルを判別します。
2. ファイルが永続的な損傷を受けた場合には、破損したファイルの最新のバックアップを識別します。メディア障害によって破損したデータ・ファイルのみをリストアします。破損していないデータ・ファイルまたはオンライン REDO ログ・ファイルはリストアしないでください。

たとえば、破損したファイルが `/oracle/dbs/tbs_10.f` のみである場合には、記録を調べて、このファイルの最新のバックアップが `/oracle/backup/tbs_10.backup` であることを確認します。特定のデータ・ファイルのバックアップがない場合には、リカバリするための空の置換ファイルを作成できます。

3. オペレーティング・システム・ユーティリティを使用して、デフォルトの場所または新しい場所にファイルをリストアします。第3章「**ユーザー管理のリストア操作の実行**」の説明に従って、必要なファイルをリストアします。たとえば、UNIX ユーザーが `/oracle/dbs/tbs_10.f` をデフォルトの場所にリストアする場合は、次のコマンドを入力できます。

```
% cp /oracle/backup/tbs_10.backup /oracle/dbs/tbs_10.f
```

次のガイドラインに従って、データ・ファイルのバックアップをリストアする場所を決定します。

状況	対応
ハードウェアの問題が解決され、デフォルトの場所にデータ・ファイルをリストアできる場合。	デフォルトの場所にデータ・ファイルをリストアし、メディア・リカバリを開始します。
ハードウェアの問題が持続し、元の場所にデータ・ファイルをリストアできない場合。	代替の記憶デバイスにデータ・ファイルをリストアします。これらのファイルの新しい場所を制御ファイルの中で指定します。必要に応じて、『Oracle9i データベース管理者ガイド』の「データ・ファイルの改名および再配置」に記載された操作を使用します。

データベースのリカバリ

最後の段階では、リストアしたデータ・ファイルのリカバリを行います。

リストアされたデータ・ファイルのリカバリの手順は、次のとおりです。

- 1. 管理者権限を使用してデータベースに接続し、新しいインスタンスを起動し、データベースをマウントしますが、オープンはしません。たとえば、次のように入力します。

```
STARTUP MOUNT
```

- 2. 現行の制御ファイルに通常付随するデータ・ファイルのリストを調べるか、V\$DATAFILE ビューを問い合せて、すべてのデータ・ファイルのデータ・ファイル名およびステータスを取得します。たとえば、次のように入力します。

```
SELECT NAME,STATUS FROM V$DATAFILE;
```

- 3. データベースのすべてのデータ・ファイルがオンライン化されていることを確認します。オフライン表領域が NORMAL モードでオフラインされたか、読取り専用表領域の一部でないかぎり、リカバリを必要とするデータベースのすべてのデータ・ファイルはオンライン化されている必要があります。たとえば、/oracle/dbs/tbs_10.f という名前のデータ・ファイルを確実にオンライン化するには、次のように入力します。

```
ALTER DATABASE DATAFILE '/oracle/dbs/tbs_10.f' ONLINE;
```

指定されたデータ・ファイルがすでにオンライン化されている場合は、この文は無視されます。次のような、すべてのデータ・ファイルを一度にオンライン化するスクリプトを作成することもできます。

```
SPOOL onlineall.sql
SELECT 'ALTER DATABASE DATAFILE '''||name||''' ONLINE;' FROM V$DATAFILE;
SPOOL OFF

SQL> @onlineall
```

4. データベース、表領域またはデータ・ファイルのリカバリのための文を発行します。たとえば、次の RECOVER コマンドのいずれかを入力します。

```
RECOVER DATABASE # recovers whole database
RECOVER TABLESPACE users # recovers specific tablespace
RECOVER DATAFILE '/oracle/dbs/tbs_10'; # recovers specific datafile
```

次のガイドラインに従って、実行する文を決定します。

目的	対応
一度の手順で破損したすべてのファイルのリカバリを行う	RECOVER DATABASE を実行します。
個々の表領域のリカバリを行う	RECOVER TABLESPACE を実行します。
破損した個々のデータ・ファイルのリカバリを行う	RECOVER DATAFILE を実行します。
データベース全体または個々のデータ・ファイルのリカバリをパラレル化する	4-27 ページの「 パラレルのメディア・リカバリの実行 」を参照してください。

5. アーカイブ・ログの適用を自動化しない場合には、必要な REDO ログが提示されるたびに、ログを受け入れるか拒否する必要があります。ユーザーがリカバリを自動化した場合は、Oracle は必要なログを自動的に適用します。Oracle はこの作業を、必要なすべてのアーカイブおよびオンライン REDO ログ・ファイルが、リストアされたデータ・ファイルに適用されるまで続行します。
6. メディア・リカバリが完了すると、Oracle はユーザーに通知します。

```
Media recovery complete.
```

完全メディア・リカバリのためにアーカイブ REDO ログ・ファイルが必要でない場合には、Oracle は必要なすべてのオンライン REDO ログ・ファイルを適用し、リカバリを終了します。

7. リカバリが終了したら、使用するためにデータベースをオープンします。

```
ALTER DATABASE OPEN;
```

関連項目： REDO ログ・ファイルの適用の詳細は、4-2 ページの「[ユーザー管理のメディア・リカバリの実行概要](#)」を参照してください。

オープン状態のデータベースでのデータ・ファイルのリカバリの実行

データベースをオープンしているときにメディア障害が発生した場合は、破損していないデータ・ファイルはオンラインのまま使用できる場合があります。データベース・ライターがファイルに書き込めない場合には、Oracle は破損したファイルを自動的にオフライン化しますが、破損したファイルを含む表領域はオフライン化しません。破損したファイルを読み取れない問合せはエラーを戻しますが、Oracle は問合せが失敗したという理由からファイルをオフライン化することはありません。たとえば、問合せを実行して、次のような出力が得られることがあります。

```
行 1 でエラーが発生しました。  
ORA-01116: データベース・ファイル 11 のオープンでエラーが発生しました。  
ORA-01110: データ・ファイル 11: '/oracle/dbs/tbs_32.f'  
ORA-27041: ファイルをオープンできません。  
SVR4 Error: 2: No such file or directory  
追加情報 : 3
```

この項で説明しているメディア・リカバリの手順を使用して、SYSTEM 表領域のデータ・ファイルの完全メディア・リカバリを実行することはできません。メディア障害により SYSTEM 表領域のデータ・ファイルが破損した場合は、データベースは自動的に停止されます。

次の段階に従ってメディア・リカバリを実行します。

1. 4-15 ページの「[オープン状態のデータベースのリカバリの準備](#)」の説明に従って、データベースがオープンしていることを確認し、リカバリが必要な表領域をオフライン化して、データベースのリカバリの準備をします。
2. 4-15 ページの「[破損または欠落したファイルのバックアップのリストア](#)」の説明に従って、影響を受けた表領域の中の必要なファイルをリストアします。
3. 4-16 ページの「[オープン状態のデータベースのオフライン表領域のリカバリ](#)」の説明に従って、影響を受けた表領域のリカバリを行います。

関連項目：

- リカバリが必要かどうかを判断するための詳細は、3-6 ページの「[リカバリが必要なデータ・ファイルの判別](#)」を参照してください。
- SYSTEM 表領域のデータ・ファイルの完全メディア・リカバリの手順については、4-10 ページの「[クローズしているデータベースのリカバリの実行](#)」を参照してください。

オープン状態のデータベースのリカバリの準備

この段階では、影響を受けた表領域をオフラインにし、問題の原因となっているメディア・デバイスを検査します。

データベースをオープンしているときにデータ・ファイルのリカバリを準備する手順は、次のとおりです。

1. リカバリが必要であると判明したときにデータベースがオープンしていた場合は、破損したデータ・ファイルを含むすべての表領域をオフライン化します。たとえば、表領域 `users` に破損したファイルが含まれている場合には、次のように入力します。

```
ALTER TABLESPACE users OFFLINE TEMPORARY;
```

2. メディア障害の原因であるハードウェアの問題を解決します。ハードウェアの問題を迅速に解決できない場合は、破損したファイルを代替の記憶デバイスにリストアしてデータベースのリカバリを続けます。

破損または欠落したファイルのバックアップのリストア

この段階では、オフラインの表領域の必要なすべてのバックアップをリストアします。

オープン状態のデータベースでデータ・ファイルをリストアする手順は、次のとおりです。

1. ファイルが永続的損傷を受けた場合には、メディア障害によって損傷したデータ・ファイルの最新のバックアップ・ファイルのみをリストアします。破損していないデータ・ファイル、オンライン REDO ログ・ファイル、または制御ファイルはリストアしないでください。ハードウェアの問題が解決され、データ・ファイルを元の場所にリストアできる場合には、そのようにしてください。ハードウェアの問題が持続する場合には、データ・ファイルを代替の記憶デバイスにリストアします。

注意： 状況によっては、特定のデータ・ファイルのバックアップがない場合に、ALTER DATABASE CREATE DATAFILE を使用して、リカバリ可能な空の置換ファイルを作成できることもあります。

2. 1つ以上の破損したデータ・ファイルを代替の場所にリストアした場合には、データベースの制御ファイル内で、データ・ファイルの名前を変更してください。たとえば、表領域 `users` のデータ・ファイルのファイル名を変更するには、次のように入力します。

```
ALTER DATABASE RENAME FILE '/d1/oracle/dbs/tbs1.f' TO '/d3/oracle/dbs/tbs1.f';
```

関連項目： ALTER DATABASE RENAME FILE の詳細は、『Oracle9i SQL リファレンス』を参照してください。

オープン状態のデータベースのオフライン表領域のリカバリ

最後の段階では、オフライン表領域の中のデータ・ファイルのリカバリを行います。

オープン状態のデータベースのオフライン表領域のリカバリの手順は、次のとおりです。

1. 管理者権限を使用してデータベースに接続します。たとえば、SYSTEM としてデータベース prod1 に接続します。

```
% sqlplus SYSTEM/manager@prod1
```
2. 一度の手順で、1 つ以上のオフライン表領域の中の破損したすべてのデータ・ファイルについて、オフライン表領域のリカバリを開始します。たとえば、次のように、users 表領域および sales 表領域のリカバリを行います。

```
RECOVER TABLESPACE users, sales # begins recovery on datafiles in users and sales
```

注意： パフォーマンスを最大化するには、パラレル・リカバリを使用してデータ・ファイルのリカバリを行います。4-27 ページの「[パラレルのメディア・リカバリの実行](#)」を参照してください。

3. Oracle は、必要な REDO ログ・ファイル（アーカイブおよびオンライン）を適用してリストアされたデータ・ファイルを再構築し、メディア・リカバリのロールフォワード・フェーズを開始します。RECOVER AUTOMATIC または SET AUTORECOVERY ON によりファイルの適用が自動化されていないかぎり、Oracle は必要な REDO ログ・ファイルごとにプロンプトを表示します。

Oracle はこの作業を、必要なすべてのアーカイブ REDO ログ・ファイルが、リストアされたデータ・ファイルに適用されるまで続行します。次に、オンライン REDO ログ・ファイルがリストアされたデータ・ファイルに自動的に適用され、メディア・リカバリが完了します。

完全メディア・リカバリにアーカイブ REDO ログ・ファイルが必要ない場合には、Oracle はプロンプトを表示しません。かわりに、必要なすべてのオンライン REDO ログ・ファイルが適用され、メディア・リカバリが完了します。

4. 破損した表領域をメディア障害が発生した時点までリカバリできたら、オフラインの表領域をオンライン化します。たとえば、表領域 users および sales をオンライン化するには、次の文を発行します。

```
ALTER TABLESPACE users ONLINE;  
ALTER TABLESPACE sales ONLINE;
```

関連項目： データ・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。

ユーザー管理の不完全メディア・リカバリの実行

この項では、各種の不完全メディア・リカバリ操作に必要な手順について説明します。この項で説明する項目は、次のとおりです。

- [不完全リカバリの準備](#)
- [不完全リカバリ実行前のデータ・ファイルのリストア](#)
- [取消ベースの不完全リカバリの実行](#)
- [時間ベースの不完全リカバリの実行](#)
- [変更ベースの不完全リカバリの実行](#)

データベースが季節的な時間変更（たとえば夏時間）の影響を受ける場合には、REDO ログの中に1つの時間が2度出現し、2番目の、つまり後の時間までリカバリを行う場合に、問題が発生します。時間の変更を処理するには、取消ベースまたは変更ベースのリカバリを実行します。

不完全リカバリの準備

このフェーズでは、メディアの問題の原因を確認します。

取消ベースのリカバリを準備する手順は、次のとおりです。

1. 不完全メディア・リカバリの実行に自信がない場合は、リカバリ手順中にエラーが発生した場合に備える事前の防護策として、データベース全体のバックアップ（データベースのすべてのデータ・ファイル、制御ファイルおよびパラメータ・ファイル）を作成します。
2. データベースをまだオープンしているときに不完全メディア・リカバリが必要になった場合は、インスタンスを終了します。

SHUTDOWN ABORT

3. メディア障害が発生した場合は、障害の原因であるハードウェアの問題を解決します。ハードウェアの問題を迅速に解決できない場合は、破損したファイルを代替の記憶デバイスにリストアしてデータベースのリカバリを続けます。

不完全リカバリ実行前のデータ・ファイルのリストア

このフェーズでは、データベース全体のバックアップをリストアします。

取消ベースのリカバリに必要なファイルをリストアし、オンライン化する手順は、次のとおりです。

1. 現行の制御ファイルが、リカバリ目標時点のデータベースの物理構造と一致していない場合には、3-9 ページの「[制御ファイルのリストアおよび再作成](#)」の説明に従って、バックアップ制御ファイルをリストアします。リストアされた制御ファイルは、不完全メディア・リカバリの終了時点のデータベースの物理ファイル構造を反映している必要があります。次の方法で、使用する制御ファイルのバックアップを決定します。
- 現行の制御ファイル、および制御ファイルの各バックアップに対応するファイルのリストを検討し、使用する正しい制御ファイルを決定します。

■ 必要であれば、データベースのすべての現行の制御ファイルを、正しい制御ファイルのバックアップで置き換えます。

■ あるいは、欠落した制御ファイルを置き換えるために新しい制御ファイルを作成します。

注意： CONTROL_FILES の場所のいずれかに制御ファイルのバックアップをリストアできない場合には、初期化パラメータ・ファイルを編集して、この CONTROL_FILES 場所を削除します。

2. データベースのすべてのデータ・ファイルのバックアップをリストアします。既存のデータ・ファイルを置き換えるために使用するすべてのバックアップは、リカバリの目標時刻の前に作成されたものである必要があります。たとえば、1月2日の 2:00 p.m. まですりかバリする場合は、この時刻の前までに完了したバックアップですべてのデータ・ファイルをリストアします。次のガイドラインに従ってください。

状況	対応
データ・ファイルのバックアップがない場合	3-8 ページの「 バックアップが利用できない場合のデータ・ファイルの再作成 」の説明に従って、リカバリ可能な空の置換ファイルを作成します。
リカバリの目標時刻の後にデータ・ファイルが追加されている場合	このファイルは、リカバリ完了後はデータベースで使用されないため、このファイルのバックアップはリストアしないでください。
障害の原因となっていたハードウェアの問題が解決され、すべてのデータ・ファイルをデフォルトの場所にリストアできる場合	3-7 ページの「 データ・ファイルのリストア 」の説明に従ってファイルをリストアし、この手順のステップ 5 をスキップします。
ハードウェアの問題が解決しない場合	破損したデータ・ファイルを代替の記憶デバイスにリストアします。

注意： 制御ファイルのバックアップを使用する場合には、読取り専用表領域の中のファイルはオフライン化する必要があります。そうでない場合、リカバリ操作は読取り専用ファイルのヘッダーの更新を試みます。

3. SQL*Plus を起動し、管理者権限を使用して Oracle に接続します。たとえば、次のように入力します。

```
% sqlplus SYS/change_on_install@prod1
```

4. 新しいインスタンスを起動し、データベースをマウントします。

```
STARTUP MOUNT
```

5. ステップ 2 で 1 つ以上の破損したデータ・ファイルが代替の場所にリストアされた場合には、関連するデータベースの制御ファイルで、これらのファイルの新しい場所を指定します。たとえば、次のように入力します。

```
ALTER DATABASE RENAME FILE '/oracle/dbs/df2.f' TO '/oracle/newloc/df2.f';
```

6. 現行の制御ファイルに通常付随するデータ・ファイルのリストを調べるか、V\$DATAFILE ビューを問い合せて、すべてのデータ・ファイルのデータ・ファイル名およびステータスを取得します。たとえば、次のように入力します。

```
SELECT NAME,STATUS FROM V$DATAFILE;
```

7. データベースのすべてのデータ・ファイルがオンライン化されていることを確認します。オフライン表領域が **NORMAL** モードでオフラインされたか、読取り専用表領域の一部でないかぎり、リカバリを必要とするデータベースのすべてのデータ・ファイルはオンライン化されている必要があります。たとえば、/oracle/dbs/tbs_10.f という名前のデータ・ファイルを確実にオンライン化するには、次のように入力します。

```
ALTER DATABASE DATAFILE '/oracle/dbs/tbs_10.f' ONLINE;
```

指定されたデータ・ファイルがすでにオンライン化されているときには、この文は無視されます。次のような、すべてのデータ・ファイルを一度にオンライン化するスクリプトを作成することもできます。

```
SPOOL onlineall.sql
SELECT 'ALTER DATABASE DATAFILE '''||name||''' ONLINE;' FROM V$DATAFILE;
SPOOL OFF
SQL> @onlineall
```

取消ベースの不完全リカバリの実行

取消ベースのリカバリでは、ユーザーに対してアーカイブ REDO ログ・ファイルのファイル名を提示するプロンプトを表示して、リカバリが進行していきます。ユーザーがファイル名のかわりに CANCEL を指定するか、すべての REDO がデータ・ファイルに適用されると、リカバリは停止します。

リカバリを停止するアーカイブ・ログをユーザーが制御する場合には、変更ベースまたは時間ベースのリカバリよりも取消ベースのリカバリが適しています。たとえば、順序 1234 以降のすべてのログが損害を受けたことがわかっている場合は、ログ 1233 を適用した後で、リカバリを取り消すことができます。

取消ベースのメディア・リカバリは次の段階に従って実行する必要があります。

1. 4-17 ページの「**不完全リカバリの準備**」の説明に従って、データベースをバックアップし、メディア障害を修復して、リカバリの準備をします。
2. 4-18 ページの「**不完全リカバリ実行前のデータ・ファイルのリストア**」の説明に従って、バックアップ・データ・ファイルをリストアします。現行の制御ファイルがある場合には、バックアップ制御ファイルをリストアしないでください。
3. 次の手順に従って、リストアされたデータベースのバックアップに対してメディア・リカバリを実行します。

取消ベースのリカバリを実行する手順は、次のとおりです。

1. SQL*Plus を起動し、管理者権限を使用して Oracle に接続します。たとえば、次のように入力します。

```
% sqlplus '/ AS SYSDBA'
```

2. 新しいインスタンスを起動し、データベースをマウントします。

```
STARTUP MOUNT
```

3. 次のコマンドを発行し、取消ベースのリカバリを開始します。

```
RECOVER DATABASE UNTIL CANCEL
```

この不完全リカバリでバックアップ制御ファイルを使用している場合には、RECOVER コマンドで USING BACKUP CONTROLFILE オプションを指定します。

```
RECOVER DATABASE UNTIL CANCEL USING BACKUP CONTROLFILE
```

注意： RECOVER コマンドの UNTIL 句を指定しなかった場合には、完全リカバリを実行するまでは、データベースをオープンできません。

4. Oracle は、リストアされたデータ・ファイルを再構築するために必要な REDO ログ・ファイルを適用します。Oracle は名前を LOG_ARCHIVE_DEST_1 から検索できるものと想定し、名前を提示し、そのログ・ファイルの適用を停止するか進めるかについて、ユーザーの指示を要求します。制御ファイルがバックアップの場合には、オンライン・ログの変更を適用する場合、ユーザーがオンライン・ログの名前を提供する必要があります。

注意： Oracle Real Application Clusters 構成を使用した場合で、ユーザーが不完全リカバリを実行するか、バックアップ制御ファイルを使用したときには、Oracle は最初のスレッドから最初のアーカイブ REDO ログ・ファイルの名前を算出することのみが可能です。他のスレッドからの最初の REDO ログ・ファイルは、ユーザーが提供する必要があります。特定のスレッド内の最初のログ・ファイルが提供されれば、Oracle はこのスレッド内の後続のログ・ファイルの名前を提示できます。

5. リストアされたデータ・ファイルに最後のログが適用されるまで、REDO ログ・ファイルの適用を続けた後、次のコマンドを実行して、リカバリを取り消します。

CANCEL

Oracle はリカバリが成功したかどうかを示すメッセージを戻します。すべてのデータ・ファイルが整合性のある SCN までリカバリされる前にリカバリを取り消し、その後データベースをオープンしようとする、ファイルでさらにリカバリが必要な場合は、ORA-1113 エラーが表示されます。V\$RECOVER_FILE を問い合わせると、さらにリカバリが必要かどうか、または不完全リカバリの開始前にデータ・ファイルのバックアップがリストアされていなかったかどうかを確認できます。

