

Oracle8 Server バックアップおよびリカバリ

リリース 8.0

1998 年 2 月

部品番号 A56815-1

ORACLE[®]

The Database for Network Computing[™]

Oracle8 Server バックアップおよびリカバリ、リリース 8.0

部品番号：A56815-1

第 1 版：1998 年 2 月

原本名：Oracle8 Backup and Recovery, Release 8.0

原本部品番号：A58396-01

原本著者：Connie Dialeris, Joyce Fee

協力者：Bill Bridge, Sandra Cheevers, John Frazzini, TimBerry-Hart, Gordon Larimer, Bill Lee, Diana Lorentz, Greg Pongracz, LynPratt, Tuomas Pystynen, Daniel Semler, Slartibartfast, Steve Wertheimer, Min Zhou

Copyright© 1997, Oracle Corporation. All rights reserved.

Printed In Japan

制限付権利の説明

プログラムの使用、複製、または開示は、オラクル社との契約に記された制約条件に従うものとします。

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

危険な用途への使用について

当社製品は、原子力、航空産業、大量輸送、又は医療の分野など、本質的に危険が伴うアプリケーションを用途として特に開発されておられません。当社製品を上述のようなアプリケーションに使用することについての安全確保は顧客各位の責任と費用により行っていただきたく、万一かかる用途での使用によりクレームや損害が発生いたしましても、当社および開発元である米国 Oracle Corporation（その関連会社も含まれます。）は一切責任を負いかねます。

ORACLE は、Oracle Corporation の登録商標です。

本文中の他社の商品名は、それぞれ各社の商標または登録商標です。

目次

はじめに	XV
------------	----

第1章 バックアップの必要性について

バックアップとは?	1-2
バックアップがなぜ重要なのか?	1-2
バックアップをとる時期	1-2
障害のタイプ	1-3
データベースの物理構造	1-4
制御ファイル	1-4
オンライン REDO ログ・ファイル	1-5
データ・ファイル	1-5
ロールバック・セグメント	1-6
アーカイブ・ログ	1-7

第2章 バックアップ対象について

オンライン REDO ログ	2-2
オンライン REDO ログ・ファイルの内容	2-2
オンライン REDO ログ・ファイルへの書き込み方法	2-3
チェックポイント	2-5
多重オンライン REDO ログ・ファイル	2-9
オンライン REDO ログのスレッドと Oracle Parallel Server	2-13
アーカイブ REDO ログ	2-13
アーカイブの仕組み	2-14
アーカイブ REDO ログ・ファイルの内容	2-15

アーカイブ REDO ログの多重化	2-15
データベースのアーカイブ・モード	2-16
NOARCHIVELOG モード（メディア回復使用禁止）	2-16
ARCHIVELOG モード（メディア回復使用可能）	2-16
制御ファイル	2-19
制御ファイルの内容	2-19
多重制御ファイル	2-20
バックアップのタイプ	2-21
データベース全体のバックアップ	2-21
表領域のバックアップ	2-24
データ・ファイルのバックアップ	2-24
制御ファイルのバックアップ	2-24
アーカイブ・ログのバックアップ	2-25
バックアップ形式	2-26
バックアップ・セット	2-26
データ・ファイルのコピー	2-27
オペレーティング・システムのバックアップ	2-27
論理バックアップ	2-28

第3章 バックアップをとる時期

データベースのバックアップのガイドライン	3-2
頻繁で定期的なバックアップの実行	3-2
構造変更時のデータベースの該当部分のバックアップ	3-3
使用頻度の高い表領域の頻繁なバックアップ	3-3
回復不能な操作またはログをとっていない操作の実行後のバックアップ	3-4
古いバックアップの保存	3-4
RESETLOGS オプション使用後のデータベースのバックアップ	3-5
保護と柔軟性の強化のためのデータベース・データのエクスポート	3-6
分散データベースのバックアップの検討	3-6
バックアップおよび回復計画のテスト	3-7
バックアップ計画の作成	3-7
NOARCHIVELOG モードでのバックアップ計画	3-7
ARCHIVELOG モードでのバックアップ計画	3-8
オンライン REDO ログのバックアップ	3-9
オンライン REDO ログのバックアップ作成の危険性	3-12

第 4 章 回復方針の選択

回復の概要と方針	4-2
回復で使用される重要なデータ構造	4-3
回復計画	4-5
回復方針を判断する要因	4-5
回復操作	4-7

第 5 章 バックアップ方法の選択

バックアップ方法および要件	5-2
Recovery Manager	5-2
オペレーティング・システム (O/S)	5-2
Export	5-3
Enterprise Backup Utility	5-3
Recovery Manager	5-3
Recovery Manager でのバックアップに Oracle Enterprise Manager を 使用する方法 5-3	
Recovery Manager と従来のオペレーティング・システムの バックアップとの違い 5-4	
ディスクへのバックアップ	5-4
順次メディアへのバックアップ	5-5
バックアップ方法の機能の比較	5-7

第 6 章 Recovery Manager のスタート・ガイド

Recovery Manager を使用する前に決めておくべきこと	6-2
リカバリ・カタログを使用するかどうか	6-2
パスワード・ファイルを使用するかどうか	6-6
init.ora ファイルとパスワード・ファイルのバックアップ方法を定める	6-7
Recovery Manager の接続オプション	6-7
リカバリ・カタログを使用せずに Recovery Manager に接続する場合	6-7
リカバリ・カタログを使用して Recovery Manager に接続する場合	6-8
Recovery Manager のコマンドの実行	6-8
対話形式による Recovery Manager のコマンドの実行：例 1	6-9
コマンド・ファイルの使用：例 2	6-9
ストアド・スクリプトの使用：例 3	6-10
Recovery Manager で時刻パラメータを指定する場合	6-10

Recovery Manager のサンプル・スクリプトと使用例	6-11
テープにバックアップをとる場合の前提条件	6-11
メディア・マネージャとのリンク	6-11
一意のファイル名の生成	6-12
メディア・マネージャにおけるファイルの最大サイズの制限	6-12

第 7 章 Recovery Manager の概要

Recovery Manager の概要	7-2
順次メディアへのバックアップ	7-4
リカバリ・カタログ	7-5
リカバリ・カタログを使った操作	7-5
リカバリ・カタログなしでの運用	7-7
スナップショット制御ファイル	7-8
ストアド・スクリプト	7-9
Recovery Manager によるバックアップのタイプ	7-9
バックアップ・セット	7-9
全体バックアップ・セットと増分バックアップ・セット	7-11
イメージ・コピー	7-16
破損の検出	7-17
チャンネル制御	7-18
パラレル化	7-19
パラレル化の度合いを左右する要素	7-19
多重バックアップ・セット	7-21
レポート生成	7-22
Recovery Manager によるバックアップのためのユーザー・タグ	7-23
バックアップの制約	7-24
復元の制約	7-24
整合性チェック	7-25
Recovery Manager でのオープン状態のデータベースのバックアップ時における 分裂ブロックの検出	7-25
アーカイブ・ログの追跡	7-25
イメージ・コピーおよびアーカイブ・ログのカatalog化	7-26

第 8 章 Recovery Manager でのバックアップおよび回復の実行

リカバリ・カタログのインストール	8-2
------------------------	-----

データベースの登録.....	8-2
リカバリ・カタログのメンテナンス	8-3
リカバリ・カタログへのターゲット・データベースの登録.....	8-3
リカバリ・カタログ内の情報のリセット.....	8-4
リカバリ・カタログとターゲット・データベースの再同期化.....	8-4
バックアップ・セットまたはファイル・コピーの可用性の変更.....	8-6
ユーザー作成のバックアップ・ファイルのカタログ化.....	8-8
消失または破損したリカバリ・カタログ用のデータベースの回復.....	8-9
チャネル制御コマンドの使用法	8-9
チャネル制御コマンド.....	8-10
レポートの生成	8-12
レポートの生成.....	8-13
リストの生成.....	8-17
スクリプトのメンテナンス	8-19
スクリプトの作成と置換.....	8-19
スクリプトの削除.....	8-19
スクリプトの出力.....	8-19
スナップショットの制御ファイルの位置の設定	8-19
ファイルのバックアップ	8-20
バックアップの実行.....	8-21
バックアップ : 使用例.....	8-28
ファイルのコピー	8-28
Copy コマンドの指定子.....	8-29
ファイルの復元	8-30
データベースの Point-in-Time 回復.....	8-30
ファイルの選択.....	8-31
データ・ファイルの復元の宛先.....	8-31
制御ファイルの復元の宛先.....	8-31
制御ファイルのレプリケート.....	8-32
アーカイブ・ログの復元の宛先.....	8-32
データ・ファイルの復元のガイドライン.....	8-32
restore コマンドのオペランド・リスト.....	8-33
データ・ファイルの切替え	8-35
データ・ファイルの回復	8-35
データ・ファイルの回復のガイドライン.....	8-36
データベースの Point-in-Time 回復.....	8-36

回復コマンド	8-37
RECOVER コマンドのオブジェクト・リスト	8-38
バックアップと復元の監視	8-38
チャンネルへのセッションの接続	8-38
進行状況の監視	8-39

第 9 章 Recovery Manager の使用例

NOARCHIVELOG モードでのバックアップ	9-2
データベースと表領域のバックアップ	9-2
データベースのバックアップ	9-2
表領域のバックアップ	9-3
個々のデータ・ファイルのバックアップ	9-3
制御ファイルのバックアップ	9-3
アーカイブ・ログのバックアップ	9-3
Parallel Server 環境でのバックアップ	9-5
データ・ファイルのコピー	9-5
増分バックアップ	9-6
エラー処理	9-6
コピーのための O/S ユーティリティの使用方法	9-7
バックアップの保存	9-7
復元と回復	9-8
オープン状態のデータベースの復元と回復	9-8
復元	9-9
データベースの Point-in-Time 回復	9-10
リカバリ・カタログへの問合せ	9-11
リカバリ・カタログへの複雑な問合せのための report コマンドの使用方法 9-12	

第 10 章 Recovery Manager による表領域の Point-in-Time 回復

Recovery Manager による表領域の Point-in-Time 回復の概要	10-2
Recovery Manager による表領域の Point-in-Time 回復	10-3
制限事項	10-3
Recovery Manager による TSPITR の計画の必要条件	10-6
Recovery Manager による表領域の Point-in-Time 回復の実行	10-8
Recovery Manager による TSPITR 実行後の表領域のバックアップ	10-9
チューニングに関する考慮事項	10-9

補助セットの表領域内のデータ・ファイルへの新しい名前の指定.....	10-9
Recovery Manager による TSPITR 用のクローン名の設定と データ・ファイル・コピーの使用	10-10
変換されたデータ・ファイル名の使用.....	10-11

第 11 章 オペレーティング・システムのバックアップの実行

バックアップの実行	11-2
バックアップ前のデータベース・ファイルのリスト表示.....	11-2
データベース全体のバックアップの実行.....	11-3
表領域、データ・ファイル、制御ファイル、アーカイブ・ログのバックアップの実行....	11-5
制御ファイルのバックアップの実行.....	11-10
オンライン表領域のバックアップの失敗からの回復	11-12
データベース保護を強化するための Export ユーティリティと Import ユーティリティの使用方法	11-13
Export の使用方法.....	11-13
Import の使用方法.....	11-14

第 12 章 データベースの回復

分散回復の調整	12-2
時間ベースおよび変更ベースの分散データベース回復の調整.....	12-2
スナップショットによるデータベースの回復.....	12-3
回復のシナリオ	12-3
クローズ状態のデータベースの回復.....	12-3
オープン状態のデータベースでのオフライン表領域の回復.....	12-4
インスタンス起動中の回復の開始.....	12-4
REDO ログ・ファイルの適用.....	12-4
ログ・ファイルの適用.....	12-5
メディア回復の中断.....	12-8
データベース全体のバックアップの復元、NOARCHIVELOG モード	12-9
パラレル回復の指定	12-11
メディア回復の準備	12-11
メディア回復コマンド.....	12-11
すべてのメディア回復操作に共通する事項.....	12-12
完全メディア回復の実行	12-15
クローズ状態のデータベースの回復.....	12-15

オープン状態のデータベースのオフライン表領域の個別回復の実行	12-17
オープン状態のデータベースのオフライン表領域の個別回復の実行	12-19
不完全メディア回復の実行	12-21
取消しベースの回復の実行	12-21
時間ベースの回復の実行	12-26
変更ベースの回復の実行	12-31
災害時回復の準備	12-35
スタンバイ・データベースの計画と作成	12-35
プライマリ・データベースの物理構造の変更	12-39
回復不能オブジェクトと回復	12-43
読取り専用表領域と回復	12-44
バックアップ制御ファイルの使用	12-44
制御ファイルの再作成	12-44
回復手順の例	12-44
メディア障害のタイプ	12-45
データ・ファイルの消失	12-45
オンライン REDO ログ・ファイルの消失	12-46
アーカイブ REDO ログ・ファイルの消失	12-51
制御ファイルの消失	12-51
ユーザー・エラーからの回復	12-53

第 13 章 表領域の Point-in-Time 回復の実行

表領域の Point-in-Time 回復の概要	13-2
表領域の Point-in-Time 回復の計画	13-3
制限事項	13-4
TSPITR の必要条件	13-6
表領域の Point-in-Time 回復の実行	13-7
ステップ 1: TSPITR 実行時に消失するオブジェクトを調べる	13-8
ステップ 2: プライマリ・データベースにおける依存性の調査と解決	13-8
ステップ 3: TSPITR 用にプライマリ・データベースを準備する	13-12
ステップ 4: クローン・データベースのパラメータ・ファイルを準備する	13-12
ステップ 5: TSPITR 用にクローン・データベースを準備する	13-13
ステップ 6: クローン・データベースを回復する	13-14
ステップ 7: クローン・データベースをオープンする	13-15
ステップ 8: エクスポート用にクローン・データベースを準備する	13-15

ステップ 9: クローン・データベースをエクスポートする.....	13-15
ステップ 10: 回復セットのクローン・ファイルをプライマリ・データベースに コピーする.....	13-15
ステップ 11: プライマリ・データベースにインポートする.....	13-15
ステップ 12: プライマリ・データベースを使用できるように準備する	13-16
ステップ 13: プライマリ・データベースの回復した表領域のバックアップをとる	13-16
パーティション表の部分的な TSPITR の実行	13-16
ステップ 1: 回復する各パーティション用にプライマリ・データベースに表を作成する	13-17
ステップ 2: 回復するパーティションの索引を削除する.....	13-17
ステップ 3: パーティションをスタンドアロン表と入れ替える	13-18
ステップ 4: 回復セット表領域をオフラインにする.....	13-18
ステップ 5: クローン・データベースに表を作成する.....	13-18
ステップ 6: 回復するパーティションの索引を削除する.....	13-18
ステップ 7: パーティションをスタンドアロン表と入れ替える	13-18
ステップ 8: クローン・データベースをエクスポートする.....	13-18
ステップ 9: 回復セットのデータ・ファイルをプライマリ・データベースにコピーする	13-19
ステップ 10: プライマリ・データベースにインポートする.....	13-19
ステップ 11: 回復セットの表領域をオンラインにする.....	13-19
ステップ 12: パーティションをスタンドアロン表と入れ替える	13-19
ステップ 13: プライマリ・データベースの回復した表領域のバックアップをとる	13-20
パーティションを削除した場合のパーティション表の TSPITR の実行	13-21
ステップ 1: 削除したパーティションの下限および上限の範囲を確認する	13-21
ステップ 2: 一時表を作成する.....	13-22
ステップ 3: パーティション表からレコードを削除する.....	13-22
ステップ 4: 回復セットの表領域をオフラインにする.....	13-22
ステップ 5: クローン・データベースに表を作成する.....	13-22
ステップ 6: 回復するパーティションの索引を削除する.....	13-22
ステップ 7: スタンドアロン表をパーティションと入れ替える	13-22
ステップ 8: クローン・データベースをエクスポートする.....	13-22
ステップ 9: 回復セットのデータ・ファイルをプライマリ・データベースにコピーする	13-23
ステップ 10: プライマリ・データベースにインポートする.....	13-23
ステップ 11: 回復セットの表領域をオンラインにする.....	13-23
ステップ 12: スタンドアロン表をパーティション表に挿入する	13-23
ステップ 13: プライマリ・データベースの回復した表領域のバックアップをとる	13-24
パーティションを分割した場合のパーティション表の TSPITR の実行	13-25

ステップ 1: プライマリ・データベースの 2 つのパーティションのうち 下位の方を削除する	13-25
ステップ 2: 回復するパーティションの索引を削除する	13-26
ステップ 3: スタンドアロン表とパーティションを入れ替える	13-26
ステップ 4: 回復セットの表領域をオフラインにする	13-26
ステップ 5: クローン・データベースに表を作成する	13-26
ステップ 6: 回復するパーティションの索引を削除する	13-26
ステップ 7: スタンドアロン表とパーティションを入れ替える	13-26
ステップ 8: クローン・データベースをエクスポートする	13-26
ステップ 9: 回復セットのデータ・ファイルをプライマリ・データベースにコピーする ..	13-27
ステップ 10: プライマリ・データベースにインポートする	13-27
ステップ 11: 回復セットの表領域をオンラインにする	13-27
ステップ 12: パーティションをスタンドアロン表と入れ替える	13-27
ステップ 13: プライマリ・データベースの回復した表領域のバックアップをとる	13-28
TSPITR のチューニングに関する考慮事項	13-28
回復セットの位置に関する考慮事項	13-28
バックアップ制御ファイルに関する考慮事項	13-29

付録 A Recovery Manager のコマンド構文

allocateForDelete	A-2
allocate	A-3
archivelogRecordSpecifier	A-4
atClause	A-5
backup	A-6
backupSpec	A-7
catalog	A-9
change	A-10
connect	A-11
copy	A-12
copyOption	A-13
createScript	A-14
deleteScript	A-15
host	A-16
inputfile	A-17
list	A-18
listObjList	A-19

needBackupOperand	A-20
primary_key	A-21
printScript	A-22
recover	A-23
register	A-24
release	A-25
replaceScript	A-26
replicate	A-27
report	A-28
reportObject	A-29
reset	A-30
restoreObject	A-31
restore	A-32
restoreSpecOperand	A-33
resync	A-34
releaseForDelete	A-35
rmanCmd	A-36
reportObsoleteOperand	A-37
run	A-38
set	A-39
setPragma	A-40
sql	A-41
switch	A-42
untilClause	A-43
validate	A-44

用語集

索引

はじめに

このマニュアルでは、Oracle のバックアップと回復について説明します。このマニュアルは、今回のバージョンアップで Oracle Server のマニュアル・セットに追加されました。新機能の Recovery Manager ユーティリティまたは既存のオペレーティング・システムのバックアップを使ってバックアップおよび復元、回復の作業を行うために必要な概念を網羅し、個々の作業についても詳しく説明します。

注意： このマニュアルには、Oracle8 および Oracle8 Enterprise Edition の 2 つの製品の特徴と機能について説明する部分もあります。Oracle8 と Oracle8 Enterprise Edition の基本機能は同じですが、Enterprise Edition でだけ使用できる高度な機能がいくつか（一部はオプション）あります。たとえば、表領域の Point-in-Time 回復を、Recovery Manager により自動的に実行するには、Enterprise Edition が必要です。

Oracle8 と Oracle8 Enterprise Edition の違いおよび利用できる機能とオプションの詳細は、『Oracle8 と Oracle8 Enterprise Edition の解説』を参照してください。

このマニュアルの構成

このマニュアルは、次の部と章で構成されています。

部 / 章	内容
第 1 部	基本概念
第 1 章：「バックアップの必要性について」	2 章以降に進む前に理解しておかなければならない基本的なデータ構造について説明します。この章で説明するデータ構造は、Oracle のバックアップおよび復元、回復の作業の基礎となります。
第 2 章：「バックアップ対象について」	Oracle のバックアップおよび回復を理解するために必要なすべての基本概念を詳しく説明します。この章で説明する概念は、Recovery Manager ユーティリティにも当てはまります。
第 3 章：「バックアップをとる時期」	バックアップの実行時に従うべきガイドラインと方針の概要を説明します。
第 4 章：「回復方針の選択」	データベースを回復するときに従うべきガイドラインと方針の概要を説明します。この章では、回復状況についての説明、および適切な回復手順についての推奨事項を示します。
第 5 章：「バックアップ方法の選択」	第 3 章で説明したガイドラインと方針を実現するための各種の方法を説明します。
第 2 部	バックアップおよびリカバリー Recovery Manager
第 6 章：「Recovery Manager のスタート・ガイド」	Recovery Manager を使用する前に考慮すべき点と、Recovery Manager を短期間で使いこなせるようになるためのヒントを説明します。
第 7 章：「Recovery Manager の概要」	Recovery Manager ユーティリティの概要を示し、Recovery Manager の基本概念について説明します。
第 8 章：「Recovery Manager を使用してのバックアップおよび回復の実行」	Oracle の Recovery Manager ユーティリティを使用して、会社全体のバックアップおよび復元、回復作業を管理するための手順を説明します。
第 9 章：「Recovery Manager の使用例」	Recovery Manager のさまざまな使用例を紹介し、それぞれの場合に使用する Recovery Manager のコマンドを示します。
第 10 章：「Recovery Manager による表領域の Point-in-Time 回復」	Recovery Manager による表領域の Point-in-Time 回復の自動実行について、実際の手順に入る前に検討すべき計画と制限事項を説明します。
第 3 部	バックアップおよびリカバリーオペレーティング・システム
第 11 章：「オペレーティング・システムのバックアップの実行」	オペレーティング・システムのバックアップを実行するための手順を順を追って説明します。この章の内容は、以前は『Oracle7 Server 管理者ガイド』に記載されていました。
第 12 章：「オペレーティング・システムの回復の実行」	オペレーティング・システムの回復を実行するための手順を順を追って説明します。この章の内容は、以前は『Oracle7 Server 管理者ガイド』に記載されていました。

部 / 章	内容
第 13 章：「表領域の Point-in-Time 回復の実行」	表領域の Point-in-Time 回復を手動で実行するときの計画作成のガイドラインを示し、実際の手順を順を追って説明します。
付録 A: 「Recovery Manager のコマンド構文」	Recovery Manager の基本的なコマンド構文図を示します。

対象読者

このマニュアルは、Oracle データベース・システムのバックアップおよび復元、回復の各操作を管理する方を対象としています。

前提となる知識

このマニュアルは、リレーショナル・データベースの概念に詳しい読者を対象として書かれています。また、Oracle 製品の実行に使用するオペレーティング・システム環境に精通していることも前提になっています。

このマニュアルの使用法

このマニュアルを読む前に、必ず『Oracle8 Server 概要』の第 1 章「Oracle Server の基礎知識」を読んでください。このマニュアルで説明する内容の基礎となる、Oracle に関する概念と用語についての説明があります。『Oracle8 Server 概要』のその他の章には、Oracle のアーキテクチャと機能、およびその動作についてのさらに詳しい説明があります。

第I部

基本概念

バックアップの必要性について

この章では、バックアップおよび回復の基礎となるデータベースの概念を紹介し、データベース操作を成功させるために、なぜバックアップをとることが重要であるかを説明します。次の項を取り挙げます。

- バックアップとは？
- データベースの物理構造

バックアップとは？

簡単にいえば、データベースのバックアップとは、データの代理コピーのことです。元のデータが失われた場合、消失した情報 (Oracle データベースを構成する物理ファイル) を再構築するためにバックアップを使用できます。このコピーには、制御ファイル、アーカイブ・ログ、データ・ファイルなどのデータベースの重要な部分、つまりこの章を含むこのマニュアル全体を通じてこれから説明する構造が含まれています。メディアに障害が起きた場合には、データベースのバックアップがデータを正常に回復するための鍵になります。

バックアップがなぜ重要なのか？

次のような場合に失う収益を想像してみてください（得意先の不満の大きさは言うまでもありません）。カタログ会社または特急配送サービス、銀行、航空会社の実働データベースが、5 分か 10 分でも突然使用できなくなったとしたらどうなるでしょう。メディア障害のためにデータ・ファイルを失って、バックアップがないために復元も回復もできないとしたらどうでしょうか。業務上、大変重大な結果になることでしょう。業務を再開するためには、データを迅速に復元し回復する必要があります。この状況で成功するための鍵になるのは、バックアップおよび回復計画が正しく定義されているかどうかです。

バックアップをとる時期

バックアップ計画は、業務上のニーズに合わせて調整する必要があります。たとえば、ディスクに障害が起きたときにデータが失われてもよい場合は、バックアップを頻繁に実行する必要はありません。1 日 24 時間、週 7 日間、データベースが常に使用可能な状態になっていなければならない場合はどうでしょうか。この場合は、データベースのバックアップを頻繁にとる必要があります。バックアップの頻度と実行するバックアップのタイプは、その大部分が業務上のニーズによって決まります。

障害のタイプ

残念なことに、どのようなデータベース・システムでも、システム障害またはハードウェア障害が起こる可能性があります。最もよく発生するタイプの障害について、次に説明します。

文障害およびプロセス障害

文障害は、Oracle プログラム内での文の取扱いに論理的な誤りがあるとき（たとえば、文が有効な SQL 文でないとき）に起こります。文障害が起こると、その文の実行結果が Oracle により自動的に取り消され、制御がユーザーに戻されます。

プロセス障害は、異常な接続切離しまたはプロセスの終了など、Oracle にアクセスするユーザー・プロセスで起こる障害です。Oracle および他のユーザー・プロセスでは処理を続けることができますが、失敗したユーザー・プロセスでは処理を続けることはできません。

インスタンス障害

インスタンス障害が起こるのは、インスタンス（システム・グローバル領域およびバックグラウンド・プロセス）が処理を継続できなくなるような問題が発生したときです。停電などのハードウェア上の問題、またはオペレーティング・システムのクラッシュなどのソフトウェア上の問題が原因で、インスタンス障害が起こる場合があります。インスタンス障害が起こった場合、システム・グローバル領域のバッファ内のデータはデータ・ファイルには書き込まれていません。データベースがオープンしたときに、Oracle は、インスタンス障害から自動的に回復します。

ユーザー・エラーまたはアプリケーション・エラー

ユーザー・エラーの場合は、エラーが起こる前の時点までデータベースを回復する必要があります。たとえば、（所得税などのような）まだ必要なデータをユーザーが誤って表から削除してしまう場合があります。ユーザー・エラーから回復や、その他特殊な状況で必要となる回復条件を適用できるように、Oracle は正確な Point-in-Time 回復を提供します。たとえば、ユーザーが誤って表を削除した場合でも、データベースは、その表が削除される直前の状態に回復できます。

メディア（ディスク）障害

データベースの操作に必要なファイルの書込みまたは読取りをしようとしたときに、エラーが起こることがあります。これは、ディスク上の物理ファイルの読取りまたは書込みを行うときの物理的な問題なので、このようなエラーをディスク障害といいます。よくある例は、ディスク・ヘッドのクラッシュで、この場合はディスク・ドライブ上のすべてのファイルが失われてしまいます。この種のディスク障害により、データ・ファイル、REDO ログ・ファイル、制御ファイルなどさまざまな種類のファイルに影響が出ることがあります。また、該当のデータベース・インスタンスはそれ以降は正常に動作しなくなるので、システム・グローバル領域のデータベース・バッファ内のデータについてはデータ・ファイルへの永続的な書込みができなくなります。

データベースの物理構造

この項では、Oracle データベースの物理構造について、簡単に説明します。これらの項目については、このマニュアルの後の章でさらに詳しく説明します。

制御ファイル

各 Oracle データベースには、制御ファイルが1つあります。制御ファイルは非常に重要なデータ・ファイルで、ここにはデータベースの物理構造を指定するエントリがあり、回復時に使用されるデータベース一貫性情報が格納されています。たとえば、次のような種類の情報があります。

- データベース名
- データベースのデータ・ファイルおよび REDO ログ・ファイルの名前と位置
- データベース作成時のタイムスタンプ
- バックアップ情報 (Recovery Manager ユーティリティを使用する場合)

REDO ログと同様に、Oracle では、制御ファイルを保護するために制御ファイルもミラー化できるようになっています。

制御ファイルの使用方法

Oracle データベースのインスタンスをマウントするたびに、そのデータベースの制御ファイルが調べられ、データベース操作を実行するためにオープンしなければならないデータ・ファイルおよび REDO ログ・ファイルが識別されます。データベースの物理的な構成が変更されると（たとえば、新しいデータ・ファイルまたは REDO ログ・ファイルの作成時など）、そのデータベースの制御ファイルが自動的に修正されて変更が反映されます。

データベースに対して構造上の変更を加えた場合は、制御ファイルのバックアップをとる必要があります。

オンライン REDO ログ・ファイル

各 Oracle データベースには、複数の REDO ログ・ファイルがあります。あるデータベース用の REDO ログ・ファイル全体が、そのデータベースの REDO ログになります。REDO ログの機能は、データに対して行われたすべての変更を記録することです。たとえば障害のために、変更済みのデータをデータ・ファイルに永続的に書き込めなくなっている場合があります。このような場合、変更済みのデータを REDO ログから取得して、そのデータをデータ・ファイルに永続的に書き込むことにより、その間の作業が無駄にならずに済みます。

REDO ログ・ファイルは、データベースを障害から保護するために重要です。REDO ログ自体にかかわる障害からデータベースを保護するために、REDO ログを多重化できます。つまり、Oracle は、REDO ログの複数のコピーを異なるディスクに保持します。

オンラインの REDO ログのバックアップをとる必要も、復元する必要もありません。

REDO ログ・ファイルの使用方法

REDO ログ・ファイル内の情報は、データベースのデータ・ファイルにデータベースのデータの書き込みができなくなるようなシステム障害またはメディア障害が起きた場合に、データベースを回復するためだけに使用されます。

たとえば、不測の停電によりデータベースの動作が突然終了し、メモリー内のデータがデータ・ファイルに書き込まれず、データが失われたとします。この場合でも、電源の復旧後にデータベースをオープンしたときに、失われたデータをすべて回復できます。最新の REDO ログ・ファイルに入っている情報をデータベースのデータ・ファイルに適用することにより、Oracle は停電発生時の状態にデータベースを復元します。

回復操作中に REDO ログをデータ・ファイルおよび制御ファイルに適用するプロセスを、ロールフォワードといいます。

データ・ファイル

各 Oracle データベースには、1 つ以上の物理データ・ファイルがあります。データベースのデータ・ファイルには、データベースのデータがすべて格納されています。表および索引のようなデータベースの論理構造のデータは、データベースに割り当てられたデータ・ファイル内に物理的に格納されます。

データ・ファイルの特性は次のとおりです。

- 1 つのデータ・ファイルは 1 つのデータベースにだけ関係付けられる。
- データベース・ファイルには、データベースの領域が足りなくなったときに自動的に拡張できるように、一定の特性を設定できる。
- 1 つ以上のデータ・ファイルによって、表領域と呼ばれるデータベース記憶域の論理単位が形成される。

データ・ファイルの使用方法

データ・ファイル内のデータは、必要に応じて、通常のデータベース操作中に読み取られ、Oracle のメモリー・キャッシュに格納されます。たとえば、ユーザーがあるデータベースの表に入っているデータにアクセスしようとしているとします。要求された情報がそのデータベース用のメモリー・キャッシュにまだ入っていなければ、該当のデータ・ファイルから情報が読み取られ、メモリーに格納されます。

変更されたデータまたは新規データは、すぐにデータ・ファイルに書き込まれるとは限りません。ディスク・アクセスの回数を減らしパフォーマンスを向上させるために、Oracle の DBW0 バックグラウンド・プロセスの判断によってデータはメモリーにプールされ、適切なデータ・ファイルに 1 度にまとめて書き込まれます。

ロールバック・セグメント

各データベースには、1 つ以上のロールバック・セグメントがあります。ロールバック・セグメントは、データベースの一部で、トランザクションがロールバックされたときにトランザクションのアクションがここに記録されます。ロールバック・セグメントは、読取りの整合性を確保し、トランザクションをロールバックし、回復処理の一環としてデータベースをトランザクションと整合のとれた状態にするために使用します。

ロールバック・セグメントの使用方法

ロールバック・セグメントは、Oracle データベースの操作の中で多数の機能に使用されています。通常、データベースのロールバック・セグメントには、進行中のトランザクション（つまりコミットされていないトランザクション）により変更されたデータの、古い値が格納されています。とりわけ、データベースの回復時に、REDO ログからデータ・ファイルに適用された「コミットされていない」変更を「取り消す」ために、ロールバック・セグメント内の情報が使用されます。したがって、データベースの回復が必要な場合、ロールバック・セグメントを使ってすべてのコミットされていないデータをデータ・ファイルから削除した後は、データは整合のとれた状態になっています。

アーカイブ・ログ

アーカイブ・ログ・ファイルは、REDO の書込みにより満杯になり、アクティブではなくなった REDO ログで、バックアップ位置にコピーまたはアーカイブされます。オンライン REDO ログ・ファイルは、再使用の前にアーカイブできます。これにより、アーカイブ・ログが作成されます。アーカイブ REDO ログの有無は、データベースが使用中のモードによって決まります。

ARCHIVELOG	オンライン REDO ログ・ファイルは満杯になるとアーカイブされた後再使用されます。
NOARCHIVELOG	オンライン REDO ログ・ファイルは、満杯になってもアーカイブされることなく、再使用されます。

ARCHIVELOG モードの場合は、インスタンス障害およびディスク障害の両方からデータベースを完全に回復できます。データベースがオープンされ使用できる状態になっているときに、バックアップをとることもできます。ただし、アーカイブ REDO ログをメンテナンスするには、追加の管理操作が必要です。

一般的に、NOARCHIVELOG モードで運用されるデータベースに対する唯一の回復オプションは、データベース全体を復元することです。バックアップ・オプションは、データベースを完全にクローズした後で、そのバックアップを作成する方法だけです。アーカイブ REDO ログは作成されないので、データベース管理者は余分な作業をする必要はありません。

注意： NOARCHIVELOG モードでの操作の場合、一番最後にバックアップをとった時に現行だったオンライン・ログ・ファイルが上書きされる前でないでデータベースを回復できません。

バックアップ対象について

この章ではデータベースを構成する各要素と、主要なバックアップおよび回復の概念について説明します。取り挙げる項目は次のとおりです。

- オンライン REDO ログ
- アーカイブ REDO ログ
- データベースのアーカイブ・モード
- 制御ファイル
- バックアップのタイプ

発生した障害のタイプおよび障害の影響を受けた構成要素、回復に使用できるバックアップのタイプによって、回復のプロセスは異なります。ファイルが消失も損傷もしていない場合は、インスタンスを再起動するだけで回復できることもあります。

Oracle データベースには、障害が発生したときにデータを安全に保護するための構成要素がいくつかあります。この章では、これらの各構成要素について簡単に紹介するとともに、バックアップおよび回復の処理でどのように使用されるかについても説明します。

オンライン REDO ログ

Oracle データベースの各インスタンスには、データベースにインスタンス障害が起こった場合にデータベースを保護するために、オンライン REDO ログが1つずつ対応付けられています。オンライン REDO ログは事前に割り当てられた2つ以上のファイルからなり、これらのファイルにはデータベースに対して行ったすべての変更が発生順に格納されています。

注意： オンライン REDO ログのバックアップは行わないでください。詳細は、2-25 ページの「オンライン REDO ログのバックアップは作成しないでください」の項を参照してください。

オンライン REDO ログ・ファイルの内容

オンライン REDO ログ・ファイルには REDO エントリが書き込まれます。REDO エントリには、ロールバック・セグメントなど、データベースに対して行われた変更を完全に再構成するためのデータが記録されます。したがって、オンライン REDO ログはロールバック・データを保護する役目も果たします。