6. RESETLOGS モードでデータベースをオープンします。不完全リカバリまたはバックアップ制御ファイルを使用したリカバリの後では、必ずオンライン・ログをリセットする必要があります。たとえば、次のように入力します。

ALTER DATABASE OPEN RESETLOGS;

関連項目： 4-28 ページ「ユーザー管理のメディア・リカバリ後のデータベースのオープン」

時間ベースの不完全リカバリの実行

この項では、時間ベースのメディア・リカバリの手順を、次の段階に従って説明します。

1. 4-17 ページの「[不完全リカバリの準備](#)」の説明に従って、データベースをバックアップし、メディア障害を修復して、リカバリの準備をします。
2. 4-18 ページの「[不完全リカバリ実行前のデータ・ファイルのリストア](#)」の説明に従って、バックアップ・データ・ファイルをリストアします。現行の制御ファイルがある場合には、バックアップ制御ファイルをリストアしないでください。
3. 次の手順を使用して、リストアされたバックアップに対してメディア・リカバリを実行します。

時間ベースのリカバリを実行する手順は、次のとおりです。

1. RECOVER DATABASE UNTIL TIME 文を発行して、時間ベースのリカバリを開始します。時間は、必ず一重引用符で区切った書式 ('YYYY-MM-DD:HH24:MI:SS') で指定されます。次の文は、指定された時刻までデータベースのリカバリを行います。

```
RECOVER DATABASE UNTIL TIME '2000-12-31:12:47:30'
```

この不完全リカバリで制御ファイルのバックアップを使用する（つまり、制御ファイルのバックアップまたは再作成された制御ファイルがリストアされた）場合には、リカバリを開始する文中でこのことを指定します。次の文は、制御ファイルのバックアップを使用して、指定された時刻までデータベースのリカバリを行います。

```
RECOVER DATABASE UNTIL TIME '2000-12-31:12:47:30' USING BACKUP CONTROLFILE
```

2. リストアされたデータ・ファイルのリカバリに必要な REDO ログ・ファイルを適用します。ファイルの適用が自動化されていないかぎり、Oracle は名前を LOG_ARCHIVE_DEST_1 から検索できるものと想定し、名前を提示し、そのログ・ファイルの適用を停止するか進めるかについて、ユーザーの指示を要求します。制御ファイルがバックアップの場合には、アーカイブ・ログを適用した後で、オンライン・ログの変更を適用するために、ユーザーがオンライン・ログの名前を提供する必要があります。
3. リストアされたデータ・ファイルに、必要な最後の REDO ログが適用されるまで、REDO ログを適用します。Oracle は正しい時刻になるとリカバリを自動的に停止し、リカバリが成功したかどうかを示すメッセージを戻します。
4. RESETLOGS モードでデータベースをオープンします。不完全リカバリまたはバックアップ制御ファイルを使用したリカバリの後では、必ずオンライン・ログをリセットする必要があります。たとえば、次のように入力します。

```
ALTER DATABASE OPEN RESETLOGS;
```

関連項目： 4-28 ページ「[ユーザー管理のメディア・リカバリ後のデータベースのオープン](#)」

変更ベースの不完全リカバリの実行

この項では、指定された SCN までのリカバリを実行する方法を、次の段階に従って説明します。

1. 4-17 ページの「[不完全リカバリの準備](#)」の説明に従って、データベースをバックアップし、メディア障害を修復して、リカバリの準備をします。
2. 4-18 ページの「[不完全リカバリ実行前のデータ・ファイルのリストア](#)」の説明に従って、バックアップ・データ・ファイルをリストアします。現行の制御ファイルがある場合には、バックアップ制御ファイルをリストアしないでください。
3. 次の手順を使用して、リストアされたバックアップに対してメディア・リカバリを実行します。

変更ベースのリカバリを実行する手順は、次のとおりです。

1. リカバリを終了する SCN を指定して、変更ベースのリカバリを開始します。SCN は、引用符のない 10 進数として指定されます。たとえば、SCN 10034 までリカバリするには、次のコマンドを発行します。

```
RECOVER DATABASE UNTIL CHANGE 10034;
```
2. Oracle は、必要な REDO ログ・ファイル（アーカイブおよびオンライン）を適用してリストアされたデータ・ファイルを再構築し、メディア・リカバリのロールフォワード・フェーズを開始します。ファイルの適用が自動化されていないかぎり、Oracle は名前を LOG_ARCHIVE_DEST_1 から検索できるものと想定し、名前を提示し、そのログ・ファイルの適用を停止するか進めるかについて、ユーザーの指示を要求します。制御ファイルがバックアップの場合には、アーカイブ・ログを適用した後で、オンライン・ログの変更を適用するために、ユーザーがオンライン・ログの名前を提供する必要があります。
3. リストアされたデータ・ファイルに必要な最後の REDO ログ・ファイルが適用されるまで、REDO ログ・ファイルの適用を続けます。Oracle は正しい SCN に達するとリカバリを自動的に停止し、リカバリが成功したかどうかを示すメッセージを戻します。
4. RESETLOGS モードでデータベースをオープンします。不完全リカバリまたはバックアップ制御ファイルを使用したリカバリの後では、必ずオンライン・ログをリセットする必要があります。たとえば、次のように入力します。

```
ALTER DATABASE OPEN RESETLOGS;
```

関連項目： 4-28 ページ「[ユーザー管理のメディア・リカバリ後のデータベースのオープン](#)」

NOARCHIVELOG モードでのデータベースのリカバリ

データベースが NOARCHIVELOG モードのときに、メディア障害によりデータ・ファイルの一部またはすべてが破損した場合は、最新のデータベース全体のバックアップをリストアすることが、リカバリの唯一の手段になるのが普通です。通常のバックアップを補うためにエクスポートを使用している場合には、データベースのエクスポートされたバックアップを、再作成されたデータベース、または古いバックアップからリストアされたデータベースにインポートしてデータベースのリストアを試みることもできます。

NOARCHIVELOG モードのデメリットは、最新の全体バックアップの時点からメディア障害の時点までデータベースのリカバリを行うために、この間に実行されたすべての変更を手動で再入力する必要があることです。一方、データベースが ARCHIVELOG モードの場合には、この期間の REDO ログを、アーカイブ・ログ・ファイルまたはオンライン・ログ・ファイルとして利用できる可能性があります。アーカイブ REDO ログを使用すれば、完全リカバリまたは不完全リカバリを使用してデータベースを再構築できるため、失われる処理を最小化できます。

NOARCHIVELOG モードで稼働中のデータベースがメディア障害による損傷を受けた場合に、最新のデータベース全体の一貫性バックアップからデータベースをリストアするには（この時点では唯一のオプション）、後述の手順を使用してください。

デフォルトの場所へのデータベースのリストア

このシナリオでは、メディア障害が修復されており、すべてのデータベース・ファイルを元の場所にリストアできます。

最新のデータベース全体のバックアップをデフォルトの場所にリストアする手順は、次のとおりです。

1. データベースがオープンしている場合は、データベースを停止します。たとえば、次のように入力します。

```
SHUTDOWN IMMEDIATE
```

2. 可能であれば、メディアの問題を解決して、バックアップ・データベース・ファイルを元の場所にリストアできるようにします。
3. 3-7 ページの「[データ・ファイルのリストア](#)」の説明に従って、オペレーティング・システム・コマンドを使用して、最新のデータベース全体のバックアップをリストアします。破損したファイルだけではなく、データベース全体のバックアップの、すべてのデータ・ファイルおよび制御ファイルをリストアします。次の例は、データベース全体のバックアップをリストアします。

```
% cp /oracle/work/BACKUP/tbs* /oracle/dbs # restores datafiles
% cp /oracle/work/BACKUP/cf.f /oracle/dbs # restores control file
```

4. オンライン REDO ログはバックアップされないため、データ・ファイルおよび制御ファイルとともにリストアすることはできません。Oracle がオンライン REDO ログをリセットできるように、いったん不完全リカバリを行う必要があります。

```
RECOVER DATABASE UNTIL CANCEL
CANCEL
```

5. RESETLOGS モードでデータベースをオープンします。このコマンドは現在の REDO ログ順序を 1 にリセットします。

```
ALTER DATABASE OPEN RESETLOGS;
```

RESETLOGS 操作は、オンライン・ログの中のすべての REDO を無効にします。データベース全体のバックアップからリストアした後、ログをリセットすると、バックアップが作成されてから障害の発生時点までの間にデータベースに対して行われたすべての変更が廃棄されます。

新しい場所へのデータベースのリストア

このシナリオでは、元の場所がメディア障害により損傷したため、代替の場所にデータベース・ファイルをリストアします。

最新のデータベース全体のバックアップを新しい場所にリストアする手順は、次のとおりです。

1. データベースがオープンしている場合は、データベースを停止します。たとえば、次のように入力します。

```
SHUTDOWN IMMEDIATE
```

2. 破損したファイルだけではなく、データベース全体のバックアップの、すべてのデータ・ファイルおよび制御ファイルをリストアします。ハードウェアの問題が解決できず、データ・ファイルの一部またはすべてを代替の場所にリストアする必要がある場合は、データベース全体のバックアップを新しい場所にリストアします。たとえば、次のように入力します。

```
% cp /disk2/BACKUP/tbs* /disk3/oracle/dbs # default location
% cp /disk2/BACKUP/cf.f /disk3/oracle/dbs # new location
% cp /disk2/BACKUP/system01.dbf /disk4/temp # new location
```

3. 必要であれば、制御ファイルの新しい場所を示すように、リストアされたパラメータ・ファイルを編集します。たとえば、次のように入力します。

```
CONTROL_FILES = "/disk3/oracle/dbs/cf.f"
```

4. リストアされ、編集されたパラメータ・ファイルを使用してインスタンスを起動し、データベースをマウントしますが、オープンはしません。たとえば、次のように入力します。

```
STARTUP MOUNT
```

5. リストアされたデータ・ファイルのファイル名が異なる場合には、制御ファイルの中で、リストアされたデータ・ファイルの名前を変更します。たとえば、次のように入力します。

```
ALTER DATABASE RENAME FILE '/disk1/oracle/dbs/system01.dbf' TO  
                             '/disk4/temp/system01.dbf';
```

6. オンライン REDO ログが破損したディスク上にある場合、ハードウェアの問題が解決されないときには、各オンライン・ログの新しい場所を指定します。たとえば、次のように入力します。

```
ALTER DATABASE RENAME FILE '/disk1/oracle/dbs/log1.f' TO  
                             '/disk3/oracle/dbs/log1.f';  
ALTER DATABASE RENAME FILE '/disk1/oracle/dbs/log2.f' TO  
                             '/disk3/oracle/dbs/log2.f';
```

7. オンライン REDO ログはバックアップされないため、データ・ファイルおよび制御ファイルとともにリストアすることはできません。Oracle がオンライン REDO ログをリセットできるように、いったん不完全リカバリを行う必要があります。

```
RECOVER DATABASE UNTIL CANCEL;  
CANCEL;
```

8. RESETLOGS モードでデータベースをオープンします。このコマンドは現在の REDO ログ順序を 1 にリセットします。

```
ALTER DATABASE OPEN RESETLOGS;
```

RESETLOGS 操作は、オンライン・ログの中のすべての REDO を無効にします。データベース全体のバックアップからリストアした後、ログをリセットすると、バックアップが作成されてから障害の発生時点までの間にデータベースに対して行われたすべての変更が廃棄されます。

関連項目： データ・ファイルの改名および再配置の詳細は、『Oracle9i データベース管理者ガイド』を参照し、ALTER DATABASE RENAME FILE の詳細は、『Oracle9i SQL リファレンス』を参照してください。

パラレルのメディア・リカバリの実行

メディア・リカバリのロールフォワード・フェーズのチューニングのために、**パラレル・メディア・リカバリ**を使用します。パラレル・メディア・リカバリでは、Oracle はロールフォワード時に各データ・ブロックに別のプロセスを割り当てる分業体制を使用することにより、処理の効率を高めています。たとえば、**PARALLEL 4** を指定してパラレル・リカバリを実行した場合、リカバリ対象のデータ・ファイルが 1 つのみである場合には、ただ 1 つのプロセスではなく、4 つの起動されたプロセスがデータ・ファイルからブロックを読み取り、レコードを適用します。

注意： 一般的にリカバリでは、データ・ブロックの読取りの I/O が中心となります。ブロック・レベルでのパラレル化は、たとえばオペレーティング・システムの非同期 I/O の制約を排除するなどして、I/O 全体を増加できる場合のみ、リカバリのパフォーマンスを向上させることができます。効率的な非同期 I/O を使用するシステムでは、パラレル・メディア・リカバリを使用した場合の改善は小規模なものになるのが一般的です。

SQL*Plus の **RECOVER PARALLEL** コマンドはパラレル・メディア・リカバリを指定します（デフォルトは **NOPARALLEL**）。このコマンドは、参加している全インスタンスで使用可能な CPU の数を **PARALLEL_THREADS_PER_CPU** 初期化パラメータの値で乗算した値に、並列度を設定します。

RECOVER PARALLEL コマンドの形式は次のとおりです。

```
RECOVER PARALLEL integer;
```

integer 変数には、メディア・リカバリに使用されるリカバリ・プロセスの数を設定します。**Real Application Clusters** 構成を使用している場合は、Oracle はこれらのリカバリ・プロセスをインスタンス間で配分する方法を決定します。*integer* が指定されていない場合は、デフォルトのリカバリ・プロセス数が使用されます。

注意： **RECOVERY_PARALLELISM** 初期化パラメータは、インスタンスまたはクラッシュ・リカバリの場合にのみ同時実行リカバリ・プロセスの数を指定します。メディア・リカバリは影響を受けません。

関連項目：

- パラレル・リカバリの詳細は、『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。
- SQL*Plus の **RECOVER ... PARALLEL** 文の詳細は、『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。

ユーザー管理のメディア・リカバリ後のデータベースのオープン

不完全リカバリまたはバックアップ制御ファイルを使用したリカバリを実行した場合には、データベースをオープンするときに、オンライン・ログをリセットする必要があります。リセットされたデータベースの新しいバージョンは、新しい**インカネーション**と呼ばれます。古いインカネーションの RESETLOGS の時点以降に生成されたアーカイブ・ログはすべて、新しいインカネーションでは無効です。

完全リカバリを実行した場合には、RESETLOGS オプションを指定してデータベースをオープンする必要はありません。データベースのこのインカネーションの存続時間中に作成された以前のバックアップおよびアーカイブ・ログはすべて有効です。

この項で説明する項目は、次のとおりです。

- [RESETLOGS 操作](#)
- [オンライン REDO ログをリセットするかの判断](#)
- [RESETLOGS 操作後の処理](#)
- [RESETLOGS 前に作成したバックアップのリカバリ](#)

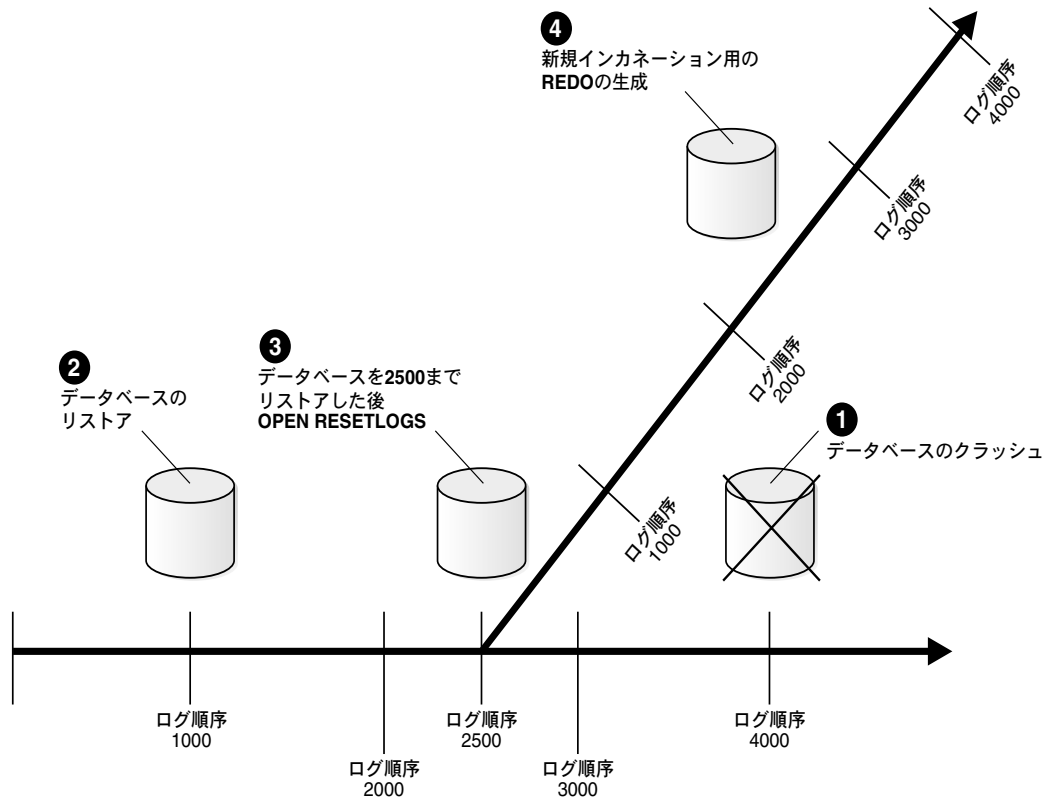
RESETLOGS 操作

RESETLOGS オプションを指定してデータベースをオープンするたびに、すべてのデータ・ファイルで新しい RESETLOGS SCN とタイム・スタンプが設定され、ログ順序番号が 1 にリセットされます。アーカイブ REDO ログでもこれら 2 つの値をファイル・ヘッダー内に保持しています。Oracle は RESETLOGS SCN とタイム・スタンプが一致しないかぎりデータ・ファイルにアーカイブ REDO ログを適用しないため、RESETLOGS 操作は、データ・ファイルが古いアーカイブ・ログにより損傷することを防止します。

[図 4-1](#) は、アーカイブ REDO ログが欠落しているため、ログ順序 2500 までのみリカバリ可能なデータベースのケースを示しています。ログ順序 4000 でデータベースがクラッシュしました。ログ順序 1000 のバックアップをリストアし、完全リカバリの準備をします。残念ながら、アーカイブ REDO ログの 1 つが破損しています。欠落しているログの前のログにログ順序 2500 が含まれているため、この時点までリカバリを行い、RESETLOGS オプションを指定してオープンします。ログ順序は 1 にリセットされます。

この図が示すように、データベースの新しいインカネーションに新しい変更を生成し、最終的にログ順序 4000 に達します。データベースの新しいインカネーションのログ順序 2500 とログ順序 4000 の間の変更は、古いインカネーションのログ順序 2500 とログ順序 4000 の間の変更とはまったく異なります。古いインカネーションから新しいインカネーションにログを適用することは許されません。古いインカネーションのログ順序 2500 より前のバックアップを新しいインカネーションにリストアすることはできません。

図 4-1 新しいデータベース・インカネーションの作成



オンライン REDO ログをリセットするかの判断

RESETLOGS オプションを指定してデータベースをオープンするには、すべてのデータ・ファイルが同じ SCN までリカバリされている必要があります。バックアップ制御ファイルをリストアした場合は、バックアップ制御ファイルも同じ SCN までリカバリする必要があります。

不完全メディア・リカバリ、またはバックアップ制御ファイルを使用したリカバリの後は、常に RESETLOGS オプションが必要です。REDO ログをリセットすると、次のような影響があります。

- リカバリ中に適用されなかった REDO 情報は、適用されることのないように、すべて廃棄されます。
- オンライン REDO ログおよび REDO スレッドに関する制御ファイルのメタデータが再初期化されます。
- オンライン REDO ログの内容が消去されます。
- 現在存在していない場合は、オンライン REDO ログ・ファイルが作成されます。
- ログ順序番号が 1 にリセットされます。

注意： REDO ログをリセットすると、オンライン・ログに含まれている、データベースに対するすべての変更が廃棄されます。このため、RESETLOGS でオープンした後で、リセットされたログ内のある時点まで再度リカバリを実行することはできません。

RESETLOGS を指定するか NORESETLOGS を指定するかは、次のルールを使用して決定してください。

- 不完全リカバリ後は必ず RESETLOGS オプションを指定します。たとえば、将来の時点ではなく、過去の時刻または SCN を指定した場合です。
- 完全リカバリを実行した場合でも不完全リカバリを実行した場合でも、リカバリに制御ファイルのバックアップを使用した場合は、必ず RESETLOGS を指定します。
- 完全リカバリを実行した後は、オプションを指定しないか、NORESETLOGS オプションを指定します（バックアップ制御ファイルを使用した場合を除きます。この場合は RESETLOGS オプションを指定してオープンする必要があります）。
- スタンバイ・データベースのためにアーカイブ・ログを使用している場合は、RESETLOGS オプションは指定しないでください。オンライン・ログのリセットが必要な場合は、スタンバイ・データベースを再作成する必要があります。

ALTER DATABASE OPEN 文の実行

メディア・リカバリ後にデータベースをオープンするときにログ順序番号を維持するには、次のいずれかの文を実行します。

```
ALTER DATABASE OPEN NORESETLOGS;  
ALTER DATABASE OPEN;
```