注意： REDO エントリには、ユーザー・アクションに対応させることができないデータベース変更の低レベル表現が格納されます。したがって、オンライン REDO ログ・ファイルの内容は絶対に編集も変更もしないでください。また、オンライン REDO ログ・ファイルを監査などのアプリケーション用に使用しないでください。

REDO エントリは「循環」方式で SGA の REDO ログ・バッファに一時格納され、Oracle のバックグラウンド・プロセスであるログ・ライター (LGWR) によって、オンライン REDO ログ・ファイルの1つに書き込まれます。トランザクションがコミットされるたびに、LGWR によりそのトランザクションの REDO エントリがシステム・グローバル領域 (SGA) の REDO ログ・バッファからオンライン REDO ログ・ファイルに書き込まれ、コミットされた各トランザクションの REDO エントリを識別するためのシステム変更番号 (SCN) が割り当てられます。

ただし、REDO エントリに対応するトランザクションがコミットされる前に、REDO エントリがオンライン REDO ログ・ファイルに書き込まれる場合があります。REDO ログ・バッファが満杯になった場合、または別のトランザクションがコミットされた場合は、LGWR は REDO ログ・バッファ内の REDO ログ・エントリをすべてオンライン REDO ログ・ファイルにフラッシュします。この場合、REDO エントリの一部はコミットされていないことがあります。

オンライン REDO ログ・ファイルへの書き込み方法

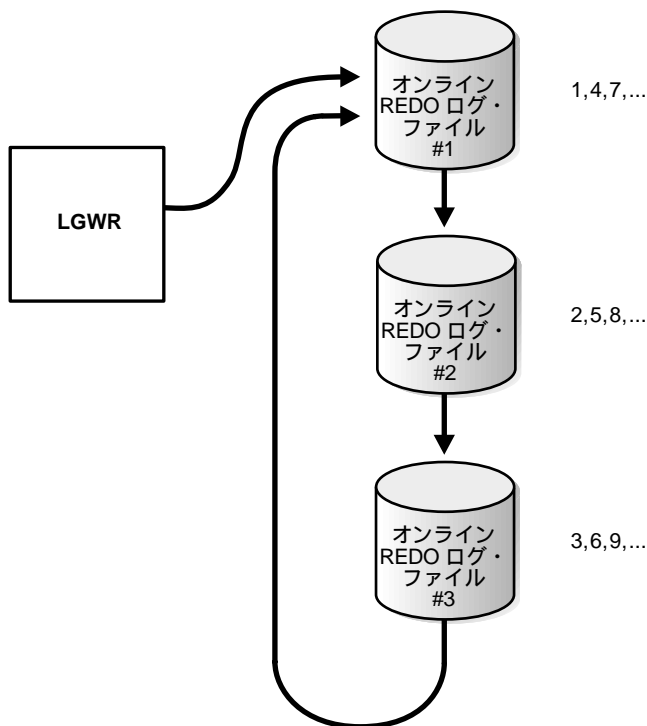
データベースのオンライン REDO ログ・ファイルは、複数のオンライン REDO ログ・ファイルで構成されます。Oracle では、一方のファイルがアーカイブ中（アーカイブを作成する場合）であっても常に他のファイルを書込みに使用できるようにするため、2つのファイルが必要です。

LGWR は、オンライン REDO ログ・ファイルに循環方式で書き込みます。現行のオンライン REDO ログ・ファイルが満杯になると、LGWR は、次の使用可能なオンライン REDO ログ・ファイルへの書き込みを開始します。使用可能な最後のオンライン REDO ログ・ファイルが満杯になると、LGWR は最初のオンライン REDO ログ・ファイルに戻ってそれに書き込むことにより、循環を繰り返します。図 2-1 に、オンライン REDO ログ・ファイルへの書き込みの循環を示します。各行の番号は、LGWR が各オンライン REDO ログ・ファイルへ書き込む順序を示しています。

満杯になったオンライン REDO ログ・ファイルは LGWR が再使用できる状態になりますが、いつ使用可能になるかはアーカイブの設定によって異なります。

- アーカイブが使用禁止の場合は、満杯のオンライン REDO ログ・ファイルは、そのファイルが関係しているチェックポイントの完了後に、使用可能になります。
- アーカイブが使用可能の場合は、満杯のオンライン REDO ログ・ファイルは、そのファイルが関係しているチェックポイントが完了し、さらにアーカイブされた後で、LGWR で使用可能になります。

図 2-1 LGWR によるオンライン REDO ログ・ファイルの循環使用



アクティブな（現行の）オンライン REDO ログ・ファイルとアクティブでないオンライン REDO ログ・ファイル

REDO ログ・バッファから書き込まれる REDO エントリを格納するために使用されるオンライン REDO ログ・ファイルは 1 度に 1 つだけです。LGWR によりアクティブに書き込みが行われているオンライン REDO ログ・ファイルを、現行のオンライン REDO ログ・ファイルといいます。

インスタンス回復に必要なオンライン REDO ログ・ファイルを、アクティブなオンライン REDO ログ・ファイルといいます。インスタンス回復に必要でないオンライン REDO ログ・ファイルを、アクティブでないオンライン REDO ログ・ファイルといいます。

アーカイブが使用可能になっている場合、ファイルの内容がアーカイブされるまで、アクティブなオンライン・ログ・ファイルの再使用も上書きもされません。アーカイブが使用禁止になっている場合は、最後のオンライン REDO ログ・ファイルが満杯になると、使用可能な最初のアクティブ・ファイルに上書きする形で書き込みが続行されます。

ログ・スイッチとログ順序番号

あるオンライン REDO ログ・ファイルへの書込みを Oracle が終了して、次のオンライン REDO ログ・ファイルへの書込みを開始する地点を、ログ・スイッチといいます。現行のオンライン REDO ログ・ファイルが満杯になり、次のオンライン REDO ログ・ファイルで書込みを続けることが必要になった時点で、常にログ・スイッチが発生します。データベース管理者が、強制的にログ・スイッチが発生させることもできます。

ログ・スイッチが発生するたびに、Oracle は各オンライン REDO ログ・ファイルに新しいログ順序番号を割り当て、LGWR がそのファイルへの書込みを開始します。オンライン REDO ログ・ファイルがアーカイブされても、アーカイブ REDO ログ・ファイルとしてそのログ順序番号が保持されます。循環後に再使用されるために再びオンライン REDO ログ・ファイルになると、使用可能な次のログ順序番号が割り当てられます。

各 REDO ログ・ファイル（オンラインおよびアーカイブ）は、そのログ順序番号に基づき、一意的に識別されます。インスタンス回復またはメディア回復時には、Oracle は必要なアーカイブ REDO ログ・ファイルおよびオンライン REDO ログ・ファイルのログ順序番号を使って、REDO ログ・ファイルを昇順で正しく適用します。

チェックポイント

Oracle のバックグラウンド・プロセス DBW0 により、SGA 内の変更されたすべてのデータベース・バッファ（コミットされたデータもコミットされていないデータも）がデータ・ファイルに書き込まれるときに、チェックポイントと呼ばれるイベントが発生します。チェックポイントが実行されるのは、次の理由からです。

- チェックポイントにより、メモリー内で頻繁に変更されるデータ・ブロックが定期的にデータ・ファイルに書き込まれます。DBW0 の最低使用頻度アルゴリズムのため、頻繁に変更されるデータ・ブロックは最低使用頻度ブロックとはみなされません。このため、使用頻度の高いファイルはチェックポイントが発生しない限りディスクに書き込まれません。
- チェックポイントが発生するまでに行ったデータベースに対する変更はすべてデータ・ファイルに記録されるので、インスタンス回復が必要になった場合でも、チェックポイントより前の REDO ログ・エントリをデータ・ファイルに適用する必要はありません。したがって、インスタンス回復の時間を短縮するという意味でも、チェックポイントは有効です。

チェックポイントによる多少のオーバーヘッドがありますが、アクティビティの中断や現行のトランザクションへの影響はありません。DBW0 はデータベース・バッファを連続的にディスクに書き込むので、チェックポイントでは必ずしも、多数のデータ・ブロックをすべて一度に書き込む必要はありません。チェックポイントが完了すると、前回のチェックポイント以降に変更されたすべてのデータ・ブロックが実際にディスクに書き込まれたことになります。

チェックポイントの発生は、満杯になったオンライン REDO ログ・ファイルがアーカイブされるかどうかには関係ありません。アーカイブが使用禁止になっている場合は、オンライン REDO ログ・ファイルに影響するチェックポイントが完了してからでないと、LGWR でその

オンライン REDO ログ・ファイルを再使用できません。アーカイブが使用可能になっている場合は、チェックポイントが完了し、しかも満杯になったオンライン REDO ログ・ファイルがアーカイブされてからでないと、LGWR でそのファイルを再使用できません。

チェックポイントには、データベースのすべてのデータ・ファイルについて発生するもの（これをデータベース・チェックポイントといいます）と、特定のデータ・ファイルについてだけ発生するものがあります。チェックポイントが発生する時期と、個々の状況でどのタイプのチェックポイントが発生するかについて、次に説明します。

- データベース・チェックポイントは、ログ・スイッチが発生するたびに自動的に発生します。前に発生したデータベース・チェックポイントの処理がまだ進行中であっても、ログ・スイッチにより発生したチェックポイントによって、進行中のチェックポイントが上書きされます。
- 初期化パラメータ `LOG_CHECKPOINT_INTERVAL` を設定することにより、事前に設定した数の REDO ログ・ブロックが前回のデータベース・チェックポイント後にディスクに書き込まれた時点で、次のデータベース・チェックポイントを強制的に発生させることができます。また、パラメータ `LOG_CHECKPOINT_TIMEOUT` を設定することにより、前回のデータベース・チェックポイントが開始されてから一定秒数が経過した後で、次のデータベース・チェックポイントを強制的に発生させることもできます。これらのパラメータは、REDO ログ・ファイルが非常に大きい場合に、ログ・スイッチとログ・スイッチの間に追加のチェックポイントを設定するために使用します。これらの初期化パラメータからの指示で開始されるデータベース・チェックポイントは、直前のチェックポイントの完了後に実行されます。
- オンラインの表領域のバックアップの開始が指示されると、バックアップ対象の表領域を構成するデータ・ファイルでだけ、チェックポイントが強制的に実行されます。以前に発生し、まだ処理中のチェックポイントがあっても、この時点で発生したチェックポイントによって上書きされます。この場合、チェックポイントはバックアップ中のデータ・ファイルに影響を与えるだけで、インスタンス回復に必要な REDO の量が減るわけではありません。
- 管理者が通常優先度または一時優先度で表領域をオフラインにすると、チェックポイントは、その表領域のオンライン・データ・ファイルだけに対して強制的に実行されます。
- データベース管理者が（通常終了または即時終了で）インスタンスを停止しようとする、Oracle はインスタンスが停止する前にデータベース・チェックポイントを強制的に完了させます。インスタンスの停止により強制的に実行されるデータベース・チェックポイントによって、その前に発生した処理中のチェックポイントが上書きされます。
- データベース管理者は、必要なときにいつでも要求を出して、データベース・チェックポイントを強制的に発生させることができます。先に実行されているチェックポイントがあっても、要求により強制実行されたチェックポイントによって上書きされます。

増分チェックポイント

増分チェックポイントによって、クラッシュからの回復やインスタンス回復（メディア回復は除く）のパフォーマンスが改善されます。増分チェックポイントでは、クラッシュからの回復またはインスタンス回復のための REDO スレッド（ログ）内の開始位置を記録します。このログの位置は、バッファ・キャッシュ内の最も古い使用済バッファによって決まります。増分チェックポイント情報のメンテナンスは、通常の処理の間、オーバーヘッドなしかまたは最低限のオーバーヘッドで定期的に行われます。

回復のパフォーマンスは、クラッシュ前にデータベースに書き込まれていなかったバッファの数のほぼ比例します。インスタンスのバッファ・キャッシュ内の使用済バッファ数の上限を指定するパラメータ `DB_BLOCK_MAX_DIRTY_TARGET` を設定することによって、クラッシュからの回復やインスタンス回復のパフォーマンスを上げたり下げたりできます。たとえば、バッファ・キャッシュが非常に大きい場合や、クラッシュやインスタンス回復の所要時間に厳しい制限がある場合、あるいはその両方の条件が適用される場合に、回復時間を長くしたり短くしたりできます。パラメータ値を小さくすると、書き込む必要のあるバッファの数が増えるため、通常の処理時のオーバーヘッドが大きくなります。一方、パラメータ値を小さくするほど、回復しなければならないブロックの数が減るため、回復のパフォーマンスは改善されます。

増分チェックポイントの情報は、Oracle8 サーバーによって、他のチェックポイント（ログ・スイッチ・チェックポイントやユーザー指定のチェックポイントなど）に影響を与えることなく自動的にメンテナンスされます。つまり、増分チェックポイントは、インスタンス内で発生する他のチェックポイントとは関係なく発生します。

増分チェックポイントは、シングル・インスタンス環境でもマルチ・インスタンス環境でも回復処理に有効です。

注意： Oracle Parallel Server を使用している場合は、上記以外の場合にもチェックポイントが発生します。詳細は、『Oracle8 Parallel Server 概要および管理』を参照してください。

チェックポイントの仕組み

チェックポイントが発生すると、チェックポイント・バックグラウンド・プロセス (CKPT) は、オンライン REDO ログ・ファイルに書き込む次のエントリの位置を記憶して、SGA 内の変更されたデータベース・バッファをディスク上のデータ・ファイルに書き込むように、データベース・ライター・バックグラウンド・プロセス (DBW0) に指示します。CKPT は、最新のチェックポイントを反映させるために、すべての制御ファイルおよびデータ・ファイルのヘッダーを更新します。

チェックポイントが発生していない場合は、DBW0 は最低使用頻度のデータベース・バッファだけをディスクに書き込んで、新しいデータに必要なバッファを解放します。しかし、チェックポイントの処理が進行中の場合は、DBW0 はそのチェックポイントと進行中のデータベース操作の両方として、データをデータ・ファイルに書き込みます。DBW0 は、チェックポイントとして一部の変更されたデータ・バッファを書き込んだ後で必要に応じて最低使用頻度のバッファを書き込み、さらにそのチェックポイントとして使用済みバッファを書き込む、という処理をチェックポイントが完了するまで続けます。

チェックポイントは、発生の要因に応じて、「通常」のチェックポイントの場合と「高速」のチェックポイントの場合があります。通常のチェックポイントの場合は、DBW0 はチェックポイントとして 1 度に少しずつデータ・バッファを書き込みます。高速チェックポイントの場合は、DBW0 はチェックポイントとして 1 度に大量のデータ・バッファを書き込みます。

したがって、通常のチェックポイントを完了させるには、高速チェックポイントの場合に比べて多くの I/O が必要です。高速チェックポイントは、必要な I/O が少ないので、非常に速く完了します。ただし、DBW0 が完了させる必要のあるデータベース処理が多い場合には、高速チェックポイントによりデータベース全体のパフォーマンスが低下することがあります。通常のチェックポイントが発生させるイベントには、ログ・スイッチおよび初期化パラメータにより設定されるチェックポイント間隔があります。高速チェックポイントが発生させるイベントには、オンラインの表領域のバックアップおよびインスタンスの停止、データベース管理者が強制的に発生させるチェックポイントがあります。

チェックポイントが完了するまでは、データベース障害が起きてチェックポイントが中断されインスタンス回復が必要になった場合に備えて、前回のチェックポイントの後で書き込まれたすべてのオンライン REDO ログ・ファイルが必要です。また、チェックポイントが完了していないために LGWR がオンライン REDO ログ・ファイルにアクセスして書き込むことができない場合は、そのチェックポイントが完了しオンライン REDO ログ・ファイルが使用可能になるまで、データベース操作が一時保留されます。この場合、できるだけ速く処理が完了するように、通常のチェックポイントであっても高速チェックポイントに切り替わります。

たとえば、オンライン REDO ログ・ファイルが2つだけ使用されていて、LGWR が別のログ・スイッチを要求した場合、前のログ・スイッチのチェックポイントが完了するまでは、LGWR は最初のオンライン REDO ログ・ファイルを使用できません。

注意： Oracle Parallel Server 構成を使用している場合、チェックポイントの一部としてデータ・ファイルと制御ファイルに記録される情報は異なります。詳細は、『Oracle8 Parallel Server 概要および管理』を参照してください。

初期化パラメータ LOG_CHECKPOINTS_TO_ALERT を設定することにより、必要な頻度でチェックポイントが発生しているかどうかを判断できます。このパラメータのデフォルト値は NO で、この場合チェックポイントはログに記録されません。このパラメータを YES に設定すると、各チェックポイントについての情報が ALERT ファイルに記録されます。

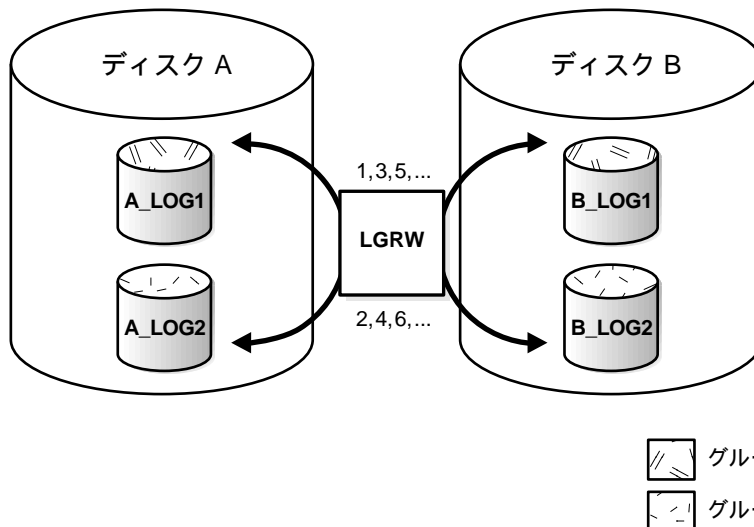
多重オンライン REDO ログ・ファイル

Oracle には、オンライン REDO ログ・ファイルが損傷した場合の保護策として、インスタンスのオンライン REDO ログ・ファイルを多重化する機能が備わっています。多重オンライン REDO ログ・ファイルを使うと、LGWR は、同じ REDO ログ情報を複数のまったく同じオンライン REDO ログ・ファイルに同時に書き込むので、単一ポイントのオンライン REDO ログ障害が全体に及ぶのを防ぐことができます。

注意： REDO ログ・ファイルは多重化しておくことをお勧めします。ログ・ファイル情報が失われた状態で回復が必要になると、重大な事態を招きかねません。

図 2-2 に、多重 (2 セット構成の) オンライン REDO ログ・ファイルを示します。

図 2-2 多重オンライン REDO ログ・ファイル



相互に対応する複数のオンライン REDO ログ・ファイルをグループといいます。グループ内の各オンライン REDO ログ・ファイルをメンバーといいます。LGWR により割り当てられた同一のログ順序番号によって指示された時点で、同じグループ内のメンバーはすべて同時にアクティブになります (LGWR により同時に書き込まれます)。多重オンライン REDO ログを使う場合、グループ内の各メンバーは同一サイズにする必要があります。

多重オンライン REDO ログの仕組み

グループ内のメンバーが 1 つか複数にかかわらず、LGWR は常にグループのすべてのメンバーを処理します。たとえば、ログ・スイッチの後、LGWR は次のグループのすべてのメンバーに同時に書き込みます。あるグループのメンバーと別のグループのメンバーに LGWR が同時に書き込むことはありません。

特定のオンライン REDO ログ・メンバーが使用できない場合、LGWR はファイルが使用できない理由に応じて、異なる対応処理を実行します。

- LGWR が、(ログ・スイッチの発生時またはグループへの書き込み処理中に) グループ内の少なくとも 1 つのメンバーに正常に書き込むことができる場合は、グループのアクセス可能メンバーへの書き込みは通常どおり続行されます。LGWR は単にグループの使用可能なメンバーに書き込み、使用できないメンバーは無視します。
- 次のグループをアーカイブする必要があるために、ログ・スイッチの発生時に LGWR がそのグループにアクセスできない場合は、そのグループが使用可能になるまで (つまりグループのアーカイブが終了するまで)、データベース操作が一時的に停止します。
- 1 つ以上のディスク障害が起こったためログ・スイッチの発生時に LGWR が次のグループのすべてのメンバーにアクセスできない状態になった場合には、エラーが戻り、問題

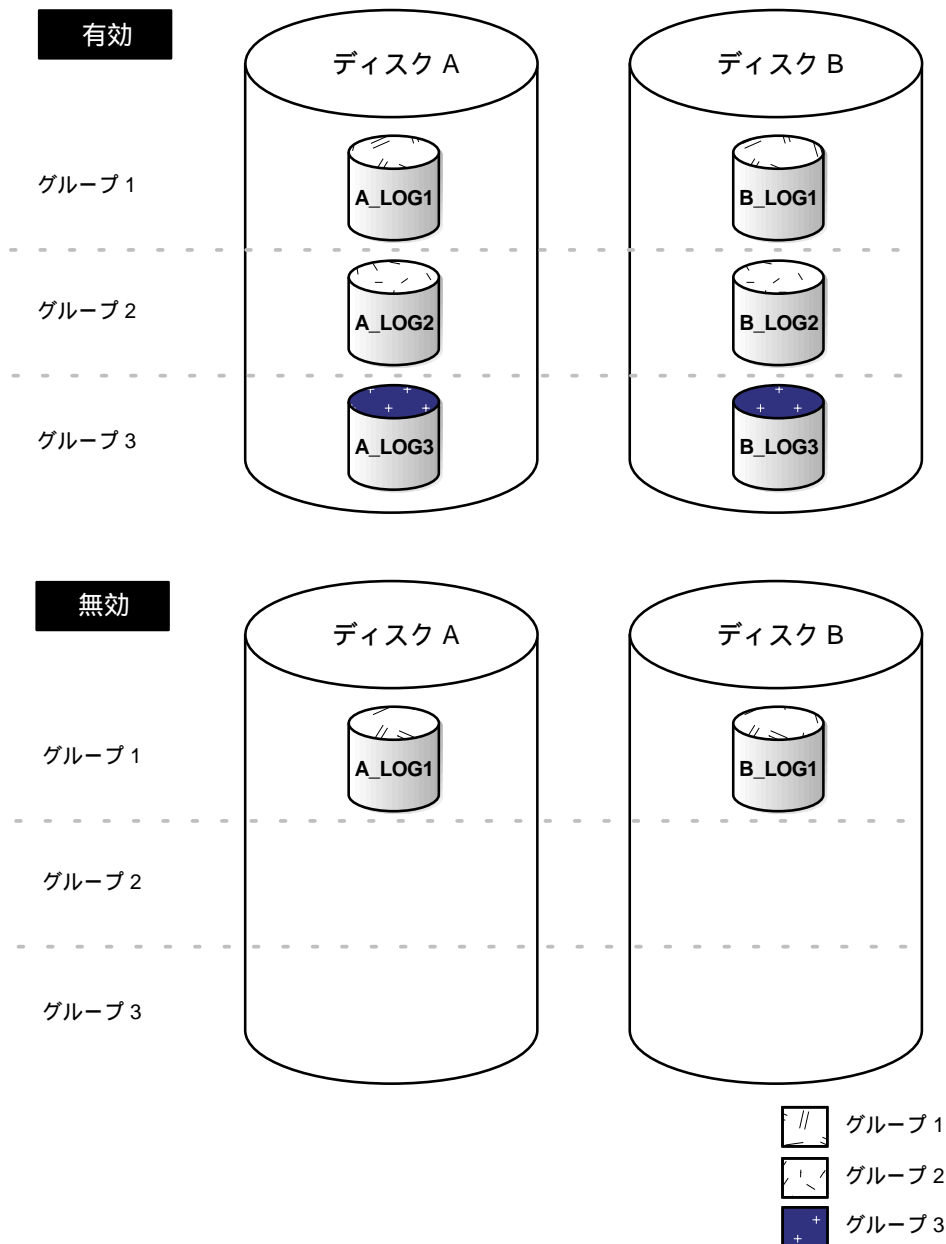
のデータベース・インスタンスがただちに停止します。この場合、失われたオンライン REDO ログ・ファイルを回復するために、データベースのメディア回復が必要になることがあります。ただし、失われたログ（この例では現行ログではない）を過ぎた時点までチェックポイントが進んでいる場合は、メディア回復は必要ありません。この場合は、アクセスできないログ・グループを削除するだけで済みます。ログがアーカイブされていない場合は、アーカイブを使用禁止にしてから、`ALTER DATABASE CLEAR UNARCHIVED LOG` コマンドを使ってログを削除します。

- 書き込み中に、LGWR がグループ内のすべてのメンバーに突然アクセスできなくなった場合は、エラーが戻り、データベース・インスタンスがただちに停止します。この場合、失われたオンライン REDO ログ・ファイルを回復するために、データベースのメディア回復が必要になることがあります。ログのあるメディアが実際に失われているわけではない場合（たとえば、ログのあるドライブが誤ってオフにされている場合など）には、メディア回復は必要ないことがあります。この場合は、ドライブをオンにしてインスタンス回復を行うことにより、すべてが解決します。

LGWR からグループ内のメンバーへの書き込みができない場合は、そのメンバーに STALE のマークが付けられ、LGWR トレース・ファイルとデータベースの ALERT ファイルに、アクセス不能なファイルの問題が起きたことを示すエラー・メッセージが書き込まれます。

単一ポイントのオンライン REDO ログの障害からデータベースを保護するには、多重オンライン REDO ログを完全に対称にする必要があります。つまり、オンライン REDO ログのすべてのグループが同数のメンバーを持つように構成しなければなりません。ただし、Oracle では、多重オンライン REDO ログを対称にすることは必須ではありません。たとえば、あるグループにはメンバーが 1 つだけで、他のグループにはメンバーが 2 つあってもかまいません。Oracle では上記の処理によって、あるオンライン REDO ログ・メンバーには一時的に影響しても他のメンバーには影響しないような状況（たとえばディスク障害）に対応できます。インスタンスのオンライン REDO ログ（多重化した、またはしていない）の必要条件は、少なくとも 2 つのグループで構成されているということだけです。図 2-3 に、多重オンライン REDO ログの有効な構成と無効な構成を示します。2 番目の構成には 1 つのグループしかないため、無効です。

図 2-3 多重オンライン REDO ログの有効な構成と無効な構成



オンライン REDO ログのスレッドと Oracle Parallel Server

各データベース・インスタンスには、専用のオンライン REDO ログ・グループがあります。このようなオンライン REDO ログ・グループ（多重化した、またはしていない）を、インスタンスのオンライン REDO の「スレッド」といいます。通常の構成では、1つのデータベース・インスタンスだけが Oracle データベースにアクセスするので、スレッドは1つだけです。しかし、Oracle Parallel Server を実行している場合は、複数のインスタンスが同時に1つのデータベースにアクセスします。このタイプの構成では、各インスタンスに専用のスレッドがあります。

このマニュアルでは、Oracle Parallel Server を使っていない場合のオンライン REDO ログの構成および管理方法について説明します。すべての説明やコマンドの例で、スレッド数を1と仮定しています。Oracle Parallel Server でのオンライン REDO ログの構成の詳細は、『Oracle8 Parallel Server 概要および管理』を参照してください。

アーカイブ REDO ログ

Oracle では、必要に応じて満杯のオンライン REDO ログ・ファイル・グループをアーカイブできます。アーカイブすると、アーカイブ (オフライン) REDO ログが作成されます。満杯のグループをアーカイブすることには、データベースのバックアップおよび回復のオプションに関して、次のような2つの大きな利点があります。

- オンライン REDO ログ・ファイルとアーカイブ REDO ログ・ファイルの両方が組み込まれているデータベースのバックアップでは、オペレーティング・システムの障害またはディスク障害の発生時にコミット済みのトランザクションをすべて回復できる。
- アーカイブ・ログが永続的に保管されていれば、データベースがオープンされシステムが通常に使用されている間にとられたバックアップを使用できる。

満杯のオンライン REDO ログ・ファイルのグループのアーカイブを使用可能にするかどうかは、データベースに対して実行中のアプリケーションに要求される可用性と信頼性によって決まります。ディスク障害が起きた場合にデータベース内のデータを1つも失わないようにするには、ARCHIVELOG モードを使う必要があります。ただし、満杯のオンライン REDO ログ・ファイルをアーカイブする場合は、データベース管理者が行う必要がある管理作業が増えます。

アーカイブの仕組み

アーカイブをどのように構成するかに応じて、オプションの Oracle のバックグラウンド・プロセス ARCH(自動アーカイブを使う場合)か、または SQL 文を発行することによってグループを手作業でアーカイブするユーザー・プロセスのどちらかの仕組みによって、REDO ログ・グループのアーカイブが実行されます。

注意： 説明を簡単にするために、以後この項では、自動アーカイブを使用可能にして、ARCH バックグラウンド・プロセスで満杯のオンライン REDO ログ・グループをアーカイブすることを前提にします。

ARCH がグループをアーカイブできるようになるのは、グループがアクティブでなくなり、次のグループへのログ・スイッチが完了したときです。ARCH プロセスは、グループのアーカイブを完了するために、必要に応じてグループ内のどのメンバーにでもアクセスできます。ARCH がグループのあるメンバーをオープンまたは読み込もうとしたときにそのメンバーに（たとえばディスク障害などで）アクセスできなかった場合は、アーカイブに使用可能なメンバーが見つかるまで、ARCH は自動的にグループ内の別のメンバーに次々にアクセスを試みます。ARCH がグループ内のメンバーをアーカイブしていて、アーカイブ処理の途中でメンバー内の情報が無効と判明した場合、またはディスク障害が起こった場合は、ARCH は、自動的にグループの別のメンバーに切り替えて、そのグループのアーカイブを中断した箇所から継続します。

ARCH がオンライン REDO ログ・ファイルのグループのアーカイブを完了するまで、LGWR はそのグループを再使用できません。この制約は、アーカイブが済んでいないグループが LGWR によって誤って上書きされないようにするために重要です。上書きされてしまうと、データベース回復時に後続のすべてのアーカイブ REDO ログ・ファイルが使用できなくなります。

注意： 1 つのアーカイブ・ログを失うと、後続のすべてのアーカイブ・ログが使用できなくなります。

アーカイブを行うには、パラメータ LOG_ARCHIVE_DEST を設定してアーカイブの宛先を指定します。この宛先は通常、データベースのデータ・ファイルおよびオンライン REDO ログファイル、制御ファイルを保管するディスク・ドライブとは別の記憶デバイスです。一般に、アーカイブの宛先には同じデータベース・サーバーの別のディスク・ドライブを指定します。この方法により、アーカイブはインスタンスにより要求される他のファイルと競合することなく迅速に完了し、LGWR がそのグループを使用できるようになります。アーカイブ REDO ログ・ファイル（およびそれに対応するデータベースのバックアップ）は、テープなどの安価なオフライン記憶メディアに永続的に移して、データベース・サーバーとは別の安全な場所に保管しておくことをお勧めします。

ログ・スイッチの時点、つまりこれ以上 REDO ログに書き込む情報がなくなった時点で、データベースの制御ファイル内にレコードが作成されます。各レコードにはスレッド番号およびログ順序番号、グループの低 SCN、アーカイブ・ログ・ファイルに後続する SCN が格

納されます。この情報は、Parallel Server 構成でデータベース回復を行うときに、REDO ログ・ファイルを自動的に適用するために使用されます。

ログのアーカイブが正常に完了した場合にも常に、レコードが作成されます。このレコードには、アーカイブ・ログの名前および低 SCN、高 SCN、ログ順序番号が格納されます。

関連項目：詳細は、『Oracle8 Parallel Server 概要および管理』を参照してください。

アーカイブ REDO ログ・ファイルの内容

アーカイブ REDO ログ・ファイルは、オンライン REDO ログ・グループを構成する満杯になった同一のメンバーを単にコピーしたものです。したがって、アーカイブ REDO ログ・ファイルには、グループがアーカイブされた時点でそのグループの同一のメンバー内にあった REDO エントリが入っています。アーカイブ REDO ログ・ファイルには、グループのログ順序番号も格納されています。

アーカイブが使用可能になっていると、オンライン REDO ログ・グループのアーカイブが完了するまでは、LGWR はそのグループを再使用できません。したがって、アーカイブ REDO ログには、アーカイブが使用可能になった後で作成されたすべてのグループ（ログ順序番号により一意に識別される）のコピーが確実に格納されます。

アーカイブ REDO ログの多重化

オンライン REDO ログ・グループをアーカイブ用に 2 つのコピーとして書き込むことができます。アーカイブ REDO ログをメディアの故障から保護する信頼性を高めるために、多数のサイトでこの方法を選択することが予想されます。

LOG_ARCHIVE_DUPLEX_DEST パラメータは、2 番目の位置を決定します。REDO ログ・ファイルをアーカイブする時点で、LOG_ARCHIVE_DEST と LOG_ARCHIVE_DUPLEX_DEST の両方にアーカイブされます。

LOG_ARCHIVE_MIN_SUCCEED_DEST パラメータが設定されている場合は、このパラメータによって、REDO ログ・グループを正常にアーカイブする必要があるアーカイブ・ログの宛先件数が決まります。詳細は、『Oracle8 Server リファレンス・マニュアル』を参照してください。

データベースのアーカイブ・モード

データベースの運用には、NOARCHIVELOG モード（メディア回復使用禁止）と ARCHIVELOG モード（メディア回復使用可能）の 2 つのモードがあります。

NOARCHIVELOG モード（メディア回復使用禁止）

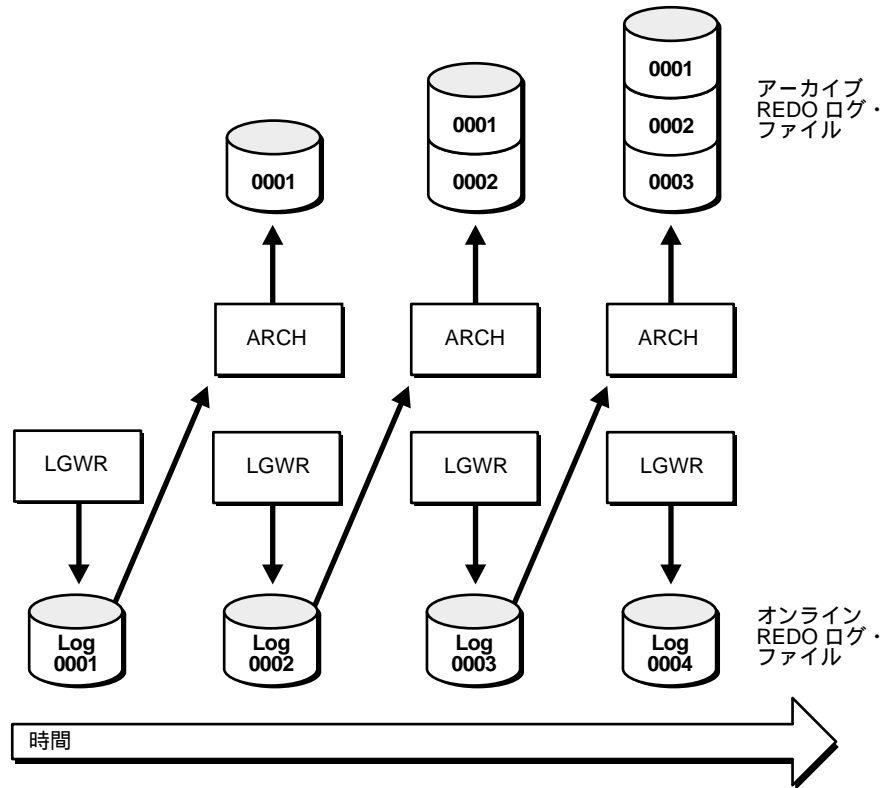
データベースを NOARCHIVELOG モードで使うと、オンライン REDO ログのアーカイブは使用禁止になります。データベースの制御ファイル内の情報は、満杯のグループのアーカイブが必要ないことを示しています。したがって、満杯のグループがアクティブでなくなり、ログ・スイッチ発生時のチェックポイントが完了した後は、LGWR がそのグループを再使用できるようになります。