リカバリ後にデータベースをオープンするときにログ順序番号をリセットし、それによりデータベースの新しいインカネーションを作成するには、次の文を実行します。

```
ALTER DATABASE OPEN RESETLOGS;
```

RESETLOGS オプションを指定してオープンした場合は、リカバリが完全か不完全かにより、戻されるメッセージが異なります。完全リカバリの場合には、`alert_SID.log` ファイルに次のメッセージが表示されます。

```
RESETLOGS after complete recovery through change scn
```

不完全リカバリの場合には、`alert_SID.log` ファイルにこのメッセージが表示されます。この `scn` は、不完全リカバリの終了位置を表します。

```
RESETLOGS after incomplete recovery UNTIL CHANGE scn
```

ユーザーが行うべきでない状況で OPEN RESETLOGS を試みるか、ログをリセットする必要がある状況でログをリセットしないと、Oracle はエラーを戻し、データベースをオープンしません。問題を解決して再度試してください。

関連項目： ALTER DATABASE OPEN RESETLOGS の失敗を招く状況の説明
は、5-2 ページの「[ユーザー管理のメディア・リカバリの問題](#)」を参照してください。

RESETLOGS 操作後の処理

この項では、RESETLOGS モードでデータベースをオープンした後に実行する必要がある処理について説明します。

データベース全体のバックアップの作成

即時にデータベースを通常に停止し、データベース全体のバックアップを作成します。そうしないと、ログをリセットした後で行った変更のリカバリができなくなります。全体バックアップを作成するまでは、リカバリの唯一の方法は、ログのリセットに至るまで、今行った手順を繰り返すことです。ログ順序をリセットしなかった場合には、データベースのバックアップをもう 1 つ作成する必要はありません。

一般的に、RESETLOGS 操作前に作成したバックアップは、新しいインカネーションで使用できません。しかし、このルールには例外があります。事前 RESETLOGS のバックアップは、Oracle がリカバリを実行するために RESETLOGS 前のアーカイブ REDO ログにアクセスする必要がない場合にかぎり、リストアできます。

関連項目： 4-33 ページ「[RESETLOGS 前に作成したバックアップのリカバリ](#)」

アラート・ログのチェック

RESETLOGS オプションを使用してデータベースをオープンした後、alert_SID.log を調べて、たとえばデータ・ディクショナリには含まれているが新しい制御ファイルにはリストされていないデータ・ファイルがあるなど、データ・ディクショナリと制御ファイルの間の矛盾が検出されていないかを確認します。

データ・ディクショナリに存在するデータ・ファイルが新しい制御ファイルに反映されていない場合には、Oracle は制御ファイル内で MISSINGnnnn（この nnnn は 10 進数で示したファイル番号）としてプレースホルダ・エントリを作成します。制御ファイルの中で MISSINGnnnn には、オフラインで、メディア・リカバリが必要であるとのフラグが立てられます。

MISSINGnnnn に対応するデータ・ファイルは、バックアップが作成された時点から RESETLOGS が発行されるまでの間に、データ・ファイルが読取り専用であったか、NORMAL モードでオフラインにされていた場合にのみ、データ・ファイルをポイントするように MISSINGnnnn を改名することにより、アクセス可能にできます。一方、MISSINGnnnn が、リカバリ期間中に読取り専用でもなく NORMAL モードでオフラインにされたものでもなかったデータ・ファイルに対応する場合、このデータ・ファイルは、RESETLOGS の結果により不可能となるメディア・リカバリを必要とするため、改名操作によりデータ・ファイルをアクセス可能にすることはできません。この場合には、データ・ファイルを含む表領域を削除する必要があります。

反対に、制御ファイルの中にあるデータ・ファイルの記述がデータ・ディクショナリの中になければ、Oracle は新しい制御ファイルからこのデータ・ファイルに対する言及を削除します。いずれの場合にも、Oracle は判明事項をユーザーに知らせるために、alert_SID.log ファイルにメッセージを記入します。

RESETLOGS 前に作成したバックアップのリカバリ

Oracle8 より前のリリースでは、DBA は RESETLOGS オプションを指定してデータベースをオープンすることを回避するために、コールドー貫性バックアップを実行するときには、オンライン・ログをバックアップするのが通常でした（即時にリストアする予定の場合）。

この方法の典型的な例は、ディスク・メンテナンスです。この作業ではデータベースをバックアップし、削除し、ディスクを再構成し、データベースをリストアする必要があります。DBA は、RESETLOGS モードで再起動しなければ、リストア後にデータベースを即時にバックアップする必要はないことを認識しました。このバックアップが必要であったのは、特にログのリセット後にエラーが発生した場合には、RESETLOGS 前に作成されたバックアップに対してリカバリを実行することが不可能であったためです。

RESETLOGS 前に作成したバックアップのリストア

RESETLOGS 前に作成した次のようなバックアップを、新しいインカネーションにリストアできます。

- 表領域が読取り専用を設定された後で作成された表領域のバックアップ（RESETLOGS 前に再度読取り / 書込みに変更されていない場合のみ）。
- 表領域が NORMAL モードでオフラインされた後で作成された表領域のバックアップ（RESETLOGS 前に再度オンライン化されていない場合のみ）。
- リカバリの終了後、かつ RESETLOGS でオープンする前に作成された読取り / 書込み表領域の一貫性バックアップ。つまり、バックアップと RESETLOGS の間にさらにリカバリを行ったり、データ・ファイルの変更を行っていないもの。ただし RESETLOGS でオープンした後で有効な制御ファイルがある場合のみ。

RESETLOGS の直前に作成されたものではない、読取り / 書込み表領域のバックアップはリストアできません。この制約は、バックアップと ALTER DATABASE OPEN RESETLOGS の間に読取り / 書込み表領域の中のデータ・ファイルが変更されていない場合にも適用されます。バックアップのデータ・ファイル・ヘッダーの中のチェックポイントは、制御ファイルの中のチェックポイントよりも古いため、Oracle は変更を適用する必要があるかどうかを判断するためにアーカイブ・ログを検索する必要があります（また、RESETLOGS の前に生成されたアーカイブ・ログは、新しいインカネーションでは無効です）。

RESETLOGS 前に作成したバックアップのリストア : シナリオ

次のシナリオは、RESETLOGS 前に作成したバックアップを使用できる状況を示しています。ハードウェア・ストライプ化の再構成を実行する場合を考えてみます。このためには、データベース・ファイルをバックアップしてから削除し、ハードウェアを再構成し、データベースをリストアする必要があります。

金曜日の夜に、次のような処理を実行します。

1. 一貫性をもってデータベースを停止します。たとえば、次のように入力します。

```
SHUTDOWN IMMEDIATE
```

2. データベース全体のバックアップを実行します。たとえば、次のように入力します。

```
% cp /oracle/dbs/* /oracle/backup
```

注意： この時点でデータベースを再オープンしないでください。

3. オペレーティング・システム・メンテナンスを実行します。
4. 作成したばかりのバックアップからデータ・ファイルおよび制御ファイルをリストアします。たとえば、次のように入力します。

```
% cp /oracle/backup/* /oracle/dbs
```

5. データベースをマウントします。たとえば、次のように入力します。

```
STARTUP MOUNT
```

6. 取消ベースのリカバリを開始します。たとえば、次のように入力します。

```
RECOVER DATABASE UNTIL CANCEL
```

7. RESETLOGS オプションを指定してデータベースをオープンします。たとえば、次のように入力します。

```
ALTER DATABASE OPEN RESETLOGS;
```

土曜日の朝に、スケジュールされたジョブが実行され、アーカイブ・ログが生成されます。土曜日の夜に、データベース全体のリストアを必要とするハードウェア・エラーが発生した場合には、RESETLOGS オプションを指定してオープンする直前に作成しておいたバックアップをリストアし、土曜日に生成されたログを使用してロールフォワードできます。

土曜日の夜には次の作業を実行します。

1. インスタンスを終了します（まだ存在している場合）。たとえば、次のように入力します。

```
SHUTDOWN ABORT
```

2. 金曜日の夜に作成したバックアップから、破損したすべてのファイルをリストアします。たとえば、次のように入力します。

```
% cp /oracle/backup/* /oracle/dbs
```

注意： 現行の制御ファイルがある場合には、これはリストアしないでください。そうでない場合、RESETLOGS を指定してデータベースをオープンした後で有効であった制御ファイルをリストアする必要があります。

3. 土曜日に生成されたアーカイブ・ログをすべて適用して、完全リカバリを開始します。ログの適用を自動化するには SET AUTORECOVERY ON を使用します。たとえば、次のように入力します。

```
SET AUTORECOVERY ON  
RECOVER DATABASE
```

4. データベースをオープンします。たとえば、次のように入力します。

```
STARTUP
```

このシナリオで、金曜日の夜のバックアップ以降、RESETLOGS を指定してデータベースをオープンするまでの間にデータベースをオープンした場合、または、データベースをオープンした後の制御ファイルがない場合には、金曜日の夜のバックアップを使用してロールフォワードすることはできません。リカバリのためには、RESETLOGS オプションを指定してデータベースをオープンした後のバックアップが必要です。

ユーザー管理のメディア・リカバリの中断

たとえばリカバリ操作を夜間中止して翌朝再開する必要があるなど、メディア・リカバリを開始した後で中断することが必要な場合には、次のいずれかの処理を実行できます。

- REDO ログ・ファイルのプロンプトに対して **CANCEL** と入力します。
- 個々のデータ・ファイルのリカバリを行っている場合、または自動リカバリの実行中に終了することが必要な場合は、オペレーティング・システムの割込みシグナルを使用します。

リカバリが取り消された後、**RECOVER** コマンドを使用してリカバリを再開できます。リカバリは、取り消された位置から再開されます。

リカバリの再開を必要とするいくつかの要因が考えられます。たとえば、別のバックアップを使用して再開する、同じバックアップを使用しても、最初に指定した時刻よりも前の時点に終了時刻を変更する場合には、バックアップをリストアして、操作全体を再開する必要があります。

RECOVER TABLESPACE または **RECOVER DATAFILE** を指定してデータベースの一部のリカバリを行う場合には、リカバリを再開して、リカバリを終了しないと、データベースのこれらの部分を使用可能にすることはできません。

データベース全体の不完全リカバリを実行している場合は、メディア・リカバリを取り消した後で、読取り専用または **RESETLOGS** でデータベースをオープンできます。この方法は、すべてのデータ・ファイルが一貫性のある SCN までリカバリされている場合と、メディア・リカバリの中断後に成功します。すべてのデータ・ファイルが一貫性のある SCN までリカバリされていない場合には、**RESETLOGS** は失敗することがあります。この場合はさらにメディア・リカバリを実行する必要があります。

ユーザー管理のメディア・リカバリの制限

メディア・リカバリを実行する前に、次の事項を理解してください。

- リカバリ不能の表および索引のユーザー管理のリカバリ
- 現行ではない制御ファイルを使用した、読取り専用表領域のユーザー管理のリカバリ

リカバリ不能の表および索引のユーザー管理のリカバリ

表および索引は CREATE TABLE AS SELECT 文で作成できます。また、**リカバリ不能**として作成するように指定することもできます。ユーザーが表または索引をリカバリ不能として作成した場合、Oracle は操作の REDO ログ・レコードを生成しません。このため、ARCHIVELOG モードで実行している場合でも、リカバリ不能として作成されたオブジェクトのリカバリを行うことはできません。

注意： リカバリ不能として作成された表または索引の破損が許されない場合には、リカバリ不能な表または索引を作成した後で、バックアップを作成します。

メディア・リカバリを実行する場合、表または索引の一部がリカバリ可能として作成され、一部がリカバリ不能として作成されているときには、リカバリ不能なオブジェクトには、RECOVER 操作により論理的に破損しているというマークが付けられます。リカバリ不能なオブジェクトにアクセスを試みると、ORA-01578 エラー・メッセージが戻されます。リカバリ不能なオブジェクトを削除し、必要な場合は再作成します。

リカバリ不能として表を作成し、その表の索引をリカバリ可能として作成することもできるため、メディア・リカバリの実行後、索引には論理的に破損しているというマークが付けられません。表はリカバリ不能であるため（リカバリ後に破損しているというマークが付けられます）、索引は破損ブロックをポイントすることになります。索引を削除し、必要な場合は表および索引を再作成する必要があります。

関連項目： リカバリ不能操作のスタンバイ・データベースへの影響の詳細は、『Oracle9i Data Guard 概要および管理』を参照してください。

現行ではない制御ファイルを使用した、読取り専用表領域のユーザー管理のリカバリ

読取り専用メディア、または低速のメディア上に読取り専用表領域がある場合には、`USING BACKUP CONTROLFILE` オプションを指定してメディア・リカバリを実行した場合に、エラーまたはパフォーマンスの低下が発生することがあります。この状況は、制御ファイルがバックアップされたときに表領域が読取り / 書き込みであったことをバックアップ制御ファイルが示している場合に発生します。この場合には、メディア・リカバリはファイルへの書き込みを試みることがあります。読取り専用メディアの場合には、ファイルへの書き込みができないことを告げるエラー・メッセージが発行されます。テープでバックアップされる階層ストレージ・システムなど、低速メディアの場合には、パフォーマンスが低下することがあります。

これらのリカバリの問題を回避するには、バックアップではなく、現行の制御ファイルを使用して、データベースのリカバリを行います。バックアップ制御ファイルを使用する必要がある場合には、読取り専用表領域がメディア障害の影響を受けていなければ、この問題を回避できます。

バックアップ制御ファイルを使用した、読取り専用または低速メディアのリカバリ

バックアップ制御ファイルを使用する場合には、読取り専用または低速メディアのリカバリに次の代替方法を使用できます。

- バックアップ制御ファイルを使用してリカバリを実行する前に、読取り専用表領域のデータ・ファイルをオフライン化し、メディア・リカバリの最後に、ファイルをオンライン化します。
- リカバリには正しいバージョンの制御ファイルを使用します。リカバリの完了時に表領域が読取り専用になる場合には、表領域が読取り専用であった時点の制御ファイルを使用する必要があります。同様に、リカバリの終了時に表領域が読取り / 書き込みになる場合には、表領域が読取り / 書き込みであった時点の制御ファイルを使用する必要があります。

再作成された制御ファイルを使用した、読取り専用ファイルのリカバリ

リカバリに現行の制御ファイルまたはバックアップ制御ファイルを使用できない場合は、3-15 ページの「[現行の制御ファイルおよびバックアップ制御ファイルのすべてが損害を受けた場合](#)」の説明に従って、`CREATE CONTROLFILE` 文を実行できます。リカバリでスキップできるように、読取り専用ファイルは `CREATE CONTROLFILE` 文に含めないでください。リストアされた読取り専用ファイルのバックアップが、ファイルが読取り / 書き込みであった時点のものでないかぎり、読取り専用ファイルのリカバリは必要ありません。

ユーザーが新しい制御ファイルを作成し、データベースのマウントおよびオープンを試みると、**Oracle** は制御ファイルに列挙されたファイルに対してデータ・ディクショナリのチェックを実行します。CREATE CONTROLFILE 文に含まれていないが、データ・ディクショナリに存在するファイルについては、制御ファイル内にエントリが作成されます。**Oracle** はこれらのファイルに MISSINGnnnn という名前を付けます。この nnnn は、0 (ゼロ) から開始する、5 桁の数字です。

データベースをオープンした後、名前の前に MISSING という接頭辞があるすべてのファイルについて、ALTER DATABASE RENAME FILE 文を実行して、読取り専用ファイルを正しいファイル名に改名します。

制御ファイルの再作成が必要となる状況に備えるために、データベースをマウントまたはオープンしているときに次の文を実行し、CREATE CONTROLFILE 構文を取得します。

```
ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
```

この SQL 文は、ユーザーが編集し、リカバリ・シナリオで制御ファイルを再作成するためのスクリプトとして使用できるトレース・ファイルを作成します。RESETLOGS または NORESETLOGS (デフォルト) キーワードを指定して、CREATE CONTROLFILE ... RESETLOGS または CREATE CONTROLFILE ... NORESETLOGS 対応のスクリプトを生成できます。

CREATE CONTROLFILE 文の読取り専用ファイルに関連するすべての制限は、データベースをオープンした後で表領域をオンライン化する必要があることを除き、NORMAL モードでオフラインされた表領域にも当てはまります。一時ファイルは CREATE CONTROLFILE 文から除外し、データベースをオープンした後で追加する必要があります。

関連項目： 制御ファイルのトレース・バックアップの作成については、2-21 ページの「[制御ファイルのトレース・ファイルへのバックアップ](#)」を参照してください。

ユーザー管理のメディア・リカバリの トラブルシューティング

この章では、ユーザー管理のメディア・リカバリのトラブルシューティングの方法について説明します。この章で説明する項目は、次のとおりです。

- [ユーザー管理のメディア・リカバリの問題](#)
- [メディア・リカバリの問題の調査 : フェーズ 1](#)
- [ブロックを破損させない修正の試行 : フェーズ 2](#)
- [リカバリによるブロックの破損を許可するかどうかの決定 : フェーズ 3](#)
- [リカバリによるブロックの破損の許可 : フェーズ 4](#)
- [試行リカバリの実行](#)

ユーザー管理のメディア・リカバリの問題

表 5-1 は、メディア・リカバリ中に発生する可能性のある問題を示しています。

表 5-1 メディア・リカバリの問題

問題	説明
アーカイブ・ログの欠落 または名前の誤り	Oracle が制御ファイルに記録されたアーカイブ・ログを見つけられないため、リカバリは停止します。
データベースをオープンしようとすると ORA-1113 エラーが、ファイルのメディア・リカバリが必要なことを示す	<p>このエラーの一般的な原因は次のとおりです。</p> <ul style="list-style-type: none">■ 不完全リカバリを実行していますが、必要なすべてのデータ・ファイル・バックアップがリストアされていません。■ データ・ファイルが一貫性のある SCN に達する前に、不完全リカバリが停止しました。■ オンライン・バックアップからデータ・ファイルのリカバリを行っていますが、データ・ファイルを一貫性のある状態にするための十分な REDO が適用されていません。■ バックアップ制御ファイルを使用してリカバリを行っていますが、必要なオンライン・ログの場所が指定されていません。■ ユーザーがデータベースをオープンしようとしたときに、データ・ファイルに対してメディア・リカバリが実行されていました。■ リカバリの必要なデータ・ファイルは、RECOVER DATABASE の実行前にオンライン化されていなかったため、リカバリが行われていません。
REDO レコードの問題	<p>2 つの状況が考えられます。</p> <ul style="list-style-type: none">■ 一貫性チェックが失敗したためにリカバリが停止しました。スタック・リカバリと呼ばれる問題です。スタック・リカバリは、基礎となるオペレーティング・システムまたはストレージ・システムが、データベースの通常稼働中に Oracle から発行された書込みを失った場合に発生することがあります。■ Oracle が REDO を適用するときに内部エラーを通知します。この問題は Oracle のバグによって発生することがあります。チェックサムが使用されていない場合には、REDO またはデータ・ブロックの破損により発生することもあります。
アーカイブ・ログの破損	格納またはストレージ・システム間でコピーされるときに、ログが破損した可能性があります。DB_BLOCK_CHECKSUM が有効になっている場合には、Oracle は通常チェックサム・エラーを通知します。チェックサムがオンになっていない場合、ログの破損が、REDO の問題として表示されることがあります。

表 5-1 メディア・リカバリの問題（続き）

問題	説明
互換性のないパラレル REDO フォーマットによるアーカイブ・ログ	<p>Oracle9i リリース 2 (9.2) でパラレル REDO 機能を使用可能にしている場合、Oracle では REDO ログが新規のフォーマットで生成されます。以前のリリースの Oracle では、パラレル REDO ログを適用できません。ただし、Oracle9i リリース 1 (9.0.1) では、パラレル REDO フォーマットを検出し、エラー・メッセージ「External error 00303, 00000, "パラレル REDO は処理できません"」によって非一貫性の存在を確認できます。</p> <p>関連項目: パラレル REDO 機能の詳細は、『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。</p>
データ・ブロックの破損	<p>データ・ファイルのバックアップに破損したデータ・ブロックが含まれているか、リカバリ中またはバックアップにコピーされるときにデータ・ブロックが破損しました。チェックサムが使用されている場合には、Oracle はチェックサム・エラーを通知します。そうでない場合には、問題が REDO の破損として表示されることがあります。</p>
稀に起きる問題	<p>リカバリ中には、メモリーの破損や、その他の一時的問題が発生することがあります。</p>

メディア・リカバリの問題は通常、リカバリ中に通知される外部エラーまたは内部エラーとして出現します。たとえば、外部エラーは、REDO ブロックまたはデータ・ブロックでチェックサム検証チェックが失敗したことを示します。内部エラーは、Oracle のバグか、基礎となるオペレーティング・システムおよびハードウェアに起因するエラーにより発生する可能性があります。

メディア・リカバリで、データベース・バックアップのリカバリ中に問題が発生した場合には、スタック・リカバリの問題であっても、REDO 適用中の問題であっても、Oracle は常に停止し、リカバリ中のデータ・ファイルを一貫性のある状態、つまり、障害前の SCN のままにします。次のいずれかを実行できます。

- データベースを読取り専用でオープンし、問題を調査する。
- RESETLOGS のオープンの要件が満たされているかぎり、RESETLOGS オプションを指定してデータベースをオープンする（4-28 ページの「[ユーザー管理のメディア・リカバリ後のデータベースのオープン](#)」の説明を参照）。スタンバイ・データベースはメディア・リカバリの形態により更新されるため、RESETLOGS の制限は、スタンバイ・データベースをオープンする場合にも適用されます。

一般的に、データベースを読取り専用でオープンしたり、RESETLOGS オプションを指定してオープンする場合には、すべてのオンライン・データ・ファイルを同じ SCN までリカバリする必要があります。この要件が満たされない場合には、ユーザーがオープンを試みると、Oracle はORA-1113 またはその他のエラーを通知することがあります。表 5-1 には、ORA-1113 の一般的な原因を示しています。

メディア・リカバリの問題に対する一般的な対処は、次のようなフェーズで行われます。

1. 問題の原因を識別してみます。必要であれば試行リカバリを実行します。
2. 問題がログの欠落に係るか、ログ、メモリーまたはデータ・ブロックの破損が疑われる場合には、表 5-2 に示された方法を使用して解決を試みます。
3. 表 5-2 に説明された方法を使用しても問題を解決できない場合は、次のいずれかを実行します。
 - データベース全体のバックアップのリカバリを行う場合は、RESETLOGS オプションを指定してデータベースをオープンします。シリアル・メディア・リカバリを実行した場合には、データベースには、破損が発生した SCN の前までのすべての変更が含まれています。この SCN 以降の変更は、データベースのリカバリされた部分に含まれていません。オンライン・バックアップをリストアした場合は、RESETLOGS によるオープンは、REDO ストリーム中のすべての ALTER ... END BACKUP 操作までリカバリが完了している場合にのみ成功します。
 - データ・ブロックを破損させることをメディア・リカバリに許可して、リカバリを続行します。メディア・リカバリが完了した後、Recovery Manager を使用してブロック・メディア・リカバリを実行してください。
 - 最後の手段として、オラクル社カスタマ・サポート・センターに連絡します。

関連項目： ブロック・メディア・リカバリについては、『Oracle9i Recovery Manager ユーザーズ・ガイド』を参照してください。

メディア・リカバリの問題の調査：フェーズ 1

メディア・リカバリで問題が起きた場合は、リカバリが停止した後、できるだけ多くの情報を入手してください。解決する問題を誤って時間を消費し、さらに問題を悪化させることを防ぐためです。

調査で最初に行うことは、問題が設定の誤りか、ログの破損か、データ・ブロックの破損か、メモリーの破損か、またはその他の問題によって発生しているかを確認することです。データ・ブロックのチェックサムにエラーがある場合には、データ・ブロックが破損しています。REDO ログ・ブロックのチェックサムにエラーがある場合には、REDO ログが破損しています。

リカバリの問題の原因は、判別が困難なことがあります。それでも、この章で紹介する方法を使用することにより、問題の原因が完全にはわからない場合にでも、データベースのリカバリを迅速に実行できる場合があります。

メディア・リカバリの問題を調査する手順は、次のとおりです。

1. alert.log を調べて、エラー・メッセージに問題の性質についての一般的な情報が記載されているかを確認します。たとえば、alert_SID.log はチェックサムのエラーを示しているでしょうか。alert_SID.log は、メディア・リカバリを続行するためにはデータ・ブロックを破損させる必要があることを示しているでしょうか。
2. リカバリ中に Oracle プロセスから生成されたトレース・ファイルを調べます。追加のエラー情報が含まれていることがあります。

ブロックを破損させない修正の試行：フェーズ 2

疑われるメディア・リカバリの問題のタイプによっては、いくつかの解決方法があります。表 5-2 に説明されている方法の 1 つ、あるいはその組合せを使用できます。これらの方法は比較的安全なものです。ほとんどすべての場合に、データベースに対する害はありません。

表 5-2 メディア・リカバリの解決方法

疑われる状況	対応
アーカイブ・ログの欠落または名前の誤り	ファイル名が正しく入力されているかを確認します。正しく入力されていた場合には、オペレーティング・システムからログが欠落しているかを確認します。欠落している場合に、バックアップがあれば、バックアップをリストアし、ログを適用します。バックアップがない場合には、可能であれば、欠落したログの時点まで不完全リカバリを実行します。
ALTER DATABASE OPEN での ORA-1113	表 5-1 でこのエラーの原因を検討します。リカバリが必要なすべての読取り / 書込みデータ・ファイルがオンライン化されていることを確認します。リカバリにバックアップ制御ファイルを使用した場合には、制御ファイルとデータ・ファイルの SCN に一貫性がないかぎり、データベースをオープンできません。必要な REDO が不在場合には、制御ファイルを再作成する必要があります。
アーカイブ・ログの破損	ログ REDO ブロックのチェックサム検証が失敗したときには、ログが破損しています。リカバリ・セッション中、またはデータベースが REDO を生成したときに DB_BLOCK_CHECKSUM が有効になっていなかった場合には、リカバリの問題の原因としてログの破損が考えられます。ログが破損し、破損したログの代替コピーが使用できる場合には、このコピーを適用して、これにより問題が修正されるかを確認します。 DB_BLOCK_CHECKSUM 初期化パラメータは、REDO ログおよびデータ・ブロックに対してチェックサムを計算するかどうかを決定します。

表 5-2 メディア・リカバリの解決方法（続き）

疑われる状況	対応
互換性のないパラレル REDO フォーマットによるアーカイブ・ログ	Oracle9i リリース 2 (9.2) より前の Oracle リリースを実行している場合、パラレル REDO フォーマットで作成された REDO ログを適用するには、次の手順に従う必要があります。 1. データベースを Oracle9i リリース 2 (9.2) にアップグレードします。 2. メディア・リカバリを実行します。 3. データベースを一貫した状態で停止し、バックアップを行います。 4. データベースを元のリリースにダウングレードします。 関連項目： パラレル REDO 機能の詳細は、『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。
メモリーの破損または一時的問題	データベースを停止し、リカバリを再起動します。この方法で問題が解決される場合もあります。2 回目にも失敗した場合には、Oracle はデータベースを一貫性のある状態にします。
データ・ブロックの破損	ユーザー管理の方法を使用して、データ・ファイルのリストアおよびリカバリを再度実行するか、Recovery Manager の BLOCKRECOVER コマンドを使用して、個々のデータ・ブロックのリストアおよびリカバリを行います。この方法で問題が解決される場合もあります。 データ・ブロックのチェックサム検証が失敗した場合には、データ・ブロックが破損しています。DB_BLOCK_CHECKING が有効化されていない場合には、データ・ブロックの破損の問題が、REDO の問題として表示されることがあります。リカバリを続行する必要がある場合には、ここでブロックを破損させてリカバリを続け、後で Recovery Manager を使用してブロック・メディア・リカバリを実行できます。

表 5-2 で説明されている方法で問題を解決できない場合には、データを失わずに、簡単に問題を解決できる方法がない可能性があります。次のようなオプションがあります。

- RESETLOGS オプションを指定してデータベースをオープンします（データベース全体のリカバリのため）。このソリューションでは、REDO の問題が発生した時点以降のすべての変更が廃棄されますが、データベースの論理的一貫性は保証されます。
- メディア・リカバリに 1 つ以上のデータ・ブロックを破損させることを許可して、リカバリを続行します。このオプションは、alert_SID.log が、データ・ブロックを破損させてもよければリカバリを続行できることを示している場合にのみ成功します。これはほとんどのリカバリの問題に当てはまります。データベースを迅速に立ち上げ、すべての変更を回復することが重要である場合には、このオプションが最良です。最後の手段としてこのオプションを検討する場合には、5-7 ページの「リカバリによるブロックの破損を許可するかどうかの決定：フェーズ 3」に進んでください。

関連項目： BLOCKRECOVER コマンドを使用してブロック・メディア・リカバリを実行する方法については、『Oracle9i Recovery Manager ユーザーズ・ガイド』を参照してください。

リカバリによるブロックの破損を許可するかどうかの決定：フェーズ 3

メディア・リカバリで問題が起きた場合、問題の原因となっているデータ・ブロックを破損させることが許可されればリカバリを続行できることが、`alert_SID.log` に示されています。また、`alert_SID.log` には常にブロックに関する情報（ブロック・タイプ、ブロック・アドレス、所属する表領域など）が含まれています。ユーザー・データを含むブロックの場合には、データ・オブジェクト番号もログで報告されることがあります。

この場合、問題のブロックに破損というマークを付けることが許可されれば、Oracle はリカバリを続行できます。しかし、この方法は常に正しいとはかぎりません。たとえば、SYSTEM 表領域の中の重要なブロックの場合には、ブロックに破損というマークを付けると、最終的にはリカバリ済みのデータベースをオープンできなくなる可能性があります。もう 1 つ考慮すべき点は、リカバリの問題を分離できるかどうかです。この問題の直後に、REDO ストリームの中でその他多数の問題が発生する場合には、RESETLOGS オプションを指定してデータベースをオープンします。

ユーザー・データを含むブロックの場合には、ブロックを所有するオブジェクトまたは表を確認するために、通常はデータベースに問い合わせることができます。データベースがオープンしていない場合には、データベース全体のバックアップのリカバリ中であっても、データベースを読取り専用でオープンできます。次の例はリカバリを取り消して、読取り専用でオープンします。

```
CANCEL
ALTER DATABASE OPEN READ ONLY;
```

`alert_SID.log` で報告されているデータ・オブジェクト番号が 8031 であるとし、次の問合せを発行して、所有者、オブジェクト名およびオブジェクト型を確認できます。

```
SELECT OWNER, OBJECT_NAME, SUBOBJECT_NAME, OBJECT_TYPE
FROM DBA_OBJECTS
WHERE DATA_OBJECT_ID = 8031;
```

リカバリの問題が分離されているかどうかを確認するために、問題を探して REDO ストリームをスキャンするが、リカバリされたデータベースに対して実際の変更は行わない、診断用の**試行リカバリ**を実行できます。試行リカバリはリカバリの問題を検出した場合、`alert_SID.log` の中に報告します。RECOVER ... TEST 文を使用して試行リカバリを起動できます。

関連項目： 5-9 ページ「[試行リカバリの実行](#)」

これらの調査を実行したあと、表 5-3 のガイドラインに従って、リカバリにブロックを破損させることを許可するかどうかを決定できます。

表 5-3 許容リカバリに破損を許可する場合のガイドライン

問題の状態	ブロック	対応
分離されていない	N/A	おそらく、RESETLOGS オプションを使用してデータベースをオープンする必要があります。スタック・リカバリはオペレーティング・システムまたはストレージ・システムが書込みを失ったために発生することがあるため、この対応はスタック・リカバリの問題にとって重要です。オペレーティング・システムまたはストレージ・システムで突然エラーが発生した場合には、複数のブロックでスタック・リカバリの問題が発生する可能性があります。
分離されている	SYSTEM 表領域内	最終的にデータベースをオープンできなくなる可能性があるため、ブロックは破損させないでください。しかし、SYSTEM 表領域の中のデータが重要でないこともあります。SYSTEM ブロックを破損させ、すべての変更のリカバリを行う必要がある場合には、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。
分離されている	索引データ	データベースのリカバリ後に索引を再作成できるため、索引ブロックを破損させることを検討してください。
分離されている	ユーザー・データ	データの重要性に基づいて決定します。データ・ファイルのリカバリを続行し、ブロックを破損させると、ブロック内のデータが失われます。しかし、データ・ファイルのリカバリが完了した後で、Recovery Manager を使用してブロック・メディア・リカバリを実行できます。RESETLOGS でオープンすると、データベースは一貫性のあるものになりますが、リカバリが停止された時点以降のすべての変更は失われます。
分離されている	ロールバックまたは UNDO データ	UNDO を生成したトランザクションがロールバックされなければ、データベースに害はないため、ロールバックまたは UNDO ブロックを破損させることを検討してください。しかし、これらのトランザクションがロールバックされた場合には、UNDO ブロックを破損させたことにより、問題が発生する可能性があります。判断できない場合は、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。

関連項目： 試行リカバリの実行方法については、5-9 ページの「[試行リカバリの実行](#)」を参照し、ブロックを破損させることに決定した場合は、5-9 ページの「[リカバリによるブロックの破損の許可：フェーズ 4](#)」を参照してください。

リカバリによるブロックの破損の許可：フェーズ 4

リカバリの続行のために、ブロックの破損を許可することを決定した場合には、`ALLOW n CORRUPTION` 句を指定して `RECOVER` コマンドを実行します。この `n` は、破損させることを許可するブロックの数です。

ブロックを破損させることをリカバリに許可する手順は、次のとおりです。

1. リカバリの通常の前提条件がすべて満たされていることを確認します。たとえば、データベースがオープンしている場合には、リカバリを試みる前に、表領域をオフライン化します。
2. 1 つの破損を許可する、`RECOVER` コマンドを実行します。必要な破損の数だけ繰り返します。次の文は有効な例を示しています。

```
RECOVER DATABASE ALLOW 1 CORRUPTION
```

試行リカバリの実行

この項では、次の項目について説明します。

- [試行リカバリ](#)
- [試行リカバリの仕組み](#)
- [RECOVER ... TEST 文の実行](#)

試行リカバリ

スタック・リカバリなどの問題が発生した場合には、ユーザーは困難な選択に直面します。ブロックが比較的重要でない場合に、問題が分離されている場合には、ブロックを破損させた方がいいはずですが、問題が分離されていない場合には、`RESETLOGS` オプションを指定してデータベースをオープンした方がいい場合もあります。

Oracle はこのような状況のために、試行リカバ리를サポートしています。試行リカバリは、通常メディア・リカバリと同じ方法で `REDO` を適用しますが、ディスクに変更を書き込むことはなく、必ず変更をロールバックします。試行リカバリはメモリー内でのみ発生します。

関連項目： 5-9 ページ「[リカバリによるブロックの破損の許可：フェーズ 4](#)」

試行リカバリの仕組み

試行リカバリはデフォルトとして、スタック・リカバリやその他の問題に遭遇すると、このアクションによりリカバリを続行できる場合には、必ずメモリー内でデータ・ブロックに破損しているというマークを付けます。Oracle は、試行リカバリ中に生成されたエラーをアラート・ファイルに書き出します。Oracle はこれらのエラーに、テスト実行のエラーであるという明確なマークを付けます。

試行リカバリは通常のメディア・リカバリと同様に、アーカイブ・ログ・ファイル名をプロンプトで指示し、ログの適用をユーザーに要求できます。試行リカバリは次の場合に終了します。

- 試行リカバリが使用を許可されているメモリー内のバッファの最大数を使用し尽くしたとき
- リカバリ不能なエラー、つまり、データ・ブロックを破損させることにより解決できないエラーが通知されたとき
- ユーザーがリカバリ・セッションを取り消したか中断したとき
- REDO ストリーム内の次の REDO レコードが制御ファイルを変更したとき
- 必要な REDO がすべて適用されたとき

試行リカバリが終了すると、Oracle はアラート・ファイル内のエラー・メッセージを除き、テスト実行のすべての影響をシステムから削除します。試行リカバリ中にインスタンスの障害が発生した場合には、試行リカバリが変更をディスクに書き込むことはないため、Oracle は試行リカバリのすべての影響をシステムから削除します。

試行リカバリを使用することにより、通常のリカバリを続行した場合に発生する可能性のある問題を予見することができます。進行中のメモリーの破損を原因とする問題の場合には、試行リカバリと通常のリカバリで遭遇するエラーが異なることがあります。

RECOVER ... TEST 文の実行

TEST オプションはすべての RECOVER コマンドで使用できます。たとえば、SQL*Plus を起動し、次のコマンドのいずれかを発行できます。

```
RECOVER DATABASE TEST
RECOVER DATABASE USING BACKUP CONTROLFILE UNTIL CANCEL TEST
RECOVER TABLESPACE users TEST
RECOVER DATABASE UNTIL CANCEL TEST
```

デフォルトでは試行リカバリは、このアクションにより試行リカバリを続行できる場合には、必ずメモリー内でブロックを破損させようとします。つまり、試行リカバリはデフォルトとして、いくつでもデータ・ブロックを破損させることができます。試行リカバリがメモリー内で破損させることができるデータ・ブロックの数を制限するために、RECOVER ... TEST 文で ALLOW *n* CORRUPTION 句を指定できます。

試行リカバリのコマンドは、通常リカバリのコマンドを使用できるすべてのシナリオで使用できます。しかし、試行リカバリを実行する必要があるのは、リカバリで問題が発生する場合のみです。

ユーザー管理のメディア・リカバリのシナリオ

この章では、一般的なメディア障害からのリカバリ方法について説明します。この章で説明する項目は、次のとおりです。

- データ・ファイルが損害を受けた後のリカバリ : 使用例
- データ・ファイルの追加を伴うリカバリ : 使用例
- トランスポータブル表領域のリカバリ : 使用例
- オンライン REDO ログ・ファイルが損害を受けた後のリカバリ : 使用例
- アーカイブ REDO ログ・ファイルが損害を受けた後のリカバリ : 使用例
- ユーザー・エラーからのリカバリ : 使用例
- 分散環境でのメディア・リカバリの実行 : 使用例

データ・ファイルが損害を受けた後のリカバリ：使用例

メディア障害がデータ・ファイルに影響を与えた場合、リカバリの手順は次の事項に依存します。

- データベースのアーカイブ・モード：ARCHIVELOG または NOARCHIVELOG
- メディア障害のタイプ
- メディア障害の影響を受けたファイル

この後の項では、データベースのアーカイブ・モードに基づいた適切なリカバリ計画について説明します。

- [NOARCHIVELOG モードでデータ・ファイルが損害を受けた場合](#)
- [ARCHIVELOG モードでデータ・ファイルが損害を受けた場合](#)

NOARCHIVELOG モードでデータ・ファイルが損害を受けた場合

永続的または一時的メディア障害により、NOARCHIVELOG モードで稼働しているデータベースのデータ・ファイルが影響を受けた場合には、Oracle はデータベースを自動的に停止します。障害のタイプに基づいて、次のリカバリ方法の 1 つを使用します。

メディア障害のタイプ	対応
一時的	ハードウェアの問題を解決してデータベースを再起動します。通常はクラッシュ・リカバリが可能です。オンライン REDO ログを使用して、コミット済みのすべてのトランザクションのリカバリを行うことができます。
永続的	4-10 ページの「 ユーザー管理の完全メディア・リカバリの実行 」の手順に従ってください。

ARCHIVELOG モードでデータ・ファイルが損害を受けた場合

永続的または一時的メディア障害により、ARCHIVELOG モードで稼働しているデータベースのデータ・ファイルが影響を受けた場合には、次のシナリオを想定できます。

損害を受けたデータ・ファイル	データベース・ステータス	解決方法
SYSTEM 表領域内のデータ・ファイルまたはアクティブなロールバック・セグメントまたは UNDO セグメントを持つデータ・ファイル	Oracle は停止します。	ハードウェアの問題が一時的な場合には、問題を解決し、データベースを再起動します。通常、クラッシュ・リカバリにより損害を受けたトランザクションがリカバリされます。ハードウェアの問題が永続的な場合には、4-10 ページの「 クローズしているデータベースのリカバリの実行 」を参照してください。
SYSTEM 表領域内のものではないデータ・ファイル、またはアクティブなロールバック・セグメントまたは UNDO セグメントを含まないデータ・ファイル	Oracle は影響を受けたデータ・ファイルをオフライン化しますが、データベースはオープンしたままです。	データベースの影響を受けていない部分を使用可能なままにしておく必要がある場合には、データベースを停止しないでください。一時オプションを使用して、問題のデータ・ファイルを含む表領域をオフラインにした後、4-14 ページの「 オープン状態のデータベースでのデータ・ファイルのリカバリの実行 」の手順に従ってください。

データ・ファイルの追加を伴うリカバリ : 使用例

バックアップ制御ファイルを使用したデータベースのリカバリが、CREATE TABLESPACE または ALTER TABLESPACE ADD DATAFILE 操作を伴ってロールフォワードされた場合には、追加されたファイルへ REDO レコードを適用するときに Oracle はリカバリを停止し、ユーザーがファイル名を確認できるようにします。

たとえば、データベース全体のバックアップを作成した後、/oracle/dbs/db2.f および /oracle/dbs/db3.f という 2 つのデータ・ファイルを含む新しい表領域を作成するとします。その後バックアップ制御ファイルをリストアし、CREATE TABLESPACE 操作を伴うメディア・リカバリを実行すると、CREATE TABLESPACE の REDO データの適用時に Oracle から次のエラーが通知されることがあります。

ORA-00283: エラーによってリカバリ・セッションは取り消されました。
ORA-01244: メディア・リカバリによって名前なしデータ・ファイルが制御ファイルに追加されました
ORA-01110: データ・ファイル 3: '/oracle/dbs/db2.f'
ORA-01110: データ・ファイル 2: '/oracle/dbs/db3.f'

ADD DATAFILE 操作を伴うリカバリの手順は、次のとおりです。

1. V\$DATAFILE から選択して、追加されたファイルを表示します。たとえば、次のように入力します。

```
SELECT FILE#,NAME
FROM V$DATAFILE;
```

FILE#	NAME
1	/oracle/dbs/db1.f
2	/oracle/dbs/UNNAMED00002
3	/oracle/dbs/UNNAMED00003