NOARCHIVELOG モードでは、データベースをインスタンス障害からは保護できますが、ディスク（メディア）障害からは保護できません。インスタンス回復に使用できるのは、データベースに対して行われた最新の変更、つまりオンライン REDO ログのグループ内に格納されているものだけです。

ARCHIVELOG モード（メディア回復使用可能）

Oracle データベースを ARCHIVELOG モードで運用している場合は、オンライン REDO ログのアーカイブが使用可能になります。データベース制御ファイル内の情報は、満杯のオンライン REDO ログ・ファイルのグループは、グループのアーカイブが済むまで LGWR では再使用できないことを指示しています。ログ・スイッチが発生する（グループがアクティブでなくなる）と、アーカイブを実行中のプロセスはただちに満杯のグループを使用できるようになります。つまり、アーカイブを実行中のプロセスは、ログ・スイッチのチェックポイントの完了を待つ必要がなく、アクティブでないグループにすぐにアクセスしてアーカイブできます。図 2-4 に、ARCHIVELOG モードでデータベースのオンライン REDO ログ・ファイルがどのように使われるか、また、満杯のグループをアーカイブするプロセス（この図では ARCH）でアーカイブ REDO ログがどのように生成されるかを示します。

図 2-4 ARCHIVELOG モードでのオンライン REDO ログ・ファイルの使用方法



ARCHIVELOG モードでは、データベースに対する変更はすべてアーカイブ REDO ログに永続的に保存されるため、インスタンス障害からディスク障害から完全に回復できます。

自動アーカイブと ARCH(アーカイバ) バックグラウンド・プロセス

オンライン REDO ログ・ファイルのグループがアクティブでなくなったときに、追加バックグラウンド・プロセスであるアーカイバ (ARCH) により自動的にそれらのグループをアーカイブするように、インスタンスを構成できます。したがって、自動アーカイブを利用すれば、データベース管理者は満杯のグループを追跡して手動でアーカイブする作業から解放されます。この点だけを見ても、アーカイブ REDO ログを備えたほとんどのデータベース・システムに自動アーカイブが適していることは明かです。

LOG_ARCHIVE_START 初期化パラメータを設定することによってインスタンスの起動時に自動アーカイブを要求すると、Oracle はインスタンスの起動中に ARCH を起動します。自動アーカイブを要求しなければ、インスタンスの起動時に ARCH は起動されません。

ただし、データベース管理者は、対話形式でいつでも自動アーカイブを起動または停止できます。インスタンスの起動時に自動アーカイブが起動するように指定しないで、後で管理者が自動アーカイブを起動した場合は、ARCH バックグラウンド・プロセスが作成されます。この場合、自動アーカイブを一時的にオフにして再びオンにしても、インスタンスの実行中は ARCH は有効なままになっています。

ARCH は必ず、最小の順序番号から順にグループをアーカイブします。また、満杯のグループがアクティブでなくなった時点で、ARCH はそれらのグループを自動的にアーカイブします。各自動アーカイブの記録、ARCH プロセスによって ARCH トレース・ファイルに書き込まれます。各エントリには、アーカイブの開始時刻と終了時刻が示されます。またこの情報は、データベースの制御ファイルにも記録され、V\$ARCHIVED_LOG ビューを問い合わせることにより表示できます。

ARCH がグループをアーカイブしようとしてエラーを検出した場合（たとえば宛先が無効または満杯の状態など）でも、ARCH はそのグループのアーカイブを続けようとします。エラーは、ARCH トレース・ファイルおよび ALERT ファイルにも書き込まれます。問題が解決されないと、最終的にはすべてのオンライン REDO ログ・グループが満杯になり、しかもアーカイブされず、LGWR で使用できるグループがなくなるためシステムが停止します。したがって、問題が検出された場合は、その問題を解決して（たとえばアーカイブ先を変更するなど）ARCH がアーカイブを続行できるようにするか、問題が解決されるまでグループを手動でアーカイブする必要があります。

手動アーカイブ

データベースを ARCHIVELOG モードで運用している場合、自動アーカイブが使用可能か使用禁止かにかかわらず、データベース管理者はアクティブでなくなったオンライン REDO ログ・ファイルの満杯のグループを必要に応じて手動でアーカイブできます。自動アーカイブが使用禁止の場合は、データベース管理者には満杯のすべてのグループをアーカイブする責任があります。

自動アーカイブの場合はグループがアクティブでなくなりアーカイブに使用できる状態になったかどうかをデータベース管理者が監視する必要がないので、ほとんどのシステムでは自動アーカイブが選択されます。さらに、自動アーカイブが使用禁止になっている場合に手動アーカイブが適切な時期に実行されていないと、アクティブでないグループが再使用可能になるまで LGWR が待たされるたびに、データベースの動作が一時中断されます。

手動アーカイブ・オプションは、データベース管理者が次の操作を実行できるように用意されています。

- 問題が発生して（たとえば、アーカイブ REDO ログの宛先として指定されたオフライン記憶デバイスに障害が起きたり満杯になった場合など）自動アーカイブが停止した場合に、グループをアーカイブする。
- 標準外の方法でグループをアーカイブする（たとえば、あるグループをあるオフライン記憶デバイスにアーカイブし、次のグループを別のオフライン記憶デバイスにアーカイブする場合など）。
- 元のアーカイブ済みグループを失ったり損傷した場合に、グループを再アーカイブする。

グループを手動でアーカイブする場合は、グループをアーカイブする文を発行したユーザー・プロセスが、実際にそのグループのアーカイブ処理を実行します。該当のインスタンス用の ARCH バックグラウンド・プロセスがある場合でも、そのグループのオンライン REDO ログ・ファイルのアーカイブはユーザー・プロセスにより実行されます。

制御ファイル

データベースの制御ファイルは、データベースの正常な起動と操作に必要な小さいバイナリ・ファイルです。制御ファイルは、データベースの使用中に Oracle により絶えず更新されます。したがって、データベースがマウントされている間は、制御ファイルは常に書込み可能になっていなければなりません。なんらかの理由で制御ファイルにアクセスできなくなると、データベースは正しく動作しなくなります。

各制御ファイルは、それぞれ 1 つの Oracle データベースだけに対応付けられます。

制御ファイルの内容

制御ファイルには、それに対応付けられたデータベースについての情報が格納されています。この情報は、データベースの起動時および通常の運用時にデータベースにインスタンスを介してアクセスできるようにするために必要です。制御ファイルの情報は、Oracle だけが更新できます。データベース管理者やエンド・ユーザーがデータベースの制御ファイルを編集することはできません。

制御ファイルに格納されている主な情報は次のとおりです。

- データベース名
- データベース作成時のタイムスタンプ
- 対応付けられたデータ・ファイルおよびオンライン REDO ログ・ファイルの名前と位置
- 現行のログ順序番号
- チェックポイント情報
- バックアップ情報 (Recovery Manager を使用する場合)

データベース名およびタイムスタンプは、データベースの作成時に作成されます。データベースの名前には、初期化パラメータ `DB_NAME` に指定された名前、または `CREATE DATABASE` 文で使用されている名前が適用されます。

データ・ファイルまたはオンライン REDO ログ・ファイルがデータベース内で追加または改名、削除されるたびに、この物理構造の変化を反映して制御ファイルが更新されます。これらの変更は次の目的のために記録されます。

- データベースの起動時にオープンするデータ・ファイルおよびオンライン REDO ログ・ファイルを Oracle が識別できるようにする。
- データベースの回復が必要になったときに必要なファイルまたは使用可能なファイルを Oracle が識別できるようにする。

したがって、データベースの物理構造を変更した場合は、ただちに制御ファイルの新しいバックアップを作成する必要があります。

制御ファイルには、チェックポイントについての情報も記録されます。チェックポイントが開始されると、オンライン REDO ログに入れる必要がある次のエントリを記憶しておくための情報が、制御ファイルに記録されます。この情報は、データベースの回復時に、そのオンライン REDO ログ・グループでそれ以前に記録された REDO エントリはすべてデータ・ファイルに書き込み済みなのでデータベース回復に使う必要がないことを、Oracle に通知するために使用されます。

多重制御ファイル

オンライン REDO ログ・ファイルの場合と同様に、Oracle では、まったく同じ複数の制御ファイルを同時にオープンし、同じデータベースについての情報を書き込むことができます。1 つのデータベースについて複数の制御ファイルを異なるディスクに格納しておくことにより、単一ポイントの障害から制御ファイルを保護できます。制御ファイルが格納されている単一のディスクがクラッシュした場合、Oracle がこの破損した制御ファイルにアクセスしようとする、現行のインスタンスにエラーが発生します。しかし制御ファイルが多重化されている場合は、現行の制御ファイルのコピーが別のディスクにあります。このため、データベースを回復しなくても、インスタンスを簡単に再起動できます。

データベースの制御ファイルのコピーがすべて永久に失われると重大な事態を招くため、その防止対策は重要です。制御ファイルが運用中に障害を起こすと、そのインスタンスは異常終了し、メディア回復が必要になります。制御ファイルを多重化していないと、回復作業はさらに複雑になります。したがって、各データベースについて多重制御ファイルを使用し、各コピーを別々の物理ディスクに格納しておくことをお勧めします。

バックアップのタイプ

ここでは、次の各タイプのバックアップを定義し、説明します。

- データベース全体のバックアップ
 - 一貫性のあるデータベース全体のバックアップ
 - 一貫性のないデータベース全体のバックアップ
- 表領域のバックアップ
- データ・ファイルのバックアップ
- 制御ファイルのバックアップ
- アーカイブ・ログのバックアップ

論理バックアップはエクスポートともいいます。詳細は、『Oracle8 Server ユーティリティ』を参照してください。

データベース全体のバックアップ

データベース全体のバックアップとは、制御ファイル、およびそのデータベースに所属するすべてのデータベース・ファイルを含むバックアップです。データベース全体のバックアップは、データベース管理者が実行する最も一般的なバックアップです。データベース全体のバックアップを（または他のどのタイプのバックアップでも）実行する前に、組織のバックアップ計画について詳しく知っておく必要があります。特に、ARCHIVELOG モードの場合と NOARCHIVELOG モードの場合のバックアップから生じる結果の違いについて注意しなければなりません。

データベース全体のバックアップでは、そのデータベースに所属しているデータ・ファイルと制御ファイルがすべてバックアップをとられます（オンライン REDO ログのバックアップを作成する必要はありません）。

データベース全体のバックアップでは、データベースを特定のアーカイブ・モードにしておく必要はありません。データベースが ARCHIVELOG モードになっているか NOARCHIVELOG モードになっているかに関係なく、データベース全体のバックアップをとることができます。

データベースが ARCHIVELOG モードになっている場合は、データベースがオープン状態でもクローズ状態でもバックアップをとることができます。NOARCHIVELOG モードで運用中のデータベースのバックアップをとるには、その前に正しく停止（たとえば、即時オプションまたは標準オプションを使って停止するなど）することによりデータベースをクローズする必要があります。NOARCHIVELOG モードで運用中の場合、正しく停止せずにとられたデータベースのバックアップは無効です。この場合、バックアップ後のファイルはある時点に対して整合がとれておらず、しかもデータベースが NOARCHIVELOG モードになっているので、データベースの整合をとるために使用できるログもありません。NOARCHIVELOG モードで運用されていて異常終了により停止したデータベースからとられたバックアップは

回復に使用できないので、Recovery Manager ではこのようなデータベースのバックアップは作成できません。

データベースに対応付けられたパラメータ・ファイルや、各インスタンスのパスワード・ファイル（インスタンスがパスワード・ファイルを使用する場合）の内容を変更したときには、必ずバックアップをとってください。

注意： ARCHIVELOG モードの場合は、複数の異なる時点に作成されたデータ・ファイルのバックアップを使って、データベース全体のバックアップを作成できます。たとえば、データベースが7つの表領域で構成され、毎晩、別の表領域のバックアップが作成される場合（制御ファイルのバックアップは毎晩作成されます）、7日後にデータベース内のすべての表領域と制御ファイルのバックアップが作成されることになります。これはデータベース全体のバックアップと見なすことができます。

データベース全体のバックアップには、一貫性バックアップと非一貫性バックアップの2種類があります。

一貫性のあるデータベース全体のバックアップ

一貫性のあるデータベース全体のバックアップとは、すべてのファイル（データ・ファイルおよび制御ファイル）が同一の時点で整合がとれている（たとえば同一のシステム変更番号 (SCN) にチェックポイントが設定されている）バックアップです。一貫性バックアップで、古いタイムスタンプ (SCN) を持つことが許されている表領域は読取り専用でオフラインの普通の表領域だけです。このような表領域は同じバックアップ内の他のデータ・ファイルと整合がとれています。

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得られる一連のファイルは、データの整合がとれているので、REDO ログを適用せずにオープンできます。復元したデータ・ファイル内のデータを正しくするための作業は必要ありません。

一貫性のあるデータベース全体のバックアップは、NOARCHIVELOG モードで実行しているデータベースの場合に唯一の有効なバックアップのオプションです。一貫性のあるデータベース全体のバックアップをとる唯一の方法は、データベースを正しく停止して、データベースがクローズしているときにバックアップをとることです。

一貫性のあるデータベースのバックアップを現行の状態にする（または後の時点にバックアップを合せる）には、バックアップに REDO を適用します。また、Recovery Manager を使ってバックアップを実行している場合には、増分バックアップと REDO の組合せを適用するオプションを使用することもできます。REDO は、アーカイブ・ログおよびオンライン・ログ（現時点まで回復する場合）に入っています。

データベースが正しく停止されなかった場合（たとえば、インスタンス障害が起きたり、SHUTDOWN ABORT 文が発行された場合）はほとんど、データベースのデータ・ファイルが不整合な状態になっています。

警告： 一貫性のあるデータベース全体のバックアップ時に作成されたバックアップ制御ファイルは、回復処理を行わずにデータベース全体のバックアップを復元する時にだけ使用してください。回復処理を行って現行の制御ファイルを使用する場合は、古い制御ファイルを復元しないでください。

一貫性のないデータベース全体のバックアップ

データベースを常時（1 日 24 時間、週 7 日間）稼働状態にしておくことが必要な状況では、一貫性のないデータベース全体のバックアップが必要となる場合が多くなります。オープン状態のデータベースのバックアップは、バックアップの処理中にデータベースの一部（したがって、データベース内のデータ・ファイル）が変更されてディスクに書き込まれるので、整合性がなくなります。オープン・バックアップを実行するには、データベースは ARCHIVELOG モードになっていなければなりません。

オープン・バックアップを実行するかどうかは、データの可用性の条件だけによって決まります。バックアップ対象のデータが常に使用可能な状態になっていなければならない場合は、オープン・バックアップしか選択できません。

システム・クラッシュ障害または異常終了停止の後でデータベースをバックアップした場合も、非一貫性バックアップが作成されます。データベースが ARCHIVELOG モードで実行されている場合にだけ、バックアップの整合性を確保するためにすべてのログが使用できるので、非一貫性バックアップが有効になります。

注意： 整合性のない状態でクローズされたデータベースのバックアップは、とらないようにしてください。

オープン（「ホット」ともいう）・バックアップまたは整合性のない状態でクローズされたデータベースのバックアップをとった後は、現行のオンライン・ログ・ファイルを必ずアーカイブしなければなりません。現行のオンライン・ログは、SQL コマンドの ALTER SYSTEM ARCHIVE ALL を発行してアーカイブします。現行のログ・ファイルをアーカイブした後は、バックアップの開始以降に作成されたすべてのアーカイブ・ログのバックアップを作成する必要があります。これは、バックアップを確実に使用できるようにするために必要な操作です。バックアップ時に作成されたすべてのアーカイブ・ログがなければ、バックアップを一貫性のあるものにするために必要なすべてのログが取得できないため、バックアップを回復できません。

一貫性のあるデータベース全体のバックアップをとっている場合を除き、BACKUP CONTROLFILE オプションを指定して ALTER DATABASE コマンドを実行することにより、制御ファイルのバックアップをとるようにしてください。データベースは、オー

ブスする前に整合のとれた状態にしておく必要があります。一貫性のないデータベース全体のバックアップは、増分バックアップと REDO ログ（アーカイブおよびオンライン）を適用することにより整合をとることができます。

注意： NOARCHIVELOG モードで運用中のデータベースの一貫性のないデータベース全体のバックアップは、そのバックアップに関連するログがある場合にだけ利用できます。NOARCHIVELOG モードで実行中のデータベースの場合、回復作業なしに使用できるバックアップをとるようにしてください。バックアップを回復するためにログから REDO を適用する必要がある場合、この方法は使用できません。データベースの一貫性を維持するために REDO が必要な場合は、オンライン・ログのバックアップをとることができる ARCHIVELOG モードでデータベースを運用する必要があります。

表領域のバックアップ

表領域のバックアップとは、データベースのサブセットのバックアップです。

表領域のバックアップが有効なのは、データベースが ARCHIVELOG モードで運用されている場合だけです。NOARCHIVELOG モードで運用中のデータベースで表領域のバックアップが有効なのは、その表領域（およびその表領域内のすべてのデータ・ファイル）が読取り専用または通常オフラインの場合だけです。

データ・ファイルのバックアップ

データ・ファイルのバックアップとは、1 つのデータ・ファイルのバックアップです。表領域はデータベースのバックアップを作成する論理単位なので、データベース管理者は通常データ・ファイルのバックアップではなく表領域のバックアップをとります。

データ・ファイルのバックアップが有効なのは、データベースが ARCHIVELOG モードで運用されている場合だけです。NOARCHIVELOG モードで運用中のデータベースでデータ・ファイルのバックアップが有効なのは、そのデータ・ファイルが表領域内の唯一のファイルである場合だけです。たとえば、バックアップは表領域のバックアップであるがその表領域にファイルが 1 つしかなく、しかもその表領域が読取り専用または通常オフラインの場合だけです。

制御ファイルのバックアップ

制御ファイルのバックアップとは、データベースの制御ファイルのバックアップです。データベースがオープン状態であれば、次の文を発行できます。

```
ALTER DATABASE BACKUP CONTROLFILE to 'location';
```

Recovery Manager を使って制御ファイルのバックアップをとることもできます。

制御ファイルの O/S のバックアップを行う場合は、最初にデータベースを停止しなければなりません。

アーカイブ・ログのバックアップ

データベースが ARCHIVELOG モードで運用されている場合、オンライン・ログは LOG_ARCHIVE_DEST にアーカイブされ、通常そこから 3 次記憶装置（磁気テープなど）にバックアップがとられます。

非常に高速の回復が必要なサイトでは、アーカイブ・ログのコピーを、別のディスク・ファイル・システムと 3 次記憶装置の両方にとります。パラメータ LOG_ARCHIVE_DUPLEX_DEST を設定することにより、各オンライン・ログの第 2 のコピーが 3 次記憶装置にアーカイブされます。ディスクにとられたアーカイブ・ログのコピーは、通常、短時間（たとえば 48 時間）保持された後、削除されます。

関連項目：2-15 ページの「アーカイブ REDO ログの多重化」

アーカイブ・ログのバックアップの頻度は次の条件により決定します。

- 生成するアーカイブ・ログの数とサイズ
- ログのアーカイブ先のファイル・システムのサイズ

オンライン REDO ログのバックアップは作成しないでください

現行のオンライン・ログの内容のバックアップ方法として最善の方法は、オンライン・ログをアーカイブし、さらにアーカイブ・ログのバックアップを作成することです。

現行のオンライン・ログはコピーしないでください。現行のオンライン・ログをコピーした後でそのコピーを復元すると、そのコピーが REDO スレッドの終わりとみなされます。ところが、そのスレッド内にさらに REDO が生成されている場合があります。REDO ログ・コピーを使用して回復を実行しようとする、回復処理中に REDO スレッドの終わりが誤って検出され、処理が途中で打ち切られます。さらに、データベースが破損するおそれもあります。

バックアップ形式

これまでの項で説明したそれぞれのタイプのバックアップは、次の形式で作成できます。

- バックアップ・セット
 - データ・ファイル
 - アーカイブ REDO ログ・ファイル
- データ・ファイルのコピー
- オペレーティング・システムのバックアップ

バックアップ・セット

バックアップ・セットは、**Recovery Manager** の **backup** コマンドを使って作成します。バックアップ・セットはアーカイブ・ログまたはデータ・ファイルのどちらか一方で構成され、両方を混在させることはできません。

バックアップ対象のデータ・ファイルの読取りおよびバックアップ・セットの作成は、**Oracle** サーバー・プロセスが行います。したがって、オープン状態のデータベース・ファイルのバックアップ・セット (**Recovery Manager** により作成されたもの) は、**ALTER TABLESPACE BEGIN BACKUP** 文で処理する必要はありません。

詳細は、7-25 ページの「**Recovery Manager** でのオープン状態のデータベースのバックアップ時における分裂ブロックの検出」の項を参照してください。

バックアップ・セットは、**Oracle** 独自の形式で書き込まれます。バックアップ・セット内のファイルは、**Recovery Manager** によりそれらのファイルがインスタンスで使用可能な形式（通常はデータ・ファイル）に復元されるまで、**Oracle** インスタンスでは使用できません。たとえば、バックアップ・セット内の表領域のバックアップは、表領域内の各ファイルを圧縮したものです。**Recovery Manager** コマンドは、バックアップ・セット内のデータ・ファイルをインスタンスで使用可能な形式に復元します。

Oracle データベースの論理単位は、どれでもバックアップ・セットとしてバックアップを作成できます。

- **Recovery Manager** にデータベース全体のバックアップの実行を指示すると、そのデータベース内のすべてのファイルおよび制御ファイルのバックアップが自動的に作成される。
- **Recovery Manager** に表領域のバックアップの実行を指示すると、その表領域内の各データ・ファイルのバックアップが作成される。
- **Recovery Manager** にデータ・ファイルのバックアップの実行を指示すると、指定したデータ・ファイルのバックアップが作成される。
- **Recovery Manager** に **ARCHIVELOG** のバックアップの実行を指示すると、指定したアーカイブ・ログのバックアップが作成される（生成された「すべての」アーカイブ・ログのバックアップも指示できます）。

データ・ファイルのバックアップ・セットに制御ファイルのバックアップを組み込むように、Recovery Manager に指示することもできます。

注意： アーカイブ・ログとデータ・ファイルを1つのバックアップ・セットに入れることはできません。

データ・ファイルのコピー

データ・ファイルのコピーは、Recovery Manager の **copy** コマンドを使って作成します。

Oracle サーバー・プロセスは OS ルーチンではなくデータ・ファイルを読み取り、データ・ファイルのコピーをディスクに書き込みます。したがって、オープン状態のデータベース・ファイル (Recovery Manager により作成されたもの) のデータ・ファイルのコピーは、ALTER TABLESPACE BEGIN BACKUP 文で処理する必要はありません。

関連項目 :7-25 ページの「Recovery Manager でのオープン状態のデータベースのバックアップ時における分裂ブロックの検出」を参照してください。

データ・ファイルのコピーは Oracle インスタンスでただちに使用できます。それらはすでにインスタンスで使用可能な（通常のデータ・ファイル）形式になっています。

注意： データ・ファイルのコピーは、ディスクにしかとることができません。

オペレーティング・システムのバックアップ

オペレーティング・システム (O/S) のバックアップは、オペレーティング・システムのコマンドを使って作成します。O/S のバックアップは、使用中の O/S のユーティリティでサポートされている形式で、ディスクまたはテープに書き込むことができます。

Recovery Manager は、ディスク上のイメージ・バックアップである O/S のバックアップをカタログ化して使用できます。

論理バックアップ

論理バックアップでは、データベース用に作成されたスキーマ・オブジェクトについての情報が格納されます。Oracle データベースから、Oracle データベースの形式を持つオペレーティング・システムのファイルにデータを書き込むには、Export ユーティリティを使用します。

Oracle の Export ユーティリティは特定のオブジェクトまたはオブジェクトの一部（たとえば、パーティション表）を選択してエクスポートできるので、データベースのバックアップ計画における保護と柔軟性の強化という観点から、データベースの一部または全部のエクスポートを検討してください。データベースのエクスポートは、物理バックアップに代わるものではなく、Oracle のビルトイン機能により提供されるほどの完全な回復上の利点也没有。

関連項目： Export ユーティリティの詳細は、『Oracle Server ユーティリティ』を参照してください。

バックアップをとる時期

この章では、バックアップを計画するときに従うべきガイドラインと方針を示します。次の項目を中心に説明します。

- データベースのバックアップのガイドライン
- バックアップ計画の作成
- オンライン REDO ログのバックアップ

データベースのバックアップのガイドライン

この項では、データベースのバックアップを実行する時期と、データベースの中でバックアップが必要な部分を決める時に役立つガイドラインについて、次の項目を中心に説明します。

- 頻繁で定期的なバックアップの実行
- 構造変更時のデータベースの該当部分のバックアップ
- 使用頻度の高い表領域の頻繁なバックアップ
- 古いバックアップの保存
- リセットログ後のデータベースのバックアップ
- 保護と柔軟性の強化のためのデータベース・データのエクスポート
- 分散データベースのバックアップの検討
- 回復不能な操作またはログをとっていない操作の実行後のバックアップ
- バックアップおよび回復計画のテスト

Oracle データベースを作成する前に、ディスク障害が起きたときのためのデータベースの保護対策、また、Point-in-Time 回復を使用可能にするかどうかを決めておく必要があります。データベースを作成する前にこのような計画を立てておかないと、ディスク障害が起きてデータベースのデータ・ファイルまたはオンライン REDO ログ・ファイル、制御ファイルが破損した場合に、データベースを回復できなくなります。

頻繁で定期的なバックアップの実行

どのような回復計画でも、データベース全体のバックアップまたは表領域のバックアップを定期的しかも頻繁に実行することが重要です。バックアップの頻度は、データベース・データの変更（既存の表内での行の挿入および更新、削除や、新規の表の追加など）の割合、すなわち頻度、に基づいて決める必要があります。データベースのデータが頻繁に変更される場合は、それに比例してデータベースのバックアップ頻度も高く設定しなければなりません。逆に、データベースが主として読取り専用で、まれにしか更新されないのなら、データベースのバックアップ頻度は低くてかまいません。

構造変更時のデータベースの該当部分のバックアップ

次に示すような構造上の変更をする場合は、変更の直前および変更完了の直後に、データベースの該当部分のバックアップをとってください。

- 表領域の作成または削除
- 既存の表領域内でのデータ・ファイルの追加または改名（再配置）
- オンライン REDO ログ・グループまたはオンライン REDO ログ・メンバーの追加または改名（再配置）、削除

データベースの該当部分のバックアップの方法は、次に説明するように、データベースのアーカイブ・モードによって異なります。

- データベースが ARCHIVELOG モードで運用されている場合は、構造変更の前後に必要なのは、制御ファイルのバックアップだけです (BACKUP CONTROLFILE オプションを指定した ALTER DATABASE コマンドを使用)。ただし、データベースの他の部分のバックアップも作成できます。
- データベースが NOARCHIVELOG モードで運用されている場合は、変更の直前および直後に、すべてのデータ・ファイルおよび制御ファイルも含めて、一貫性のあるデータベース全体のバックアップをとる必要があります。

使用頻度の高い表領域の頻繁なバックアップ

データベースが ARCHIVELOG モードで運用されている場合は、個々の表領域内の全データ・ファイル、または 1 つのデータ・ファイルのバックアップをとることができます。データベースの一部（たとえば SYSTEM 表領域や、ロールバック・セグメントが格納されている表領域など）を他の部分よりも頻繁に使う場合には、このオプションが便利です。データベースの中でも使用頻度の高いデータ・ファイルのバックアップを頻繁にとることによって、それらのデータ・ファイルのコピーを常に最新の状態にしておくことができます。その結果、ディスク障害により使用頻度の高いデータ・ファイルが破損した場合でも、最新のバックアップを使って破損ファイルを復元できます。復元後のファイルを障害時点の状態まで、または必要な Point-in-Time 回復の時点までロールフォワードするためのデータ変更も少なくても済むので、データベースの回復にかかる時間を短縮できます。

回復不能な操作またはログをとっていない操作の実行後のバックアップ

UNRECOVERABLE オプションを使って表または索引を作成している場合は、そのオブジェクトの作成後にバックアップをとることを検討してください。表および索引を UNRECOVERABLE として作成すると、REDO ログは作成されず、既存のバックアップを基にしてこのようなオブジェクトを回復することもできません。

注意： Recovery Manager を使用する場合は、増分バックアップをとることができます。

関連項目： UNRECOVERABLE オプションについては、『Oracle8 Server SQL リファレンス』の CREATE TABLE...AS SELECT コマンドおよび CREATE INDEX コマンドの説明を参照してください。

古いバックアップの保存

古いデータベースのバックアップを保存しておく期間は、データベース回復に必要な条件によって異なります。過去のある時点まで回復する必要がある場合、その時点より前にとったデータベースのバックアップが必要です。NOARCHIVELOG モードで運用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必要です。

ARCHIVELOG モードで運用されているデータベースの場合は、一貫性はなくてもよいが、回復する時点より前にとったデータベース全体のバックアップ（制御ファイルが回復する時点のデータベース構造を反映している必要がある）と、必要な時点までデータ・ファイルを回復するために要求されるすべてのアーカイブ・ログがなくてはなりません。

保護を強化するために、現行バックアップの直前のバックアップ（およびそのバックアップに関連するすべてのアーカイブ・ログ）を複数保存することを検討してください。これにより、最新のバックアップが使用できない場合（たとえば、バックアップ用に使用したテープ・ドライブに不良バックアップが書き込まれた場合）、すべてのデータが失われてしまうという事態を避けることができます。

警告： RESETLOGS オプションを使ってデータベースをオープンした後は、ログのリセット後の時点に回復するために、既存のバックアップは使用できません。したがって、データベースを停止して一貫性のあるデータベース全体のバックアップを実行する必要があります。そうすれば、RESETLOGS オプションの使用後に行ったデータベースの変更を回復できます。

RESETLOGS オプション使用後のデータベースのバックアップ

RESETLOGS オプションを指定してデータベースをオープンした場合は、オープン直後にデータベース全体のバックアップを実行してください。バックアップをとらないと、障害が発生してデータベースの一部または全体の復元が必要になったときに、データベースをオープンした後に行った作業内容はすべて失われます。

注意： この規則には、1 つだけ例外があります。詳細は、3-10 ページの「RESETLOGS オプション使用後の回復」の項を参照してください。

RESETLOGS オプションを指定してデータベースをオープンした場合、Oracle は自動的に次のことを実行します。

- データベースの「インカーネーション」を新規作成する。
- ログ順序番号を 1 にリセットする。
- オンライン REDO ログ・ファイルがあれば、そのファイルを再フォーマットする（なければ、新規作成する）。

これら 3 つの作業は、データベースのインカーネーションに適用するアーカイブ REDO ログを識別するために実行されます。

NOARCHIVELOG モードでのバックアップ・オプション

NOARCHIVELOG モードの操作の場合、一貫性のあるデータベース全体のコールド・バックアップしか実行できません。

ARCHIVELOG モードでのバックアップ・オプション

ARCHIVELOG モードでの操作の場合、一貫性のある（クローズ状態の）データベース全体のバックアップ、または一貫性のない（オープン状態の）データベースのバックアップのどちらも実行できます。オプションの選択は、次の要因に依存します。

- データベースが使用可能になっていない時刻までの時間
- 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に入力したデータの重要性

最も重要な基準が、データベースを起動して、使用可能にしておくことである場合には、オープン状態のデータベースのバックアップが唯一のオプションです。

オープン状態のデータベースのバックアップに伴うリスクは、バックアップ作業が完了する前に別のメディア障害が起こった場合に、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に行った変更内容がすべて失われることです。その理由は、RESETLOGS

オプションを指定してデータベースをオープンする前にとられたバックアップは、このデータベースのインカーネーションの回復には使用できないためです。

注意： この規則には、1 つだけ例外があります。3-10 ページの「RESETLOGS オプション使用後の回復」の項を参照してください。

別の障害が起きたときに復元できるようにすることが最も重要な基準の場合は、一貫性のある（クローズ状態の）データベースのバックアップをとるようにしてください。

時間が許す場合は、一貫性のあるデータベース全体のバックアップをとるようにしてください。

保護と柔軟性の強化のためのデータベース・データのエクスポート

Oracle の Export ユーティリティを使うと、特定のオブジェクトを選択してエクスポートできるので、データベースのバックアップ計画において保護と柔軟性を強化するために、データベースの一部または全部のエクスポートを検討する価値があります。データベースのエクスポートは、データベース全体のバックアップにかわるものではないので、Oracle のビルトイン機能ほどの完全な回復機能は提供されません。

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

分散データベースのバックアップの検討

データベースが分散データベースのノードの 1 つである場合は、次のガイドラインを検討してください。

- 分散データベース・システム内のすべてのデータベースは、同じアーカイブ・モードで実行する必要がある。
- 分散データベース・システム内のデータベースが ARCHIVELOG モードで運用されている場合は、各ノードのバックアップは自律的に（同期をとらずに個別に）実行できる。
- 分散データベース・システム内のデータベースが NOARCHIVELOG モードで運用されている場合は、グローバルな分散データベース回復を計画するには、一貫性のあるデータベース全体のバックアップを同じ時点で（グローバルに）実行しなければならない。たとえば、ニューヨークのデータベースのバックアップをとるのが東部標準時の午前零時なら、サンフランシスコのデータベースのバックアップは太平洋標準時の午後 9 時にとる必要があります。

バックアップおよび回復計画のテスト

実働システムへの移行前と移行後に、作成したバックアップおよび回復計画をテスト環境でテストしてください。そうすることによって、計画の完全性をテストし、実際の稼働状況で問題が発生する前に、問題を最小限に留めることができます。

テスト回復を定期的に行うことにより、アーカイブおよびバックアップ、回復作業の動作を確認できます。また、回復作業に日頃から慣れておくことにもなるので、実際に緊急事態が起こった場合にもミスをする危険性が少なくなります。

バックアップ計画の作成

Oracle データベースを作成する前に、障害が起きた場合にデータベースを保護するための計画を決めておいてください。バックアップ計画を展開する前に、次の質問に答えてください。

- データベースを構成するファイルの一部がディスク障害によって破損した場合、データを失ってもかまいませんか？データを1つも失ってはならない状況の場合は、データベースを ARCHIVELOG モードで（理想的には多重オンライン REDO ログを実現して）運用する必要があります。ディスク障害が起きたときにある程度の量のデータなら失ってもよいという場合は、データベースを NOARCHIVELOG モードで運用しても構いません。これにより、REDO ログ・ファイルが満杯になったときに必要となるアーカイブ作業を省くことができます。
- 過去のある時点への回復が必要となる可能性がありますか？過去のある時点まで回復して、データベースに行った操作エラーまたはプログラム変更エラーを訂正する必要がある場合は、必ず ARCHIVELOG モードで実行し、構造を変更するたびに制御ファイルのバックアップをとるようにしてください。過去のある時点でのデータベース構造を反映するバックアップ制御ファイルがあれば、その時点の状態に回復できます。
- データベースは常時（1日24時間、週7日間）使用可能にしておく必要がありますか？その必要がある場合は、データベースを NOARCHIVELOG モードで運用しないでください。NOARCHIVELOG モードでは、データベース全体のバックアップはデータベースの停止中にとる必要がありますが、常時使用可能にしておく必要がある場合には、まったくとはいえないまでも、頻繁にはバックアップをとることができないからです。したがって、可用性が高いデータベースは、オープン状態のデータ・ファイルのバックアップを利用できるように、常に ARCHIVELOG モードで運用します。

NOARCHIVELOG モードでのバックアップ計画

データベースが NOARCHIVELOG モードで運用されている場合、オンライン REDO ログ・ファイルのグループは、満杯になってもアーカイブされません。したがって、ディスク障害に対する唯一の保護手段は、データベース全体の最新のバックアップだけです。

失ってもよい作業量に応じて、データベース全体のバックアップを定期的にとる計画を立ててください。たとえば、1週間分の作業量なら失ってもかまわないという場合は、1週間に一度、データベース全体のクローズ状態のバックアップをとります。1日分の作業の消失し

が許されない場合は、データベース全体のクローズ状態のバックアップを毎日とる必要があります。作業量の多い大規模データベースでは、一般に、作業の消失は許されません。したがって、データベースを ARCHIVELOG モードで運用し、適切なバックアップ計画に従う必要があります。

NOARCHIVELOG モードで運用されているデータベースの物理構造を変更した場合は、必ず変更直後に一貫性のあるデータベース全体のバックアップをとってください。これにより、データベース全体のバックアップに、データベースの新しい構造が完全に反映されます。

ARCHIVELOG モードでのバックアップ計画

データベースが ARCHIVELOG モードで運用されている場合は、満杯になったオンライン REDO ログ・ファイルのグループはアーカイブされます。したがって、アーカイブ REDO ログをオンライン REDO ログおよびデータ・ファイルのバックアップと組み合わせることによって、ディスク障害からデータベースを保護し、ディスク障害が発生したインスタンス（または指定した過去のある時点）まで、その障害から完全に回復させることができます。

ARCHIVELOG モードで運用されているデータベースについての一般的なバックアップ計画を次に示します。

- データベースを最初に作成したときに、データベース全体に関してのクローズ状態のバックアップを実行してください。この最初のデータベース全体のバックアップには、このデータベースに関連するすべてのデータ・ファイルおよび制御ファイルのバックアップが含まれるので、今後のバックアップの基礎になります。

注意： この最初のデータベース全体のバックアップを実行する前に、データベースが ARCHIVELOG モードになっていることを確認してください。ARCHIVELOG モードでないと、バックアップされた制御ファイルに NOARCHIVELOG モードの設定が組み込まれることになります。

- その後は、データベース全体のバックアップは必須ではありません。データベースを常にオープンしたままにしておかなければならない場合は、データベース全体のクローズ状態のバックアップは実行できません。かわりに、オープン状態のデータベースまたは表領域のバックアップを実行することにより、データベースのバックアップを更新できます。
- データベースに関するバックアップ済み情報を更新する（そして最初のデータベース全体のバックアップを補足する）ために、オープン状態またはクローズ状態のデータ・ファイルのバックアップをとってください。特に、万一回復が必要になった場合に、データベースの回復にかかる時間を短縮するために、使用頻度の高い表領域のデータ・ファイルのバックアップを頻繁にとるようにしてください。破損したデータ・ファイルの復元に使用するデータ・ファイルのバックアップが最新であればあるほど、復元後のデータ・ファイルを障害発生時点までロールフォワードするために適用しなければならない REDO（または増分バックアップ）が少なく済みます。

データ・ファイルのバックアップをオープン状態でとるかクローズ状態でとるかは、データの可用性に関する必要条件に依存します。バックアップ対象のデータを常に使用

可能にしておく必要がある場合は、オープン状態のデータ・ファイルのバックアップしか選択できません。

データ・ファイルの回復には、データ・ファイルのコピーも使用できますが、そのデータ・ファイル・コピーは、データベースがオープン状態で表領域がオンラインになっているときにとったものでなくてはなりません。ただし、復元後のデータ・ファイル内のデータには一貫性がありません。したがって、復元されたデータ・ファイルには、適切な（オンラインおよびアーカイブ）REDO ログ・ファイルを再度適用して、データに一貫性を持たせ、指定時点の状態までロールフォワードする必要があります。

- データベースの構造を変更するたびに、制御ファイルのバックアップをとってください。ARCHIVELOG モードで運用されており、データベースがオープン状態の場合は、Recovery Manager、または BACKUP CONTROLFILE オプションを指定した ALTERDATABASE コマンドを使用してください。

注意： 制御ファイルのバックアップ後は、データ・ファイルが追加された時点までの REDO を適用してください。次に、ALTER DATABASECREATE DATAFILE 文を発行して、回復を継続してください。ARCHIVELOG モードで制御ファイルのバックアップをとるときは、（クローズ状態のバックアップを実行するとき以外は）オペレーティング・システムのユーティリティを使用しないでください。

オンライン REDO ログのバックアップ

次の理由から、オンライン・ログのバックアップはとる必要はありません。

- メディア障害に対してオンライン・ログを保護する方法は、オンライン・ログを多重化することです（異なるディスクおよび異なるディスク・コントローラに、グループ毎に複数のログ・メンバーを持つことです）。
- データベースが ARCHIVELOG モードの場合、それぞれのログはアーカイブ時にバックアップがとられます。
- データベースが NOARCHIVELOG モードの場合、実行できるバックアップのタイプは、クローズ状態で、一貫性のあるデータベース全体のバックアップしかありません。バックアップに一貫性を持たせるには、即時停止またはトランザクションの停止、正常停止により、データベースをクローズしなければなりません。このタイプのバックアップのファイルはすべて一貫性があり、回復の必要がないので、オンライン・ログは必要ありません。

データベースがクラッシュし、オンライン REDO ログの多重化されたコピーがすべて失われたときには、次のようにします。

- データベースが ARCHIVELOG モードの場合、データベース全体を復元し、最後のアーカイブ・ログまでデータベースを回復させる必要があります。回復後、RESETLOGS オプションを指定して、データベースをオープンしなければなりません。

このような状況では、オンライン・ログの古いバックアップは使用されません。そのオンライン・ログの情報は回復に必要なからです（古いオンライン・ログ・ファイル内の情報は、アーカイブ・ログにすでにあります）。

- データベースが NOARCHIVELOG モードの場合は、バックアップ方法は、一貫性のあるデータベース全体のバックアップになります。バックアップ後、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。データベースにはすでに一貫性があるので、回復は必要ありません。

注意： Recovery Manager は、データベースが正しく停止されていない限り、NOARCHIVELOG モードのデータベースのバックアップは実行しません。

RESETLOGS オプション使用後の回復

Oracle8 より前のリリースでは、ただちに復元する場合には、RESETLOGS オプションを使わずにデータベースをオープンできるように、DBA は、一貫性のあるコールド・バックアップを実行するときにオンライン・ログのバックアップをとっていました。このようなバックアップ例は、ディスク・メンテナンス中（データベースのバックアップと削除およびディスクの再構成、データベースの復元）に発生していました。RESETLOGS オプションを使わない理由は、データベースを復元した後すぐにデータベース全体のバックアップをとらないで済むようにするためです。RESETLOGS オプションを使用すると、ログのリセット後にエラーが発生した場合、RESETLOGS を使用する前にとったバックアップによる回復ができないため、データベース全体のバックアップが必要でした。

今では、このようなことはありません。RESETLOGS オプションを指定してデータベースをオープンする直前に、一貫性のあるデータベースのバックアップを実行し、RESETLOGS オプションを指定してデータベースをオープンした後に有効な制御ファイルがあれば、そのバックアップを使ってロールフォワードできます。

これが可能な場合を、次の状況を例にして説明します。

DBA がハードウェアのストライプ化を再構成する場合を考えてみます。これには、データベース・ファイルのバックアップと削除、およびハードウェアの再構成、データベースの復元が必要です。

金曜日の夜に、DBA は次の作業を行います。

- データベースを正しく停止する (SHUTDOWN IMMEDIATE)
- データベース全体のバックアップ（制御ファイルおよびデータ・ファイル）を実行する

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。

- O/S のメンテナンスを実行する
- データベースと制御ファイルを復元する
- STARTUP MOUNT DATABASE
- OPEN DATABASE RESETLOGS

土曜日の朝に、次の処理が発生します。

- スケジュールされていたバッチ・ジョブを実行し、REDO ログに多数のレコードが生成される

データベース全体の復元が必要なハードウェア障害が土曜日の夜に発生した場合、RESETLOGS オプションを指定してデータベースをオープンする直前にとったバックアップを復元し、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。

土曜の夜に、次の処理が発生します。

- SHUTDOWN ABORT INSTANCE(インスタンスが存在する場合)
- すべての破損ファイルを、金曜日の夜のバックアップから復元する

注意： 現行の制御ファイルがある場合は、その制御ファイルは復元しないでください。復元した場合、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に有効だった制御ファイルも復元しなければなりません。

- ロールフォワード

警告： 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に回復できるのは、次のものがある場合だけです。

- データベースのオープン直前にとった一貫性バックアップ。つまり、バックアップをとった後、RESETLOGS オプションを指定してデータベースをオープンするまでは、データベースをオープンしていなかったことが必要条件になります。
 - RESETLOGS オプションを指定してデータベースをオープンした後に有効だった制御ファイル。
-

上記の例では、DBA が金曜日の夜のバックアップから RESETLOGS オプション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のオープン後に制御ファイルがない場合、金曜日の夜のバックアップを使ってロールフォワードできません。その場合、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にバックアップをとっておかないと、回復できません。

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は、データベースの完全バックアップをとるようにしてください。

関連項目： 詳細は、3-5 ページの「RESETLOGS オプション使用後のデータベースのバックアップ」の項を参照してください。

オンライン REDO ログのバックアップ作成の危険性

オンライン・ログのバックアップを作成することは、危険なことではありません。危険なのは、意図せずに、オンライン・ログを誤って復元してしまうことです。オンライン・ログを復元した場合、回復プロセスで重大な問題が発生する場合があります。オンライン・ログのバックアップの復元による、回復への重大な影響を次の 2 つの例で示します。

オンライン REDO ログの意図せぬ復元

障害が発生すると、単純なミスを犯しがちです。DBA やシステム管理者がよく経験する危険の 1 つは、データベースの復元中に起こります。データベース全体の復元中に、誤ってオンライン REDO ログも復元してしまうことがあります。これによって、現行のログが古い（無効な）バックアップで上書きされてしまいます。この操作によって、完全回復を実行しようとしているにもかかわらず、不完全回復が実行されてしまい、大切なトランザクションを回復できなくなります。

複数のパラレル REDO ログの時間線の作成エラー

シングル・インスタンスのデータベースに対して、パラレル REDO ログの時間線を誤って複数作成してしまうことがあります。オンライン・ログが復元できないように設定されていれば、このミスは防ぐことができます。この設定により、データベースを RESETLOGS オプ

ションを指定してオープンしなければならなくなり、このオプションによって新しいログと新しいデータベースのインカーネーションが有効に作成されるからです。

発生した問題に対する最善の方法が、一貫性バックアップでデータベースを復元し、回復は行わないことである場合、オンライン・ログを復元して、**RESETLOGS** オプションを指定したデータベースのオープンは避けるのが安全とされる場合があります。

これを実行した場合、データベースによって前回にすでに生成されていたログの順序番号が、データベースによって生成されてしまいます。この状況で別の問題が発生し、このバックアップから復元してロールフォワードすることが必要になった場合、実際にどのログ順序番号が正しい番号か特定するのが困難になります。この例では、ログがリセットされていた場合、データベースの新たなインカーネーションが作成されます。この新たなインカーネーションで作成されたログは、このインカーネーションにしか適用されません。

注意： 上記の理由から、**Recovery Manager** と **EBU** は、オンライン・ログのバックアップをとりません。

回復方針の選択

この章では、データベースの回復を準備するときに従うべきガイドラインと方針を示します。次の項目を取り挙げます。

- 回復の概要と方針
- 回復計画

回復の概要と方針

データベースを回復する前に、Oracle の回復の基本的なデータ構造および概念、方針について十分に知っておく必要があります。この項では、回復の基本的な問題点について説明します。次の項目を取り挙げます。

- 回復で使用される重要なデータ構造
- 回復計画
- 回復操作

当然のこととして、回復の操作において重要なデータ構造を学習する前に、最初にデータの復元と回復の意味を理解する必要があります。

ファイルの復元とは、元のファイルのコピーをバックアップから再構築するか、または取り出すことです。

ファイルの回復とは、復元されたファイルを最新のファイルにすること、またはそのファイルを特定の時点での最新のファイルにすることです。回復プロセスは、「ロールフォワード」ともいいます。

回復コマンドには、以下の 2 つのタイプがあります。

- SQL

SQL の RECOVER コマンドは、REDO(アーカイブ・ログおよびオンライン REDO ログ・ファイル)を適用して、復元済みのファイルをロールフォワードします。

- Recovery Manager

Recovery Manager の **recover** コマンドは、増分バックアップ（存在する場合）を適用してから、REDO(アーカイブ REDO ログおよびオンライン REDO ログ)を適用して、復元済みのデータ・ファイルをロールフォワードします。

回復で使用する重要なデータ構造

Table 4-1 に、回復プロセスで使用する重要なデータ構造を示します。回復操作を開始する前に、これらのデータ構造についてよく理解しておいてください。

表 4-1 回復で使用するデータ構造

データ構造	説明
制御ファイル	制御ファイルには、データベースの物理構造を記録し、その物理構造についての情報をメンテナンスするレコードが格納されています。制御ファイルは、データベースの使用中に継続的に更新されるので、データベースのオープン中は常に書き込み可能になっていなければなりません。制御ファイルにアクセスできないと、データベースは正常に動作しません。
システム変更番号 (SCN)	システム変更番号は、Oracle データベース用のクロック値で、データベースのコミット済みバージョンが記述されます。システム変更番号 (SCN) は、データベースのシーケンス・ジェネレータとして機能し、並列性および REDO レコードの順序付けを制御します。SCN は、トランザクションの整合性を確実にするためのタイムスタンプとみなすことができます。
ロールバック・セグメント	ロールバック・セグメント内の情報は、データベース回復時に、REDO ログからデータ・ファイルに適用済ではあるが、まだコミットされていない変更を取り消すために使用されます。ロールバック・セグメントを使ってコミットされていないすべてのデータをデータ・ファイルから削除した後は、データは整合のとれた状態になっています。
REDO レコード	REDO レコードは、変更ベクトルのグループで、データベースに対する単一の最小単位の変更が記録されます。すべてのデータ・ブロック変更について、REDO レコードが記録され、ディスク上の REDO ログに保存されます。REDO レコードを使うと、障害があった場合に、複数のデータベース・ブロックを変更して、すべての変更を有効にするか変更がまったくなかった状態に戻すことができます。
REDO ログ	Oracle データベースに対する変更はすべて REDO ログに記録されます。REDO ログは、少なくとも 2 つの REDO ログ・ファイルで構成され、データ・ファイルとは分離されています。インスタンス障害またはメディア障害からのデータベース回復時には、Oracle はそのデータベースの REDO ログに記録されている該当の変更内容をデータ・ファイルに適用します。これによって、データベースのデータが更新され、障害発生時のインスタンスまで回復できます。
バックアップ	データベースのバックアップとは、Oracle データベースを構成している物理ファイルのコピーです。元のデータが失われた場合、バックアップを使って、失われた情報を再構築できます。

表 4-1 回復で使用されるデータ構造（続き）

増分バックアップ (Recovery Manager を使 う場合のみ)	増分バックアップは、1 つ以上のデータ・ファイルのバック アップです。これには、前回のバックアップ以降に変更された ブロックだけが含まれます。
リカバリ・カタログ (Recovery Manager を使 う場合のみ)	Recovery Manager が使用し、管理する情報のリポジトリ。これ には、次の情報が含まれます。 データ・ファイルおよびアーカイブ・ログのバックアップ・ セットおよびバックアップ・ピース データ・ファイルのコピー アーカイブ REDO ログおよびそのコピー ターゲット・データベースにある表領域およびデータ・ファイ ル ユーザー作成ストアド・スクリプト
チェックポイント	チェックポイントは、制御ファイル内のデータ構造で、REDO ログのすべてのスレッド全体で整合のとれた、データベースの 1 つの時点です。チェックポイントはシステム変更番号 (SCN) に似ていますが、対応する SCN にどのスレッドが存在するか についても記録されています。チェックポイントは、Oracle が REDO を適用するためのログ・スレッドの読取りを正しい時点 から開始できるようにするため、回復処理で使用されます。 Parallel Server の場合は、各チェックポイントに専用の REDO 情報が記録されます。

回復計画

データベースを回復する前に、回復計画を作成する必要があります。この項では、計画を定義するときに考慮しなければならない重要な点について説明します。

回復方針を判断する要因

回復方針は、次の各質問に対する答えの内容によって決まります。

- データベースは ARCHIVELOG モードで実行中ですか？
- どのファイルが消失または破損しましたか？
- 誤ってデータを破棄しましたか？
- バックアップおよび回復計画をテストしましたか？

これらの質問に答えた後、その特定の状況に適した回復の方針を選択できます。次の項からは、特定の状況で実行する回復のタイプについて説明します。

データベースは ARCHIVELOG モードで実行中ですか？

データベースが ARCHIVELOG モードで運用中でない場合は、一貫性のあるデータベースのバックアップからデータベースを復元する必要があります。データベースが ARCHIVELOG モードでない場合は、これが唯一のオプションです。制御ファイルとすべてのデータ・ファイルは一貫性バックアップから復元され、データベースがオープンされます。バックアップの後に行ったすべての変更内容は、失われます。

注意： NOARCHIVELOG モードで運用中の場合、一番最後にバックアップをとった時に現行だったオンライン・ログ・ファイルが上書きされる前でないとはデータベースを回復できません。

どのファイルが消失または破損しましたか？

1 つ以上のデータ・ファイルが消失したけれども、データベースがオープンされたままの場合は、データベースのオープン中に表領域（またはデータ・ファイル）の回復を実行してください。表領域またはデータ・ファイルはオフラインに設定され、バックアップから復元され、回復した後にオンラインに設定されます。変更内容が失われることはなく、データベースは回復中でも使用可能です。

回復に必要なアーカイブ・ログまたはオンライン・ログも消失した場合は、データベースまたは表領域の Point-in-Time 回復を実行する必要があります。

1 つ以上のデータ・ファイルまたはすべての制御ファイルが消失し、しかもデータベースがオープン状態でない場合は、この方法は使用できません。その場合は、消失したすべてのファイルをバックアップから復元、回復してから、データベースをオープンします。変更内容が失われることはありませんが、回復中はデータベースを使用できません。回復に必要な

アーカイブ・ログまたはオンライン・ログも消失した場合は、Point-in-Time データベース回復を実行する必要があります。

誤ってデータを破棄しましたか？

回復に必要なアーカイブ・ログまたはオンライン・ログが消失した場合、またはユーザー・エラーから回復する場合は、Point-in-Time データベース回復を実行する必要があります。すべてのデータ・ファイルと制御ファイルは、それ以前の時点で実行されたバックアップから復元し、その時点の状態に回復してから、データベースをオープンします。その時点以降に行われたすべての変更内容は、失われます。データベースがオープン状態のままで、ロールバック・セグメントが入っている表領域が消失していない場合も、表領域の Point-in-Time 回復を実行できます。

表領域の Point-in-Time 回復 回復した表領域に対してその時点以降に行っていたすべての変更内容は、失われます。データベースは、回復した表領域を除き、回復中も使用可能です。

バックアップおよび回復計画をテストしましたか？

バックアップおよび回復計画は、稼働システムに移行する前に、テスト環境でテストする必要があります。また、その後も定期的にシステムをテストしてください。こうすることによって、計画の完全性をテストし、その後の実稼動環境での危機を避けることができます。テスト回復を定期的に行うことで、アーカイブおよびバックアップの作業が正常に動作することを確認できます。また、定期的にテストを実行することにより、回復作業に慣れるので、実際に緊急事態が起こった場合にもミスをする危険性が少なくなります。

回復操作

メディア回復では、ディスク障害前の時点の状態にデータベースのデータ・ファイルが復元され、障害のために消失した、メモリー内のコミット済みデータも書き込まれます。次に示すようなメディア回復操作があります。

- 完全メディア回復

完全メディア回復には、回復対象のデータベースの特定のインカーネーションについてそれまでに生成されたすべての必要な REDO または増分バックアップが適用されます。完全メディア回復は、データベースがオープン状態の間にオフライン・データ・ファイルに対して実行できます。回復対象にバックアップの制御ファイルが含まれる場合、または新規制御ファイルを作成するリセットログ操作を直前に実行した場合は、完全メディア回復で、リセットログ・オープン操作が必要になることがあります。完全メディア回復には次のタイプがあります。

- クローズ状態のデータベース回復
- オープン状態のデータベースのオフライン表領域回復
- オープン状態のデータベースのオフライン表領域内の個別データ・ファイル回復

- 不完全メディア回復

不完全メディア回復では、過去のある時点の状態のデータベースが作成されます。不完全メディア回復の後で、完全メディア回復を行うか、データベースの新規のインカーネーションを作成するリセットログ・オープン操作を実行しなければなりません。不完全メディア回復操作をするには、データベースをクローズする必要があります。不完全メディア回復には次のタイプがあります。

- 時間ベースの回復
回復する時点を指定します。
- 取消しベースの回復
回復操作中に取消しを決めます。
- 変更ベースの回復

Recovery Manager では、不完全メディア回復には次のタイプがあります。

- 時間ベースの回復
- 変更ベースの回復
- ログ順序番号の回復（上記の取消しベースの回復に相当します）
指定されたログ順序番号まで回復させます。

バックアップ方法の選択

この章では、使用可能なさまざまなバックアップ方法と、状況に応じたバックアップ方法の選択について説明します。主な項目は次のとおりです。

- バックアップ方法および要件
- Recovery Manager
- バックアップ方法の機能の比較

バックアップ方法および要件

この項では、Oracle のさまざまなバックアップ方法と、それぞれに対応する要件について説明します。次の項目を中心に説明します。

- 表 5-1 に、バックアップ方法の種類と要件を示します。
- オペレーティング・システム (O/S)
- Enterprise Backup Utility (EBU)
- Export

表 5-1 に、バックアップ方法の種類と要件を示します。

表 5-1 Requirements for Different Backup Methods

バックアップ方法	使用可能なバージョン	要件
Recovery Manager	Oracle8	メディア・マネージャ（テープにバックアップをとる場合）
O/S	Oracle のすべてのバージョン	O/S のバックアップ・ユーティリティ（例：UNIX dd）
Export	Oracle のすべてのバージョン	該当せず
Enterprise Backup Utility (EBU)	Oracle7	メディア・マネージャ（テープにバックアップをとる場合）

Recovery Manager

Recovery Manager ユーティリティは、Oracle データベースのバックアップおよび復元、回復操作を管理します。Recovery Manager は、データベースに関する情報を使って、自動的に位置を特定してから、データ・ファイルおよび制御ファイル、アーカイブ REDO ログのバックアップおよび復元、回復を実行します。

オペレーティング・システム (O/S)

O/S のバックアップの場合は、使用中のオペレーティング・システムに固有のコマンドを使って、Oracle データベース・ファイルのバックアップを作成します。たとえば、UNIX O/S のバックアップを作成するには、'dd' または 'tar' コマンドを使用します。O/S のバックアップは、データベース管理者とシステム管理者が記述およびメンテナンスする UNIX スクリプトによって制御されます。

関連項目： 使用しているオペレーティング・システムで使用可能なユーティリティの情報は、それぞれのオペレーティング・システム用のマニュアルを参照してください。

Export

Oracle データベースから、オペレーティング・システムのファイルに Oracle データベースの形式でデータを書き込むには、Oracle の Export ユーティリティを使用します。エクスポート・ファイルには、データベース用に作成されたスキーマ・オブジェクトについての情報が格納されます。Oracle の Export ユーティリティでは特定のオブジェクトを選択してエクスポートできるので、保護と柔軟性を強化するために、データベースの一部または全部のエクスポートを検討してください。データベースのエクスポートは、データベース全体のバックアップに代わるものではなく、回復機能としては、Oracle のビルトイン機能ほど優れていません。

関連項目：バックアップ計画の一部としてエクスポートを使用する方法については、『Oracle8 Server ユーティリティ』を参照してください。

Enterprise Backup Utility

Oracle の Enterprise Backup Utility は、Oracle7 データベースのバックアップをとるときに使用します。Oracle8 データベースとの互換性はありません。

関連項目：Enterprise Backup Utility(EBU) を使って Oracle7 のデータベースのバックアップをとる方法については、『Oracle7 Enterprise バックアップ・ユーティリティ管理者ガイド』を参照してください。

Recovery Manager

Recovery Manager ユーティリティは、Oracle データベースのバックアップおよび復元、回復操作を管理します。Recovery Manager は、データベースに関する情報を使って、自動的に位置を特定してから、データ・ファイルおよび制御ファイル、アーカイブ REDO ログをバックアップおよび復元、回復します。

Recovery Manager は必要な情報を、データベースの制御ファイルから取得するか、または Recovery Manager がメンテナンスするリカバリ・カタログと呼ばれる情報の中央リポジトリを介して取得します。

Recovery Manager でのバックアップに Oracle Enterprise Manager を使用する方法

Oracle Enterprise Manager を使用して、Recovery Manager を使ったバックアップを実行できます。Oracle Enterprise Manager の Backup Manager は、Recovery Manager の GUI インタフェースで、これによりポイント・アンド・クリック方式でバックアップおよび回復を実行できます。

関連項目：Oracle Enterprise Manager を使ったバックアップおよび回復については、『Oracle Enterprise Manager 管理者ガイド』を参照してください。

Recovery Manager と従来のオペレーティング・システムのバックアップとの違い

Recovery Manager は、Oracle サーバー・プロセスに対して、接続先のデータベースのバックアップまたは復元、回復を指示するコマンド行インタフェース (CLI) です。

Recovery Manager プログラムは、Oracle サーバー・プロセスにコマンドを発行します。Oracle サーバー・プロセスは、バックアップ対象のデータ・ファイルまたは制御ファイル、アーカイブ REDO ログを読み込んだり、復元または回復対象のデータ・ファイルまたは制御ファイル、アーカイブ REDO ログを書き込みます。

Oracle サーバー・プロセスは、データ・ファイルの読み込みで、分割ブロックを検出すると、一貫性のあるブロックを取得するために再読み込みを行います。したがって、オープン・バックアップに Recovery Manager を使用する場合は、表領域をホット・バックアップ・モードに設定しないでください。

関連項目 :7-25 ページの「Recovery Manager でのオープン状態のデータベースのバックアップ時における分裂ブロックの検出」

バックアップと回復の作業が Recovery Manager によって実行されるので、バックアップや回復のときに管理者が行う作業を大幅に省力化できます。Recovery Manager を使うと、たとえば次のようなことができます。

- 頻繁に実行するバックアップ操作を設定する。
- バックアップおよび回復にかかわるすべてのアクションについて印刷可能ログを生成する。
- リカバリ・カタログを使って、メディア復元操作および回復操作の両方を自動化する。
- バックアップおよび復元を自動パラレル化する。
- REDO の量についてユーザーが指定した適用必須の制限に基づき、バックアップが必要なデータ・ファイルを検索する。
- データベースまたは個々の表領域、個々のデータ・ファイルのバックアップをとる。

ディスクへのバックアップ

Recovery Manager を使用して、ディスクへのバックアップを実行できます。ディスクへのバックアップの書込みだけを実行する場合は、Oracle にメディア・マネージャを組み込む必要はありません。

順次メディアへのバックアップ

注意： Oracle データベースのバックアップを磁気テープ記憶装置に格納する場合、Recovery Manager によって、メディア管理製品をインストールするように要求されます。Oracle の Backup Solutions Program(BSP) は、Oracle Media Management インタフェース仕様に適合したメディア管理製品を備えています。その 1 つに Legato Storage Manager(LSM) があります。これは、多数のプラットフォームで使用でき、各プラットフォーム用の Oracle ソフトウェアに付属しています。

この Legato 製品が使用しているプラットフォームで使えるかどうかは、それぞれのプラットフォーム用のマニュアルを参照してください。この製品でサポートされる機能については、『Legato Storage Manager 管理者ガイド』を参照してください。BSP の他のメディア管理製品が使用中の Oracle 製品に付属している場合は、それぞれのプラットフォーム用のマニュアルにそのメディア管理製品の説明があります。

他にも、メディア管理ソフトウェア・ベンダーから直接入手できる、各プラットフォーム用のメディア管理製品がいくつかあります。Oracle の Backup Solutions Program(BSP) ウェブ・サイトにアクセスすると、利用できるメディア管理製品の最新の情報を参照できます。URL は次のとおりです。

<http://www.oracle.com/st/products/features/backup.html>

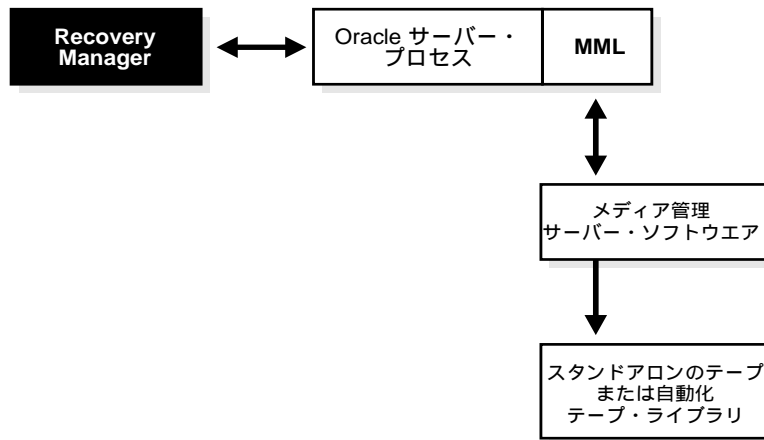
あるいは、この情報が必要な場合は、オラクル社の営業担当員に問い合わせてください。

特定のメディア管理製品を使用する場合は、その製品が Oracle の Backup Solutions Program に属する製品かどうかをそのベンダーに直接問い合わせて確認してください。

Recovery Manager を使ってテープなどの順次メディアへのバックアップを実行するには、Oracle ソフトウェアが統合されたメディア管理ソフトウェアが必要です。

図 5-1 に、Oracle に統合されたメディア管理ソフトウェアのアーキテクチャを示します。

図 5-1 Oracle に統合された MML のアーキテクチャ



Oracle 実行可能ファイルは、ユーザーがデータベースに接続した時に起動されるものと同じです。上記のダイアグラムの MML(メディア管理ライブラリ)は、ベンダーが提供するメディア管理ソフトウェアで、これは Oracle とインタフェースをとることができます。Oracle は MML ソフトウェア・ルーチンをコールして、メディア・マネージャが制御するメディアをターゲットまたはソースとして使い、データ・ファイルのバックアップおよび復元を実行します。

次の Recovery Manager スクリプトは、メディア・マネージャが制御するテープ・ドライブに対してデータ・ファイルのバックアップを実行します。

```

run {
  allocate channel t1 type 'SBT_TAPE';
  backup
    format 'df_%s_%t'
    (datafile 10);
}

```

Recovery Manager は上記のコマンドを実行するときに、バックアップを実行する Oracle サーバーに対して、バックアップ要求を送信します。Oracle サーバー・プロセスは出力チャンネルをタイプ 'SBT_TAPE' として識別し、メディア管理ライブラリに対して、テープをロードし、指定された出力を書き込むように要求します。

Oracle が特定のファイルの復元を要求した場合、メディア・マネージャはファイルが常駐するテープを識別し、テープを読み取り、(ファイルをディスクに書き込む)Oracle サーバー・プロセスに情報を渡します。

メディア・マネージャは、テープにラベルを付け、テープを追跡して記録し、また各テープ上でのファイル名を追跡して記録します。ロボット・アーム付の自動化テープ・ライブラリがあるサイトの場合、メディア・マネージャは Oracle で必要なテープを自動的にロードおよ

びアンロードします。ライブラリがない場合、メディア・マネージャのオペレータに対して指定テープをドライブにロードするように要求します。

バックアップ・データの書込み先がディスクの場合は、Oracle サーバーを MML ソフトウェアに接続する必要はありません。Oracle 実行可能ファイルは、サード・パーティの統合ソフトウェアを介さずに、バックアップ・データをディスクに書き込みます。

バックアップ方法の機能の比較

Table 5-2 では、この章で説明したバックアップ方法の機能を比較します。

表 5-2 各バックアップ方法の機能の比較

機能	Recovery Manager	オペレーティング・システム	Export
クローズ状態のデータベースのバックアップ	サポートされている。インスタンスをマウントする必要があります。	サポートされている。	サポートされていない。
オープン状態のデータベースのバックアップ	BEGIN/END BACKUP コマンドは必要ない。	BEGIN/END BACKUP コマンドを使用すると、生成される REDO が増える。	一貫性バックアップを生成するには、RBS が必要。
増分バックアップ	サポートされている。すべての変更ブロックのバックアップをとりまします。	サポートされていない。	サポートされているが、真の増分ではない。1つのブロックだけが変更された場合でも表全体のバックアップがとられます。
破損ブロックの検出	サポートされている。破損ブロックを特定し、V\$BACKUP_CORRUPTION または V\$COPY_CORRUPTION に書き込みます。	サポートされていない。	サポートされている。エクスポート・ログ内で、破損ブロックを特定します。
データの自動的なバックアップ	サポートされている。バックアップ対象のすべてのファイル（データベース全体または表領域、データ・ファイル、制御ファイルのバックアップ）の名前と位置を設定します。	サポートされていない。バックアップ対象のファイルは、手動で指定しなければなりません。	サポートされている。全体バックアップまたはユーザー・バックアップ、表バックアップを実行します。
実行済みのバックアップのカタログ化	サポートされている。バックアップ・データはリカバリ・カタログと制御ファイルにカタログ化されるか、または制御ファイルだけにカタログ化されます。	サポートされていない。	サポートされていない。

表 5-2 各バックアップ方法の機能の比較（続き）

機能	Recovery Manager	オペレーティング・システム	Export
テープへのバックアップ	サポートされている。メディア・マネージャとインタフェースをとります。	サポートされている。テープへのバックアップは手動で行われるか、またはメディア・マネージャによって管理されます。	サポートされている。
init.ora ファイルとパスワード・ファイルのバックアップ	サポートされていない。	サポートされている。	サポートされていない。
オペレーティング・システムに依存しない言語	O/S に依存しないスクリプト言語。	O/S に依存する。	O/S に依存しないスクリプト言語。

第 II 部

バックアップおよびリカバリ
－ Recovery Manager

Recovery Manager のスタート・ガイド

この章は Recovery Manager の使用方法を説明する「クイック・スタート」で、次の項目を取り挙げます。

- Recovery Manager を使用する前に決めておくべきこと
- Recovery Manager の接続オプション
- Recovery Manager のコマンドの実行
- Recovery Manager のサンプル・スクリプトと使用例
- テープにバックアップをとる場合の前提条件

Recovery Manager を使用する前に決めておくべきこと

Recovery Manager の使用を開始する前に、次のことを決めておく必要があります。

- リカバリ・カタログを使用するかどうか
- パスワード・ファイルを使用するかどうか
- init.ora ファイルとパスワード・ファイルのバックアップ方法を決める

リカバリ・カタログを使用するかどうか

リカバリ・カタログを使用する場合と使用しない場合では、それぞれコストと利点が異なります。これらのコストと利点について概要を簡単に説明します。

リカバリ・カタログを使用する場合のコストと利点

リカバリ・カタログを使用すると、Recovery Manager で自動バックアップと回復の拡張機能を実行できます。しかし、Recovery Manager を使用する場合、リカバリ・カタログのスキーマと、そのスキーマで使用する、対応付けられた領域をメンテナンスする必要があります。