2. 名前のないファイルが複数存在する場合には、次の方法の 1 つを使用して、どの名前なしファイルがどのデータ・ファイルに対応するかを確認します。
 - 名前のない各ファイルについて、元のファイルの場所についてのメッセージを含む alert_SID.log をオープンします。
 - エラー・メッセージおよび V\$DATAFILE から、名前のない各ファイルの元のファイルの場所を導出します。名前のないファイルは、エラー・メッセージの中の同じファイル番号のファイルに対応します。
3. ALTER DATABASE RENAME FILE 文を発行してデータ・ファイルを改名します。たとえば、次のように入力します。

```
ALTER DATABASE RENAME FILE '/db/UNNAMED00002' TO '/oracle/dbs/db3.f';
ALTER DATABASE RENAME FILE '/db/UNNAMED00003' TO '/oracle/dbs/db2.f';
```

4. 前述のリカバリ文を発行してリカバリを続行します。たとえば、次のように入力します。

```
RECOVER AUTOMATIC DATABASE USING BACKUP CONTROLFILE UNTIL CANCEL
```

トランスポータブル表領域のリカバリ：使用例

Oracle のトランスポータブル表領域機能を使用すると、ユーザーは表領域のセットを 1 つのデータベースから別のデータベースにトランスポートできます。表領域をデータベースにトランスポートすることは、あらかじめロードされたデータを使用して表領域を作成することに似ています。この機能を使用することには、次のメリットがあります。

- データ・ファイルのコピーとメタデータの統合のみが行われるため、エクスポートまたは SQL*Loader ユーティリティを使用した場合よりも高速です。
- 索引データを移動できます。それにより索引を再作成する必要がなくなります。

トランスポータブル表領域は、通常の表領域と同じように、リカバリ可能です。通常の表領域はバックアップなしにリカバリできますが、移動された表領域のリカバリには、移動されたデータ・ファイルのバージョンが必要です。

トランスポータブル表領域のリカバリの手順は、次のとおりです。

1. データベースがオープンしている場合は、移動された表領域をオフライン化します。たとえば、users 表領域のリカバリの場合には、次のコマンドを発行します。

```
ALTER TABLESPACE users OFFLINE IMMEDIATE;
```

2. オペレーティング・システム・ユーティリティを使用して、トランスポートされたデータ・ファイルのバックアップをリストアします。このバックアップは、トランスポートされたデータ・ファイルの最初のバージョンでも、表領域がトランスポートされた後で作成されたバックアップでもかまいません。たとえば、次のように入力します。

```
% cp /backup/users.dbf /oracle/dbs/users.dbf
```

3. 表領域を通常にリカバリします。たとえば、次のように入力します。

```
RECOVER TABLESPACE users
```

Oracle は、CREATE TABLESPACE 操作を伴うリカバリの場合と同じように、トランスポータブル表領域の操作によるリカバリでも ORA-01244 を通知することがあります。この場合には、6-3 ページの「[データ・ファイルの追加を伴うリカバリ：使用例](#)」の手順を使用して、名前のないファイルを正しい場所に改名してください。

関連項目： トランスポータブル表領域機能の使用方法の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

オンライン REDO ログ・ファイルが損害を受けた後のリカバリ：使用例

データベースのオンライン REDO ログがメディア障害の影響を受けたときの正しいリカバリ手順は、次の条件に依存します。

- オンライン REDO ログの構成：ミラー化されているかいらないか
- メディア障害のタイプ：一時的か永続的か
- メディア障害の影響を受けたオンライン REDO ログ・ファイルのタイプ：現行、アクティブ、アーカイブされていない、または非アクティブ

表 6-1 は、オンライン REDO ログを含むリカバリ状況にとって重要な、V\$LOG のステータス情報を示しています。

表 6-1 V\$LOG の STATUS 列

ステータス	説明
UNUSED	このオンライン REDO ログは書き込まれたことはありません。
CURRENT	このログはアクティブです。つまり、インスタンス・リカバリに必要です。また、Oracle が現在書込みを行っているログです。REDO ログはオープンまたはクローズです。
ACTIVE	このログはアクティブです。つまり、インスタンス・リカバリに必要です。しかし、Oracle が現在書込みを行っているログではありません。ブロック・リカバリに使用されている可能性があります。また、アーカイブされていることもあればされていないこともあります。
CLEARING	このログは、ALTER DATABASE CLEAR LOGFILE 文の後で、空のログとして、再作成中です。ログが消去されると、ステータスは UNUSED に変更されます。
CLEARING_CURRENT	現在のログはクローズ・スレッドの消去中です。新しいログ・ヘッダーへの書込み時の I/O エラーなど、なんらかのスイッチの障害が発生した場合には、ログがこのステータスのままになることがあります。
INACTIVE	このログはインスタンス・リカバリに必要とされなくなりました。メディア・リカバリで使用されることがあります。アーカイブされることもあればされないこともあります。

この後の項では、これらの状況での正しいリカバリ計画について説明します。

- 多重オンライン REDO ログ・グループの一部のメンバーが損害を受けた後のリカバリ
- オンライン REDO ログ・グループのすべてのメンバーが損害を受けた後のリカバリ

多重オンライン REDO ログ・グループの一部のメンバーが損害を受けた後のリカバリ

データベースのオンライン REDO ログが多重化されている場合、各オンライン REDO ログ・グループの少なくとも 1 つのメンバーがメディア障害の影響を受けていなければ、データベースは通常どおりに機能し続けることができます。Oracle は、データベースの LGWR トレース・ファイルおよび alert_SID.log にエラー・メッセージを書き込みます。

次のアクションの 1 つを実行して、問題を解決します。

- ハードウェアの問題が一時的な場合は、問題を解決します。LGWR は、問題が存在しなかったかのように、以前は使用不可能であったオンライン REDO ログ・ファイルにアクセスします。
- ハードウェアの問題が永続的な場合には、次の手順に従って、破損したメンバーを削除し、新しいメンバーを追加します。

注意： 新しく追加されたメンバーは、ログ・グループが再使用されるまでは、冗長性を提供しません。

REDO ログ・グループの破損したメンバーを置き換える手順は、次のとおりです。

1. V\$LOGFILE で破損したメンバーのファイル名を探します。ファイルにアクセスできない場合、ステータスは INVALID になります。

```
SELECT GROUP#, STATUS, MEMBER
FROM V$LOGFILE
WHERE STATUS='INVALID';
```

GROUP#	STATUS	MEMBER
0002	INVALID	/oracle/dbs/log2b.f

2. 破損したメンバーを削除します。たとえば、メンバー log2b.f をグループ 2 から削除するには、次のコマンドを発行します。

```
ALTER DATABASE DROP LOGFILE MEMBER '/oracle/dbs/log2b.f';
```

3. グループに新しいメンバーを追加します。たとえば、log2c.f をグループ 2 に追加するには、次のコマンドを発行します。

```
ALTER DATABASE ADD LOGFILE MEMBER '/oracle/dbs/log2c.f' TO GROUP 2;
```

追加するファイルがすでに存在している場合には、グループの他のメンバーと同じサイズである必要があります。また、REUSE を指定する必要があります。たとえば、次のように入力します。

```
ALTER DATABASE ADD LOGFILE MEMBER '/oracle/dbs/log2b.f' REUSE TO GROUP 2;
```

オンライン REDO ログ・グループのすべてのメンバーが損害を受けた後のリカバリ

オンライン REDO ログ・グループのすべてのメンバーがメディア障害の損害を受けた場合には、障害の影響を受けたオンライン REDO ログ・グループのタイプと、データベースのアーカイブ・モードによって、想定されるシナリオが異なります。

破損したグループが非アクティブの場合には、クラッシュ・リカバリは必要ありません。アクティブな場合には、クラッシュ・リカバリが必要です。

グループの状態	対応	ユーザーが取るべき処置
非アクティブ	クラッシュ・リカバリは必要ありません	アーカイブされたグループまたはアーカイブされていないグループを消去します。
アクティブ	クラッシュ・リカバリが必要です	チェックポイントを発行してログを消去します。できない場合は、バックアップをリストアし、使用可能な最新のログの時点まで、不完全リカバリを実行する必要があります。
現行	Oracle が現在書込みを行っているログです	ログを消去します。消去できない場合は、バックアップをリストアし、使用可能な最新のログの時点まで、不完全リカバリを実行する必要があります。

最初に行うことは、破損したグループがアクティブか非アクティブかを判別することです。

破損したグループがアクティブかどうかを判別する手順は、次のとおりです。

1. V\$LOGFILE で損害を受けた REDO ログのファイル名を探し、これに対応するグループ番号を検索します。たとえば、次のように入力します。

```
SELECT GROUP#, STATUS, MEMBER FROM V$LOGFILE;
```

GROUP#	STATUS	MEMBER
-----	-----	-----
0001		/oracle/dbs/log1a.f
0001		/oracle/dbs/log1b.f
0002	INVALID	/oracle/dbs/log2a.f
0002	INVALID	/oracle/dbs/log2b.f
0003		/oracle/dbs/log3a.f
0003		/oracle/dbs/log3b.f

2. どのグループがアクティブかを判別します。たとえば、次のように入力します。

```
SELECT GROUP#, MEMBERS, STATUS, ARCHIVED FROM V$LOG;
```

GROUP#	MEMBERS	STATUS	ARCHIVED
0001	2	INACTIVE	YES
0002	2	ACTIVE	NO
0003	2	CURRENT	NO

3. 影響を受けたグループが非アクティブの場合には、6-9 ページの「[非アクティブのオンライン REDO ログ・グループの損害](#)」の手順に従ってください。影響を受けたグループがアクティブな場合には（前述の例のように）、6-11 ページの「[アクティブなオンライン REDO ログ・グループの損害](#)」の手順に従ってください。

非アクティブのオンライン REDO ログ・グループの損害

INACTIVE ステータスのオンライン REDO ログ・グループのすべてのメンバーが損害を受けた場合には、非アクティブ REDO ログ・グループに損害を与えたメディアの問題を解決できるかどうかで、手順が異なります。

障害の状態	対応
一時的	問題を解決します。LGWR は、必要な場合に REDO ログ・グループを再使用できます。
永続的	破損した非アクティブのオンライン REDO ログ・グループは最終的に、データベースの通常の稼働を停止させます。この項の説明に従って ALTER DATABASE CLEAR LOGFILE 文を発行し、破損したグループを手動で再初期化します。

データベースをオープンまたはクローズした状態で、アクティブな REDO ログ・グループを消去できます。手順は、破損したグループがアーカイブされているかどうかで異なります。

アーカイブされている、非アクティブのオンライン REDO ログ・グループを消去する手順は、次のとおりです。

1. データベースが停止している場合は、新しいインスタンスを起動してデータベースをマウントします。

```
STARTUP MOUNT
```

2. 破損したログ・グループを再初期化します。たとえば、REDO ログ・グループ 2 を消去するには、次の文を発行します。

```
ALTER DATABASE CLEAR LOGFILE GROUP 2;
```

アーカイブされていない、非アクティブのオンライン REDO ログ・グループを消去する手順は、次のとおりです。

アーカイブされていないログを消去することにより、アーカイブせずに、ログを再使用できます。ログの中の最初の変更の前にファイルがオフライン化されておらず、ログの中の最後の変更前にバックアップが開始されている場合には、このアクションによりバックアップは使用できなくなります。このため、バックアップのリカバリのために消去されたログ・ファイルが必要となった場合は、このバックアップはリカバリできません。また、ログの欠落により、バックアップからの完全リカバリもできません。

1. データベースが停止している場合は、新しいインスタンスを起動してデータベースをマウントします。

```
STARTUP MOUNT
```

2. UNARCHIVED キーワードを使用してログを消去します。たとえば、ログ・グループ 2 を消去するには、次のコマンドを発行します。

```
ALTER DATABASE CLEAR LOGFILE UNARCHIVED GROUP 2;
```

オンライン化するために、消去された、アーカイブされていないログを必要とするオフライン・データ・ファイルがある場合には、UNRECOVERABLE DATAFILE キーワードが必要です。オンライン化するために必要な REDO が消去され、コピーもないため、そのデータ・ファイルと、その表領域全体は削除する必要があります。たとえば、次のように入力します。

```
ALTER DATABASE CLEAR LOGFILE UNARCHIVED GROUP 2 UNRECOVERABLE DATAFILE;
```

3. 2-4 ページの「[データベース全体のユーザー管理バックアップの作成](#)」の説明に従って、オペレーティング・システム・ユーティリティを使用して即時にデータベースをバックアップします。これで、消去されたログ・グループに依存せずに、このバックアップを使用して完全リカバリを実行できます。たとえば、次のように入力します。

```
% cp /disk1/oracle/dbs/*.f /disk2/backup
```

4. 2-20 ページの「[制御ファイルのバイナリ・ファイルへのバックアップ](#)」の説明に従って、ALTER DATABASE 文を使用して、データベースの制御ファイルをバックアップします。たとえば、次のように入力します。

```
ALTER DATABASE BACKUP CONTROLFILE TO '/oracle/dbs/cf_backup.f';
```

CLEAR LOGFILE 操作の失敗 次のことができない場合には、メディア障害による I/O エラーにより ALTER DATABASE CLEAR LOGFILE 文が失敗することがあります。

- 現在構成されている REDO ログのファイル名を元に再作成することにより、REDO ログ・ファイルを別のメディアに再配置する
- 名前自体が無効または使用不可能なため（たとえばメディア障害により）、現在構成されているログ・ファイル名を再使用して REDO ログ・ファイルを再作成する

これらの場合には、ALTER DATABASE CLEAR LOGFILE 文は（I/O エラーを受け取る前に）、ログが消去され、アーカイブが必要ななかったことを、制御ファイルに正しく通知します。CLEAR LOGFILE 文が新しい REDO ログ・ファイルを作成して 0（ゼロ）を書き込もうとした段階で I/O エラーが発生します。この事実は V\$LOG.CLEARING_CURRENT に反映されません。

アクティブなオンライン REDO ログ・グループの損害

データベースがまだ稼働中であり、損害を受けたアクティブなログが現行ログでない場合には、ALTER SYSTEM CHECKPOINT 文を発行します。成功すると、アクティブ・ログがアクティブでなくなるため、6-9 ページの「[非アクティブのオンライン REDO ログ・グループの損害](#)」の手順を実行できます。成功しなかったか、データベースが停止した場合には、アーカイブ・モードに従って、この項の手順の 1 つを実行します。

現行ログとは、LGWR によって現在書込みが行われているログです。LGWR の I/O が失敗すると、LGWR は終了し、インスタンスがクラッシュします。この場合には、バックアップをリストアし、不完全リカバリを実行し、RESETLOGS オプションを指定してデータベースをオープンする必要があります。

NOARCHIVELOG モードの場合の、アクティブなオンライン REDO ログ・グループの損害のリカバリ手順は、次のとおりです。

1. メディア障害が一時的な場合には、問題を解決して、必要なときに Oracle がグループを再使用できるようにします。
2. 3-7 ページの「[データ・ファイルのリストア](#)」の説明に従って、データベース全体の一貫性バックアップ（データ・ファイルと制御ファイル）からデータベースをリストアします。たとえば、次のように入力します。

```
% cp /disk2/backup/*.f /disk1/oracle/dbs
```

3. データベースをマウントします。

```
STARTUP MOUNT
```

4. オンライン REDO ログはバックアップされないため、データ・ファイルおよび制御ファイルとともにリストアすることはできません。Oracle がオンライン REDO ログをリセットできるように、いったん不完全リカバリを行う必要があります。

```
RECOVER DATABASE UNTIL CANCEL  
CANCEL
```

5. RESETLOGS オプションを使用してデータベースをオープンします。

```
ALTER DATABASE OPEN RESETLOGS;
```

6. 一貫性をもってデータベースを停止します。たとえば、次のように入力します。

```
SHUTDOWN IMMEDIATE
```

7. 2-4 ページの「データベース全体のユーザー管理バックアップの作成」の説明に従って、データベース全体のバックアップを作成します。たとえば、次のように入力します。

```
% cp /disk1/oracle/dbs/*.f /disk2/backup
```

ARCHIVELOG モードの場合の、アクティブなオンライン REDO ログ・グループの損害のリカバリ手順は、次のとおりです。

メディア障害が一時的な場合には、問題を解決して、必要なときに Oracle がグループを再使用できるようにします。メディア障害が一時的なものでない場合には、次の手順を使用します。

1. 不完全メディア・リカバリを開始します。4-17 ページの「ユーザー管理の不完全メディア・リカバリの実行」の手順を使用して、破損したログの前のログまでリカバリします。
2. 損害を受けた REDO ログの現在の名前を、新しく作成したファイルに使用できることを確認します。使用できない場合は、破損したオンライン REDO ログ・グループのメンバーの名前を新しい場所に変更します。たとえば、次のように入力します。

```
ALTER DATABASE RENAME FILE "/oracle/dbs/log_1.rdo" TO "/temp/log_1.rdo";  
ALTER DATABASE RENAME FILE "/oracle/dbs/log_2.rdo" TO "/temp/log_2.rdo";
```

3. RESETLOGS オプションを使用してデータベースをオープンします。

```
ALTER DATABASE OPEN RESETLOGS;
```

注意： 不完全リカバリのエンドポイントから現在までの間に実行されたすべての更新を、再度実行する必要があります。

複数の REDO ログ・グループの損害

オンライン REDO ログの複数のグループが損害を受けた場合には、リカバリの最も困難なログのリカバリ方法を使用します。リカバリが困難な順に示します。

1. 現行のオンライン REDO ログ
2. アクティブなオンライン REDO ログ
3. アーカイブされていないオンライン REDO ログ
4. 非アクティブのオンライン REDO ログ

アーカイブ REDO ログ・ファイルが損害を受けた後のリカバリ：使用例

データベースが ARCHIVELOG モードで稼動しているときに、アーカイブ REDO ログ・ファイルの唯一のコピーが破損した場合には、破損したファイルが、データベースの現在の稼働に影響を与えることはありません。しかし、REDO ログの書き込まれたタイミングおよびデータ・ファイルのバックアップが作成されたタイミングによっては、次の状況が発生することがあります。

バックアップ対象	対応
満杯になったオンライン REDO ログ・グループ（現在アーカイブされている）が書き込まれた後のすべてのデータ・ファイル	満杯になったオンライン REDO ログ・グループのアーカイブされたバージョンは、完全メディア・リカバリ操作に必要ありません。
満杯になったオンライン REDO ログ・グループが書き込まれる前の特定のデータ・ファイル	対応するデータ・ファイルが永続的なメディア障害により破損した場合には、破損したデータ・ファイルの最新のバックアップを使用して、破損したログに至るまで、不完全リカバリを実行します。

注意： アーカイブ REDO ログ・グループが破損したことがわかっている場合には、破損したアーカイブ REDO ログを必要としない、データベース全体のバックアップを作成するために、即時にすべてのデータ・ファイルをバックアップします。

ユーザー・エラーからのリカバリ : 使用例

データベースに対する誤った操作またはプログラムによる変更により、データの損害や破損が発生することがあります。リカバリでは、エラー前の状態に戻すことが必要になります。

注意： 強力な権限（DROP ANY TABLE など）を特定の、適切なユーザーのみに付与することにより、データベース・リカバリを必要とするユーザー・エラーを最小化できます。

誤って削除された表のリカバリの手順は、次のとおりです。

1. 可能であれば、ユーザー・エラーが発生したデータベースをオンラインのまま保ち、使用できるようにします。この手順の後の手順でエラーが発生した場合に備えて、既存のデータベースのすべてのデータ・ファイルをバックアップします。
2. 代替の場所にデータベースのバックアップをリストアし、リストアされたバックアップ制御ファイルを使用して、表が削除された直前まで、このバックアップの不完全リカバリを実行します（4-17 ページの「[ユーザー管理の不完全メディア・リカバリの実行](#)」の説明を参照）。
3. Oracle のエクスポート・ユーティリティを使用して、データベースの一時的にリストアされたバージョンから、損害を受けたデータをエクスポートします。この場合、誤って削除された表がエクスポートされます。

注意： システム監査オプションがエクスポートされます。

4. インポート・ユーティリティを使用して、本番データベースにデータをインポートして戻します。
5. スペースの節約のためにデータベースの一時コピーのファイルを削除します。

関連項目： インポートおよびエクスポート・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。

分散環境でのメディア・リカバリの実行：使用例

メディア・リカバリの実行方法は、データベースが分散データベース・システムの一部であるかどうかにより異なります。Oracle 分散データベース・アーキテクチャは自律型です。このため、1つの、破損したデータベースに対して選択されたリカバリ操作のタイプによっては、分散システム中のすべてのデータベースでグローバルにリカバリ操作を調整することが必要な場合があります。

表 6-2 は、リカバリ操作の様々なタイプと、分散データベース・システムのノード間の調整が必要かどうかを要約しています。

表 6-2 分散データベース環境でのリカバリ操作

状況 ..	対応 ..
リモート・ノードからアクセスされたことのないデータベースの全体バックアップをリストアする場合	無調整式の自律型データベース・リカバリを使用します。
NOARCHIVELOG モードのデータベースのリモート・ノードからアクセスされたことのあるデータベースの全体バックアップをリストアする場合	すべてのデータベースを停止し、同じ調整式全体バックアップを使用してリストアします。
分散データベース中の 1 つ以上のデータベースの完全メディア・リカバリを実行する場合	無調整式の自律型データベース・リカバリを使用します。
リモート・ノードからアクセスされたことのないデータベースの不完全メディア・リカバリを実行する場合	無調整式の自律型データベース・リカバリを使用します。
リモート・ノードからアクセスされたことのあるデータベースの不完全メディア・リカバリを実行する場合	調整式の不完全リカバリを使用して、分散システム中のすべてのデータベースでグローバルに同じ時点までリカバリします。

時間ベースおよび変更ベースの分散データベースのリカバリの調整

特別な状況では、分散データベース中の 1 つのノードで、過去の時点までリカバリすることが必要な場合があります。グローバルなデータ整合性を保つには、システム中の他のすべてのノードを同じ時点までリカバリすることが必要な場合があります。この操作は、**分散データベースの時間ベースの調整式リカバリ**と呼ばれます。次のタスクを、この章で説明した、時間ベースおよび変更ベースのリカバリの標準的な手順で実行する必要があります。

1. 4-22 ページの「**時間ベースの不完全リカバリの実行**」の説明に従って、時間ベースのリカバリを使用して、リカバリ操作に必要なデータベースのリカバリを実行します。たとえば、ユーザー・エラー（誤った表の削除など）のためにデータベースのリカバリが必要な場合には、最初に時間ベースのリカバリを使用してこのデータベースをリカバリします。この時点では他のデータベースのリカバリは行いません。

2. データベースをリカバリし、RESETLOGS オプションを指定してオープンした後で、データベースの `alert_SID.log` で RESETLOGS メッセージを探します。

メッセージが「RESETLOGS after complete recovery through change xxx」の場合には、データベースのすべての変更が適用され、完全リカバリが実行されています。分散システム内の他のデータベースのリカバリは実行しないでください。実行すると、変更が不必要に削除されます。リカバリは完了しています。

メッセージが「RESETLOGS after incomplete recovery UNTIL CHANGE xxx」の場合には、不完全リカバリが正常に実行されています。メッセージの変更番号を記録し、次の手順に進みます。

3. 手順 2 の変更番号 (SCN) を指定し、変更ベースのリカバリを使用して、分散データベース・システム中の他のすべてのデータベースをリカバリします。

ユーザー管理の TSPITR の実行

この章では、トランスポータブル表領域の機能を使用して、ユーザー管理の表領域の Point-in-Time リカバリ（TSPITR）を実行する方法について説明します。

この章で説明する項目は、次のとおりです。

- ユーザー管理の表領域の Point-in-Time リカバリ（TSPITR）の概要
- 表領域の Point-in-Time リカバリの準備：基本手順
- 補助データベースのリストアおよびリカバリ：基本手順
- トランスポータブル表領域を使用した TSPITR の実行
- パーティション表の部分 TSPITR の実行
- パーティションが削除された場合のパーティション表の部分 TSPITR の実行
- パーティションが分割されている場合のパーティション表の TSPITR の実行

ユーザー管理の表領域の Point-in-Time リカバリ (TSPITR) の概要

トランSPORTABLE表領域機能で表領域の Point-in-Time リカバリ (TSPITR) を使用すると、1 つ以上の表領域 (SYSTEM 以外の表領域) を、データベースの残りの部分よりも前の時点まで迅速にリカバリできます。

ユーザー管理の TSPITR は、次の場合のリカバリに特に有用です。

- 誤った DROP TABLE または TRUNCATE TABLE 操作
- DROP TABLESPACE 操作のエラー
- 論理的に破損した表
- データベースの一部に対してのみ影響する、不正なバッチ・ジョブまたはその他の DML 文の実行
- 1 つの物理データベースの別個の表領域に複数のスキーマが存在するときに、ある論理スキーマの物理データベースの残りの部分とは異なる時点までのリカバリ
- バックアップからデータベース全体をリストアしてロールフォワードするよりも TSPITR が効率的な場合の、VLDB (非常に大規模なデータベース) の表領域 (判断する前に 7-4 ページの「[表領域の Point-in-Time リカバリの準備: 基本手順](#)」を参照)

TSPITR の用語

この章で使用される、次の用語と略称を覚えておいてください。

TSPITR (Tablespace point-in-time recovery)

表領域の Point-in-Time リカバリ。

プライマリ・データベース

過去の時点までリカバリしようとしている表領域を含むデータベース。

補助データベース

バックアップからリストアされる現在のデータベースのコピー。次のもののリストアされたバックアップが含まれます。

- SYSTEM 表領域に属するデータ・ファイル
- リカバリ対象の表領域のセット中のデータ・ファイル
- システム管理の UNDO 表領域に属するデータ・ファイル (自動 UNDO 管理モードでデータベースを実行するとき)、またはロールバック・セグメントを含む表領域 (手動 UNDO 管理モードでデータベースを実行するとき)

すべてのバックアップは、リカバリの目標時刻よりも前のものである必要があります。

リカバリ・セット

Point-in-Time リカバリを実行する必要があるすべての表領域。

リカバリ・セットの自己完結チェック

リカバリ・セットの一部であるすべてのオブジェクトは、自己完結型であることが必要です。リカバリ・セットの外部のオブジェクトへの依存性がないようにしてください。たとえば、リカバリ・セットの一部に表が含まれ、その索引が別個の表領域内にある場合には、索引を含む表領域がリカバリ・セット内に含まれている必要があります。あるいは索引を削除することもできます。リカバリ・セットの表領域が自己完結型かどうかは、DBMS_TTS.TRANSPORT_SET_CHECK プロシージャを使用してチェックできます。

補助セット

次のものを含む、補助データベースのリストアに必要なその他のアイテム。

- バックアップ制御ファイル
- SYSTEM 表領域のデータ・ファイル
- UNDO 表領域内のデータ・ファイルまたはロールバック・セグメントを含むデータ・ファイル

トランスポートابل表領域

ソース・データベースから取り出し、ターゲット・データベースに組み込むことにより、データベース間で表領域を迅速に移動するための方法。取出しおよび組込みはエクスポートおよびインポート・ユーティリティにより行われます。表データが実際にエクスポートおよびインポートされるわけではなく、内部メタデータがエクスポートおよびインポートされるのみであることに注意してください。この手順中に、移動された表領域のデータ・ファイルは、ターゲット・データベースの一部となります。

TSPITR の方法

Oracle9i よりも前のリリースでは、ユーザー管理の TSPITR を実行するために次の 2 つの方法を使用できました。

- クローン・データベースと呼ばれる、特殊なタイプのデータベースを作成する必要があった、伝統的なユーザー管理の TSPITR
- トランスポートابل表領域機能を使用したユーザー管理の TSPITR

Oracle9i の TSPITR は、トランスポートابل表領域機能を使用して実行する必要があります。この方法は、現在は使用を勧奨していない（サポート停止ではない）伝統的な方法に比べて、簡単に使用でき、エラーの可能性も低くなっています。

TSPITR は概念的には、リカバリ対象の表領域をプライマリ・データベースから削除し、**補助データベース**と呼ばれるデータベースのコピーをリストアし、このコピーを目的の時点までリカバリし、関連する表領域を補助データベースからプライマリ・データベースの現行バージョンに移送することにより、実行されます。

使用しやすいように、プライマリ・データベースと補助データベースは別のホスト上に配置することをお薦めします。これらのデータベースが同じホスト上に配置されている場合も TSPITR を実行できます。

ユーザー管理の TSPITR を実行する基本手順は、次のとおりです。

1. TSPITR を必要とする表領域をオフライン化します。
2. 補助データベースの設定を計画します。
3. 補助データベースを作成し、目的の時点までリカバリします。
4. TSPITR を必要とする表領域をプライマリ・データベースから削除します。
5. トランスポートابل表領域機能を使用して、表領域のセットを、補助データベースからプライマリ・データベースに移送します。

関連項目： トランスポートابل表領域機能の使用方法的詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

表領域の Point-in-Time リカバリの準備 : 基本手順

TSPITR では慎重な計画が必要です。先に進む前に、この章全体を確認してください。

この項で説明する項目は、次のとおりです。

- [手順 1: TSPITR の要件の検討](#)
- [手順 2: リカバリ・セットおよび補助セットの表領域内のすべてのファイルの識別](#)
- [手順 3: 失われるオブジェクトの判別](#)
- [手順 4: 補助インスタンスへの接続方法の選択](#)
- [手順 5: 補助インスタンスの Oracle パスワード・ファイルの作成](#)
- [手順 6: 補助インスタンスの初期化パラメータ・ファイルの作成](#)

注意： 最初から本番システムで TSPITR を実行しないでください。また、時間の制約があるときにも実行しないでください。

手順 1: TSPITR の要件の検討

TSPITR を実行する前に、次の要件を満たす必要があります。

- リカバリ・セットおよび補助セットの表領域のすべてのデータ・ファイルのバックアップが作成されていることを確認します。データ・ファイルのバックアップは、TSPITR の目標時刻の前に作成されている必要があります。
- 使用可能な制御ファイルのバックアップが補助データベースにあることを確認します。制御ファイルが使用可能であるためには、次の要件が満たされている必要があります。
 - 制御ファイルのバックアップは、TSPITR の目標時刻の前に作成されている必要があります。
 - 制御ファイルは、次の SQL 文を使用してバックアップされている必要があります。この `cf_name` は、完全指定のファイル名です。

```
ALTER DATABASE BACKUP CONTROLFILE TO 'cf_name';
```

- リカバリ・セットの表領域を構成するすべてのファイルが、補助データベース上のリカバリ・セットに含まれていることを確認します。含まれていない場合、表領域の移動中にエクスポート・フェーズが失敗になります。
- 補助データベースを格納するだけの十分なディスク領域を補助ホストに割り当てます。
- 補助インスタンスを起動するための十分な実メモリを提供します。

関連項目： 7-7 ページ「[手順 6: 補助インスタンスの初期化パラメータ・ファイルの作成](#)」

手順 2: リカバリ・セットおよび補助セットの表領域内のすべてのファイルの識別

補助データベースを作成する前に、管理者権限を使用してプライマリ・データベースに接続し、プライマリ・データベースに関して次のすべての情報を入手していることを確認します。

- リカバリ・セットの表領域内のデータ・ファイルのファイル名
- SYSTEM 表領域内のデータ・ファイルのファイル名
- UNDO 表領域内のデータ・ファイルのファイル名、またはロールバック・セグメントを含むデータ・ファイルのファイル名
- 制御ファイルのファイル名

次の便利な問合せは、データベース内のすべてのデータ・ファイル、制御ファイルおよびオンライン REDO ログのファイル名を表示します。

```
SELECT NAME FROM V$DATAFILE  
UNION ALL
```

```
SELECT MEMBER FROM V$LOGFILE
UNION ALL
SELECT NAME FROM V$CONTROLFILE;
```

SYSTEM およびリカバリ・セットの表領域内のデータ・ファイルのファイル名を判別するために、次の問合せを実行し、RECO_TBS_1、RECO_TBS_2などを、リカバリ・セットの表領域の名前で置き換えます。

```
SELECT t.NAME AS "reco_tbs", d.NAME AS "dbf_name"
FROM V$DATAFILE d, V$TABLESPACE t
WHERE t.TS# = d.TS#
AND t.NAME IN ('SYSTEM', 'RECO_TBS_1', 'RECO_TBS_2');
```

手動 UNDO 管理モードでデータベースを実行している場合、次の問合せは、ロールバック・セグメントを含む表領域の名前と表領域内のデータ・ファイルの名前を表示します。

```
SELECT r.TABLESPACE_NAME AS "rbs_tbs", d.FILE_NAME AS "dbf_name"
FROM DBA_ROLLBACK_SEGS r, DBA_DATA_FILES d
WHERE r.TABLESPACE_NAME=d.TABLESPACE_NAME;
```

自動 UNDO 管理モードでデータベースを実行している場合、次の問合せは、UNDO 表領域の名前と表領域内のデータ・ファイルの名前を表示します。

```
SELECT u.TABLESPACE_NAME AS "undo_tbs", d.FILE_NAME AS "dbf_name"
FROM DBA_UNDO_EXTENTS u, DBA_DATA_FILES d
WHERE u.TABLESPACE_NAME=d.TABLESPACE_NAME;
```

手順 3: 失われるオブジェクトの判別

表領域に対して TSPITR を実行すると、リカバリ時刻の後に作成されたすべてのオブジェクトは失われます。失われるオブジェクトを判別するために、プライマリ・データベースで TS_PITR_OBJECTS_TO_BE_DROPPED ビューを問い合わせます。このビューの内容を表 7-1 に示します。

表 7-1 TS_PITR_OBJECTS_TO_BE_DROPPED ビュー

列名	意味
OWNER	削除されるオブジェクトの所有者
NAME	TSPITR の結果として失われるオブジェクトの名前
CREATION_TIME	オブジェクトの作成タイム・スタンプ
TABLESPACE_NAME	オブジェクトを含む表領域の名前

このビューを問い合わせるときには、日付フィールドのすべての要素を指定しないかぎり、デフォルト設定が使用されます。また、TO_CHAR および TO_DATE 関数を使用します。たとえば、sales_1 および sales_2 から構成されるリカバリ・セットを使用し、リカバリ

Point-in-Time が '2000-06-02:07:03:11' のときには、次の SQL スクリプトを使用します。

```
SELECT OWNER, NAME, TABLESPACE_NAME, TO_CHAR(CREATION_TIME, 'YYYY-MM-DD:HH24:MI:SS')
FROM SYS.TS_PITR_OBJECTS_TO_BE_DROPPED
WHERE TABLESPACE_NAME IN ('SALES_1', 'SALES_2')
AND CREATION_TIME > TO_DATE('00-JUN-02:07:03:11', 'YY-MON-DD:HH24:MI:SS')
ORDER BY TABLESPACE_NAME, CREATION_TIME;
```

関連項目： TS_PITR_OBJECTS_TO_BE_DROPPED ビューの詳細は、『Oracle9i データベース・リファレンス』を参照してください。

手順 4: 補助インスタンスへの接続方法の選択

補助インスタンスに接続できる必要があります。Oracle Net またはオペレーティング・システムの認証を使用できます。ネットワーク・ファイルの構成方法の詳細は、『Oracle9i Net Services 管理者ガイド』を参照してください。

手順 5: 補助インスタンスの Oracle パスワード・ファイルの作成

Oracle パスワード・ファイルの作成およびメンテナンスの詳細は、『Oracle9i データベース管理者ガイド』を参照してください。パスワード・ファイルを使用しない場合は、この手順をスキップできます。

手順 6: 補助インスタンスの初期化パラメータ・ファイルの作成

本番データベースの初期化パラメータ・ファイルをコピーして編集するのではなく、新しい初期化パラメータ・ファイルを作成します。次のようなパラメータには低い設定値を使用して、メモリーを節約します。

- DB_CACHE_SIZE
- SHARED_POOL_SIZE
- LARGE_POOL_SIZE

これらのパラメータの値を低く設定することにより、他の依存パラメータが高く設定されすぎた場合に（たとえば、共有プールからのメモリーを割り当てる初期化パラメータ ENQUEUE_RESOURCES）、補助データベースが起動されないようにできます。

補助データベースは、プライマリ・データベースと同じホスト上に配置しても別のホスト上に配置してもかまいません。補助制御ファイルの中の補助データベースのファイル名はプライマリ・データベースのファイル名と同じであるため、リストア場所をポイントするように、この制御ファイルの中でファイル名を変更する必要があります。補助データベースがプライマリ・データベースと同じマシン上にある場合、または補助データベースが別のパス名を使用する別のマシン上にある場合には、制御ファイル、データ・ファイルおよびオンライン REDO ログを改名する必要があります。補助データベースが同じパス名を使用する別のマシン上にある場合には、オンライン REDO ログを改名するだけで済みます。

注意： 補助データベースとプライマリ・データベースが同じマシン上にある場合には、オンライン REDO ログ・ファイルを改名しないと、プライマリ・データベースが破損することがあります。

補助の初期化パラメータ・ファイルで、表 7-2 に示すパラメータを設定します。

表 7-2 補助の初期化パラメータ

パラメータ	用途
DB_NAME	補助データベースを命名します。補助データベースの名前はプライマリ・データベースと同じにします。
CONTROL_FILES	補助制御ファイルを識別します。補助制御ファイルのファイル名に設定されます。この制御ファイル名は必ず、プライマリ・データベースの制御ファイル名と異なる名前にしてください。
LOCK_NAME_SPACE	プライマリ・データベースと同じ名前であっても、補助データベースを起動できるようにします。たとえば=AUX など、一意の値に設定します。このパラメータは、補助データベースとプライマリ・データベースが同じホスト上にある場合にのみ必要です。
DB_FILE_NAME_CONVERT	補助データベースのデータ・ファイルのファイル名を変換するためにパターンを使用します。このパラメータは、補助データベースをプライマリ・データベースと同じホスト上にリストアするか、補助データベースをプライマリ・ホストとは別のパス名を使用する別のホスト上にリストアする場合にのみ必要です。
LOG_FILE_NAME_CONVERT	補助データベースのオンライン REDO ログのファイル名を変換するためにパターンを使用します。このパラメータは必須です。
LOG_ARCHIVE_DEST_1	リカバリに必要なアーカイブ REDO ログを含むデフォルト・ディレクトリを指定します。このパラメータは、アーカイブ・ログが置かれる補助ホスト上の場所を指定します。
LOG_ARCHIVE_FORMAT	アーカイブ・ログの書式を指定します。プライマリの初期化パラメータ・ファイルで使用したのと同じ書式設定を使用する必要があります。

Oracle Net を通じて SYSDBA として接続できるようにするパラメータも含めて、その他のパラメータを必要に応じて設定します。

たとえば、プライマリ・データベースと同じホスト上のデータベースの補助パラメータ・ファイルは、次のようになります。

```
DB_NAME = prod1
CONTROL_FILES = /oracle/aux/cf1.f
LOCK_NAME_SPACE = aux
DB_FILE_NAME_CONVERT=("/oracle/dbs/", "/oracle/aux/")
LOG_FILE_NAME_CONVERT=("/oracle/dbs/", "/oracle/aux/")
LOG_ARCHIVE_DEST_1 = 'LOCATION=/oracle/work/arc_dest/arc'
LOG_ARCHIVE_FORMAT = r_%t_%s.arc
```

たとえば、プライマリ・データベースと同じパス名を使用する別のホスト上のデータベースの補助パラメータ・ファイルは、次のようになります。

```
DB_NAME = prod1
CONTROL_FILES = /oracle/aux/cf1.f
LOG_FILE_NAME_CONVERT=("/oracle/dbs/", "/oracle/aux/")
LOG_ARCHIVE_DEST_1 = 'LOCATION=/oracle/work/arc_dest/arc'
LOG_ARCHIVE_FORMAT = r_%t_%s.arc
```

補助データベースのリストアおよびリカバリ：基本手順

補助データベースのリストアおよびリカバリの手順は、補助データベースがプライマリ・データベースと同じホスト上にあるかどうかで異なります。この項の例では、次のように想定しています。

- ホスト prim_host 上の prod1 という名前の本番データベースで TSPITR を実行します。
- リカバリ・セットの表領域は sales_1 および sales_2 です。表領域 sales_1 はデータ・ファイル /oracle/dbs/sales_1.f を含み、表領域 sales_2 はデータ・ファイル /fs2/sales_2.f を含みます。
- 補助セットは、SYSTEM 表領域のデータ・ファイル /oracle/dbs/system.f、UNDO 表領域のデータ・ファイル /oracle/dbs/undo.f、および制御ファイル /oracle/dbs/cf1.f を含みます。
- オンライン REDO ログの名前は /oracle/dbs/log1.f および /oracle/dbs/log2.f です。
- プライマリ・データベースのすべてのファイルは /oracle/dbs に入っています。

この後の項で、様々なケースについて説明します。

- [同じホスト上での補助データベースのリストアおよびリカバリ](#)
- [同じパス名を使用する別のホスト上での補助データベースのリストア](#)
- [異なるパス名を使用する別のホスト上での補助データベースのリストア](#)

同じホスト上での補助データベースのリストアおよびリカバリ

次の例では、プライマリ・データベースと同じホスト上に補助データベースをリストアするケースを想定しています。この例では、プライマリ・データベースのすべてのファイルは /oracle/dbs に入っています。補助データベースは /oracle/dbs/aux にリストアします。このため、DB_FILE_NAME_CONVERT および LOG_FILE_NAME_CONVERT を設定し、ファイル名を /oracle/dbs から /oracle/dbs/aux に変換します。

補助データベースのリストアおよびリカバリのために次のタスクを実行します。

1. 補助セットおよびリカバリ・セットを、プライマリ・データベースの場所とは別の場所にリストアします。たとえば、補助セットが次のファイルから構成されているとします。

```
/oracle/dbs/cf1.f      # control file
/oracle/dbs/undo.f     # datafile in undo tablespace
/oracle/dbs/system.f   # datafile in SYSTEM tablespace
```

また、リカバリ・セットは次のデータ・ファイルから構成されているとします。

```
/oracle/dbs/sales_1.f # datafile in sales_1 tablespace
/oracle/dbs/sales_2.f # datafile in sales_2 tablespace
```