リカバリ・カタログを使用する場合は、リカバリ・カタログのスキーマをインストールするために使用するデータベースと、そのデータベースのバックアップ方法を決める必要があります。

リカバリ・カタログのスキーマのサイズは次のようになります。

- カタログでモニターするデータベースの件数に依存する。
- カタログに格納された Recovery Manager スクリプトの件数とサイズに依存する。
- 各データベースのアーカイブ・ログとバックアップの件数の増加に従って、増加する。

Recovery Manager を使って多数のデータベースのバックアップをとる場合、リカバリ・カタログ用のデータベースを別に作成して、そのデータベース内に Recovery Manager ユーザーを作成できます。その場合には、このデータベースを ARCHIVELOG モードで運用するかどうかを決める必要があります。

リカバリ・カタログを個別のデータベースに格納する場合は、次のそれぞれに対してディスク領域が少しずつ必要になります。

- SYSTEM 表領域
- 一時表領域
- ロールバック・セグメント表領域
- オンライン REDO ログ

リカバリ・カタログを個別にメンテナンスした場合、DBA の自由裁量で使用不可にできるという別の利点があります。このデータベースで使用する領域のほとんどは、表領域のサポート (SYSTEM、一時、ロールバックなど) に使われます。

Table 6-1 に、1 年間に必要となる領域の一般的な例を示します。

表 6-1 1 年間に必要となるリカバリ・カタログの領域の一般的な例

領域のタイプ	必要な領域
SYSTEM	50MB
一時	5MB
ロールバック	5MB
リカバリ・カタログ	10MB
オンライン・ログ	各 1MB(3つのグループがあり、各グループには2つのメンバーがある場合)

バックアップをとるデータベースが複数ある場合は、複数のリカバリ・カタログを作成して、各データベースを別のデータベースのリカバリ・カタログ用に使用できます。たとえば、2つの稼働データベースがあり、一方を "prd1"、他方を "prd2" と呼ぶとします。その場合、"prd1" 用のリカバリ・カタログを "prd2" データベースにインストールし、"prd2" データベース用のリカバリ・カタログを "prd1" にインストールできます。これにより、リカバリ・カタログ用の個別のデータベースをメンテナンスする場合に余分に必要となる領域やメモリーのオーバーヘッドを節約できます。ただし、両方のリカバリ・カタログ用のデータベースが、同じ物理ディスク上の表領域に常駐する場合には、この方法は実用的ではありません。

注意： また、リカバリ・カタログのスキーマは、バックアップ対象のターゲット・データベースとは別のデータベースにインストールしなければなりません。そうしないと、データベースが消失したために復元が必要になったときに、リカバリ・カタログを使用できません。

警告： リカバリ・カタログとターゲット・データベースが、同じディスク上に常駐していないことを確認してください。同じディスク上にあると、どちらか一方が消失した場合に、他方も失うことになります。

関連項目：詳細は、第7章の「Recovery Manager の概要」を参照してください。

リカバリ・カタログを使用しない場合のコストと利点

リカバリ・カタログを使用しない場合、次の回復機能は使用できません。

- 表領域の Point-in-Time 回復
- ストアド・スクリプト
- 制御ファイルが消失または破損したときの回復

リカバリ・カタログを使用せずに、データベースを復元して回復させる場合は、次のことを実行してください。

- メディア障害に対する保護対策として、多重制御ファイル（少なくとも2つのファイル）を使用し、各制御ファイルを別々のディスクにインストールする。
- バックアップをとったファイルの名前およびバックアップをとった日付、各ファイルを書き込んだバックアップ・ピースの名前のレコードを確実に保管しておく。また、Recovery Manager のすべてのバックアップ・ログを保管しておく。

警告： 制御ファイルが消失し、リカバリ・カタログも使用していない場合は、復元と回復は困難です。すべての制御ファイルが消失し、しかもデータ・ファイルの復元と回復が必要な場合、唯一の方法は、オラクル社カスタマー・サポートに連絡することです。カスタマー・サポートには、次のことを知らせる必要があります。

- データベースの現行のスキーマ
- バックアップをとってあるファイルの名前
- ファイルのバックアップをとった時刻
- 各ファイルが含まれているバックアップ・ピースの名前

Recovery Manager は、リカバリ・カタログとともに使用することをお勧めします。

リカバリ・カタログのスキーマの設定

リカバリ・カタログを使用する場合は、そのスキーマを設定する必要があります。リカバリ・カタログのスキーマは、リカバリ・カタログの表領域に設定することをお薦めしますが、必要な場合は、SYSTEM システム表領域にインストールできます。

リカバリ・カタログのスキーマの設定

1. リカバリ・カタログが入っているデータベースに対して、Server Manager(LineMode) の connect internal を (SYSDBA として) 使用します。
2. 次のコマンドを発行します。

```
spool create_rman.log
connect internal
create user rman identified by rman
temporary tablespace temp
default tablespace rcvcat quota unlimited on rcvcat;
grant recovery_catalog_owner to rman;
connect rman/rman
@?/rdlms/admin/catrman
```

3. 続行する前に、エラーがないか create_rman.log ファイルをチェックします。

ここで作成したリカバリ・カタログのスキーマに、次のことがありと仮定します。

- リカバリ・カタログを格納する "rcvcat" と呼ばれる表領域が存在する。
- "temp" と呼ばれる表領域が存在する。
- このデータベースは、「通常の」すべてのデータベースと同じ方法で構成されている (たとえば、catalog.sql と catproc.sql が正常に実行された)

注意： catrman.sql スクリプトを SYS スキーマで実行しないでください。catrman.sql スクリプトは、リカバリ・カタログのスキーマ内で実行してください。

また、リカバリ・カタログのスキーマは、バックアップ対象のターゲット・データベースとは別のデータベースにインストールしなければなりません。そうしないと、データベースが消失したために復元が必要になった場合に、リカバリ・カタログを使用できません。

警告： リカバリ・カタログとターゲット・データベースが同じディスク上に常駐していないことを確認してください。同じディスク上に常駐していると、どちらか一方が消失した場合に、他方も失うことになります。

リカバリ・カタログのバックアップをとる時期と方法を決める

リカバリ・カタログは、定期的にバックアップをとることが大切です。できれば、毎日バックアップをとるようにしてください。

また、リカバリ・カタログのバックアップには、2つの異なる方法を使用してください。たとえば、Recovery Manager と Export ユーティリティの組合せ、または Recovery Manager とオペレーティング・システムのユーティリティの組合せを使用してください。

Recovery Manager を使用してリカバリ・カタログ用のデータベースのバックアップをとる場合は、別のデータベース（バックアップをとる稼働データベースなど）内に 2 番目のリカバリ・カタログを作成してください。リカバリ・カタログ用のデータベース自体にリカバリ・カタログのバックアップをとると、メディア故障が原因でそのデータベースが消失した場合に、復元が難しくなります。

リカバリ・カタログの再同期化方法を決める

リカバリ・カタログの再同期化を自動化して、少なくとも毎日一回は再同期化が実行されるようにしてください。毎日多数のアーカイブ・ログを生成しているサイトでは、1 時間に 1 度のカタログ再同期化が必要です。

実際には、カタログの再同期化にかかる時間は短く、再同期化の周期を短くするほど、個々の再同期化プロセスは高速化されます。

設定したカタログ自動再同期化プロセスが正しく動作するか、少なくとも週に 1 度は検証してください。

注意： データベースの構造的な変更（ファイルの追加や表領域の削除など）を行った場合は、必ず手動でカタログを再同期化してください。

パスワード・ファイルを使用するかどうか

一般的に、セキュリティが施されていない Net8 接続を介してターゲット・データベースに接続するとき、特に次に示すいずれかを行うときは、パスワード・ファイルを使用する必要があります。

- Recovery Manager をリモート位置（バックアップをとるデータベースとは異なるマシン）から実行する場合。たとえば、Oracle EnterpriseManager の Backup Manager の GUI を使用する場合
- Recovery Manager を使用し、またターゲット・データベースの接続文字列で TNS の別名を使用する場合
- Parallel Server データベースにおいて、Recovery Manager のセッションを 1 つだけ使用しているときに、クラスタ内の複数のノードから同時にそのデータベースのバックアップをとる場合

関連項目： 『Oracle8 Parallel Server 概要および管理』 (OPS クラスタの 2 つのノードに分散したバックアップの例)

init.ora ファイルとパスワード・ファイルのバックアップ方法を決める

Recovery Manager は、init.ora ファイルとパスワード・ファイルのバックアップはとりません。これらのファイルをメディア故障から保護する方法を計画する必要があります。

Recovery Manager の接続オプション

この項では、次のように、Recovery Manager の接続オプションのサンプルを示します。

- リカバリ・カタログを使用せずに Recovery Manager に接続する場合
- リカバリ・カタログを使用して Recovery Manager に接続する場合

注意： この項に示す例では、Oracle Enterprise Manager の Backup Manager の GUI は使用していません。また、ここに示す例以外にも、Recovery Manager には、多数の接続オプションがあります。

Recovery Manager への一般的な接続を行う場合 Recovery Manager の接続文字列の一般的な形式を、次に示します。

```
rman target internal/<pwd>@prdl rcvcat rman/rman@rcat
```

リカバリ・カタログを使用せずに Recovery Manager に接続する場合

この例では、次のことを前提としています。

- ターゲット・データベースを "prod1" と呼ぶ（また、同じ TNS の別名を持つ）
- "scott" には SYSDBA の権限が付与されている（パスワードは "tiger"）

ターゲット・データベースにパスワード・ファイルがない場合は、ログインに使ったユーザー名が有効かどうかを、O/S の認証機能を使用して検証する必要があります（これは svrmgrl や connect internal の場合と同じです）。

パスワード・ファイルを使用せずに Recovery Manager に接続する場合 ORACLE_SID をターゲット・データベースに設定して、次の文を発行します。

```
rman nocatalog
RMAN> connect target
```

パスワード・ファイルを使用して RecoveryManager に接続する場合 ターゲット・データベースにパスワード・ファイルがある場合、次の文を使用して接続できます。

```
rman target internal/kernel@prod1 nocatalog
```

または

```
rman nocatalog
RMAN> connect target internal/kernel@prod1
```

Recovery Manager は自動的に、SYSDBA としてターゲット・データベースへの接続を要求します。

SYSDBA の権限を持つ DBA ユーザーとして接続する場合は、次の文を発行します。

```
rman target scott/tiger@prod1 nocatalog
```

リカバリ・カタログを使用して Recovery Manager に接続する場合

この例では、次を前提としています。

- リカバリ・カタログ・データベースを "rcat" と呼ぶ（また、同じ TNS の別名を持つ）
- リカバリ・カタログが含まれているスキーマは、"rman" である（パスワードは、"rman"）
- ターゲット・データベースを "prod1" と呼ぶ（また、同じ TNS の別名を持つ）
- "scott" には SYSDBA の権限が付与されている（パスワードは "tiger"）

パスワード・ファイルを使用して接続する場合 次の文では、管理者は、ターゲット・データベースのパスワード・ファイルを使用して接続を行っており、また、内部接続のパスワードは 'kernel' です。

```
rman target internal/kernel@prod1 rcvcat rman/rman@rcat
```

パスワード・ファイルを使用せずに接続する場合 次の文では、Recovery Manager はターゲット・データベースと同じマシン上で動作しています。管理者は、ORACLE_SID をターゲット・インスタンスとして設定後、次の文を発行して Recovery Manager を起動します。

```
rman rcvcat rman/rman@rcat  
RMAN> connect target
```

SYSDBA を持つユーザーとして接続する場合 ここでは、Recovery Manager は、パスワード・ファイルは使わずに O/S の認証機能を使用して、SYSDBA としてターゲット・データベースに接続するように自動的に要求します。

```
rman target scott/tiger@prod1 rcvcat rman/rman@rcat
```

Recovery Manager のコマンドの実行

この項では、次の項目について説明します。

- 対話形式による Recovery Manager のコマンドの実行：例 1
- コマンド・ファイルの使用：例 2
- ストアド・スクリプトの使用：例 3
- Recovery Manager で時刻パラメータを指定する場合

カタログ・メンテナンス用のコマンドは、次に示すように、**run** コマンドの外部で実行されます。

```
rman target internal/kernel@prod1 rcvcat rman/rman@rcat
RMAN> resync catalog;
```

backup および **restore**、**allocate** などのコマンドは、次に示すように、**run** コマンドの内部で実行する必要があります。

```
rman target internal/kernel@prod1 rcvcat rman/rman@rcat
RMAN> run {
2> allocate channel c1 type disk;
3> copy datafile 5 to '/dev/backup/prod1/prod1_tab7.dbf';
4> }
```

対話形式による Recovery Manager のコマンドの実行：例 1

Recovery Manager のコマンドを対話形式で実行するには、Recovery Manager の起動後、コマンド行インタフェースに次のようにコマンドを入力する必要があります。

```
rman target internal/kernel@prod1 rcvcat rman/rman@rcat
RMAN> register database;
RMAN> run {
2>   allocate channel d1 type disk;
3>   allocate channel d2 type disk;
4>   allocate channel d3 type disk;
5>   backup
6>     incremental level 0
7>     tag db_level_0
8>     filesperset 6
9>     format '/dev/backup/prod1/df/df_t%t_s%s_p%p'
10>    (database);
11>   sql 'alter system archive log current';
12>   backup
13>     filesperset 20
14>     format '/dev/backup/prod1/al/al_t%t_s%s_p%p'
15>    (archivelog all
16>     delete input);
17> }
```

コマンド・ファイルの使用：例 2

Recovery Manager のコマンドをファイルに入力しておく、コマンド・ファイル名をコマンド行で指定することにより、そのコマンド・ファイルを実行できます。

コマンド・ファイルの内容は、コマンド行に入力したものと同一になります。

ここでは、例 1 の Recovery Manager のスクリプトが、ファイル ("b_whole_10.rcv" と呼ぶ) に保管されました。このファイルは、次のように、コマンド行から実行できます。

```
rman target internal/kernel@prodl rcvcat rman/rman@rcat cmdfile b_whole_10.rcv
```

ストアド・スクリプトの使用：例 3

ストアド・スクリプトとは、リカバリ・カタログに格納された Recovery Manager の一連のコマンド（大カッコで囲まれています）のことです。1つのストアド・スクリプトに対応付けられているデータベースは、1つだけです。リカバリ・カタログを使用していない場合は、ストアド・スクリプトは使用できません。

ストアド・スクリプトを作成するには、スクリプトを対話形式で Recovery Manager のコマンド行インタフェースに入力するか（例 1）、または Recovery Manager のコマンドをコマンド・ファイルに入力して、そのコマンド・ファイルを実行します（例 2）。

ストアド・スクリプトの例を次に示します。

```
replace script b_whole_10 {
# Backup Whole database, and archived logs
allocate channel d1 type disk;
allocate channel d2 type disk;
allocate channel d3 type disk;
backup
  incremental level 0
  tag b_whole_10
  filesperset 6
  format '/dev/backup/prodl/df/df_t%t_s%s_p%p'
  (database);
sql 'alter system archive log current';
backup
  filesperset 20
  format '/dev/backup/prodl/al/al_t%t_s%s_p%p'
  (archivelog all
  delete input);
}
```

Recovery Manager で時刻パラメータを指定する場合

Recovery Manager を起動する前に、NLS_DATE_FORMAT 環境変数と NLS_LANG 環境変数を設定してください。これらの変数は、次のような Recovery Manager コマンドの時刻パラメータで使用する形式を指定します。

- **restore** および **recover**
- **report**

次の例は、代表的な環境変数を示しています。

```
NLS_LANG=american
NLS_DATE_FORMAT='Mon DD YYYY HH24:MI:SS'
```

注意： NLS_LANG と NLS_DATE_FORMAT は、使用する NLS_DATE_FORMAT に合わせて設定してください。

Recovery Manager のサンプル・スクリプトと使用例

RDBMS デモ・ディレクトリには多数のサンプル・スクリプトがあります。これらのファイルは、Recovery Manager の実行可能なコマンド・ファイルで、各コマンドの機能がわかるようにコメントが付いています。これらのファイルは編集できるので、サイトにあわせてカスタマイズできます。

case1.rcv ファイルは、データベースのバックアップおよび復元、回復を実行する多数のストアド・スクリプトを作成します。これらのスクリプトは、DBA がデータベースのバックアップを作成する一般的な方法を示しており、ユーザーはこれを使ってバックアップの作成や、スクリプトの復元および回復を実行できます。

次の各ファイルには、バックアップまたは回復の使用例が入っています。これらのスクリプトは、サンプル・スクリプトよりもさらに柔軟な Recovery Manager の構文や機能を示しています。

case2.rcv - 不完全回復
 case3.rcv - 障害発生時の回復
 case4.rcv - 一貫性バックアップ

関連項目： その他のバックアップと回復の例については、Chapter 9, “Recovery Manager の使用例” を参照してください。

テープにバックアップをとる場合の前提条件

テープなどの順次メディアをターゲットまたはソースとして使って、バックアップや復元を実行する場合は、Oracle にメディア・マネージャを組み込む必要があります。

この項では、次の項目を取り挙げます。

- メディア・マネージャとのリンク
- 一意のファイル名の生成
- メディア・マネージャにおけるファイルの最大サイズの制限

メディア・マネージャとのリンク

Oracle にメディア・マネージャを組み込む場合は、次のことを実行しなければなりません。

- メディア・マネージャのソフトウェアとハードウェアのインストールと構成
- メディア管理ソフトウェア・ベンダーからのライブラリ・インタフェース・ソフトウェア（メディア管理ライブラリ）の取得

次に、メディア管理ライブラリ (MML) を Oracle のカーネル・ソフトウェアにリンクします。これによって、Oracle サーバー・プロセスがメディア・マネージャをコールできるようになります。

注意： プラットフォーム上でこれを行う方法は、それぞれのオペレーティング・システム用の Oracle マニュアルと、メディア・マネージャが提供するマニュアルを参照してください。

一意のファイル名の生成

バックアップ・データをメディア・マネージャに書き込む時は、**RecoveryManager** が提供する置換変数を使用して、一意のバックアップ・ピース名を生成する必要があります。バックアップ・ピース名は、**backup** コマンド、または **allocate channel** コマンドで指定された形式文字列によって判別されます。

メディア・マネージャは、バックアップ・ピース名を「バックアップ済ファイル名」と認識するため、この名前はメディア・マネージャのカatalog内で一意の名前でなければなりません。

警告： バックアップ時に、メディア・マネージャのカatalog内にすでに存在する名前を持つファイルが生成されると、その名前を持つ元のファイルが削除されてしまいます。

次の置換変数を使用して、一意の形式文字列を作成できます。

%d - データベース名 (大文字)

%t - バックアップ・セット・スタンプ

%s - バックアップ・セット番号

%p - バックアップ・セット内でのピース番号

%u - バックアップ・セット番号と、バックアップ・セットの作成時間を圧縮形式で表す 8 文字の id

注意： 一部のメディア・マネージャでは、14 文字のバックアップ・ピース名しかサポートされません。使用中のメディア管理ソフトウェアのマニュアルを参照して、使用しているメディア・マネージャの制限を確認してください。

メディア・マネージャにおけるファイルの最大サイズの制限

一部のメディア・マネージャでは、バックアップまたは復元できるファイルの最大サイズに制限があります。

Recovery Manager は複数のデータ・ファイルを 1 つのファイル (1 つのバックアップ・ピース) に多重化するため、バックアップ・ピースのサイズが、多くのメディア・マネージャやファイル・システムで格納できるサイズを超えてしまうことがあります。

後で問題が起こらないように、ファイル・サイズの操作可能最大値を調べ、RecoveryManager で書き込んだファイルがその最大値を超えていないことを確認してください。バックアップ・ピース・ファイルのサイズを制限するには、**setlimit channel** コマンドでキーワード **kbytes** を使用してください (入力例は、デモ・ディレクトリ内の使用例 `case1.rcv` を参照してください)。

Recovery Manager の概要

この章では、Oracle の Recovery Manager ユーティリティの基本概念を説明します。主な項目は次のとおりです。

- Recovery Manager の概要
- 順次メディアへのバックアップ
- Recovery Manager によるバックアップのタイプ
- 破損の検出
- パラレル化
- レポート生成
- Recovery Manager によるバックアップのためのユーザー・タグ
- バックアップの制約
- 復元の制約
- 整合性チェック
- Recovery Manager でのオープン状態のデータベースのバックアップ時における分裂ブロックの検出
- イメージ・コピーおよびアーカイブ・ログのカタログ化

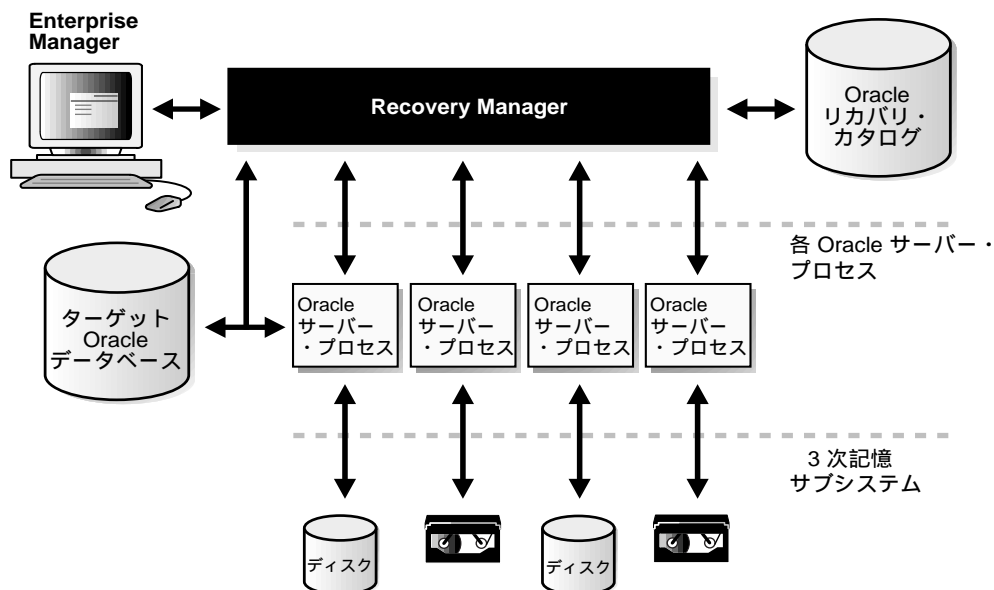
Recovery Manager の概要

Recovery Manager は、データベース・ファイルのバックアップ、復元、回復に使う Oracle のユーティリティです。Recovery Manager は、バックアップまたは復元の対象となるデータベース（ターゲット・データベースといいます）に対して Oracle サーバー・プロセスを起動します。バックアップや復元を実際に行うのはこれらの Oracle サーバー・プロセスです。たとえば、バックアップ時には、サーバー・プロセスが、バックアップ対象のファイルを読み取って 3 次記憶デバイスに書き込みます。

バックアップと回復の重要な作業が Recovery Manager によって実行されるので、バックアップや回復のときに管理者が行う作業を大幅に省力化できます。Recovery Manager を使うと、たとえば次のようなことができます。

- 頻繁に実行するバックアップ操作を設定する。
- バックアップおよび回復にかかわるすべてのアクションについて印刷可能ログを生成する。
- リカバリ・カタログを使って、復元操作および回復操作の両方を自動化する。
- バックアップおよび復元を自動パラレル化する。
- REDO の量についてユーザーが指定した適用必須の制限に基づき、バックアップが必要なデータ・ファイルを検索する。
- データベース全体または選択した表領域、選択したデータ・ファイルのバックアップをとる。

図 7-1 Recovery Manager



Recovery Manager は、コマンド言語インタプリタ (CLI) を備えており、対話モードでもバッチ・モードでも実行できます。コマンド行でログ・ファイルを指定して、そのログ・ファイルに Recovery Manager の重要なアクションを記録することもできます。Enterprise Manager の Recovery Manager アプリケーションを介して Recovery Manager を使うこともできます。

バッチ・モードでは、Recovery Manager はコマンド・ファイルの内容を入力として読み取り、ログ・ファイル（指定されている場合）に出力メッセージを書き込みます。このコマンド・ファイル全体の解析後に、コマンドがコンパイルされて実行されます。オペレーティング・システムのジョブ制御機能を使って、計画に基づき定期的にバックアップをとる場合は、バッチ・モードが最適です。

Recovery Manager のコマンドには、次の種類があります。

- **backup** および **restore**、**copy**、**recover**
- リカバリ・カタログのメンテナンス用コマンド
- ストアド・スクリプトのメンテナンス用コマンド
- **report** コマンドおよび **list** コマンド

順次メディアへのバックアップ

テープなどの順次メディアへバックアップをとるために Recovery Manager を使う場合は、Oracle ソフトウェアが統合されたメディア管理ソフトウェアが必要です。

図 7-2 MML のアーキテクチャ

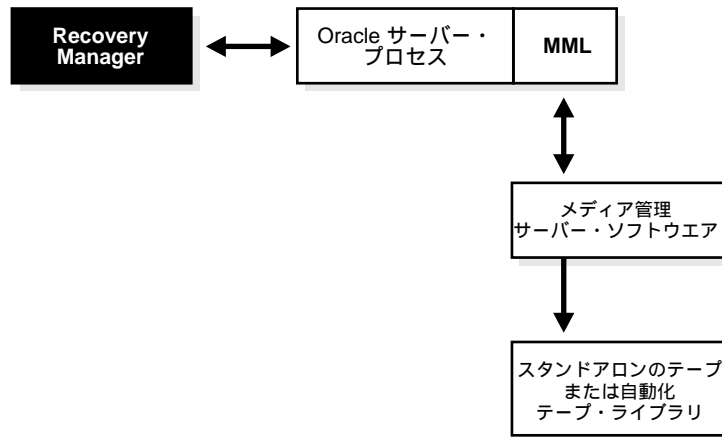


図 7-2 では、Oracle 実行可能ファイルは、ユーザーがデータベースに接続したときに起動されたのと同じ実行可能ファイルです。MML とは、メディア管理ライブラリ (Media Management Library) のことで、これはベンダーが提供するメディア管理ソフトウェアです。MML 内のソフトウェア・ルーチンは、Oracle からコールされ、バックアップと復元を実行します。

次の例では、Recovery Manager スクリプトによってデータ・ファイルのバックアップを実行します。

```
run {
  allocate channel t1 type 'SBT_TAPE';
  backup
    format 'df_%s_%t'
  (datafile 10);
}
```

Recovery Manager でこのコマンドを実行すると、Recovery Manager から Oracle サーバー・プロセスにバックアップ要求が送られ、その Oracle サーバー・プロセスでバックアップが実行されます。Oracle サーバー・プロセスでは、出力チャンネルがタイプ 'SBT_TAPE' として識別され、MML に対し、テープをロードして出力を書き込むよう要求が出されます。

チャンネルのタイプがディスクの場合、Oracle ではサード・パーティの統合ソフトウェアを介さなくてもディスクに書き込めるので、Oracle は MML ソフトウェアに接続されません。

関連項目 :Recovery Manager を統合したソフトウェアを提供しているベンダーについては、オラクル社の営業担当員に問い合せてください。

Enterprise Manager を使ったバックアップ

Recovery Manager へのグラフィカル・ユーザー・インタフェースである Oracle Enterprise Manager の Backup Manager ソフトウェアを使って、Recovery Manager のバックアップを実行できます。

関連項目 : 詳細は、それぞれのオペレーティング・システム用のマニュアルを参照してください。

リカバリ・カタログ

リカバリ・カタログは情報のリポジトリで、Recovery Manager が使用し、管理します。Recovery Manager は、リカバリ・カタログの情報に基づいて、要求されたバックアップと復元の作業を実行する方法を判別します。

リカバリ・カタログには、次のものに関する情報が入っています。

- データ・ファイルおよびアーカイブ・ログのバックアップ・セットとバックアップ・ピース
- データ・ファイルのコピー
- アーカイブ REDO ログおよびそのコピー
- ターゲット・データベースにある表領域およびデータ・ファイル
- ストアド・スクリプトと呼ばれる、ユーザーが作成した名前付きの一連の Recovery Manager コマンドおよび SQL コマンド

リカバリ・カタログを使った操作

Oracle では、特にデータ・ファイルが 20 個以上ある場合に、Recovery Manager でリカバリ・カタログを使うことをお勧めします。リカバリ・カタログは、ターゲット・データベース制御ファイルと頻繁に再同期させて、常に最新の状態にしておく必要があります。リカバリ・カタログを最新の状態にしておくほど、回復が簡単になります。たとえば、前回の再同期化後に作成された 25 個のアーカイブ・ログがあり、データベース全体と制御ファイルを回復しなければならないような障害が起きた場合、それらのファイルを使えるようにするには、まずリカバリ・カタログを使ってそれらのファイルをカタログ化する必要があります。

リカバリ・カタログは Recovery Manager だけがメンテナンスします。リカバリ・カタログがターゲット・データベースから直接アクセスされることはありません。Recovery Manager は、データベース構造およびアーカイブされた REDO ログ、バックアップ・セット、データ・ファイル・コピーに関する情報をターゲット・データベースの制御ファイルからリカバリ・カタログの中に波及させます。

リカバリ・カタログ専用のデータベースを追加する必要はありません。リカバリ・カタログは、既存のデータベースに格納できます。リカバリ・カタログのデータベースを **Recovery Manager** から使用できるようにするのは、データベース管理者の責任です。また、リカバリ・カタログのバックアップをとるのも、データベース管理者の責任です。リカバリ・カタログは **Oracle** データベースにあるので、管理者は、**Recovery Manager** を使ってリカバリ・カタログのデータベースとターゲット・データベースの役割を入れ替えることにより、リカバリ・カタログのバックアップをとることができます。つまり、ターゲット・データベースはリカバリ・カタログのデータベースとして使用でき、リカバリ・カタログのデータベースはターゲット・データベースとして使用できます。

1 つのリカバリ・カタログに、複数のターゲット・データベースに関する情報を格納できません。

リカバリ・カタログが壊れた場合に、有効なバックアップがなくても、現行の制御ファイルまたは制御ファイルのバックアップを使用すれば、カタログの一部は回復できます。ただし、リカバリ・カタログの有効な最新のバックアップがあるようにしてください。

制御ファイルからの情報の伝播

ターゲット・データベースの制御ファイルのサイズは、次に示すものの数に応じて大きくなります。

- バックアップを実行した回数
- 作成したアーカイブ・ログの数
- 制御ファイル内にその情報を格納しておく最低日数

CONTROL_FILE_RECORD_KEEP_TIME パラメータを使って、制御ファイル内に情報を保持する最低日数を指定できます。この日数より古いエントリは、新しい情報によって上書きされる候補となります。**CONTROL_FILE_RECORD_KEEP_TIME** の設定値が大きいほど、制御ファイルのサイズも大きくなります。

リカバリ・カタログは、最低でも **CONTROL_FILE_RECORD_KEEP_TIME** の設定値より短い間隔で再同期化してください。その理由は、この日数の経過後、制御ファイル内の情報は最も新しく作成された情報で上書きされるからです。再同期化しないうちに情報が上書きされてしまった場合には、その情報をリカバリ・カタログに波及させることはできません。

注意： 制御ファイルの最大サイズは、ポートによって異なります。それぞれのオペレーティング・システム用の **Oracle** マニュアルを参照してください。

関連項目： **CONTROL_FILE_RECORD_KEEP_TIME** パラメータの詳細は、『**Oracle8 Server** リファレンス・マニュアル』を参照してください。

リカバリ・カタログなしでの運用

注意： Recovery Manager では、リカバリ・カタログを必ず使用しなければならぬわけではありませんが、リカバリ・カタログを使うことをお薦めします。

リカバリ・カタログ内のほとんどの情報は、ターゲット・データベースの制御ファイルからも取得できるので、Recovery Manager ではリカバリ・カタログのかわりにターゲット・データベースの制御ファイルを使う運用モードをサポートしています。データベースのサイズが小さい場合は、リカバリ・カタログだけのために別のデータベースをインストールして管理するのはかなりの負担です。データベースのサイズが小さい場合には、この運用モードが適しています。

この運用モードでは次の機能はサポートされないので、注意してください。

- 表領域の Point-in-Time 回復
- ストアド・スクリプト
- 制御ファイルが消失または破損したときの復元と回復

リカバリ・カタログを使わずにデータベースの復元と回復を行うには、次のことを実行してください。

- 複数の多重制御ファイル（最低でも 2 つ）を使い、メディア障害に備えて個々のファイルを別々のディスクに入れておく。
- どのファイルのバックアップをとったか、バックアップをとった日付、個々のファイルが書き込まれたバックアップ・ピースの名前を確実に記録しておく。Recovery Manager のバックアップ・ログをすべてとっておく。

警告： 制御ファイルが消失し、リカバリ・カタログも使用していない場合は、復元と回復は困難です。すべての制御ファイルをなくしてしまい、データ・ファイルを復元して回復する必要がある場合、復元と回復を可能にする唯一の方法は、オラクル社カスタマー・サポートに連絡することです。カスタマー・サポートでは、次の情報が必要になります。

- データベースの現行のスキーマ
 - どのファイルのバックアップをとったか
 - ファイルのバックアップをとった時刻
 - 各ファイルが入っているバックアップ・ピースの名前
-

制御ファイル内での情報の保持

ターゲット・データベースの制御ファイルのサイズは、次に示すものの数に応じて大きくなります。

- バックアップを実行した回数
- 作成したアーカイブ・ログの数
- 制御ファイル内にその情報を格納しておく最低日数

CONTROL_FILE_RECORD_KEEP_TIME パラメータを使って、制御ファイル内に情報を保持する最低日数を指定できます。この日数より古いエントリは、新しい情報によって上書きされる候補となります。CONTROL_FILE_RECORD_KEEP_TIME が大きいほど、制御ファイルも大きくなります。

リカバリ・カタログを使わない場合でも、CONTROL_FILE_RECORD_KEEP_TIME パラメータにバックアップ保持期間（日数）を設定しておくことができます。こうすると、バックアップが古くなる前に制御ファイル内のバックアップ情報が上書きされるのを防止できます。

注意： 制御ファイルの最大サイズは、ポートによって異なります。それぞれのオペレーティング・システム用の Oracle マニュアルを参照してください。

関連項目： CONTROL_FILE_RECORD_KEEP_TIME パラメータの詳細は、『Oracle8 Server リファレンス・マニュアル』を参照してください。

スナップショット制御ファイル

Recovery Manager は、リカバリ・カタログをリフレッシュするたびに、一時的なバックアップ制御ファイルであるスナップショット制御ファイルを生成します。このスナップショット制御ファイルにより、リカバリ・カタログをリフレッシュするとき、または制御ファイルに問い合わせるとき（リカバリ・カタログを使っていない場合）に、制御ファイルの一貫性のあるビューが保証されます。スナップショット制御ファイルは、Recovery Manager が一時的に使用するためのものなので、リカバリ・カタログには登録されません。Recovery Manager は、スナップショット制御ファイル・チェックポイントをリカバリ・カタログに記録することにより、リカバリ・カタログがどの程度まで最新の状態であるかを正確に示します。