次のようにして、補助セットのファイルおよびリカバリ・セットのファイルのバックアップを新しい場所にリストアできます。

```
cp /backup/cf1.f /aux/cf1.f
cp /backup/undo.f /aux/undo.f
cp /backup/system.f /aux/system.f
cp /backup/sales_1.f /aux/sales_1.f
cp /backup/sales_2.f /aux/sales_2.f
```

2. 補助データベースをマウントせずに起動します。必要であれば初期化パラメータ・ファイルを指定します。たとえば、次のように入力します。

```
STARTUP NOMOUNT PFILE=/aux/initAUX.ora
```

3. CLONE キーワードを指定して補助データベースをマウントします。

```
ALTER DATABASE MOUNT CLONE DATABASE;
```

CLONE キーワードが指定されたため、Oracle は自動的にすべてのデータ・ファイルをオフライン化します。

4. DB_FILE_NAME_CONVERT および LOG_FILE_NAME_CONVERT により改名されていない場合にのみ、新しい場所を反映するように、補助データベースのすべてのファイルを手動で改名します。このシナリオでは、すべてのデータ・ファイルおよびオンライン REDO ログは初期化パラメータにより改名されるため、手動の改名は必要ありません。

5. 次の SQL スクリプトを実行して、すべてのデータ・ファイルが正しく命名されていることを確認します。

```
SELECT NAME FROM V$DATAFILE
UNION ALL
SELECT MEMBER FROM V$LOGFILE
UNION ALL
SELECT NAME FROM V$CONTROLFILE
/
```

正しく命名されていない場合には、前の手順に従って、ファイルを手動で改名します。

6. 補助セットおよびリカバリ・セットの表領域内のデータ・ファイルのみをオンライン化します。たとえば、リカバリ・セットおよび補助セット内の 4 つのデータ・ファイルをオンライン化します。

```
ALTER DATABASE DATAFILE /oracle/dbs/aux/system.f ONLINE;
ALTER DATABASE DATAFILE /oracle/dbs/aux/sales_1.f ONLINE;
ALTER DATABASE DATAFILE /oracle/dbs/aux/sales_2.f ONLINE;
ALTER DATABASE DATAFILE /oracle/dbs/aux/undo.f ONLINE;
```

注意： リカバリ・セットの各表領域のすべてのファイルがオンライン化されていないかぎり、TSPITR のエクスポート・フェーズは機能しません。

この時点で、補助データベースがマウントされ、メディア・リカバリの準備が整います。

7. USING BACKUP CONTROLFILE オプションを使用して、補助データベースを指定された時点までリカバリします。4-17 ページの「[ユーザー管理の不完全メディア・リカバリの実行](#)」の説明に従って、いずれかの形式の不完全リカバリを使用します。次の例は、取消ベースの不完全リカバリを使用しています。

```
RECOVER DATABASE UNTIL CANCEL USING BACKUP CONTROLFILE
```

8. 次の文を使用して、RESETLOGS オプションを指定して補助データベースをオープンします。

```
ALTER DATABASE OPEN RESETLOGS;
```

同じパス名を使用する別のホスト上での補助データベースのリストア

次の例では、aux_host という名前の別のホスト上に補助データベースを作成することになります。補助ホストは、プライマリ・ホストと同じパス名を使用しています。このため、補助データベースのデータ・ファイルを改名する必要はありません。このため、DB_FILE_NAME_CONVERT を設定する必要はありませんが、LOG_FILE_NAME_CONVERT は設定する必要があります。

補助データベースのリストアおよびリカバリの手順は、次のとおりです。

1. 補助セットおよびリカバリ・セットを補助ホスト上にリストアします。たとえば、補助セットが次のファイルから構成されているとします。

```
/oracle/dbs/cf1.f      # control file
/oracle/dbs/undo.f     # datafile in undo tablespace
/oracle/dbs/system.f   # datafile in SYSTEM tablespace
```

また、リカバリ・セットは次のデータ・ファイルから構成されているとします。

```
/oracle/dbs/sales_1.f  # 1st datafile in sales_1 tablespace
/oracle/dbs/sales_2.f  # 2nd datafile in sales_2 tablespace
```

これらのファイルは、補助ホスト上で同じ場所を占有します。

2. 補助データベースをマウントせずに起動します。必要であれば初期化パラメータ・ファイルを指定します。たとえば、次のように入力します。

```
STARTUP NOMOUNT PFILE=/aux/initAUX.ora
```

3. CLONE キーワードを指定して補助データベースをマウントします。

```
ALTER DATABASE MOUNT CLONE DATABASE;
```

CLONE キーワードが指定されたため、Oracle は自動的にすべてのデータ・ファイルをオフライン化します。

4. DB_FILE_NAME_CONVERT および LOG_FILE_NAME_CONVERT により改名されていない場合にのみ、新しい場所を反映するように、補助データベースのすべてのファイルを改名します。このシナリオでは、データ・ファイルの改名は必要ありません。ログは LOG_FILE_NAME_CONVERT により改名されます。このため、手動の改名は必要ありません。
5. SQL*Plus で次のスクリプトを実行して、すべてのデータ・ファイルが正しく命名されていることを確認します。

```
SELECT NAME FROM V$DATAFILE
UNION ALL
SELECT MEMBER FROM V$LOGFILE
UNION ALL
SELECT NAME FROM V$CONTROLFILE
/
```

正しく命名されていない場合には、前の手順に従って手動で改名します。

6. 補助セットおよびリカバリ・セットの表領域内のすべてのデータ・ファイルをオンライン化します。たとえば、リカバリ・セットおよび補助セット内の 4 つのデータ・ファイルをオンライン化します。

```
ALTER DATABASE DATAFILE /oracle/dbs/system.f ONLINE;  
ALTER DATABASE DATAFILE /oracle/dbs/sales_1.f ONLINE;  
ALTER DATABASE DATAFILE /oracle/dbs/sales_2.f ONLINE;  
ALTER DATABASE DATAFILE /oracle/dbs/undo.f ONLINE;
```

注意： リカバリ・セットの各表領域のすべてのファイルがオンライン化されていないかぎり、TSPITR のエクスポート・フェーズは機能しません。

この時点で、補助データベースがマウントされ、メディア・リカバリの準備が整います。

7. USING BACKUP CONTROLFILE オプションを使用して、補助データベースを指定された時点までリカバリします。4-17 ページの「[ユーザー管理の不完全メディア・リカバリの実行](#)」の説明に従って、いずれかの形式の不完全リカバリを使用します。次の例は、取消ベースの不完全リカバリを使用しています。

```
RECOVER DATABASE UNTIL CANCEL USING BACKUP CONTROLFILE
```

8. 次の文を使用して、RESETLOGS オプションを指定して補助データベースをオープンします。

```
ALTER DATABASE OPEN RESETLOGS;
```

異なるパス名を使用する別のホスト上での補助データベースのリストア

このケースは、7-10 ページの「[同じホスト上での補助データベースのリストアおよびリカバリ](#)」と同じように処理する必要があります。いずれの場合にも、ファイルの改名について同じガイドラインが適用されます。

トランスポータブル表領域を使用した TSPITR の実行

準備段階が終了した後、『Oracle9i データベース管理者ガイド』の説明に従って、実際の TSPITR の手順を開始します。この手順は次のように行われます。

- 手順 1: 補助データベースからの表領域の取出し
- 手順 2: 表領域のプライマリ・データベースへのトランスポート

手順 1: 補助データベースからの表領域の取出し

ここでは、補助データベースを現在時点以外の目的の時刻までリカバリし、目的の表領域を取り出します。

補助データベースの表領域を取り出す手順は、次のとおりです。

1. ALTER TABLESPACE ... READ ONLY 文を実行して、リカバリ・セット内の表領域を読取り専用にします。たとえば、次のようにして、sales_1 および sales_2 を読取り専用にします。

```
ALTER TABLESPACE sales_1 READ ONLY;  
ALTER TABLESPACE sales_2 READ ONLY;
```

2. リカバリ・セットが自己完結型であることを確認します。たとえば、次のように入力します。

```
EXECUTE SYS.DBMS_TTS.TRANSPORT_SET_CHECK('sales_1,sales_2',TRUE,TRUE);
```

3. 依存性を管理するためにトランスポータブル表領域の違反表を問い合わせます。たとえば、次のように入力します。

```
SELECT * FROM SYS.TRANSPORT_SET_VIOLATIONS;
```

すべての依存性が管理された場合、この問合せは行を戻しません。この表の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

4. 『Oracle9i データベース管理者ガイド』の説明に従って、エクスポート・ユーティリティを実行してトランスポータブル・セットを生成します。次の例のように、リカバリ・セット内のすべての表領域を含めます。

```
% exp SYS/pwd TRANSPORT_TABLESPACE=y TABLESPACES=(sales_1,sales_2) TTS_FULL_  
CHECK=y
```

このコマンドは、expdat.dmp という名前のエクスポート・ファイルを生成します。

手順 2: 表領域のプライマリ・データベースへのトランスポート

ここでは、リカバリ・セットの表領域をプライマリ・データベースに移送します。

リカバリ・セットの表領域をプライマリ・データベースに組み込む手順は、次のとおりです。

1. プライマリ・データベース（補助データベースではない）で DROP TABLESPACE 文を使用してリカバリ・セット内の表領域を削除します。たとえば、次のように入力します。

```
DROP TABLESPACE sales_1 INCLUDING CONTENTS;  
DROP TABLESPACE sales_2 INCLUDING CONTENTS;
```

2. 補助データベースのリカバリ・セットのデータ・ファイルを、プライマリ・データベース内のリカバリ・セット・ファイルの場所にリストアします。たとえば、次のように入力します。

```
% cp /net/aux_host/aux/sales_1.f /net/primary_host/oracle/dbs/sales_1.f  
% cp /net/aux_host/aux/sales_2.f /net/primary_host/oracle/dbs/sales_2.f
```

3. エクスポート・ファイル expdat.dmp をプライマリ・ホストに移動します。たとえば、次のように入力します。

```
% cp /net/aux_host/aux/expdat.dmp /net/primary_host/oracle/dbs/expdat.dmp
```

4. 『Oracle9i データベース管理者ガイド』の説明に従って、インポートを実行してプライマリ・データベースにトランスポータブル・セットを組み込みます。たとえば、次のように入力します。

```
% imp TRANSPORT_TABLESPACE=y FILE=expat.dmp  
DATAFILES=('/oracle/dbs/sales_1.f','/oracle/dbs/sales_2.f')
```

5. ALTER TABLESPACE READ WRITE 文を発行して、リカバリされた表領域を読み取り / 書き込みにします。たとえば、次のように入力します。

```
ALTER TABLESPACE sales_1 READ WRITE;  
ALTER TABLESPACE sales_2 READ WRITE;
```

6. 2-8 ページの「[オンラインの表領域およびデータ・ファイルのユーザー管理バックアップの作成](#)」の説明に従って、オペレーティング・システム・ユーティリティを使用してリカバリされた表領域をバックアップします。

注意： 表領域が失われる場合があるため、表領域をバックアップする必要があります。たとえば、メディア障害が発生した場合に、データベースの最後のバックアップからのアーカイブ・ログがリカバリされた表領域に論理的にリンクされていないことがあります。TSPITR 前に作成したバックアップからリカバリ・セットの表領域をリカバリしようとする、リカバリは失敗します。

パーティション表の部分 TSPITR の実行

パーティション表は、複数の表領域にまたがることができます。この手順は、リカバリ・セットがすべてのパーティションを完全に含んでいない場合にのみ使用してください。

この項では、範囲が変更または拡張されていないパーティション表の部分 TSPITR を実行する方法を、次の手順で説明します。

- [手順 1: プライマリ・データベースにおける個々のパーティションをリカバリするための表の作成](#)
- [手順 2: リカバリするパーティションの索引の削除](#)
- [手順 3: スタンドアロンの表とパーティションとの交換](#)
- [手順 4: リカバリ・セットの表領域の削除](#)
- [手順 5: 補助データベースでの表の作成](#)
- [手順 6: リカバリするパーティションの索引の削除](#)
- [手順 7: 補助データベースのパーティションとスタンドアロン表の交換](#)
- [手順 8: リカバリ・セットの表領域のトランスポート](#)
- [手順 9: プライマリ・データベースのパーティションとスタンドアロン表の交換](#)
- [手順 10: プライマリ・データベースのリカバリされた表領域のバックアップ](#)

注意： 範囲が拡張されたパーティションのリカバリでは、削除されたパーティションのリカバリも必要になることがよくあります。7-19 ページの「[パーティションが削除された場合のパーティション表の部分 TSPITR の実行](#)」を参照してください。

手順 1: プライマリ・データベースにおける個々のパーティションをリカバリするための表の作成

この表は、リカバリするパーティション表と正確に同じ列名および列データ型にする必要があります。次のテンプレートを使用して表を作成します。

```
CREATE TABLE new_table AS
  SELECT * FROM partitioned_table
  WHERE 1=2;
```

これらの表は、リカバリ・セットの各パーティションと交換するために使用されます（7-17 ページの「[手順 3: スタンドアロンの表とパーティションとの交換](#)」を参照）。

注意： 表とパーティションは同じスキーマに属する必要があります。

手順 2: リカバリするパーティションの索引の削除

リカバリするパーティション上の索引を削除するか、リカバリするパーティション上に存在する、パーティションされていない同じ索引を作成します。リカバリするパーティション上の索引を削除した場合には、補助データベースからも索引を削除する必要があります（7-17 ページの「[手順 6: リカバリするパーティションの索引の削除](#)」を参照）。TSPITR が完了した後で索引を再作成します。

手順 3: スタンドアロンの表とパーティションとの交換

変数を適切なオブジェクトの名前で置き換えて次の文を発行し、リカバリ・セット内の各パーティションを、関連するスタンドアロン表（手順 1 で作成した）と交換します。

```
ALTER TABLE table_name EXCHANGE PARTITION partition_name WITH TABLE table_name;
```

手順 4: リカバリ・セットの表領域の削除

プライマリ・データベースで、リカバリ・セット内の各表領域を削除します。たとえば、`tablespace_name` を表領域の名前で置き換えて、次のように入力します。

```
DROP TABLESPACE tablespace_name INCLUDING CONTENTS;
```

手順 5: 補助データベースでの表の作成

補助データベースをリカバリし、RESETLOGS オプションを指定してオープンした後、リカバリするパーティション表と同じ列名および列データ型の表を SYSTEM 表領域内に作成します。表を SYSTEM 表領域内に作成しないと、Oracle は ORA-01552 エラーを出力します。

リカバリするパーティションごとに表を 1 つ作成します。これらの表は、後でリカバリ・セットの各パーティションと交換するために使用されます。

注意： 表とパーティションは同じスキーマに属する必要があります。

手順 6: リカバリするパーティションの索引の削除

リカバリするパーティション上の索引を削除するか、リカバリするパーティション上に存在する、パーティションされていない同じ索引を作成します（手順 1 で作成した表上に）。

手順 7: 補助データベースのパーティションとスタンドアロン表の交換

変数を適切なオブジェクト名で置き換えて次の SQL スクリプトを実行し、補助データベースのリカバリ・セット内の各パーティションについて、パーティションとスタンドアロン表（手順 5 で作成した）を交換します。

```
ALTER TABLE partitioned_table_name  
EXCHANGE PARTITION partition_name  
WITH TABLE table_name;
```

手順 8: リカバリ・セットの表領域のトランスポート

7-14 ページの「[トランスポート可能な表領域を使用した TSPITR の実行](#)」の説明に従って、補助データベースからリカバリ・セットの表領域をエクスポートし、プライマリ・データベースにインポートします。

手順 9: プライマリ・データベースのパーティションとスタンドアロン表の交換

変数を適切なオブジェクト名で置き換えて次の文を発行し、プライマリ・データベース上のリカバリされた各パーティションを関連するスタンドアロン表と交換します。

```
ALTER TABLE table_name EXCHANGE PARTITION partition_name WITH TABLE table_name;
```

関連する索引が削除されている場合には、再作成します。

手順 10: プライマリ・データベースのリカバリされた表領域のバックアップ

プライマリ・データベースのリカバリされた表領域をバックアップします。バックアップしておかないと、メディア障害時にデータが失われます。

パーティションが削除された場合のパーティション表の部分 TSPITR の実行

この項では、パーティションが削除された場合にパーティション表で TSPITR を実行する方法を、次の手順で説明します。

- 手順 1: 削除されたパーティションの下限および上限の検出
- 手順 2: 一時的な表の作成
- 手順 3: パーティション表のレコードの削除
- 手順 4: リカバリ・セットの表領域の削除
- 手順 5: 補助データベースでの表の作成
- 手順 6: リカバリするパーティションの索引の削除
- 手順 7: スタンドアロンの表とパーティションとの交換
- 手順 8: リカバリ・セットの表領域のトランスポート
- 手順 9: パーティション表へのスタンドアロン表の挿入
- 手順 10: プライマリ・データベースのリカバリされた表領域のバックアップ

手順 1: 削除されたパーティションの下限および上限の検出

パーティションが削除された場合には、削除されたパーティションの上側のパーティションの範囲が下方に拡張されます。このため、リカバリ後には、上側のパーティションの中に、実際は削除されたパーティションにあるべきレコードがあることがあります。これを確認するには、変数を適切な値で置き換えて、プライマリ・データベースで次の SQL スクリプトを実行します。

```
SELECT * FROM partitioned_table
WHERE relevant_key
      BETWEEN low_range_of_partition_that_was_dropped
      AND high_range_of_partition_that_was_dropped;
```

手順 2: 一時的な表の作成

レコードが戻された場合には、後で必要な場合にリカバリされたパーティションに挿入できるように、これらのレコードを格納する一時的な表を作成します。

手順 3: パーティション表のレコードの削除

パーティション表から、一時的な表に格納されたすべてのレコードを削除します。

手順 4: リカバリ・セットの表領域の削除

プライマリ・データベースで、リカバリ・セット内の各表領域を削除します。たとえば、`tablespace_name` を表領域の名前で置き換えて、次のように入力します。

```
DROP TABLESPACE tablespace_name INCLUDING CONTENTS;
```

手順 5: 補助データベースでの表の作成

RESETLOGS オプションを指定して補助データベースをオープンした後、リカバリするパーティション表と同じ列名および列データ型の表を SYSTEM 表領域内に作成します。表を SYSTEM 表領域内に作成しないと、Oracle は ORA-01552 エラーを出力します。リカバリするパーティションごとに表を 1 つ作成します。これらの表は、後でリカバリ・セットの各パーティションと交換するために使用されます。

手順 6: リカバリするパーティションの索引の削除

リカバリするパーティション上の索引を削除するか、リカバリするパーティション上に存在する、パーティションされていない同じ索引を作成します。

手順 7: スタンドアロンの表とパーティションとの交換

変数を適切な値で置き換えて次の文を実行し、補助リカバリ・セット内の各パーティションについて、パーティションと手順 5 で作成したスタンドアロン表を交換します。

```
ALTER TABLE partitioned_table_name  
EXCHANGE PARTITION partition_name  
WITH TABLE table_name;
```

手順 8: リカバリ・セットの表領域のトランスポート

7-14 ページの「[トランスポートابل表領域を使用した TSPITR の実行](#)」の説明に従って、補助データベースからリカバリ・セットの表領域をエクスポートし、プライマリ・データベースにインポートします。

手順 9: パーティション表へのスタンドアロン表の挿入

この時点で、パーティション表にスタンドアロン表を挿入する必要があります。これを行うには、最初に、変数を適切な値で置き換えて次の文を発行します。

```
ALTER TABLE table_name SPLIT PARTITION partition_name AT (key_value) INTO  
(PARTITION partition_1_name TABLESPACE tablespace_name,  
PARTITION partition_2_name TABLESPACE tablespace_name);
```

この時点では、この範囲のキーはすでに表から削除されているため、パーティション 2 は空です。

変数を適切な値で置き換えて次の文を発行して、スタンドアロン表をパーティションと交換します。

```
ALTER TABLE EXCHANGE PARTITION partition_name WITH TABLE table_name;
```

ここで、手順 2 で保存しておいたレコードを、必要に応じてリカバリ済みのパーティションに挿入します。

注意： 削除されたパーティションが表の最後のパーティションである場合は、ALTER TABLE ADD PARTITION 文を使用して追加します。

手順 10: プライマリ・データベースのリカバリされた表領域のバックアップ

プライマリ・データベースのリカバリされた表領域をバックアップします。バックアップしておかないと、メディア障害時にデータが失われます。

パーティションが分割されている場合のパーティション表の TSPITR の実行

この項では、パーティションが分割された場合のパーティション表のリカバリ方法について説明します。ここで説明する項目は、次のとおりです。

- **手順 1:** プライマリ・データベースの分割された 2 つのパーティションのうち下位にあるパーティションの削除
- **手順 2:** パーティション表の部分 TSPITR の実行と同じ手順の実行

手順 1: プライマリ・データベースの分割された 2 つのパーティションのうち下位にあるパーティションの削除

範囲が分割されている、リカバリする各パーティションで、上位のパーティションが下方に拡張されるように、2 つのパーティションの下位のパーティションを削除します。つまり、上位のパーティションは分割前と同じ範囲になります。たとえば、P1 を 2 つのパーティション、P1A および P1B に分割した場合には、P1B を削除する必要があります。これにより、P1A の範囲は P1 と同じになります。