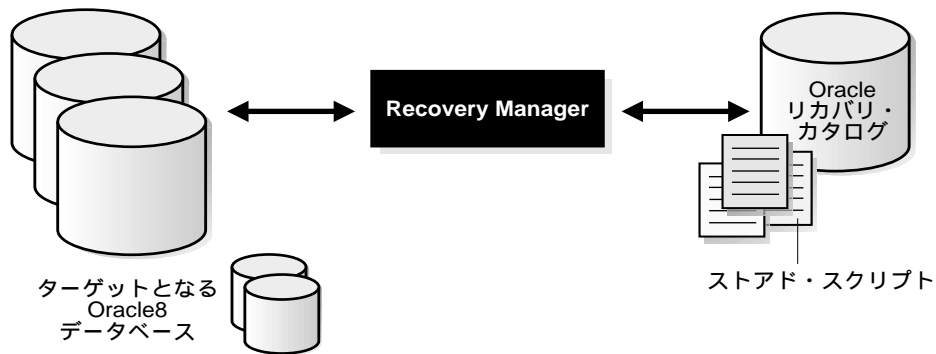
Oracle8 サーバーは、1つのスナップショット制御ファイルに対しては、1度に1つの Recovery Manager セッションしかアクセスできないよう制限しています。これは、2つの Recovery Manager セッションの間でスナップショット制御ファイルの使用に関する競合が発生するのを防ぐためです。

注意： スナップショット制御ファイルの名前と位置を指定できます。その方法については、8-19 ページの「スナップショットの制御ファイルの位置の設定」の項を参照してください。

ストアド・スクリプト

データベース管理者は、ストアド・スクリプトと呼ばれる一連の Recovery Manager コマンドをリカバリ・カタログに格納しておいて、後で実行できます。この方法によって、管理者は、データベースのバックアップおよび復元、回復のための一連のコマンドを計画および開発、テストできます。ストアド・スクリプトを使うと、オペレータ・エラーを最小限に抑えることができます。1つのストアド・スクリプトは1つのデータベースだけに関連付けられます。

図 7-3 Recovery Manager のストアド・スクリプト



Recovery Manager によるバックアップのタイプ

Recovery Manager は、次の 2 種類の基本的なバックアップをサポートしています。

- バックアップ・セット
- イメージ・コピー

バックアップ・セット

バックアップ・セットには、1つ以上のデータ・ファイルかアーカイブ・ログのいずれかが入っています。ただし、1つのバックアップ・セットに、データ・ファイルとアーカイブ・ログを混在させることはできません。データ・ファイルのバックアップ・セットには、制御ファイルのバックアップも入れることができます。

バックアップ・セットは、Oracle 専用形式（エクスポート・ファイルと同様）になっています。バックアップ・セットからファイルを取り出すには、復元操作が必要です。

バックアップ・セットは、**backup** コマンドで指定したオブジェクトの全体バックアップまたは増分バックアップを構成するバックアップ・ピースの完全なセットです。

1つのバックアップ・セット内の複数のピースへの書き込みは直列で行われます。(1つのバックアップ・セットを複数の出力デバイスに分割することはできません。) 複数の出力デバイ

スが使用できる場合は、バックアップをパーティション化して、複数のバックアップ・セットをパラレルで作成できます。Recovery Manager を使えば、必要な場合には、このバックアップのパーティション化を自動的に実行できます。

バックアップ・セットは、

- ディスクまたは 3 次記憶装置に書き込みます。
- 全体バックアップまたは増分バックアップの場合があります。
- 複数の O/S ファイルにまたがることもできます（個々のファイルがバックアップ・セットの 1 つのピースになります。通常、1 つのセットは 1 つのピースからなります）。

バックアップ・セットは、通常のアペレーティング・システム・ディスク・ファイル、または順次出力メディア（たとえば、磁気テープ）に書き込むことができます。この書込みには、使用中のアペレーティング・システムと Oracle の両方がサポートしている順次出力デバイスまたはメディア管理システムを使用します。使用中のアペレーティング・システム、またはメディア・マネージャがサポートしているデバイス・タイプを知りたい場合は、V\$BACKUP_DEVICE ビューで問い合せてください。

アーカイブ・ログが入っているバックアップ・セットを、アーカイブ・ログのバックアップ・セットといいます。テープに直接アーカイブすることはできません。ただし、Recovery Manager を使用すれば、ディスクからテープにアーカイブ・ログのバックアップを作成できます。また、回復時には、Recovery Manager により、必要なアーカイブ・ログをテープからディスクに自動的に移せます。

関連項目： V\$BACKUP_DEVICE の詳細は、『Oracle8 Server リファレンス・マニュアル』を参照してください。

バックアップ・ピース

バックアップ・セットは 1 つ以上のバックアップ・ピースで構成されています。各ピースはそれぞれ 1 つの出力ファイルです。バックアップ・ピースのサイズは、使っているメディア・マネージャまたは O/S がサポートするファイル・サイズのどちらか大きい方の最大サイズに制限してください（ディスクに書き込む場合）。このサイズを制限しないと、バックアップ・セットはただ 1 つのファイルから構成されます。

各バックアップ・ピースには制御情報およびチェックサム情報があるので、Oracle サーバー・プロセスは、これらの情報に基づいて、復元時にバックアップ・ピースの正確性を検査できます。

バックアップ・セットの圧縮

未使用のデータ・ファイル・ブロックは、バックアップ・セットには書き込まれません。データ・ファイルのイメージ・コピー・バックアップには、常にすべてのデータ・ファイル・ブロックが含まれています。

ファイルの多重化

同じバックアップ・セットに含まれるデータ・ファイル・ブロックはまとめて多重化されます。つまり、セット内では、すべてのファイルのブロックが他のファイルのブロックと一緒に散在しています。

filesperset パラメータを使って、1つのバックアップ・セットに同時にバックアップをとるデータ・ファイルの数を指定できます。1つのデータ・ファイルが多数の読取り要求で一杯になることなく（一杯になるとオンライン・パフォーマンスが低下します）、テープ・デバイスのストリーミングを続ける必要がある場合は、並行性を指定してください。読取り率は、**set limit readrate channel** コマンドを使って制限できます。

関連項目： 詳細は、7-21 ページの「多重バックアップ・セット」の項を参照してください。

全体バックアップ・セットと増分バックアップ・セット

データ・ファイルのバックアップ・セットには、全体と増分の2種類があります。全体バックアップは、1つ以上のデータ・ファイルのバックアップで、これには該当するデータ・ファイルのすべてのブロックが含まれます。増分バックアップも1つ以上のデータ・ファイルのバックアップですが、これには、前回のバックアップ以降に変更されたブロックだけが含まれます。これらのバックアップ・セットの概念については、次の各項で詳しく説明します。

全体バックアップ・セット

全体バックアップでは、未使用のデータ・ファイル・ブロックだけがスキップされ、すべてのブロックがバックアップ・セットにコピーされます。アーカイブ・ログまたは制御ファイルのバックアップをとるときには、どのブロックもスキップされません。

全体バックアップはデータベース全体のバックアップとは異なります。全体とは、このバックアップが増分バックアップでないことを示します。

また、全体バックアップの実行はその後に行う増分バックアップに対しては効果がなく、全体バックアップは増分バックアップ計画の一部とはみなされません（つまり、全体バックアップによって、その後に行う増分バックアップに含まれるブロックは影響を受けません）。

Oracle では、次のものの全体バックアップを作成および復元できます。

- データ・ファイル
- データ・ファイルのコピー
- 表領域
- 制御ファイル（現行またはバックアップ）
- データベース

アーカイブ・ログのバックアップ・セットは、常に全体バックアップです。

増分バックアップ・セット

増分バックアップは、1 つ以上のデータ・ファイルのバックアップで、これには、同じレベルかそれより低いレベルでの前回のバックアップ以降に変更されたブロックだけが含まれます。未使用のブロックは書き込まれません。

増分バックアップ・セットには、制御ファイルを入れることができますが、制御ファイルは常に完全な形で書き込まれます（つまり、常に制御ファイル全体が書き込まれ、ブロックがスキップされることはありません）。

Oracle では、次のものの増分バックアップを作成および復元できます。

- データ・ファイル
- 表領域
- データベース

マルチレベル増分バックアップ機能を使うと、いくつかの異なるレベルの増分バックアップを作成できます。各レベルはそれぞれ 1 つの整数で表されます。デフォルトでは、あるレベルの増分バックアップには、そのレベルまたはそれより低いレベルでの最後のバックアップ以降に変更されたブロックだけが含まれます。

レベル 0 の増分バックアップでは、データが入っているブロックすべてがコピーされます。レベル 0 の増分バックアップは、それ以降の増分バックアップの基礎となります。

レベル 0 より上のレベルの増分バックアップでは、同じレベル以下での前回の増分バックアップ以降に変更されたブロックだけがコピーされます。したがって、増分バックアップを行う利点は、常にすべてのブロックのバックアップをとらなくて済むことです。0 より上のレベルの増分バックアップでは、変更されたブロックだけがコピーされます。したがって、バックアップ・サイズはかなり小さくなり、必要な時間も大幅に短縮されます。バックアップ・ファイルのサイズは、DBA が選択した増分バックアップ・レベルと、変更されたブロックの数だけに影響されます。

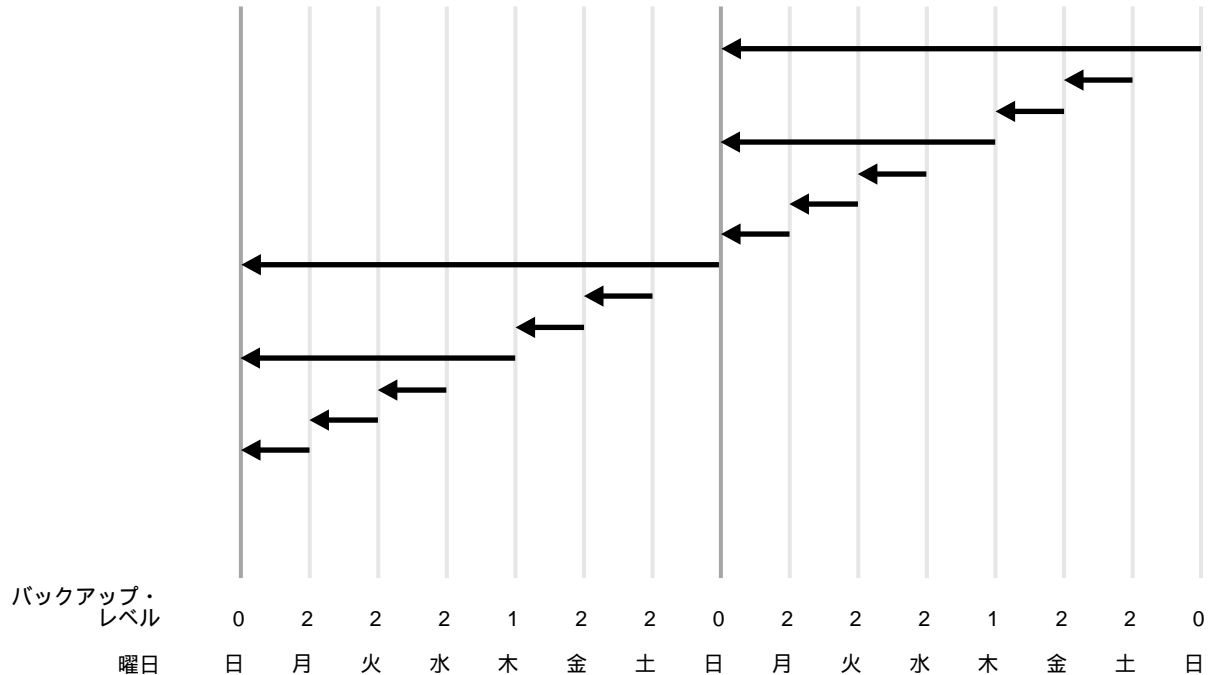
マルチレベル増分バックアップ機能を使うと、できるだけ短い時間内にデータベースを復元できるようにバックアップ方式を工夫できます。たとえば、レベル 0 のバックアップで月に一度全体バックアップを実行し、レベル 1 のバックアップで週に 1 度増分バックアップを実行し、レベル 2 で日に 1 度増分バックアップを実行するといった、3 つのレベルのバックアップ方式を実現できます。

累積増分バックアップ・セット

Oracle には、1 以上のレベルでの累積増分バックアップを作成するためのオプションがあります。累積増分バックアップを使うと、復元時に必要になる特定レベルからの増分バックアップが 1 つだけで済むので、復元に必要な作業が少なくなります。ただし、累積バックアップの場合は、同じレベルでのこれまでのバックアップの作業を繰り返すことになるので、必要な領域と時間が増加します。

累積増分バックアップでは、より低いレベルでの前回の増分バックアップ以降に変更されたすべてのブロックがコピーされます。

図 7-4 非累積増分バックアップ（増分バックアップのデフォルト・タイプ）



上記の例では次のようになります。

- 日曜日
 増分レベル0のバックアップが実行されます。これで、このデータベース内でこれまでに使われたことのあるすべてのブロックのバックアップがとられます。
- 月曜日
 増分レベル2のバックアップが実行されます。これで、レベルn以下の最後の増分バックアップ以降に変更されたすべてのブロックのバックアップがとられます。この例では、レベル2以下の最後の増分バックアップは日曜日のレベル0バックアップなので、日曜日以降に変更されたブロックだけのバックアップがとられます。
- 火曜日
 増分レベル2のバックアップが実行されます。これで、レベルn以下の最後の増分バックアップ以降に変更されたすべてのブロックのバックアップがとられます。この例では、レベル2以下の最後の増分バックアップは月曜日のレベル2バックアップなので、月曜日以降に変更されたブロックだけのバックアップがとられます。

- 水曜日

増分レベル 2 のバックアップが実行されます。これで、レベル n 以下の最後の増分バックアップ以降に変更されたすべてのブロックのバックアップがとられます。この例では、レベル 2 以下の最後の増分バックアップは火曜日のレベル 2 バックアップなので、火曜日以降に変更されたブロックだけのバックアップがとられます。

- 木曜日

増分レベル 1 のバックアップが実行されます。これで、レベル n 以下の最後の増分バックアップ以降に変更されたすべてのブロックのバックアップがとられます。この例では、レベル 1 以下の最後の増分バックアップは日曜日のレベル 0 バックアップなので、日曜日のレベル 0 のバックアップ以降に変更されたブロックだけのバックアップがとられます。

- 金曜日

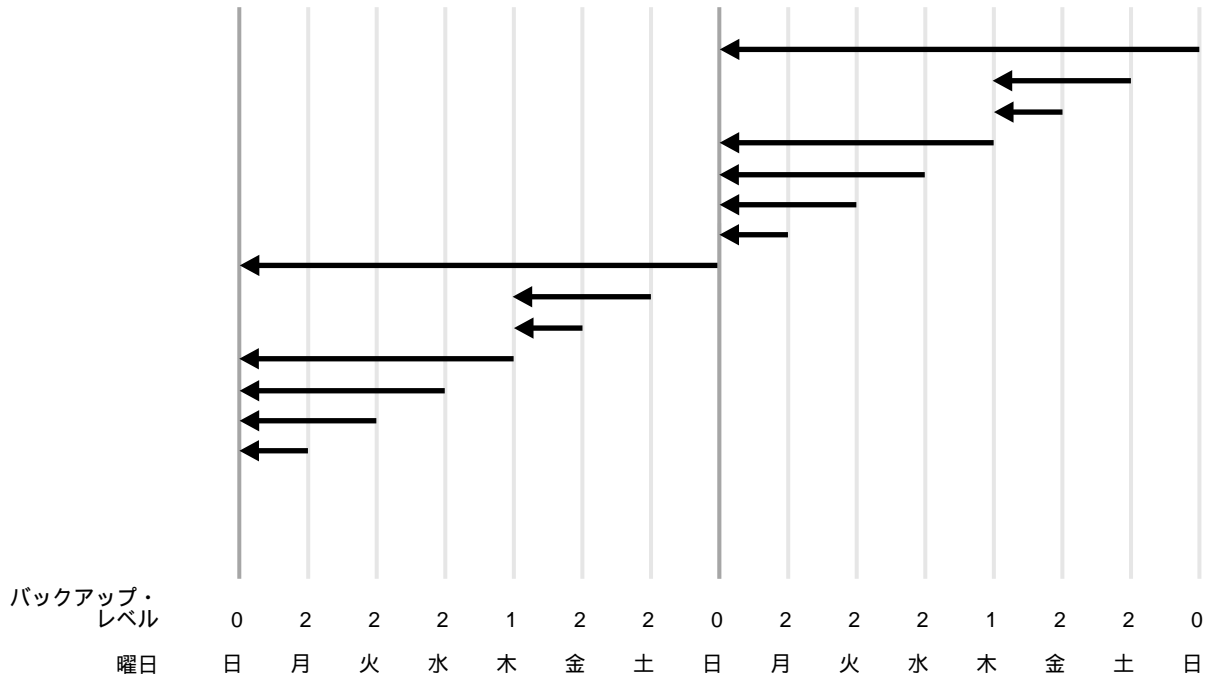
増分レベル 2 のバックアップが実行されます。これで、レベル n 以下の最後の増分バックアップ以降に変更されたすべてのブロックのバックアップがとられます。この例では、レベル 2 以下の最後の増分バックアップは木曜日のレベル 1 バックアップなので、木曜日のレベル 1 バックアップ以降に変更されたブロックだけのバックアップがとられます。

- 土曜日

増分レベル 2 のバックアップが実行されます。これで、レベル n 以下の最後の増分バックアップ以降に変更されたすべてのブロックのバックアップがとられます。この例では、レベル 2 以下の最後の増分バックアップは金曜日のレベル 2 バックアップなので、金曜日のレベル 2 バックアップ以降に変更されたブロックだけのバックアップがとられます。

- このサイクルが繰り返されます。

図 7-5 累積増分バックアップ



上記の例では次のようになります。

- 日曜日
増分レベル0のバックアップが実行されます。これで、このデータベース内でこれまでに使われたことのあるすべてのブロックのバックアップがとられます。
- 月曜日
累積増分レベル2のバックアップが実行されます。これで、レベル n-1 以下の最後の増分バックアップ以降に変更されたすべてのブロックのバックアップがとられます。この例では、レベル 2-1 以下の最後の増分バックアップは日曜日のレベル0 バックアップなので、日曜日以降に変更されたブロックだけのバックアップがとられます。
- 火曜日
累積増分レベル2のバックアップが実行されます。これで、レベル n-1 以下の最後の増分バックアップ以降に変更されたすべてのブロックがバックアップされます。この例では、レベル 2-1 以下の最後の増分バックアップは日曜日のレベル0 バックアップなので、日曜日以降に変更されたブロックのバックアップがとられます。(このバックアップは「累積」なので、月曜日にコピーされたブロックが含まれる他、現行のバックアップと同じ増分レベルでとられたバックアップによりコピーされたブロックが含まれます)。

- 水曜日
累積増分レベル 2 のバックアップが実行されます。これで、レベル n-1 以下の最後の増分バックアップ以降に変更されたすべてのブロックがバックアップされます。この例では、レベル 2-1 以下の最後の増分バックアップは日曜日のレベル 0 バックアップなので、日曜日以降に変更されたブロックのバックアップがとられます。（このバックアップは累積なので、月曜日および火曜日にコピーされたブロックの他、現行のバックアップと同じ増分レベルでとられたバックアップによりコピーされたブロックが含まれます）。
- 木曜日
累積増分レベル 1 のバックアップが実行されます。これで、レベル n-1 以下の最後の増分バックアップ以降に変更されたすべてのブロックのバックアップがとられますこの例では、レベル 1-1 以下の最後の増分バックアップは日曜日のレベル 0 バックアップなので、日曜日以降に変更されたブロックのバックアップがとられます。
- 金曜日
累積増分レベル 2 のバックアップが実行されます。これで、レベル n-1 以下の最後の増分バックアップ以降に変更されたすべてのブロックのバックアップがとられます。この例では、レベル 2-1 以下の最後の増分バックアップは木曜日のレベル 1 バックアップなので、木曜日以降に変更されたすべてのブロックのバックアップがとられます。
- 土曜日
累積増分レベル 2 のバックアップが実行されます。これで、レベル n-1 以下の最後の増分バックアップ以降に変更されたすべてのブロックのバックアップがとられます。この例では、レベル 2-1 以下の最後の増分バックアップは木曜日のレベル 1 バックアップなので、日曜日以降に変更されたすべてのブロックのバックアップがとられます。
- このサイクルが繰り返されます。

イメージ・コピー

イメージ・コピーには、回復のために現状のままで使用できる 1 つのファイル（データ・ファイルまたはアーカイブ・ログ、制御ファイル）が含まれています。イメージ・コピーのバックアップは、1 つのファイルの O/S コピーに似ていますが、Oracle サーバー・プロセスで作成される点が異なります。サーバー・プロセスは、ファイル内のブロックの妥当性検査や、制御ファイル内のコピーの登録などの追加のアクションを実行します。イメージ・コピーは多重化されず、追加のヘッダーやフッターの制御情報がコピー内に格納されないのので、バックアップ・セットとは異なります。

イメージ・コピーのバックアップはディスクにしか書き込めません。

データ・ファイルのイメージ・コピーの場合、復元操作は必要ありません。**switch** コマンドを使ってコピーで制御ファイルを指定し、リカバリ・カタログを更新してそのコピーを「使用した」ことを示します。これは、SQL 文の **ALTER DATABASE RENAME DATAFILE** と同等です。その後でメディア回復を実行し、そのコピーを現行のデータ・ファイルにすることができます。

イメージ・コピーは、コピーの「使用」を防ぐために別の位置に復元できますが、データ・ファイルを改名してイメージ・コピーにできるので、必ずしも復元は必要ありません。

ユーザー作成のイメージ・コピー

Recovery Manager 以外の機能によって作成されたイメージ・コピー (O/S コピーとも呼ばれる) は、サポートされています。ただし、このような O/S コピーを復元操作または切替え操作でできるようにするために、データベース管理者は、O/S コピーを Recovery Manager を使ってカタログ化しておく必要があります。

クローズ状態とオープン状態の両方の O/S コピーがサポートされています。このサポートは、Recovery Manager 以外の機能を使ってデータ・ファイルのイメージ・コピーを作成するユーザーのためのものです。たとえば、ミラー化されたディスクにデータ・ファイルを格納しているサイトがあるとします。この場合、ミラー化を解除するだけでイメージ・コピーを作成できます。ミラー化を解除した後で、Recovery Manager に新しいコピーの存在を知らせて、それを復元に使用できるようにします。そのコピーでは復元できなくなったときは、データベース管理者はそのことを Recovery Manager に通知する必要があります (**change ... uncatalog** コマンドを使用します)。この例では、ミラーを（ミラー化が解除される前に含まれていた他のコピーを含めずに）復元した場合、**change ... uncatalog** を使ってリカバリ・カタログを更新し、このコピーが使用できなくなったことを通知する必要があります。

破損の検出

バックアップまたはコピーを実行しているのは Oracle サーバー・プロセスであり、サーバー・プロセスは多数の種類の破損ブロックを検出できます。前回のバックアップ操作やコピー操作で検出されなかった新しい破損ブロックは、制御ファイルとアラートログの中に記録されます。

Recovery Manager は、バックアップ操作の完了時にこの情報について問い合わせ、リカバリ・カタログにその情報を格納します。V\$BACKUP_CORRUPTION ビューおよび V\$COPY_CORRUPTION ビューを使って、制御ファイルからこの情報について問い合わせることができます。

データベースによって破損のマークが付けられているデータ・ファイル・ブロックがバックアップ中に見つかり、その破損ブロックがバックアップにコピーされ、論理破損またはメディア破損のどちらかとして制御ファイルにレポートされます。

破損ヘッダーがあるデータ・ファイル・ブロックが見つかった場合、データベースによってそのヘッダーにまだ破損のマークが付けられていなければ、そのブロックは、ブロックにメディア破損があることを示すヘッダーを付け直した上で、バックアップに書き込まれます。

注意： この時点では、すべての種類の破損を検出できるわけではありません。

チャンネル制御

backup または **copy**、**restore**、**recover** コマンドを発行する前に、**channel** を割り当てておく必要があります。各 **allocate channel** コマンドは、ターゲット・データベース・インスタンス上でサーバー・プロセスを開始することにより、**Recovery Manager** からそのターゲット・データベース・インスタンスへの接続を確立します。**allocate channel** コマンドは、サーバー・プロセスがバックアップ操作や復元操作を実行するために使う I/O デバイスのタイプも指定します。通常、1 つのチャンネルは 1 つの出力デバイスに対応します。ただし、使っている **MML**(メディア管理ライブラリ) がハードウェアの多重化をサポートしている場合を除きます。**Oracle** のバックアップのハードウェアの多重化は推奨しません。ターゲット・データベースに接続されたサーバー・プロセスの数に関係なく、そのサーバー・プロセスと通信する **Recovery Manager** プロセスは、1 つだけです。

複数のチャンネルを割り当てることができるので、1 つの **Recovery Manager** コマンドで複数のバックアップ・セットまたはファイル・コピーを平行で読み書きできます。したがって、1 つのコマンドの中での並列度は、割り当てられているチャンネル（接続）の数によって決まります。

各 **allocate channel** コマンドは、別々の接続を使ってターゲット・データベースに接続します。ターゲット・データベースのさまざまなインスタンスに接続するために、チャンネルごとに異なる接続文字列を指定できます。これにより、**Oracle Parallel Server** 構成の場合に複数のノードに作業負荷を分散できます。

チャンネル制御コマンドを使って次のことができます。

- **backup** および **copy**、**restore**、**recover** の各コマンドを実行するときに **Recovery Manager** が使う O/S リソースを制御する。
- 並列度を変える (**filesperaset** とともに使う)
- I/O 帯域幅の消費に関する制限を指定する (**set limit read rate**)
- バックアップ・ピースのサイズに関する制限を指定する (**set limit kbytes**)
- 同時にオープンできるファイルの数に関する制限を指定する (**set limit maxopenfiles**)

一部のプラットフォームでは、これらのコマンドで、使用する I/O デバイスの名前またはタイプを指定します。また、これらのコマンドで、どの O/S アクセス方法または I/O ドライバを使うかを指定する場合があります。すべてのプラットフォームが、このインターフェースによる I/O デバイスの選択をサポートしているわけではありません。一部のプラットフォームでは、I/O デバイスの選択は、プラットフォーム固有の機構により制御されます。

allocate channel コマンドの結果として、実際に O/S リソースが割り当てられるかどうかは、O/S によって異なります。一部のオペレーティング・システムでは、コマンドが発行された時点でリソースが割り当てられます。その他のオペレーティング・システムでは、読取りまたは書き込みのためにファイルがオープンされるまで、リソースは割り当てられません。また、**type disk** を指定した場合は、サーバー・プロセスを作成する目的以外に O/S リソースは割り当てられません。

change ... delete コマンドでは、O/S がコールされファイルが削除されるので、このコマンドを発行する前にチャンネルを割り当てておく必要があります。この方法で割り当てられたチャンネルは、ファイルの削除だけに使用されます。これは、ジョブ用の入力チャンネルや出力チャンネルとしては使用できません。この種のチャンネルは一度に 1 つだけ割り当てることができます。

パラレル化

Recovery Manager では、操作のパラレル化や複数のログオン・セッションの確立ができ、複数のバックアップ・セットまたはファイル・コピーをパラレルに進めることもできます。同時バックアップ・セットまたはイメージ・コピーは、別個のデータベース・ファイル・セットに対して実行する必要があります。

パラレル化の度合いを左右する要素

backup および **copy**、**restore** の各コマンドのパラレル化は、Recovery Manager により内部的に処理されます。指定する必要があるのは、次のものだけです。

- 複数の **allocate channel** コマンド
- バックアップまたはコピー、復元の対象にするオブジェクト

バックアップ・セット作成のパラレル化

通常、バックアップ・セットには、1 セット当たりのファイル数を指定しなければなりません。たとえば、20 個のチャンネルを割り当ててあるときに、他のオプションを指定せずに、20 個のファイルがあるデータベースの単純バックアップ（データベース）を実行すると、20 個のチャンネルが割り当てられていても、そのうちの 1 つのチャンネルしか使用されません。

'**filesperset 4**' を追加したとすると、Recovery Manager は、5 つのチャンネルを使用します（各チャンネルのファイル数は 4 つに制限されます）。'**filesperset 1**' を指定すると、各チャンネルがそれぞれ 1 つのファイルだけに運用されるので、20 個のチャンネルすべてが使用されることになります。

上記の例で、同じデータベース（20 個のファイル）のバックアップをとるときに、各ファイルが 1 対 1 で表領域に対応するとして、次のような **backup** コマンドを記述したとすると、パラレル化のために **filesperset** を指定する必要はありません。

```
backup (tablespace 1 tablespace 2 ... tablespace 20)
```

Recovery Manager は、コマンドを順番に実行します。つまり、現行のコマンドが完了してから、その次のコマンドを開始します。並列性は、1 つのコマンド内だけに適用されます。したがって、データ・ファイルの 10 個のコピーが必要な場合は、**copy** コマンドを個別に 10 回発行するより、10 個のコピーすべてを指定した **copy** コマンドを 1 回発行の方が効率的です。

ファイル・コピーのパラレル化

ファイルのコピーの場合は、1つの **copy** コマンドでコピーするには、複数のデータ・ファイルを指定します。たとえば次の **Recovery Manager** スクリプトでは、ファイルのコピーがシリアル化されます。一度にアクティブになるチャンネルは1つだけです。

```
run {
  allocate channel c1 type disk;
  allocate channel c2 type disk;
  allocate channel c3 type disk;
  allocate channel c4 type disk;
  allocate channel c5 type disk;
  allocate channel c6 type disk;
  allocate channel c7 type disk;
  allocate channel c8 type disk;
  allocate channel c9 type disk;
  allocate channel c10 type disk;
  copy datafile 22 to '/dev/prod/backup1/prod_tab5_1.dbf';
  copy datafile 23 to '/dev/prod/backup1/prod_tab5_2.dbf';
  copy datafile 24 to '/dev/prod/backup1/prod_tab5_3.dbf';
  copy datafile 25 to '/dev/prod/backup1/prod_tab5_4.dbf';
  copy datafile 26 to '/dev/prod/backup1/prod_tab6_1.dbf';
  copy datafile 27 to '/dev/prod/backup1/prod_tab7_1.dbf';
  copy datafile 28 to '/dev/prod/backup1/prod_tab5_2.dbf';
  copy datafile 29 to '/dev/prod/backup1/prod_ndx5_1.dbf';
  copy datafile 30 to '/dev/prod/backup1/prod_ndx5_2.dbf';
  copy datafile 31 to '/dev/prod/backup1/prod_ndx5_3.dbf';
}
```

次の文は、同じ例をパラレル化するためのものです。10個のファイルをコピーする1つの **Recovery Manager** コマンドがあり、10個のチャンネルが使用可能です。10個のチャンネルすべてが、同時にアクティブになり、各チャンネルがそれぞれ1つのファイルをコピーします。

```
run {
  allocate channel c1 type disk;
  allocate channel c2 type disk;
  allocate channel c3 type disk;
  allocate channel c4 type disk;
  allocate channel c5 type disk;
  allocate channel c6 type disk;
  allocate channel c7 type disk;
  allocate channel c8 type disk;
  allocate channel c9 type disk;
  allocate channel c10 type disk;
  copy datafile 22 to '/dev/prod/backup1/prod_tab5_1.dbf',
    datafile 23 to '/dev/prod/backup1/prod_tab5_2.dbf',
    datafile 24 to '/dev/prod/backup1/prod_tab5_3.dbf',
    datafile 25 to '/dev/prod/backup1/prod_tab5_4.dbf',
    datafile 26 to '/dev/prod/backup1/prod_tab6_1.dbf',
    datafile 27 to '/dev/prod/backup1/prod_tab7_1.dbf',
```

```
datafile 28 to '/dev/prod/backup1/prod_tab5_2.dbf',
datafile 29 to '/dev/prod/backup1/prod_ndx5_1.dbf',
datafile 30 to '/dev/prod/backup1/prod_ndx5_2.dbf',
datafile 31 to '/dev/prod/backup1/prod_ndx5_3.dbf';
}
```

多重バックアップ・セット

1つのサーバー・プロセスは、1つ以上のファイルまたは表領域（あるいはその両方）を操作対象にできます。複数のファイルまたは表領域（あるいはその両方）を指定した場合、または1つ以上の複数ファイル表領域を指定した場合は、データ・ファイルは同時にバックアップがとられ、出力はまとめて多重化されます。したがって、同じバックアップ・セットに入っていたファイルのブロックもバックアップ・セット全体に分散されます。

多重バックアップ（同じバックアップ・セットへの複数のデータ・ファイルの書込み）を実行するときは、データ・ファイルを明示的に複数のバックアップ・セットにパーティション化することができます。あるいは、**Recovery Manager** に自動的にパーティション化させることもできます。

高パフォーマンスの順次出力デバイスでは、十分な数のデータ・ファイルをバックアップに含めることによって、ストリームを続けることができます。これは、バックアップ操作とオンライン・システムとの間でI/O帯域幅に競合が起こるようなオープン状態のデータベースのバックアップの場合に重要です。

データ・ファイルのバックアップ・セットには、制御ファイルを含めることができます。**Oracle** では、多重バックアップでのオブジェクトの論理ブロックサイズは、どれも同じでなければなりません。このため、データ・ファイルとアーカイブ・ログを同じセットに書き込むことはできません。

レポート生成

report コマンドおよび **list** コマンドを使って、バックアップおよびイメージ・コピーに関する情報を取得できます。**list** コマンドでは、単にリカバリ・カタログの内容がリスト表示されますが、**report** コマンドでは、リカバリ・カタログのさらに詳しい分析が行われます。これらのコマンドの出力は、メッセージ・ログ・ファイルに書き込まれます。

report コマンドを使うと、次のような質問に対する答を示すレポートを生成できます。

- どのファイルのバックアップが必要か？
- 最近バックアップがとられていないのはどのファイルか？
- どのバックアップ・ファイルを削除できるか？
- UNRECOVERABLE 操作が原因で回復不能になっているファイルはどれか？
- 過去のある時点でのデータベースの物理スキーマは何であったか？

report need backup コマンドおよび **report unrecoverable** コマンドを定期的に使って、回復に必要なバックアップが常に使用可能で、妥当な時間内に回復を実行できる状態にしておいてください。**report** コマンドを使うと、冗長であったり **recover** コマンドで使われないために削除される可能性のあるバックアップ・セットおよびデータ・ファイル・コピーのリストが出力されます。

最後のバックアップ以降に、データ・ファイル内のオブジェクトに対して回復不能操作が実行された場合、そのデータ・ファイルは回復不能と見なされます。

注意： バックアップを持たないデータ・ファイルは、回復不能とはみなされません。このようなデータ・ファイルは、ファイルの作成時以降のログがまだ存在していれば、**create datafile** コマンドを使って回復できます。

リカバリ・カタログに対する問合せを出し、その内容のリストを作るには、**list** コマンドを使用します。これにより、どのバックアップまたはコピーが使用できるかを確認できます。

- 指定したデータ・ファイル・リストのバックアップが入っているバックアップ・セット
- 指定したデータ・ファイル・リストのコピー
- 指定した表領域リストのメンバーであるデータ・ファイルのバックアップが入っているバックアップ・セット
- 指定した表領域リストのメンバーであるデータ・ファイルのコピー
- 指定した名前を持つ、または指定した範囲内にある（あるいはその両方の条件を備えた）アーカイブ・ログのバックアップが入っているバックアップ・セット
- 指定した名前を持つ、または指定した範囲内にある（あるいはその両方の条件を備えた）アーカイブ・ログのコピー
- 指定したデータベースのインカーネーション

関連項目： Recovery Manager の **report** コマンドの詳細は、Appendix A, “Recovery Manager のコマンド構文” を参照してください。

Recovery Manager によるバックアップのためのユーザー・タグ

バックアップ・セットおよびイメージ・コピー (Oracle 作成のコピーまたはユーザー作成のコピー) には、オプションとして、タグと呼ばれるユーザー指定の文字列を割り当てることができます。タグは、実質的にはバックアップ・セットまたはファイル・コピーの記号名であり、**restore** コマンドまたは **change** コマンドの実行時にこの名前を指定できます。タグを使うと、ファイル・コピーまたはバックアップ・セットの集合を記号で参照できます。タグの最大長は 30 文字です。

タグは一意である必要はありません。複数のバックアップ・セットまたはイメージ・コピーに同じタグがあってもかまいません。あるタグが一意でない場合は、特定のデータ・ファイルについて言えば、そのタグは、最新の適正ファイルを指すとみなされます（そのファイルを含む最新の適正バックアップは、必ずしも最新のバックアップではありません。この状況が当てはまるのは、Point-in-Time 回復を実行した場合です）。たとえば、毎週月曜日の夜にデータ・ファイルのコピーが作成され、常に "mondayPMcopy" というタグが付くとすれば、そのタグは、最も新しく作成されたコピーを指すとみなされます。次のコマンドは、データ・ファイル 3 を、最も新しく作成された月曜日の夜のコピーと入れ替えます。

```
switch datafile 3 to datafilecopy tag mondayPMcopy;
```

タグによって、バックアップまたはファイル・コピーの各種クラスについて、それぞれ意図した目的または用途を示すことができます。たとえば、**switch** での使用に適したデータ・ファイル・コピーには、**restore** だけのために使われるファイル・コピーとは異なるタグを付けることができます。

注意： タグまたは **until** 句で修飾した場合を除き、デフォルトでは、最新のバックアップが選択されて復元されます。

警告： タグを使って、**restore** コマンドまたは **switch** コマンドの入力ファイルを指定できます。タグを指定すると、Recovery Manager は、どのバックアップ・セットまたはイメージ・コピーを使うかを決めるときに、一致するタグを持つバックアップ・セットだけを候補と見なします。

関連項目： **switch** コマンドおよび **restore** コマンドの詳細は、付録 A 「Recovery Manager のコマンド構文」 を参照してください。

バックアップの制約

バックアップ操作は、データベースがマウントされているかオープンされているインスタンスでだけ実行できます。Oracle のパラレル・サーバー環境では、バックアップ操作の対象のインスタンスでデータベースがオープンされていない場合、そのデータベースはどのインスタンスでもオープンできません。

バックアップは、表領域とデータ・ファイルの両方のレベルでサポートされています。また、制御ファイルのバックアップも作成できます。オンライン・ログのバックアップはサポートされていません。(オンライン・ログのバックアップが必要な場合は、ARCHIVELOG モードでデータベースを運用する必要があります。)

- データベースが NOARCHIVELOG モードになっている場合は、データベースが正しく停止された場合に限り、データベース全体のバックアップをとることができる。バックアップをとるデータ・ファイルがオフラインの通常表領域内にあるか、表領域が読取り専用モードになっている場合は、表領域のバックアップまたはデータ・ファイルのバックアップをとることができます。
- データベースが ARCHIVELOG モードになっている場合は、データベースがオープン状態であり使用されているときに、バックアップをとることができる。バックアップをとることができるのは現行のデータ・ファイルだけです。バックアップから復元したデータ・ファイルのバックアップは、メディア回復を実行して現行の状態にしてからでないとれません。正しく停止されていないデータベースのバックアップをとることもできますが、お薦めしません。

Recovery Manager でバックアップをとることができないものは次のとおりです。

- データベースに対応付けられたパラメータ・ファイル
- データベースに対応付けられたパスワード・ファイル
- オペレーティング・システムのファイル
- オンライン REDO ログ

関連項目：3-9 ページの「オンライン REDO ログのバックアップ」

復元の制約

復元操作の対象は起動済みのインスタンスでなければなりませんが、そのインスタンスにはデータベースがマウントされていなくてもかまいません。したがって、制御ファイルが失われていても、復元操作を実行できます。復元する表領域やデータ・ファイルは、オフラインになっているか、データベースがクローズされていなければなりません。

復元では、既存のデータ・ファイルを上書きすることも、**set newname** コマンドを使って出力を新しいファイルに送ることもできます。

整合性チェック

Oracle では、結果として使用不能なバックアップ・ファイルができたり、復元済みのデータ・ファイルが破れるような操作は実行できません。Oracle Server による主な整合性チェックには次のものがあります。

- データベースの直前のインカーネーションからバックアップを適用することにより、復元操作でデータベースが破損しないようにする。
- 増分バックアップが正しい順序で適用されるようにする。
- 復元または回復中のデータ・ファイルへのアクセスを禁止する。
- 1つのデータ・ファイルにつき1度に1つの復元操作しか許可しない。
- 未回復のバックアップ・ファイルのバックアップを禁止する。
- バックアップに格納されている制御情報により、破損したバックアップ・ファイルを検出する。

Recovery Manager でのオープン状態のデータベースのバックアップ時における分裂ブロックの検出

Recovery Manager を使わないでオープン・バックアップを実行するときは、DBW0 により書込み中であるため整合性がない状態にあるブロックを、オペレーティング・システムがバックアップ用に読み取る場合に備えて、表領域をホット・バックアップ・モードにしておく必要があります。このように整合性がなくなったブロックを「分割ブロック」または「分裂ブロック」といいます。

Recovery Manager を使ってバックアップを実行する場合は、オペレーティング・システムのユーティリティではなく、Oracle サーバー・プロセスがデータ・ファイルを読み取ります。Oracle サーバー・プロセスはすべての Oracle ブロックを読み取り、各ブロックのヘッダーおよびフッターに格納されている制御情報と比較して、ブロックが「分裂」しているかどうかを確認します。分裂ブロックが検出されると、Oracle サーバー・プロセスはそのブロックをもう一度読み取ります。Recovery Manager を使ってデータベース・ファイルのバックアップまたはコピーをとるときに、表領域をホット・バックアップ・モードにしなくてもよいのは、この理由からです。

関連項目：ホット・バックアップ・モードの詳細は、Chapter 11, “オペレーティング・システムのバックアップの実行” を参照してください。

アーカイブ・ログの追跡

制御ファイルには、ARCH バックグラウンド・プロセスによりアーカイブされたログの位置が記録されています。ログが正常にアーカイブされるたびに、制御ファイルにレコードが1つ追加されます。このレコードには、アーカイブ・ログ・ファイル名およびその他の関連情報が格納されています。

制御ファイルの作成時に維持するアーカイブ・ログ・レコードの数を指定できます。制御ファイルの全体サイズの制限を超過しない範囲内で、任意の数を指定できます。これらのレコードは、循環方式で再使用されます。Recovery Manager は、これらのレコードを問い合わせ、結果をリカバリ・カタログに波及します。

オンライン・ログ内にメディア障害がある場合は、ログのアーカイブができないことがあります。したがって、ALTER DATABASE CLEAR UNARCHIVED LOG 文の発行を選択できます。この場合でもアーカイブ・ログ・レコードは作成されますが、そのレコードには、このログ順序番号やシステム変更番号 (SCN) 範囲についてのアーカイブ・ログが存在しないことが示されます。

イメージ・コピーおよびアーカイブ・ログのカタログ化

Recovery Manager は、イメージ・コピーをカタログ化し、メタデータを読み取ることができます。これは、リカバリ・カタログが失われ、災害時回復を実行する必要がある場合に、重要な役割を果たします。カタログ化できるのは、イメージ・コピーとアーカイブ・ログだけです。

Recovery Manager でのバックアップおよび回復の実行

この章では、**Recovery Manager** を使って、データベースのバックアップ、復元、回復などの作業を管理する方法を説明します。取り挙げる項目は次のとおりです。

- リカバリ・カタログのインストール
- リカバリ・カタログのメンテナンス
- チャンネル制御コマンドの使用法
- レポートの生成
- スクリプトのメンテナンス
- スナップショットの制御ファイルの位置の設定
- ファイルのバックアップ
- ファイルのコピー
- ファイルの復元
- データ・ファイルの切替え
- データ・ファイルの回復
- バックアップと復元の監視

リカバリ・カタログのインストール

Recovery Manager の操作には、リカバリ・カタログを使っても使わなくてもかまいません。リカバリ・カタログを使用する場合は、Recovery Manager を使う前にリカバリ・カタログをインストールします。

リカバリ・カタログをインストールする手順

1. カタログの所有者となるユーザーを登録（指定）します。必要であれば、リカバリ・カタログを入れる独立した表領域を作成できます。

```
create user rman identified by rman
temporary tablespace temp
default tablespace rcvcat
quota unlimited on rcvcat;
grant recovery_catalog_owner to rman;
```

2. 次のようにステップ 1 で作成したユーザー ID を使って接続し、リカバリ・カタログのスキーマを作成します。

```
connect rman/rman
@?/rdbs/admin/catrman
```

3. ネットワークを介してターゲット・データベース（たとえば、ローカルでないターゲット・データベース）に接続するには、パスワード・ファイルを構成する必要があります。ローカル・データベースには、オペレーティング・システムの認証を使用できます。パスワード・ファイルの作成とメンテナンスの詳細は、『Oracle8 Server 管理者ガイド』を参照してください。
4. rman を起動します。

これで、Recovery Manager を使用できる状態になりました。

データベースの登録

これから新しいデータベースを作成する場合または既存の Oracle7 データベースを Oracle8 に移行する場合は、次のコマンドを発行します。

```
register database;
```

既存のバージョン 8.x のデータベース用に Recovery Manager をインストールする場合は、次のコマンドを発行します。

```
register database;
```

バージョン 8.x でユーザーが作成したバックアップがディスク内にある場合は、次のコマンドを発行することにより、リカバリ・カタログにバックアップを追加してください。

```
catalog datafilecopy 'filename';
```

Oracle8 データベースの回復に Oracle7 のバックアップを使用できるのは、そのバックアップが、データベースを移行したときに通常オフラインであったか読取り専用とされていた表領域の一部だった場合だけです。

既存のアーカイブ・ログがある場合は、次のようにしてカタログ化します。

```
catalog archivelog 'filename';
```

関連項目：リカバリ・カタログの詳細は、Chapter 3, “バックアップをとる時期” を参照してください。

リカバリ・カタログのメンテナンス

ここでは、リカバリ・カタログをメンテナンスおよび操作する方法を説明します。取り挙げる項目は次のとおりです。

- リカバリ・カタログへのターゲット・データベースの登録
- リカバリ・カタログ内の情報のリセット
- リカバリ・カタログとターゲット・データベースの再同期化
- バックアップ・セットまたはファイル・コピーの可用性の変更
- リカバリ・カタログおよびオペレーティング・システムのバックアップからの削除および変更

リカバリ・カタログへのターゲット・データベースの登録

特定のターゲット・データベースに対して初めて **Recovery Manager** を使うときは、ターゲット・データベースをリカバリ・カタログに登録しておく必要があります。**Recovery Manager** は、ターゲット・データベースの登録に必要なすべての情報を、そのターゲット・データベース自体から入手します。したがって、この操作のためにはターゲット・データベースがマウントされていなければなりません。

Oracle では、データベース同士を区別するため、「db 識別子」という一意的に生成された内部番号を使います。この番号は、ユーザーがデータベースを作成したときに生成されます。通常、それぞれのデータベースには一意の識別子がありますが、(**create database** 文を使わずに) 既存のデータベースからファイルをコピーして作成したデータベースがあると、例外が生じます。そのような場合は、重複したデータベース識別子が検出され、**register database** コマンドは失敗します。その場合は、別の Oracle Userid から **catrman.sql** スクリプトを再実行することにより、別のユーザーのスキーマ内に 2 番目のリカバリ・カタログを作成できま

す。その後、データベース識別子が重複したデータベースを、この新スキーマ内に新たに作成したりリカバリ・カタログに登録できます。

注意： 同じデータベース名と識別子が付いた異なるターゲット・データベースを **Recovery Manager** で使う場合は、**Recovery Manager** を起動するときに必ず正しいリカバリ・カタログのスキーマを指定するよう最大限の注意を払う必要があります。

リカバリ・カタログ内の情報のリセット

RESETLOGS オプションによりオープンしたターゲット・データベースに対して再度 **Recovery Manager** を使う前に、データベースのインカーネーションがリセットされたことを **Recovery Manager** に通知する必要があります。リカバリ・カタログ内に新しいデータベース・インカーネーション・レコードを作成するよう、**reset database** コマンドを使って **Recovery Manager** に指示します。この新しいインカーネーション・レコードが " 現行の " インカーネーションになります。ターゲット・データベースが行う、後続のすべてのバックアップおよびログ・アーカイブは、新しいデータベースのインカーネーションに対応付けられます。

ALTER DATABASE OPEN RESETLOGS コマンドを発行しただけで、データベースをリセットしなかった場合は、**Recovery Manager** は、**RESETLOGS** コマンドと、古い制御ファイルが誤って復元された場合との区別ができないため、リカバリ・カタログにアクセスしません。データベースをリセットすると、**Recovery Manager** はデータベースが **RESETLOGS** オプションを使ってオープンされたことを認識します。

RESETLOGS オプションを使ったオープンの効果を取り消すために前のデータベース・インカーネーションのバックアップを復元する場合は、**reset database to incarnation key** コマンドを使って現行のインカーネーションを古いインカーネーションに変更できます。そのためにユーザーは戻り先のデータベース・インカーネーション用の dbinc レコードの主キーを指定する必要があります。キー値を取得するには、**list incarnation of database identifier** コマンドを発行します。**reset database to incarnation** コマンドの後で、**restore** コマンドと **recover** コマンドを発行して、前のインカーネーションからデータベース・ファイルを復元し、回復します。

リカバリ・カタログとターゲット・データベースの再同期化

リカバリ・カタログは、ログ・スイッチが発生したりログがアーカイブされても自動的に更新されないため、リカバリ・カタログを定期的に再同期化する必要があります。ログ・スイッチおよびアーカイブ・ログの情報は制御ファイルに格納されるので、制御ファイルをリカバリ・カタログに定期的に波及させる必要があります。リカバリ・カタログを再同期化する頻度は、ログがアーカイブされる頻度によって決まります。この運用コストは、制御ファイルの中の、前回の再同期化以降に挿入または変更されたレコード数に比例します。挿入または変更されたレコードがまったくない場合は、再同期化のコストは非常に低くなります。

したがって、この操作を頻繁に（たとえば1時間ごとに）実行しても、コストがかかりすぎるようなことはありません。

注意： **backup**、**copy**、**restore**、**switch** の各コマンドでは、次の場合、リカバリ・カタログが自動的に更新されます。

- ターゲット・データベースの制御ファイルが使用可能である
 - コマンドの実行時にリカバリ・カタログ用のデータベースが使用可能である
-

バックアップまたはコピーのときにリカバリ・カタログが使用可能でない場合は、リカバリ・カタログを再同期化する必要があります。

ターゲット・データベースの物理構造になんらかの変更を加えた場合も、リカバリ・カタログを再同期化する必要があります。ログ・アーカイブ操作の場合と同様に、物理スキーマを変更した場合も、リカバリ・カタログは自動的に更新されません。次の操作は、どれも物理スキーマの変更とみなされます。

- 表領域の追加または削除
- 既存の表領域への新しいデータ・ファイルの追加
- ロールバック・セグメントの追加または削除

リカバリ・カタログを再同期化する場合、**Recovery Manager** は、ターゲット・データベースの現行の制御ファイルまたはバックアップ制御ファイルとリカバリ・カタログを比較し、欠けている情報または変更された情報を組み込むことにより、リカバリ・カタログを更新します。現行の制御ファイルから再同期化する場合、**Recovery Manager** は、制御ファイル内の更新されたレコードを自動的に検出し、それらのレコードだけを再同期化します。ターゲット・データベースがオープン状態の場合は、リカバリ・カタログ内のロールバック・セグメントに関する情報（この情報は表領域 **Point-in-Time** 回復に使われます）も更新されます。バックアップ制御ファイルから再同期化する場合、**Recovery Manager** は、バックアップ・ピースまたはファイル・コピーが実際に存在しているかどうかの検証はしません。したがって、**change...uncatalog** コマンドを使用して、存在していないファイルのレコードを削除しなければならないことがあります。

次に示すクラスのリカバリ・カタログ・レコードが再同期化されます。

ログ履歴

ログ・スイッチが発生したときに作成されるレコードです。**Recovery Manager** はこの情報を追跡して、どのアーカイブ・ログを検出すべきかを判断します。

アーカイブ・ログ

アーカイブ・ログに関連したレコードで、オンライン・ログをアーカイブしたときまたは既存のアーカイブ・ログをコピーしたとき、アーカイブ・ログのバックアップ・セットを復元したときに作成されます。

バックアップ履歴

バックアップ・セットおよびバックアップ・ピース、バックアップ・セット・メンバー、ファイル・コピーに関連したレコードです。**resync catalog** コマンドを使うと、**backup** コマンドまたは **copy** コマンドを実行したときに、これらのレコードが更新されます。

物理スキーマ

データ・ファイルおよび表領域に関連したレコードです。ターゲット・データベースがオープンしている場合は、ロールバック・セグメントの情報も更新されます。

リカバリ・カタログ内の物理スキーマ情報が更新されるのは、ターゲット・データベースに現行の制御ファイルがマウントされている場合だけです。ターゲット・データベースに、バックアップ制御ファイル、または新たに作られた制御ファイル、前に検出された制御ファイルより古い制御ファイルがマウントされている場合は、リカバリ・カタログ内の物理スキーマ情報は更新されません。**resync catalog from backup controlfile** コマンドを使ったときには、物理スキーマ情報は更新されません。

再同期化のときにターゲット・データベースがオープン状態の場合は、リカバリ・カタログ内に新しい一貫性ポイントが作成されます。

バックアップ・セットまたはファイル・コピーの可用性の変更

Recovery Manager の **change** コマンドを使用すると、次のレコード変更を実行できます。

- リカバリ・カタログから、バックアップ・ピースまたはデータ・ファイル・コピー、アーカイブ・ログに対する参照を削除する (**change ... uncatalog**)
 - リカバリ・カタログのある場合のみ可能
- 制御ファイルおよびリカバリ・カタログから、バックアップ・ピースまたはデータ・ファイル・コピー、アーカイブ・ログに対する参照を削除し、オペレーティング・システムからファイルを物理的に削除する (**change ... delete**)
 - リカバリ・カタログの有無に関係なく可能
- 制御ファイルがディスク上に存在しなくなった場合に、制御ファイルおよびリカバリ・カタログから、バックアップ・ピースまたはデータ・ファイル・コピー、アーカイブ・ログに対する参照を削除する (**change ... validate**)
 - リカバリ・カタログの有無に関係なく可能
- バックアップ・ピースまたはデータ・ファイル・コピー、アーカイブ・ログに使用不能のマークを付ける (**change ... unavailable**)

- リカバリ・カタログのある場合のみ可能
- バックアップ・ピースまたはデータ・ファイル・コピー、アーカイブ・ログに使用可能のマークを付ける (**change ... available**)
- リカバリ・カタログのある場合のみ可能

存在していないファイルを参照しているレコードを、リカバリ・カタログから削除するには、**change ... uncatalog** オペランドを使用します。**change...delete** コマンド以外の手段により、バックアップ・ピースまたはファイル・コピー、アーカイブ・ログを削除した場合は、このコマンドを使って削除の事実を Recovery Manager に通知することが重要です。**change ... uncatalog** は、リカバリ・カタログを使って運用している場合にだけ使用できます。

delete オペランドの機能は **uncatalog** オペランドに似ていますが、それに加えて、**delete** オペランドを使用すると、バックアップ・ピースまたはバックアップ・セットを削除するためにオペレーティング・システムまたはメディア・マネージャがコールされます。このコマンドの前に **allocate channel for delete** コマンドを指定することにより、削除するファイルに適したデバイス・タイプを指定する必要があります。

ディスクから多数のファイルを削除した場合、ディスク上に残っている内容とリカバリ・カタログとを一致させる最も効率的な方法は、**change ... validate** コマンドを使うことです。このコマンドは、ディスク上になければならない指定されたバックアップ・ピースおよびファイル・コピー、アーカイブ・ログが物理的に存在するかどうかを検査するためのコマンドです。ファイルが存在しない場合、そのファイルへの参照はリカバリ・カタログおよび制御ファイルから削除されます。

unavailable オペランドは、ファイルが見つからない場合、または別のサイトに移された場合に使用します。使用不能のマークが付けられたファイルは、**restore** コマンドや **recover** コマンドでは使用できません。後でそのファイルが見つかるか、メイン・サイトに戻った場合は **available** オペランドを使用して再び使用可能のマークを付けることができます。

change コマンドの対象となるファイルは、リカバリ・カタログ（または制御ファイル）に記録されていて、現行のデータベース・インカーネーションに属しているものだけです。**catalog** コマンドおよび **resync from controlfilecopy** コマンドを除く他のコマンドについても同じです。

カタログ・レコードの削除：例

カタログ化されたファイルが、Recovery Manager 以外のなんらかの手段で削除されたか、その他の理由で消失または破損した場合は、カタログ・レコードを削除します。

ファイルがディスク上にある場合は、次の文を発行します。

```
change datafilecopy <primary_key> delete;
change archivelog <primary key> delete;
```

ファイルがメディア・マネージャによって格納されている場合は、次の文を発行します。

```
allocate channel for delete type 'tape';
change backuppiece <primary_key> delete;
```

```
release channel;
```

削除するレコードの主キーを取得するには、**list** コマンドを発行します。

多数のカタログ・レコードの削除：例

多数のファイルを削除した後に、ディスク上の内容とリカバリ・カタログを一致させる必要がある場合は、次の **change ... validate** 文を使用できます。

```
change datafilecopy <primary_key> validate;  
change archivelog all validate;
```

ユーザー作成のバックアップ・ファイルのカタログ化

多くの場合、Recovery Manager 以外の手段で作成したファイル・コピーの存在を Recovery Manager に認識させておくくと便利です。

データ・ファイルのコピーまたはアーカイブ・ログ、制御ファイルのコピーに関する情報をリカバリ・カタログと制御ファイルに追加するには、**catalog** コマンドを使います。

catalog コマンドを使って、データ・ファイルのコピーをレベル 0 のバックアップとしてカタログ化することもできます。これにより、後でそのデータ・ファイルのコピーを基礎として、増分バックアップを実行できます。

Oracle 8.x では、オープン状態のデータベースのオペレーティング・システムのバックアップを作成できる **ALTER TABLESPACE BEGIN/END BACKUP** コマンドが引き続きサポートされています。Recovery Manager ではこのようなバックアップが作成されることはありませんが、この種のバックアップをリカバリ・カタログに追加すると、Recovery Manager で認識できます。

この種のバックアップはディスク内でアクセス可能であって、しかも 1 つのファイルの完全なイメージ・コピーであることが必要です。データ・ファイルのバックアップは、一貫性のあるまたは一貫性のないデータベース全体のバックアップ、表領域のバックアップ、データ・ファイルのバックアップのどれでも構いません。Recovery Manager は、これらのバックアップをすべてデータ・ファイルのコピーと認識します。

たとえば、データ・ファイルがミラー化されたディスク・ドライブに格納されている場合は、ミラー化を解除するだけでオペレーティング・システム・コピーを作成できます。この場合、ミラー化を解除した後で、**catalog** コマンドを使って Recovery Manager にオペレーティング・システム・コピーの存在を通知します。ミラーの復元前に、**change... uncatalog** コマンドを発行することにより、Recovery Manager にファイル・コピーが削除されたことを通知する必要があります。

catalog コマンドには、データベースの現行のインカーネーションに対応付けられたファイルだけをカタログ化するという制限があります。バージョン 7.x またはそれ以前に作成されたアーカイブ・ログおよび制御ファイルのコピーについては、カタログ化できません。

バージョン 7.x またはそれ以前に作成されたデータ・ファイルのコピーをカタログ化できるのは、そのファイルが、データベースを Oracle8 に移行したときに通常オフラインか読取り専用だった表領域に属している場合です。

消失または破損したリカバリ・カタログ用のデータベースの回復

リカバリ・カタログ用のデータベースが消失または破損し、Oracle の標準的な回復方法によりそのリカバリ・カタログ用のデータベースを回復できない場合は、リカバリ・カタログの内容を部分的に再作成するオプションが 2 つあります。

1. **catalog** コマンドを発行して、アーカイブ・ログおよびバックアップ制御ファイル、データ・ファイル・コピーを再カタログ化する。
2. **resync catalog from backup controlfile** コマンドを使って、バックアップ制御ファイルから情報を抽出し、その情報を基にしてリカバリ・カタログを再構築する。

catalog コマンドは、バックアップ・ピースやバックアップ・セットの再カタログ化をサポートしていないので、**resync catalog from backup controlfile** コマンドを使う以外には、バックアップ・セットの情報を再作成する方法はありません。**Recovery Manager** は、再カタログ化しようとしているファイルがまだ存在しているかどうかを検証しないので、もう存在していないファイルのレコードが再同期化によって追加されることがあります。その場合は、**change...uncatalog** コマンドを発行して余分なレコードを削除する必要があります。

チャネル制御コマンドの使用法

チャネル制御コマンドを使うことにより、次の作業を実行できます。

- **backup** および **copy**、**restore**、**recover** の各コマンドを実行するときに **Recovery Manager** が使うオペレーティング・システム・リソースを制御する。
- 並列度を制御する。
- I/O 容量の消費量およびバックアップ・ピースのサイズに関する制限を指定する。

各チャネル割当てによって、**Recovery Manager** からターゲット・データベース・インスタンスへの接続が確立されます。複数のチャネルの同時割当てができるので、複数のバックアップ・セットまたはファイル・コピーを 1 つのジョブでパラレルで読み書きできます。したがって、1 つのジョブの中での並列度は、割り当てられているチャネル（接続）の数によって決まります。

各チャネル割当てについて、1 つずつ接続が使用されます。各チャネルごとに異なる接続文字列を指定できます。これは、複数のノードに作業負荷を分散させる OPS(Oracle Parallel Server) 構成の場合に便利です。

allocate channel コマンドの結果として、実際にオペレーティング・システム・リソースが割り当てられるかどうかは、オペレーティング・システムによって異なります。一部のプラットフォームでは、コマンドの発行時にオペレーティング・システム・リソースが割り当てられます。一方、ファイルが読み書き用にオープンされるまでオペレーティング・システム・

リソースが割り当てられないプラットフォームもあります。また、**type disk** を指定した場合は、このコマンドではオペレーティング・システム・リソースは割り当てられません。

backup または **copy**、**restore**、**recover** の各コマンドの前には、**allocate channel** コマンドを少なくとも 1 回発行する必要があります。**copy** コマンドで使用できるのは、**type disk** を指定しているチャネルだけです。**copy** コマンドの実行時に割り当てられるこの他のデバイス・タイプは無視されます。

関連項目： **channel** コマンドの詳細は、それぞれのオペレーティング・システム用の Oracle マニュアルを参照してください。

チャネル制御コマンド

allocate channel

ターゲット・データベース・インスタンスへの接続を確立するには **allocate channel** コマンドを使用します。1 つの接続で、1 つのバックアップ・セット (**backup** または **restore**、**recover** の場合)、または 1 つのファイル・コピー (**copy** の場合) に対応します。複数の接続が確立されている場合は、各接続で、それぞれバックアップ・セットまたはファイル・コピーに対応します。

バックアップ・セットをディスクへ書き込むかディスクから読み取る場合は **type disk** 割当てオペランドを使う必要があります。このオペランドを指定しないと、デバイスは順次 I/O デバイス（たとえば、テープ）とみなされます。コピーは常にディスクに書き込まれるので、**type disk** を指定してあるチャネル以外は使用できません。

割り当てたチャネルには、名前 (**channel_id**) を付ける必要があります。この名前は、チャネルを解放するとき、および I/O エラーをレポートするときに使われます。**channel_id** には、ユーザーにわかりやすい、ブランクで区切った文字列（キーワードを除く）を指定します。

次に、**allocate channel** コマンドの各オペランドを説明します。

type

このオペランドには、**disk** または引用符付き文字列を指定します。**disk** を指定した場合は、この接続で書き込まれる、または読み取られるバックアップ・セットは、すべてディスク上にあります。引用符付き文字列を指定した場合、それは、なんらかのタイプの順次 I/O デバイスまたはアクセス方法の、プラットフォーム固有の指定とみなされます。

順次 I/O デバイス・タイプの構文と方法はプラットフォームによって異なります。このオペランドを指定しなかった場合は、**name** オペランドを指定して、順次 I/O デバイスを特定する必要があります。

name

このポート固有の文字列には、割り当てる順次 I/O デバイスの名前を指定します。これを指定しなかった場合は、指定したタイプの任意の使用可能なデバイスが使用されます。**type disk** を指定する場合は、このオペランドは使わないでください。

parms

このポート固有の文字列には、割り当てるデバイスに関するパラメータを指定します。**type disk** を指定した場合は、このオペランドは使わないでください。

connect

このパラメータには Recovery Manager がバックアップ操作または復元を実行するターゲット・データベース・インスタンスへの接続文字列を指定します。このパラメータは、バックアップまたは復元の作業を複数の OPS インスタンスに分散させる OPS インストレーションに使うためのものです。このパラメータを指定しなかった場合は、**target** コマンド行パラメータによって指定したターゲット・データベース・インスタンスで、すべての操作が実行されます。

format

このパラメータには、このチャンネル上で作られるバックアップ・ピースの名前に使う形式を指定します。この形式が使われるのは、**backup** コマンドで **format** オプションが指定されていない場合だけです。**backup** コマンドの **format** オペランドの説明を参照してください。

release channel

順次 I/O デバイスを解放するには、**release channel** コマンドを使用します。しかし、追加のチャンネル割当てコマンド用の接続は維持されます。オペランドは、チャンネルの割当て時に指定する *channel_id* だけです。

set limit channel

set limit channel コマンドは、このデバイスを使って実行する **backup** コマンドまたは **copy** コマンドに適用する制限を指定します。

次に、**set limit channel** コマンドの各オペランドを説明します。

readrate

この制限には、個々の入力データ・ファイルからバックアップ用またはコピー用に読み取られる 1 秒あたりの最大バッファ数 (個々のバッファ・サイズは `db_blocksize * db_file_direct_io_count`) を指定します。このパラメータは、コマンドによるディスクの消費を妥当な範囲内に抑え、オンライン・パフォーマンスの低下を防止するために使用します。

kbytes

このチャンネル上で作成されるバックアップ・ピースの最大サイズを指定します。

maxopenfiles

1 つのインスタンスについて、バックアップ操作でオープンできる入力ファイルの最大数の制限を指定します。このパラメータは、多数のファイルのバックアップを 1 つのバックアップ・セットとして作成するときに、「オープンしているファイルの数が多すぎる」というオペレーティング・システムのエラーの発生を防ぐために使用します。

maxopenfiles を指定しなかった場合は、最大 32 個の入力ファイルを同時にオープンできます。

レポートの生成

report コマンドを使って、次のような質問に答える 2 種類のレポートを生成できます。

- どのファイルのバックアップが必要か？
- 最近バックアップをとっていないファイルはどれか？
- UNRECOVERABLE 操作が原因で回復不能になっているファイルはどれか？
- どのバックアップ・ファイルを削除できるか？
- 過去のある時点でのデータベースの物理スキーマは何であったか？

リカバリ・カタログに対する問合せを出し、その内容のリストを作るには、**list** コマンドを使用します。次の情報をリストとして出力できます。

- 指定したデータ・ファイル・リストのバックアップが入っているバックアップ・セット
- 指定したデータ・ファイル・リストのコピー
- 指定した表領域リストのメンバーであるデータ・ファイルのバックアップが入っているバックアップ・セット

- 指定した表領域リストのメンバーであるデータ・ファイルのコピー
- データベース内のすべてのバックアップ・セットまたはすべてのデータ・ファイルのコピー（必要に応じて、時刻またはデータ・ファイル・コピーのファイル名、デバイス名、タグにより限定したもの）
- 指定した名前を持つ、または指定した範囲内にある（あるいはその両方の条件を備えた）アーカイブ・ログのバックアップが入っているバックアップ・セット
- 指定した名前を持つ、または指定した範囲内にある（あるいはその両方の条件を備えた）アーカイブ・ログのコピー
- 指定したデータベース・インカーネーション、またはリカバリ・カタログにより認識されるすべてのデータベース・インカーネーション

report コマンドと **list** コマンドでは、よく似た出力が生成されます。**list** コマンドでは、単にリカバリ・カタログの内容がリスト表示されます。**report** コマンドでは、リカバリ・カタログのさらに詳しい分析が行われます。**report** コマンドおよび **list** コマンドの出力は、メッセージ・ログ・ファイルに書き込まれます。

report need backup コマンドおよび **report unrecoverable** コマンドを定期的に使って、回復に必要なバックアップが常に使用可能で、妥当な時間内に回復を実行できる状態にしておいてください。**list** コマンドは、使用可能なバックアップまたはコピーを問い合わせるために使用できます。

レポートの生成

report コマンドを使うと、次のいずれかのタイプのレポートを生成できます。

- **report unrecoverable** *report_object_list device_type_list*
- **report need backup** *need_backup_operand report_object_list device_type_list*
- **report obsolete** *report_obsolete_operand_list device_type_list*
- **report schema** *at_clause*

このとき、

report_object_list

レポートの対象とするデータ・ファイルを指定します。レポートには、データベース全体（必要に応じて特定の表領域をスキップ）または表領域のリスト、データ・ファイルのリストを組み込むことができます。

device_type_list

ファイルが回復不能かどうか、またはより新しいバックアップを必要としているかどうかを判断するときに、候補とするバックアップ・セットを制限するために使用します。これを指定した場合、指定したタイプのデバイスのいずれか

に存在しているバックアップ・セットだけが候補とみなされます。指定しなかった場合は、使用可能なすべてのバックアップ・セットが候補とみなされます。データ・ファイルのコピーは常に候補とみなされます。

回復不能データ・ファイルのレポート

回復不能なすべてのデータ・ファイルをリスト表示するには、**report unrecoverable** コマンドを使用します。最後のバックアップ以降に、データ・ファイル内のオブジェクトに対して UNRECOVERABLE 操作が実行されている場合、そのデータ・ファイルは回復不能とみなされます。

データ・ファイルのバックアップのいずれかが存在しないというだけでは、回復不能とみなされる十分な理由にはなりません。このようなデータ・ファイルは、そのファイルの作成時に開始されたログがまだ存在していれば、**create datafile** コマンドを使って回復できます。

例：新規バックアップが必要なデータ・ファイル REDO が失われているため既存のバックアップから完全には回復できないすべてのデータ・ファイルをリスト表示するには、次のコマンドを使用します。

```
report unrecoverable;
```

バックアップを作成する必要があるデータ・ファイルのレポート

report need backup コマンドは、新しくバックアップを作成する必要があるすべてのデータ・ファイルをリスト表示します。レポートでは、復元時に最新のバックアップが使われるものと想定されます。

次に、*need_backup* の各オペランドを説明します。

incremental

増分バックアップ数の最小値を指定する整数。データ・ファイルの完全な回復のために、この数より多くの増分バックアップを適用する必要がある場合は、そのデータ・ファイルは新規の全体バックアップが必要であるとみなされます。ここでは、可能な限り低いレベルの増分バックアップが使う最も効率的な方法の採用を前提としています。これは、**recover** コマンドによってファイルが実際に回復されるときに使われる方法と同じです。バックアップがないファイルは、このリストには入りません。このようなファイルは、**report need backup days** コマンドを使って検索できます。

days

このファイルの回復時に適用する必要があるログ・ファイルの日数のしきい値を指定する整数。オンライン・ファイルの場合は、これは、ファイルの最後の全体バックアップまたは増分バックアップ以後の日数です。バックアップ・セットの経過日数を数えるときに、時刻は考慮されません（つまり、昨日中にとられたバックアップは、どれも1日経過したものとみなされます）。バック