範囲が分割された、リカバリする各パーティションについて、リカバリ対象のパーティション表と正確に同じ列名および列データ型を持つ表を作成します。たとえば、変数を適切な値で置き換えて、次の文を実行します。

```
CREATE TABLE new_table  
AS SELECT * FROM partitioned_table  
WHERE 1=2;
```

これらの表は、手順 3 でリカバリ・セットの各パーティションと交換するために使用されます。

手順 2: パーティション表の部分 TSPITR の実行と同じ手順の実行

7-16 ページの「パーティション表の部分 TSPITR の実行」と同じ手順を実行しますが、手順 1 をスキップします。つまり、「**手順 2: リカバリするパーティションの索引の削除**」から開始して、残りのすべての手順を実行します。

索引

A

ABORT オプション
 SHUTDOWN 文, 3-10, 3-11, 4-17, 4-24, 4-25
ALLOW ... CORRUPTION 句
 RECOVER コマンド, 5-9
ALTER DATABASE 文
 BACKUP CONTROLFILE 句, 2-21
 TO TRACE オプション, 2-21
 CLEAR LOGFILE 句, 6-10
 END BACKUP 句, 2-13
 NORESETLOGS オプション, 4-31
 RECOVER 句, 3-17, 4-8
 RESETLOGS オプション, 4-25, 4-26, 4-31
ALTER SYSTEM 文
 RESUME 句, 2-19
 SUSPEND 句, 2-19
ALTER TABLESPACE 文
 BEGIN BACKUP 句, 2-8, 2-11
 END BACKUP オプション, 2-11
ARCHIVELOG モード
 データ・ファイルの消失, 6-3
AS SELECT 句
 CREATE TABLE 文, 4-37
AUTORECOVERY オプション
 SET 文, 4-3

B

BACKUP CONTROLFILE TO TRACE 句
 ALTER DATABASE 内, 2-3, 2-21
BACKUP CONTROLFILE 句
 ALTER DATABASE 内, 2-3
BEGIN BACKUP 句
 ALTER TABLESPACE 文, 2-8

C

CLEAR LOGFILE 句
 ALTER DATABASE 内, 6-10
CONTROL_FILES 初期化パラメータ, 3-11
CREATE DATAFILE 句
 ALTER DATABASE 内, 3-8
CREATE TABLESPACE 文, 6-3
CREATE TABLE 文
 AS SELECT 句, 4-37

D

DB_FILE_NAME_CONVERT 初期化パラメータ, 7-8
DBA_DATA_FILES ビュー, 2-6, 2-9, 2-15
DBVERIFY ユーティリティ, 2-29

L

LOG_ARCHIVE_DEST_*n* 初期化パラメータ, 4-6, 7-8
LOG_ARCHIVE_FORMAT 初期化パラメータ, 4-6, 7-8
LOG_FILE_NAME_CONVERT 初期化パラメータ, 7-8
LOGSOURCE 変数
 SET 文, 3-17, 4-8

M

MOUNT オプション
 STARTUP 文, 4-19, 4-20

N

NOARCHIVELOG モード

- データ・ファイルの消失, 6-2
- デメリット, 4-24
- リカバリ, 4-24

O

ORA-01578 エラー・メッセージ, 4-37

Oracle 管理ファイル, 1-3

P

Point-in-Time リカバリ, 4-17

- 表領域, 7-1 ~ 7-15

R

RAW デバイス

- UNIX のバックアップ, 2-25
- Windows のバックアップ, 2-27
- バックアップ先, 2-24
- リストア先, 3-7

RECOVERY_PARALLELISM 初期化パラメータ, 4-27

RECOVER 句

- ALTER DATABASE 内, 3-17, 4-8

RECOVER コマンド

- PARALLEL オプション, 4-27
- UNTIL TIME オプション, 4-22
- USING BACKUP CONTROLFILE 句, 4-38
- リカバリ不能なオブジェクトおよびスタンバイ・データベース, 4-37

REDO レコード

- 適用の問題, 5-2

REDO ログ

- ネーミング, 4-6
- バックアップ用のファイルのリスト, 2-2
- パラレル REDO, 5-3
- 非互換フォーマット, 5-3

RESETLOGS オプション

- ALTER DATABASE 内, 4-25, 4-26, 4-28, 4-31
- 使用後のデータベースのリカバリ, 4-33

RESETLOGS 操作

- 処置, 4-32
- その後のバックアップ, 4-32
- 必要な場合, 4-28

RESUME 句

- ALTER SYSTEM 文, 2-19

S

SCN (システム変更番号)

- 分散リカバリで使用, 6-16

SET 文

- AUTORECOVERY オプション, 4-3
- LOGSOURCE 変数, 3-17, 4-8

SHUTDOWN 文

- ABORT オプション, 3-10, 3-11, 4-17, 4-24, 4-25

STARTUP 文

- MOUNT オプション, 4-19, 4-20

SUSPEND 句

- ALTER SYSTEM 文, 2-19

U

UNDO 表領域

- バックアップ, 2-17

UNTIL TIME オプション

- RECOVER コマンド, 4-22

USING BACKUP CONTROLFILE オプション

- RECOVER コマンド, 4-20

V

V\$ARCHIVED_LOG ビュー

- すべてのアーカイブ・ログのリスト, 2-24

V\$BACKUP ビュー, 2-3

V\$DATAFILE ビュー, 2-2

- バックアップ用のファイルのリスト, 2-2

V\$LOG_HISTORY ビュー

- すべてのアーカイブ・ログのリスト, 3-17

V\$LOGFILE ビュー, 6-7, 6-8

- オンライン REDO ログのリスト, 2-2
- バックアップ用のファイルのリスト, 2-2

V\$RECOVER_FILE ビュー, 3-6

V\$RECOVERY_LOG ビュー

- リカバリに必要なログのリスト, 3-17

V\$TABLESPACE ビュー, 2-2

あ

- アーカイブ REDO ログ
 - 損害, 6-13
 - 適用の自動化, 4-4, 4-5
 - デフォルトの場所の変更, 4-7
 - 破損, 5-2
 - 非互換フォーマット, 5-3
 - メディア・リカバリ時の適用, 4-2
 - リカバリ後の削除, 3-18
 - リカバリ時の位置, 4-2
 - リカバリ中のエラー, 4-9
 - リカバリに使用
 - デフォルト以外の場所, 4-8
 - デフォルトの場所, 4-6
 - リストア, 3-17
- アクティブなオンライン REDO ログ
 - グループの損害, 6-11, 6-12
- アラート・ログ, 6-16
 - RESETLOGS 後のチェック, 4-32

い

- 一時停止 / 再開モード, 1-5, 2-17
- 一貫性バックアップ
 - データベース全体, 2-4
- インスタンス障害
 - バックアップ・モードの, 2-12
- インポート・ユーティリティ, 2-30
 - 使用手順, 2-32
 - データベース・リカバリ, 2-32

え

- エクスポート
 - モード, 2-31
- エクスポート・ユーティリティ, 2-30
 - バックアップ, 2-30
 - 読み込み一貫性, 2-31

お

- オンライン REDO ログ, 6-9
 - アーカイブ・グループ, 6-7, 6-8
 - アクティブ・グループ, 6-7, 6-8
 - アクティブ・ログの判別, 6-8
 - 現行グループ, 6-7, 6-8
 - 消去
 - 障害, 6-10
 - 損害
 - アクティブ・グループ, 6-11, 6-12
 - グループ, 6-8
 - すべてのメンバー, 6-8
 - ミラー・メンバー, 6-7
 - リカバリ, 6-6
 - 破損したメンバーの置換, 6-7
 - バックアップ用のログ・ファイルのリスト, 2-2
 - 非アクティブのグループ, 6-7, 6-8
 - 非アクティブのログの消去
 - アーカイブ, 6-9
 - 未アーカイブ, 6-10
 - 複数のグループの損害, 6-12
 - メディア・リカバリ時の適用, 4-2
 - メンバーのステータス, 6-7, 6-8

か

- 完全リカバリ
 - 手順, 4-10

く

- グループ
 - アーカイブ REDO ログ, 6-7, 6-8
 - オンライン REDO ログ, 6-7, 6-8
- クローン・データベース
 - TSPITR の準備, 7-10, 7-12
 - パラメータ・ファイルの準備, 7-7

け

- 警告
 - 一貫性およびバックアップのエクスポート, 2-31

こ

コールド・フェイルオーバー・クラスタ
 定義, 2-13
コマンド、SQL
 ALTER DATABASE, 3-17, 4-8
コマンド、SQL*Plus
 RECOVER
 UNTIL TIME オプション, 4-22
 SET, 3-17, 4-3, 4-8

し

時間ベースの調整式リカバリ
 分散データベース, 6-15
時間ベースのリカバリ, 4-22
 分散データベースの調整, 6-15
試行リカバリ
 概要, 5-9
 説明, 5-10
時刻書式
 RECOVER DATABASE UNTIL TIME 文, 4-22
システム時刻
 変更
 リカバリへの影響, 4-17
自動 UNDO 管理
 表領域のバックアップ, 2-17
初期化パラメータ
 CONTROL_FILES, 3-11
 LOG_ARCHIVE_DEST_*n*, 4-6
 LOG_ARCHIVE_FORMAT, 4-6
 RECOVERY_PARALLELISM, 4-27
新機能, xv ~ xix

す

スタック・リカバリ
 定義, 5-2

せ

制御ファイル
 作成, 3-16
 時間ベースのリカバリ, 4-18
 損害, 3-9
 すべてのコピー, 3-15

多重

 損害, 3-10
 トレース・ファイルへのバックアップ, 2-22
バックアップ, 2-3, 2-20
 トレース・ファイル, 2-21
 バイナリ, 2-20
 ファイル名の検索, 2-2
リストア
 デフォルト以外の場所へ, 3-11
 デフォルトの場所へ, 3-10

そ

損害
 非アクティブのログ・グループ, 6-9

た

多重ファイル
 制御ファイル
 損害, 3-10

ち

中断後のリカバリの再開, 4-36

て

データ・ディクショナリ・ビュー, 2-6, 2-9, 2-15
データ・ファイル
 改名
 リカバリ後, 6-4
 再作成, 3-8
 ステータスの判別, 2-3
 損害, 6-2
 ARCHIVELOG モードの, 6-3
 NOARCHIVELOG モードの, 6-2
 バックアップ
 オフライン, 2-6
 リカバリ
 バックアップなし, 3-8
 リスト
 バックアップ用, 2-2
 リストア, 3-7
 デフォルトの場所へ, 3-8
データ・ブロック
 破損, 5-3

- データベース
 - 一時停止, 1-5, 2-17
 - バックアップ用のリスト, 2-2
 - メディア・リカバリ手順, 4-1 ~ 4-23
 - メディア・リカバリの使用例, 6-1
 - リカバリ
 - OPEN RESETLOGS オプション後, 4-33
 - 制御ファイルの破損後, 3-10, 3-11
- データベース・インカネーション, 4-28
- データベース全体のバックアップ
 - ARCHIVELOG モード, 2-4
 - NOARCHIVELOG モード, 2-4
 - 準備, 2-5
 - 非一貫性, 2-4
 - リストア元, 4-24
- データベースの Point-in-Time リカバリ (DBPITR)
 - ユーザー管理, 4-17
- データベースの一時停止, 1-5, 2-17

と

- トランスポートابل表領域
 - TSPITR および, 7-3
 - リカバリ, 6-5
- 取消ベースのメディア・リカバリ
 - 手順, 4-14, 4-22
- トレース・ファイル
 - 制御ファイルのバックアップ, 2-22
 - 制御ファイルのバックアップ先, 2-21

は

- パーティション表
 - 削除されたパーティション, 7-19
 - パーティションの分割, 7-22
 - 部分 TSPITR の実行, 7-16
- バックアップ
 - DBVERIFY ユーティリティ, 2-29
 - RESETLOGS 後のバックアップ, 4-32
 - 一貫性, 2-4
 - オフライン・データ・ファイル, 2-6
 - オフライン表領域, 2-6
 - 記録の保管, 3-3
 - クローズ, 2-5
 - 検証, 2-29

- 制御ファイル, 2-20
 - トレース・ファイル, 2-21
 - バイナリ, 2-20
- データ・ファイルのステータスの判別, 2-3
- データベース全体
 - 準備, 2-5
 - データベース全体のリストア, 4-24
 - 非一貫性, 2-4
 - 必要なファイルのリスト, 2-2
 - 表領域, 2-10
 - ユーザー管理
 - 概要, 1-3
 - リストア, 3-7
 - ユーザー管理のリストア, 3-2
 - 読取り専用表領域, 2-15
 - 論理, 2-30
- バックアップ・モード
 - ALTER DATABASE END BACKUP を使用して終了, 2-13
 - インスタンス障害, 2-12
 - オンラインのユーザー管理バックアップ, 2-9
- パラレル・ブロック・リカバリ
 - 定義, 4-27
- パラレル・リカバリ, 4-27

ひ

- 非アクティブのオンライン REDO ログ
 - 損害, 6-9
- 表
 - 削除のリカバリ, 6-14
- 表領域
 - オープン状態のデータベースでのオフラインのリカバリ, 4-14
 - バックアップ, 2-10
 - オフライン, 2-6
 - オンライン, 2-10
 - 読取り / 書込み
 - バックアップ, 2-9
 - 読取り専用
 - バックアップ, 2-15
- 表領域の Point-in-Time リカバリ
 - 概要, 1-7, 7-2
 - クローン・データベース, 7-2
 - 計画, 7-4
 - 実行, 7-1 ~ 7-15
 - 実行方法, 7-3

トランスポータブル表領域機能を使用する手順,
7-14, 7-15
トランスポータブル表領域による方法, 7-3
ユーザー管理, 7-3
要件, 7-5
用語, 7-2

ふ

ファイル名
バックアップ用のリスト, 2-2
不完全メディア・リカバリ, 4-17
Oracle Real Application Clusters 構成内, 4-6
時間ベース, 4-22
バックアップ制御ファイルの使用, 4-6
変更ベース, 4-23
分散データベース
時間ベースの調整式リカバリ, 6-15
変更ベースのリカバリ, 6-15
リカバリ, 6-15

へ

変更ベースのメディア・リカバリ, 4-23
分散データベースの調整, 6-15

ほ

ホット・バックアップ
失敗, 2-12
ALTER DATABASE END BACKUP を使用して
終了, 2-13
ホット・バックアップ・モード
オンラインのユーザー管理バックアップ, 2-10

み

ミラー化ファイル
オンライン REDO ログ
損害, 6-7
分割, 1-5, 2-17
一時停止 / 再開モード, 1-5, 2-17
ミラーの分割
一時停止 / 再開モード, 1-5, 2-17

め

メディア障害

NOARCHIVELOG モード, 4-24
アーカイブ REDO ログ・ファイルの損害, 6-13
オンライン REDO ログ・グループの損害, 6-8
オンライン REDO ログの損害, 6-7
オンライン REDO ログ・メンバーの損害, 6-7
完全リカバリ, 4-10 ~ 4-16, 4-16
制御ファイルの損害, 3-9, 3-15
データ・ファイルの損害, 6-2
リカバリ, 4-10 ~ 4-23
分散データベース, 6-15
リカバリ手順
例, 6-2

メディア・リカバリ, 4-1 ~ 4-35

ADD DATAFILE 操作, 6-3
NOARCHIVELOG モード, 4-24
OPEN RESETLOGS 操作後, 4-33
REDO ログ適用不可, 4-9
アーカイブ REDO ログの適用, 4-2
インポート・ユーティリティの使用方法, 2-32
エラー, 4-9, 5-3
オープン状態のデータベースのオフライン表領域,
4-14
オンライン REDO ログ・ファイル, 6-6
オンラインの破損していない表領域, 4-14
完全, 4-10 ~ 4-16, 4-16
クローズ状態のデータベース, 4-10
完了, 4-13, 4-16
再起動, 4-36
時間ベース, 4-17
試行, 5-9
概要, 5-9
説明, 5-10

使用例, 6-1
制御ファイルの破損後, 3-10, 3-11
制限, 4-37
前提条件, 4-37
損害を受けたファイル
損害を受けたアーカイブ REDO ログ・ファイル,
6-13
損害を受けた制御ファイル, 3-9
損害を受けたデータ・ファイル, 6-2
損害を受けたミラー化制御ファイル, 3-10

タイプ

分散データベース, 6-15

- 中断, 4-36
- 中断後の再開, 4-36
- データ・ファイル
 - バックアップなし, 3-8
- トラブルシューティング, 5-2
 - 基本的方法, 5-4
- トランスポートابل表領域, 6-5
- 取消ベース, 4-14, 4-17, 4-22
- 破損
 - 許可する, 5-7
- パラレル, 4-27
- 不完全, 4-17
- 分散データベース, 6-15
 - 時間ベースの調整式, 6-15
- 変更ベース, 4-17, 4-23
- 問題, 5-2, 5-3
 - 修正, 5-5
 - 調査, 5-4
- リカバリ後のデータベースのオープン, 4-28, 4-31
- リストア
 - アーカイブ REDO ログ・ファイル, 3-17
 - データベース全体のバックアップ, 4-24
 - ロールフォワード・フェーズ, 4-2
- メディア・リカバリの中断, 4-36

も

- モード
 - NOARCHIVELOG
 - 障害からのリカバリ, 4-24

ゆ

- ユーザー・エラー
 - リカバリ, 6-14
- ユーザー管理のバックアップとリカバリ
 - 定義, 1-2
 - 理由, 1-2
- ユーザー管理のリカバリ, 4-17
 - ADD DATAFILE 操作, 6-3
 - アーカイブ REDO ログの適用, 4-2
 - 完全, 4-10
 - 使用例, 6-1
 - 中断, 4-36
 - 不完全, 4-17
 - リカバリ後のデータベースのオープン, 4-28

- ユーザー管理のリストアおよびリカバリ
 - 概要, 1-6
- ユーザー管理のリストア操作, 3-2
- ユーザー管理バックアップ, 2-4
 - オフライン・データ・ファイル, 2-6
 - オフライン表領域, 2-6
 - 基本的方法, 1-4
 - 検証, 2-29
 - 事前にファイルを列挙, 2-2
 - 制御ファイル, 2-20
 - トレース・ファイル, 2-21
 - バイナリ, 2-20
 - 定義, 1-3
 - データ・ファイルのステータスの判別, 2-3
 - データベース全体, 2-5
 - データベース全体のリストア, 4-24
 - バックアップ・モード, 2-12
 - 表領域, 2-10
 - ホット・バックアップ, 2-13
 - 読取り専用表領域, 2-15
 - リストア, 3-7

よ

- 読込み一貫性
 - エクスポート・ユーティリティ, 2-31
- 読取り専用表領域
 - バックアップ, 2-15

り

- リカバリ
 - ADD DATAFILE 操作, 6-3
 - アーカイブ・ログの自動適用, 4-3
 - インポート・ユーティリティ, 2-32
 - エラー, 5-3
 - オンライン REDO ログ, 6-6
 - グループの損害, 6-8
 - メンバーの損害, 6-7
 - 完全, 4-10 ~ 4-16
 - オフライン表領域, 4-14
 - クローズ状態のデータベース, 4-10
 - 削除された表, 6-14
 - 時間ベース, 4-22
 - 試行, 5-9
 - 概要, 5-9
 - 説明, 5-10

- 使用するプロセス数の設定, 4-27
- スタック, 5-2
- 制御ファイル, 3-9
- 制限, 4-37
- 前提条件, 4-37
- 中断, 4-36
- データ・ファイル, 6-2
 - ARCHIVELOG モード, 6-3
 - NOARCHIVELOG モード, 6-2
- 適用できない場合の対応, 4-9
- デフォルト以外の場所のログの使用, 4-7, 4-8
- デフォルトの場所のログの使用, 4-6
- トラブルシューティング, 5-2
- トランスポートابل表領域, 6-5
- 取消ベース, 4-14, 4-22
- 破損
 - 意図的に許可, 5-7
- パラレル, 4-27
- パラレル処理, 4-27
- 複数の REDO スレッド, 4-6
- 変更ベース, 4-23
- メディア, 3-1, 4-1, 5-1, 6-1
- 問題, 5-2
 - 修正, 5-5
 - 調査, 5-4
- ユーザー・エラー, 6-14
- ユーザー管理, 1-6, 3-1, 4-1, 5-1, 6-1
- リカバリ後のデータベースのオープン, 4-28
- リカバリの必要なファイルの判別, 3-6
- リカバリ不能なオブジェクト
- RECOVER 操作, 4-37
- リカバリ
 - リカバリ不能なオブジェクトとリカバリ, 4-37
- リストア
 - RAW デバイスへ, 3-7
 - アーカイブ REDO ログ, 3-17
 - 制御ファイル, 3-9
 - デフォルト以外の場所へ, 3-11
 - デフォルトの場所へ, 3-10
 - データ・ファイル
 - デフォルトの場所へ, 3-8
 - データベース
 - 新しい場所へのリストア, 4-25
 - デフォルトの場所へ, 4-24
 - データベース全体のバックアップ, 4-24
 - ユーザー管理バックアップ, 1-6, 3-2
 - 記録の保管, 3-3

ろ

- ログ順序番号
 - リカバリ時に要求, 4-2
- 論理バックアップ, 2-30