アップ・セットのコピーが複数ある場合は、元のバックアップ・セットの完了時点に基づいて計算されます。このファイルの最新のバックアップの経過日数がこの日数を超過している場合は、このファイルは新規バックアップが必要であるとみなされます。

ターゲット・データベース制御ファイルがマウントされていて現行の状態にある場合は、このレポートについて次のような最適化の処置がとられます。

1. オフラインであって、最新のバックアップにすべての変更内容が保存されているファイルは含まれません。
2. 前はオフラインだったが現在はオンラインであり、最新のバックアップにオフラインのときまでのすべての変更内容が保存されているファイルは、オンラインの期間が指定した日数を超えている場合に限りレポートに含まれます。

redundancy

必要と考えられる冗長性の最低レベルを指定する整数。**redundancy 2** は、各データ・ファイルについて少なくとも2つはバックアップがない限り、そのファイルについて新規バックアップが必要とみなされることを意味します。

例：バックアップを作成する必要があるデータ・ファイル データベース内のデータ・ファイルのうち、現行の状態に回復するために3つ以上の増分バックアップを適用する必要があるデータ・ファイルをすべて表示するには、次のコマンドを使用します。

```
report need backup incremental 3 database;
```

表領域 "system" 内のデータ・ファイルのうち、5日以上にわたってバックアップ（全体または増分）がとられていないデータ・ファイルをすべて表示するには、次のコマンドを使用します。

```
report need backup days 5 tablespace system;
```

古くなったデータ・ファイルのバックアップのレポート

report obsolete コマンドを使うと、冗長なために削除できるバックアップ・セットとデータ・ファイル・コピーのリストを表示できます。

バックアップは、次のいずれかの基準を満たすときに古くなったとみなされます。

- すでに存在しなくなったファイルのバックアップであるか、*until-time* に指定された時点で存在しなかったファイルのバックアップである
- 同じファイルについて、より新しいバックアップが *n* 個存在し、それらのバックアップが使用可能である (*n* は、必要な冗長性を表す)
- **orphan** オプションが指定されている場合に、親なしのインカーネーションからのバックアップである

次に、*report_obsolete* の各オペランドを説明します。

redundancy

必要と考えられる冗長性の最低レベルを指定する整数。あるデータ・ファイルに、このレベルより多くの全体バックアップまたはコピーが存在する場合、余分なバックアップは、古くなったものとなされます。値はゼロ以外でなければなりません。デフォルト値は 1 です。

orphan

バックアップおよびコピーがデータベースのインカーネーションに属し、このインカーネーションが現行のインカーネーションに先行するものではないため、使用できないバックアップとコピーを、古くなったものとみなすように指定します。

until 句

until 句を指定した場合、指定した時刻の後の変更が含まれている、バックアップは、古くなったバックアップあるいは冗長なバックアップとは見なされません。このオプションは、データベースが、現在時刻より前の時点に回復する必要がありますがある場合に便利です。

現時点のデータ・ファイルのレポート

指定した時点または現時点でのすべてのデータ・ファイルおよび表領域の名前をリスト表示するには、**report schema** コマンドを使用します。時点には時刻または SCN、REDO ログを指定できます。

at_clause には次の構造があります。

at time

時刻を指定する引用符付き文字列。

at scn

SCN が指定されたことを表す整数。

at logseq *integer* thread *integer*

候補となるログ順序番号およびスレッド番号を指定する整数。これは、指定したログおよびスレッドが最初にオープンされた時刻です。

リストの生成

リカバリ・カタログにより認識されるバックアップ・セットまたはコピーの中から、指定したグループについてのすべての情報の詳細レポートを生成するには、**list** コマンドを使用します。次の **list** コマンドが使用できます。

- **list copy of list_object_list list_qualifier_list**

データ・ファイル・コピーおよびアーカイブ・ログの情報がリスト表示されます。

- **list backupset of list_object_list list_qualifier_list**

バックアップ・セットの情報がリスト表示されます。

- **list incarnation of identifier list_qualifier_list**

データベース・インカーネーションの情報がリスト表示されます。リストには、指定したデータベース名に該当するすべてのデータベース・インカーネーション・レコードの主キーが含まれます。**Recovery Manager** が現行とみなしているインカーネーションを前のインカーネーションに変更するには、**resetdatabase** コマンドでこのキーを指定します。識別子を指定しなかった場合は、リカバリ・カタログに登録されているデータベースがすべてリスト表示されます。

list_object_list

これには、バックアップ・セットかコピーをリスト表示する、表領域またはデータ・ファイル、アーカイブ・ログを指定します。

database skip_clause

現行のデータベース内のファイルのバックアップ・セットまたはデータ・ファイル・コピーをすべてリスト表示します。オプションとして、*skip_clause* を使って表領域をスキップできます。

tablespace

表領域名のリスト。指定した表領域内のデータ・ファイルを少なくとも 1 つ含むバックアップ・セットまたはデータ・ファイル・コピーがリスト表示されます。

datafile

データ・ファイルの名前または番号のリスト。指定したデータ・ファイルを少なくとも 1 つ含むバックアップ・セットまたはデータ・ファイル・コピーがリスト表示されます。

archivelog_record_specifier

list_qualifier_list

リスト修飾子は、バックアップ・セットまたはコピーのリストを表示するオブジェクトを制限するために使用します。

tag tag

データ・ファイル・コピーおよびバックアップ・セットを、そのコピーまたはバックアップのタグを指定することによって制限できます。*tag* を指定した場合は、指定したタグを持つコピーまたはバックアップだけがリスト表示されます。

like file_name_pattern

データ・ファイル・コピーおよびアーカイブ・ログを、ファイル名パターンを指定することによって制限できます。パターンは、Oracle のパターン照合文字 '%' および '.' を使用できます。*file_name_pattern* を指定した場合は、パターンに一致する名前を持つファイルだけがリスト表示されます。

from/until time

すべてのファイルを時間範囲で修飾できます。**from** または **until** を指定した場合は、指定した時間範囲内に完了したコピーまたはバックアップだけがリスト表示されます。これらのオプションは、一方だけでも両方同時にでも指定できます。

device_type_list

このオプションは、**list backupset** コマンドだけに適用されます。これを指定した場合は、指定したタイプのデバイスのいずれかに存在しているバックアップ・セットだけがリスト表示されます。指定しなかった場合は、使用可能なバックアップ・セットがすべてリスト表示されます。

例：データ・ファイルのバックアップのリスト

データ・ファイル '?/dbs/foo.f' の既知のすべてのバックアップをリスト表示するには、次のコマンドを使用します。

```
list backupset of datafile "?/dbs/foo.f";
```

例：表領域のデータ・ファイルのコピーのリスト表示

表領域 "system" 内のすべてのデータ・ファイルのコピーをリスト表示するには、次のコマンドを使用します。

```
list copy of tablespace system;
```

スクリプトのメンテナンス

ストアド・スクリプトは名前付きのエンティティです。ストアド・スクリプトには、意味のある任意の名前を付けることができます。ストアド・スクリプトを実行するには、**execute script** コマンドを使用します。**execute script** コマンドは、*job_command_list* の中で使った場合に限って有効です。

ストアド・スクリプトのコマンドは次の 4 つです。

- ストアド・スクリプトの作成
- ストアド・スクリプトの置換
- ストアド・スクリプトの削除
- ログ・ファイル (CLI) または画面へのストアド・スクリプトの出力

ストアド・スクリプト機能は、主として、実行頻度が高い **Recovery Manager** コマンドを集めた共通リポジトリを提供するためのものです。たとえば、夜間バックアップを実行するために必要な **Recovery Manager** コマンドをまとめて、"nightlybackup" という名前の 1 つのスクリプトに入れることができます。スクリプトを、オペレーティング・システムのテキスト・ファイルではなく、リカバリ・カタログに格納しておくと、**Recovery Manager** がどのマシン上で実行されているかに関係なく、どのデータベース管理者でも **Recovery Manager** を使ってスクリプトにアクセスできるので便利です。

スクリプトの作成と置換

create コマンドまたは **replace** コマンドを使うと、ストアド・スクリプトが作成または置換され、将来の参照用としてリカバリ・カタログに格納されます。スクリプトはすぐには実行されません。ストアド・スクリプトを実行するには、**execute script** コマンドを使う必要があります。スクリプトがまだ存在していない場合は、**replace script** コマンドを使ってもスクリプトを作成できます。

スクリプトの削除

ストアド・スクリプトをリカバリ・カタログから削除するには、**delete script** コマンドを使用します。

スクリプトの出力

名前付きスクリプトを **Recovery Manager** ログ・ファイルに出力するには、**print script** コマンドを使います。

スナップショットの制御ファイルの位置の設定

Recovery Manager では、制御ファイルの読取り一貫性があるバージョンを読み取る必要がある場合、制御ファイルの一時的なバックアップが作成されます。デフォルトでは、スナップショットの制御ファイルの名前と位置はポートに固有です。

しかし、**set snapshot controlfile name to** コマンドを使うと、スナップショットの制御ファイルの書き込み先を選択できます。それ以後に作成するスナップショットの制御ファイルは、このコマンドで指定した名前と位置に従って作成されます。たとえば、次のようになります。

```
set snapshot controlfile name to '/oracle/dba/prod/snap_prod.ctl';
```

スナップショットの制御ファイル名をロー・デバイスに設定することもできます。このことは、クラスタ内の複数のインスタンスが **Recovery Manager** を使う **OPS** データベースの場合に重要です。その理由は、個々のノードのサーバー・プロセスごとに、名前と位置が同じスナップショット制御ファイルを1つずつ作成できなければならないからです。たとえば、次のようになります。

```
set snapshot controlfile name to '/dev/vgd_1_0/rlvt5';
```

関連項目：『Oracle8 Parallel Server 概要および管理』

ファイルのバックアップ

ファイルのバックアップをとるときには、ターゲット・データベースを起動しマウントする必要があります。制御ファイルは、バックアップではなく、現行の制御ファイルでなければなりません。

個々のバックアップ・ピースには、**format** オペランドを使って一意の名前を指定する必要があります。一意の名前の生成には、いくつかの置換変数を使用できます。**format** オペランドは、**backup** コマンドまたは **backup_specification** レベル、**allocate channel** コマンドの中で指定できます。

1つのバックアップ・セットに入れるファイルの数を制限できます。データ・ファイルまたはデータ・ファイル・コピーのバックアップの場合は、一般に、出力テープ・デバイスのストリーミングを維持するために必要な量を確保するために、または、バックアップによって特定のデータ・ファイルの消費が過大にならないように複数のデータ・ファイルをグループ化して1つのバックアップ・セットにします。バックアップ・セット内のファイル数が少ないほど、個々のファイルの復元が速くなります。これは、他のデータ・ファイルに属していてスキップしなければならないデータの量が少なくなるからです。アーカイブ・ログのバックアップ・セットの場合は、同じ時間範囲内のログをグループ化して、1つのバックアップ・セットに入れることをお勧めします。同じ時間範囲内のログは同時に復元する必要がある場合が多いからです。

ファイルの読取りまたはバックアップ・ピースへの書き込みのときに I/O エラーが見つかったと、ジョブは異常終了します。エラー時に書き込み中だったバックアップ・ピースは、書き込みを始めからやり直す必要があります。異常終了の前に正常に書き込まれていたバックアップ・セットは、そのまま保持されます。

データ・ファイルのバックアップ・セット

データベースが **ARCHIVELOG** モードになっている場合は、ターゲット・データベースがオープンしていても、クローズしていてもバックアップをとれます。データベースを正しく

クローズする必要はありません。データベースが **NOARCHIVELOG** モードの場合は、バックアップをとる前に、データベースを正しくクローズする必要があります。

注意： データベースが **ARCHIVELOG** モードの場合は、コールド・バックアップをとるためにデータベースを正しく停止する必要はありません。ただし、バックアップに一貫性を持たせるために、正しく停止することをお薦めします。

オフラインまたは読取り専用の表領域のバックアップも作成できます。

破損データ・ファイル・ブロックはサーバー・プロセスにより破損と設定され、バックアップ書き込まれます。Oracle は、破損ブロックのアドレスと破損のタイプを、制御ファイルに記録します。これらの制御ファイル・レコードには、**V\$BACKUP_CORRUPTION** ビューでアクセスできます。**set maxcorrupt** を使うと、バックアップ対象のデータ・ファイル内で許容できる最大破損数を指定できます。

set maxcorrupt

以前に検出されておらず、指定したデータ・ファイルまたはデータ・ファイル・リストの中で許容できるブロック破損の数に制限を設定します。バックアップ・コマンドまたはコピー・コマンドでこの数を超える破損が検出された場合、そのコマンドは異常終了します。デフォルトの制限はゼロで、これは破損ブロックを 1 つも許容できないことを意味しています。

バックアップの実行

backup コマンドは 1 つ以上のバックアップ・セットを作成するときに使用します。作成されたバックアップ・セットには、ターゲット・データベースからの 1 つ以上のデータ・ファイルまたはデータ・ファイル・コピー、アーカイブ・ログが含まれます。制御ファイルのバックアップをデータ・ファイルのバックアップ・セットに入れることもできます。1 つのファイルを複数の異なるバックアップ・セットに分割することはできません。アーカイブ・ログとデータ・ファイルを、1 つのバックアップ・セットに混在させることはできません。

バックアップ中に生成されるバックアップ・セットの数は、コマンドの中の *backup_specification* の数値、および各 *backup_object_list* 内で指定または暗黙指定されるファイルの数、**FILESERSET** 制限の値によって決まります。1 つの *backup_specification* で、1 つ以上のバックアップ・セットが生成されます。*backup_object_list* の中で指定または暗黙指定されているファイルの数が **FILESERSET** 制限を超えている場合は、*backup_specification* により複数のバックアップ・セットが生成されます。制限が指定されていない場合は、各 *backup_specification* でバックアップ・セットが 1 つずつ生成されます。

複数のバックアップ・セットを作成する場合に、複数のチャネルが割り当てられていると、**Recovery Manager** が自動的に操作をパラレル化し、パラレルで複数のバックアップ・セットを書き込みます。1 つのバックアップ・セットを複数のチャネルに分散できません。**Recovery Manager** は、自動的に 1 つのバックアップ・セットを 1 つのデバイスに割り当てま

す。*backup_specification* のすべてのバックアップ・セットを、特定のチャンネルに書き込むように指定できます。

Recovery Manager バックアップのタイプ *backup_type* は、*backup_specification_list* の中のすべての *backup_specification* に適用されます。バックアップには、次の2つのタイプがあります。

full

full と **incremental** のどちらも指定しなかった場合は、これがデフォルトです。全体バックアップでは、未使用のデータ・ファイル・ブロックだけがスキップされ、すべてのブロックがバックアップ・セットにコピーされます。アーカイブ・ログまたは制御ファイルのバックアップをとるときには、どのブロックもスキップされません。

全体バックアップは後続の増分バックアップに対しては効力がなく、増分バックアップ計画の一部とはみなされません。

incremental

レベル0より上のレベルの増分バックアップでは、前回の増分バックアップ以降に変更されたブロックだけがコピーされます。レベル0の増分バックアップは、内容は全体バックアップと同じですが、レベル0のバックアップは増分計画の一部とみなされます。

レベル0より上のレベルの増分バックアップをとろうとすると、一定のチェックが実行されます。このチェックによって、その増分バックアップが後続の **recover** コマンドで利用可能であることが確認されます。このチェックには次のようなものがあります。

-**backup** コマンドに指定した各データ・ファイルについて、レベル0のバックアップ・セットがある、またはレベル0のデータ・ファイル・コピーがあること。また、これらのものに使用不能のマークが付いていないこと。

- 作成中の増分バックアップを利用できるようにするために十分な増分バックアップが、レベル0以降に作成され、使用可能であること。

複数レベルの増分バックアップもサポートされています。レベルNの増分バックアップでは、レベルN以下の最新の増分バックアップ以降に変更されたブロックだけがコピーされます。

incremental を指定した場合は、コマンド内のすべての *backup_object_lists* に、**datafile** または **datafilecopy**、**tablespace**、**database** のいずれかを指定しなければなりません。制御ファイルまたはアーカイブ・ログ、バックアップ・セットの増分バックアップはサポートされていません。

バックアップ・コマンドのオペランド・リスト

backup コマンド全体に適用されるいくつかのオペランドを指定できます。これらのオペランドには、*backup_specification* レベルで指定できるものもあります。

tag

バックアップ・セットには、タグと呼ばれるユーザー指定の識別子を与えることができます。タグは予約語以外の文字列で、一般に、*"monday_evening_backup"* や *"weekly_full_backup"* などのようにわかりやすい名前を持ちます。タグは 30 文字以下でなければなりません。

構文を使って、*backup_command* レベルまたは *backup_specification* レベルでタグを指定できます。コマンド・レベルで指定した場合は、このコマンドにより作られるすべてのバックアップ・セットに、このタグが与えられます。*backup_specification* レベルで指定した場合は、異なるバックアップ指定により作られたバックアップ・セットには、異なるタグが与えられることがあります。両方のレベルで指定した場合は、*backup_specification* の中のタグが優先されます。

cumulative

増分バックアップにより、同じレベルの前のバックアップ以降に変更されたブロックに加えて、前回のバックアップでコピーされたすべてのブロックが再度コピーされます。

nochecksum

ブロック・チェックサムが抑止されます。このオプションを指定しなかった場合は、各ブロックについてチェックサムが計算され、バックアップに格納されます。バックアップからの復元時に、このチェックサムが検証され、復元後にデータ・ファイルに書き込まれます。

データベースですでにブロック・チェックサムが維持されている場合は、このフラグは無効です。この場合は、常にチェックサムが検証されバックアップに格納されます。

filesperset

1 つのバックアップ・セットに入れるファイルの最大数を指定します。バックアップ指定により指定または暗黙指定されているファイルの数が **filesperset** より大きい場合は、そのバックアップ指定では複数のバックアップ・セットが作成されます。

このパラメータを指定すると、**Recovery Manager** により計算値の最小値と指定値が使用されます。これにより、すべてのデバイスが使用されることになります。このパラメータを指定しないと、**Recovery Manager** により計算値の最小値

と 64 が使用されます。この場合でも、すべてのチャンネルは使用されますが、バックアップ・セット内のファイル数は限定されます。

Recovery Manager は常に、すべての割当てチャンネルが動作できるように、十分な数のバックアップ・セットを作成しようとします。ただし、バックアップをとるファイルの数よりチャンネルの数が多い場合は例外です。

setsize

このオプションを使うと、ユーザーがバックアップ・セットの最大サイズを指定できます。この制限は、1K(1024 バイト)単位で指定します。たとえば、バックアップ・セットの大きさを 3MB に制限するには、**setsize=3000** と指定します。この指定により、Recovery Manager はすべてのバックアップ・セットをこのサイズに制限します。各バックアップ・セットを 1 つのテープに収めたい場合に特に有効なオプションです。

バックアップ指定リスト

backup_specification_list は、1 つ以上の *backup_specification* からなります。

backup_specification には、少なくとも、バックアップを作成するオブジェクトのリストと、バックアップ・ピースのファイル名テンプレートを指定する *format* オペランドが含まれます。

1 つの *backup_specification* で、1 つ以上のバックアップ・セットが作成されます。

backup_object_list に指定または暗黙指定されているデータ・ファイルの数が **filesperset** の制限を超えている場合は、1 つの *backup_specification* で複数のバックアップ・セットが作成されます。

バックアップ・オブジェクト・リスト

各 *backup_specification* には、*backup_object_list* が 1 つずつ入っています。*backup_object_list* には、バックアップを作成するオブジェクトを指定します。

database

制御ファイルとすべてのデータ・ファイルがバックアップされていることを示します。

tablespace

バックアップを作成する 1 つ以上の表領域を列記して指定します。指定した表領域の中に現在あるすべてのデータ・ファイルのバックアップが作成されます。表領域はいくつでも指定できます。

キーワードの **database** および **tablespace** は、単に指定を簡単にするために提供されています。これらのフォームは、内部ではデータ・ファイルのリストに変換されます。

datafile

バックアップを作成する 1 つ以上のデータ・ファイルを列記して指定します。データ・ファイルは、ファイル名またはデータ・ファイル番号により指定できます。ファイル名を指定する場合は、リカバリ・カタログ内に表示されている（リカバリ・カタログを使う場合）、もしくは制御ファイル内に表示されている現行のデータ・ファイルの名前を指定しなければなりません。

file1(システム表領域の最初のファイル) のバックアップを作成する場合は、自動的に制御ファイルが組み込まれます。

datafilecopy

バックアップを作成する 1 つ以上のデータ・ファイルのコピーを列記して指定します。ファイルは、ファイルまたはタグにより指定できます。タグを指定し、そのタグを持つデータ・ファイル・コピーが複数ある場合は、そのデータ・ファイルの最新データ・ファイル・コピーのバックアップだけが作成されます。

archivelog

バックアップに含めるアーカイブ・ログを選択するための、ファイル名パターンまたは、時間範囲かログ順序範囲などを指定します。指定に一致するすべてのアーカイブ・ログが、バックアップに含まれます。時間範囲を指定した場合は、開始時刻と終了時刻に現行の状態にあったログがバックアップに含まれます。

current control file

現行の制御ファイルのバックアップを作成することを指定します。

backup controlfile

バックアップを作成するバックアップ制御ファイルのファイル名を指定します。

backupset

バックアップを作成するバックアップ・セットの主キーを指定します。バックアップ・セットはディスク内になければなりません。

バックアップ・オペランド・リスト

この *backup_specification* により作られるバックアップ・セットおよびバックアップ・ピースの属性を指定するオペランドのリストを次に示します。

tag

指定した場合、指定したタグがバックアップ・セットに与えられます。指定しなかった場合は、タグ値は **NULL** です。

parms

これは、OS に固有の情報を示す引用符付き文字列です。バックアップ・ピースが1つ作られるたびに、この文字列が **OSD** レイヤーに渡されます。

format

これには、バックアップ・ピースに使うファイル名を指定します。この名前は引用符で囲まなければなりません。各バックアップ・ピースがそれぞれ一意の名前を持っていれば、プラットフォーム上で順次ファイル名として有効と認められるどのような名前でも使用できます。ディスクへのバックアップの場合は、一意の名前であれば、有効と認められるどのようなディスク・ファイル名でも使用できます。

format オペランドは、次のどの場所にも指定できます。

- *backup_specification*

- **backup** コマンド

- **allocate channel** コマンド

これらの複数の場所に指定した場合、**Recovery Manager** は上記の順序で **format** オペランドを検索します。

次の置換変数は、**format** 文字列の中で一意のファイル名を生成するために使用します。

%p

バックアップ・セット内でのバックアップ・ピース番号。この値は、各バックアップ・セットごとに1から始まり、バックアップ・ピースが1つ作成されるたびに1つずつ増加します。

%s

バックアップ・セット番号。これは制御ファイル内のカウンタで、それぞれのバックアップ・セットごとに増加します。このカウンタ値は1から始まり、制御ファイルの存続期間中は一意です。バックアップ制御ファイルが復元された場合には、結果として重複値が生じます。また、**CREATE CONTROLFILE** では、このカウンタが1に初期化されます。

%d

データベース名。

%n

埋込みデータベース名。

%t

バックアップ・セット・スタンプ。これは、固定基準日時以降の経過秒数を表す 4 バイトの値です。バックアップ・セットの一意の名前を作るには、**%s** と **%t** の組合せを使用します。

%u

バックアップ・セット番号とそのバックアップ・セットの作成時刻を圧縮形で表した 8 文字の名前。

include current control file

このオペランドを指定すると、この *backup_specification* により生成される各バックアップ・セットに、現制御ファイルのスナップショットが作成され、入れられます。

filesperset

これには、1 つのバックアップ・セットに入れるデータ・ファイルの最大数を指定します。バックアップ指定により指定または暗黙指定されているデータ・ファイルの数が **filesperset** より大きい場合は、そのバックアップ指定では複数のバックアップ・セットが作成されます。

channel

この *backup_specification* でバックアップ・セットを作成するときに使うチャンネルの名前。このオペランドを指定しなかった場合、**Recovery Manager** により、この *backup_specification* のバックアップ・セットが、ジョブの実行時に使用可能なチャンネルに動的に割り当てられます。

delete input

このオペランドを使うと、バックアップ・セットの作成が正常に完了した時点で、入力ファイルが削除されます。これを指定できるのは、アーカイブ・ログまたはデータ・ファイル・コピーのバックアップを作成するときだけです。こ

れは、すべての入力ファイルを対象として **change...delete** コマンドを発行するのと同じです。

バックアップ：使用例

FOO という名前のデータベースがあり、データベース管理者がこのデータベースのバックアップを作成しようとしています。バックアップに使用できるテープ・ドライブは3つあり、データベースには26個のデータ・ファイルが入っています。管理者は、バックアップを多重化する計画を立てていて、各バックアップ・セットに4ファイルずつ入れればテープ・ドライブのストリーミングを維持するには十分なので、4の値を選択します。管理者は、データ・ファイルをどのようにグループ化してバックアップ・セットに入れるかについては、特に関心を抱いていません。

管理者は次のコマンドを発行します。

```
create script foo_full {
  allocate channel t1 type 'SBT_TAPE';
  allocate channel t2 type 'SBT_TAPE';
  allocate channel t3 type 'SBT_TAPE';
  backup full filesperset 4
    database format 'FOO.FULL.%n.%s.%p';
  run {
    execute script foo_full;
  }
}
```

このバックアップ・スクリプトでは、すべてのデータ・ファイルと制御ファイルを含めて、データベース全体のバックアップが作成されます。バックアップ対象のファイルは27個（データ・ファイルが26個と制御ファイルが1個）あり、1つのバックアップ・セットのファイル数は最大4個なので、7つのバックアップ・セットが作成されます。バックアップ・ピースのファイル名は、**FOO.FULL.database_name.x.y** の形式をとります。このジョブより前にリカバリ・カタログにバックアップ・セットが1つも記録されていなければ、*x* の値範囲は1～7になります。*y* は、それぞれのバックアップ・セットごとに1から始まり、バックアップ・ピースが1つ作成されるたびに増加します。

ファイルのコピー

多くの場合、データ・ファイルのバックアップを作成するより、コピーする方が便利です。ファイルをコピーした場合の出力は、追加の処理をしなくても使用に適したものになるからです。これに対して、バックアップ・セットは、**restore** コマンドにより処理してからでなければ使用できません。つまり、データ・ファイル・コピーに対してはメディア回復を実行できますが、バックアップ・セットの場合は、データ・ファイルが1つだけで、単一のバックアップ・ピースからなる場合であっても、直接回復を実行できません。

ファイルのコピーを作成するには、**copy** コマンドを使用します。出力ファイルは常にディスクに書き込まれます。

次のタイプのファイルをコピーできます。

- 現行のデータ・ファイル
- 前の COPY コマンドまたは他の方法で作成されたデータ・ファイル・コピー
- アーカイブ・ログ
- 現行の制御ファイル
- バックアップ制御ファイル

Copy コマンドの指定子

copy コマンドには 1 つ以上の *copy_specifiers* があり、それぞれに 1 つの入力ファイルおよび 1 つの出力ファイルを指定します。**copy** コマンドの前には、**type disk** を指定した **allocate channel** コマンドが少なくとも 1 つ必要です。

copy コマンドに、*copy_specifiers* が複数指定されていて、複数のチャンネルが割り当てられている場合、このコマンドは **allocate channel** コマンドの発行回数によって決まる並列度で、パラレルで実行されます。

次に、**copy** コマンドの各指定子について説明します。

datafile

現行データ・ファイルのコピーが作成されます。データ・ファイルは、ファイル名またはファイル番号により指定できます。ファイル名を使う場合は、制御ファイル内にリスト表示されているデータ・ファイルの名前を使わなければなりません。

datafilecopy

既存のデータ・ファイル・コピーのコピーが作成されます。その既存コピーは、前回の **copy** コマンドにより作成されたものでも、なんらかの外部オペレーティング・システム機能により作成されたものでも構いません。入力ファイルは、ファイル名またはタグによって指定できます。制御ファイルに内にリスト表示されている現行のデータ・ファイル名は使用できません。

archivelog

アーカイブ・ログのコピーが作成されます。アーカイブ・ログは、Oracle ログ・アーカイブ・プロセスにより作成されたものでも、前回の **copy** コマンドにより作成されたものでもかまいません。アーカイブ・ログはファイル名で指定しなければなりません。

current control file

現行の制御ファイルのコピーが作成されます。

backup control file

既存のバックアップ制御ファイルのコピーが作成されます。制御ファイルはファイル名またはタグにより指定できます。

オプションとして、コピー・コマンドに次のキーワードを指定できます。

tag

このオプションを指定すると、指定したタグが出力ファイルに割り当てられます。

level 0

このオプションを指定すると、データ・ファイル・コピーが増分バックアップ計画に組み込まれ、後続の増分バックアップ・セットの基礎として使用できるようになります。このオプションを指定しなかった場合は、データ・ファイル・コピーは増分バックアップ計画に影響しません。

ファイルの復元

Recovery Manager を使って、バックアップ・セット、またはディスク内のコピーから、データ・ファイルを復元できます。復元先の場所は、現在のデータ・ファイルの位置でも（現在そこにあるファイルにオーバーレイ）、新しい位置 (**set newname** コマンドを使用) でも構いません。データ・ファイルを新しい位置に復元した場合は、データ・ファイル・コピーとみなされ、制御ファイルおよびリカバリ・カタログにデータ・ファイル・コピーとして記録されます。

アーカイブ・ログが含まれるバックアップ・セットも復元できますが、Recovery Manager は、回復時に必要に応じてこの復元を自動的に実行するので、通常はこの復元は不要です。しかし、回復時に必要になるアーカイブ・ログのバックアップ・セットをあらかじめ復元しておけば、回復時のパフォーマンスが上がります。

データベースの Point-in-Time 回復

復元と回復の時刻を指定する最も簡単な方法は、**set until** コマンドを使うことです。このコマンドは、同じ **run** コマンド内の後続の **restore** コマンドおよび **switch** コマンド、**recover** コマンドに影響を及ぼします。

set until

これには、後続の **restore** コマンドまたは **recover** コマンドに使用する時点を指定します。次のキーワードを使用して指定できます。

- **time string**

- **logseq integer thread integer**

- *scn integer*

time キーワードを指定する場合、*string* には、現在有効な NLS 日付書式の文字列を指定してください。ほとんどのプラットフォームでは、この書式は NLS_DATE_FORMAT 環境変数に設定されています。

ファイルの選択

Recovery Manager は、リカバリ・カタログ（または、使用可能なリカバリ・カタログがない場合はターゲット・データベース制御ファイル）を使って、使用可能な最善のバックアップ・セットまたはコピーを、復元に使うために選択します。バックアップ・セットよりコピーの方に優先権が与えられます。また、複数の選択が可能な場合は、*until_clause* が指定されていればそれを考慮しながら、最新のバックアップ・セットまたはコピーが使用されます。

復元のためにバックアップ・セットまたはファイル・コピーを選択するには、すべての指定 (*from_tag* および *from_type*、*until_clause*) が満たされていなければなりません。自動選択実行時には、割り当てられたチャンネルのデバイス・タイプも復元において考慮されます。リカバリ・カタログ内に、指定したすべての基準を満たす使用可能なバックアップまたはコピーがない場合、Recovery Manager は、復元ジョブのコンパイル時にエラーを戻します。ジョブで割り当てられたデバイス・タイプと互換性のあるメディアの中に、バックアップ・セットまたはデータ・ファイル・コピーがないために、ファイルを復元できない場合、ジョブを取り消してください。その上で、既存のバックアップ・セットまたはデータ・ファイル・コピーと互換性があるデバイスに対するチャンネル割当てを指定して、新しいジョブを作成できます。

データ・ファイルの復元の宛先

デフォルトの **restore** コマンドでは、データ・ファイルは、リカバリ・カタログ内で指定されているそれぞれの現在位置に復元されます（たとえば、現行のデータ・ファイルがオーバーレイされます）。これが好ましくない場合は、復元の前に **set newname** を発行します。その場合、復元されたデータ・ファイルはデータ・ファイル・コピーとみなされるので、それを現行のデータ・ファイルにするための切替えを実行する必要があります。指定したファイル名が作成されるか、その名前がすでに存在している場合は上書きされます。

set newname

これには、指定したデータ・ファイルに影響を与える、後続のすべての **restore** コマンドまたは **switch** コマンドのデフォルト名を指定します。データ・ファイルの復元の前にこのコマンドを使用しなかった場合、上に説明したように、Recovery Manager はデフォルトの位置にそのファイルを復元します。

制御ファイルの復元の宛先

制御ファイルを復元するときは、宛先名を指定しなければなりません。指定したファイル名が作成されるか、その名前がすでに存在している場合は上書きされます。

制御ファイルのレプリケート

制御ファイルを複数の宛先にコピーするには、**replicate** コマンドを使用します。入力制御ファイルは名前前で指定し、出力宛先ファイルはターゲット・データベースの **control_files** 初期化パラメータに指定します。

データベースをマウントできるようにするには、制御ファイルを復元する **restore** コマンドの後で **replicate** コマンドを使用します。これは、複数の **copy controlfile** コマンドを使うのと同じです。**replicate** コマンドの前には、**type disk** を指定した **allocate channel** コマンドが必要です。

アーカイブ・ログの復元の宛先

アーカイブ・ログは、ターゲット・データベースの LOG_ARCHIVE_DEST および LOG_ARCHIVE_FORMAT 初期化パラメータを使って作成された名前を持つファイルに復元されます。復元されるアーカイブ・ログ・ファイルの名前を導出するには、これらのパラメータをポート固有の方式で組み合わせます。復元の前に **set archivelog destination to** コマンドを使ってアーカイブ・ログの宛先を指定することにより、LOG_ARCHIVE_DEST を上書きできます。復元されたアーカイブ・ログが作成されるか、すでに存在している場合は上書きされます。

set archivelog destination を使うと、データベースの復元が行われている間に、多数のアーカイブ・ログを多数の異なる場所に手動で移動できます。Recovery Manager は新しく復元されたアーカイブ・ログがどこにあるかを認識しており、LOG_ARCHIVE_DEST の中にアーカイブ・ログが入っていないか、それらのログを検出して回復できます。

set archivelog destination

これにより、後続の **restore** コマンドおよび **recover** コマンドで復元したアーカイブ・ログに使う名前を作るときに、ターゲット・データベース内の **log_archive_dest** 初期化パラメータが上書きされます。

たとえば、init.ora LOG_ARCHIVE_DEST パラメータに入っているものと異なる宛先を指定してアーカイブ・ログのバックアップを復元した場合、その後の復元操作と回復操作では新しい位置が検出されるので、LOG_ARCHIVE_DEST 内のファイルは検索されません。

データ・ファイルの復元のガイドライン

restore では、全データ・ファイルのバックアップ・セットまたは増分レベル 0 のバックアップ、データ・ファイル・コピーが復元されます。0 より上のレベルの増分バックアップは、**restore** コマンドでは、復元されません。レベル 0 のバックアップに増分バックアップを適用するためには、回復を実行する必要があります。一般に、復元を実行するのは、メディアの障害によってデータ・ファイルの現行コピーが破損した場合です。また、Point-in-Time 回復を実行する前にも復元を実行します。

後で回復を実行する予定で、**set newname** コマンドを発行して、データ・ファイルを新しい位置に復元する場合は、復元後、回復の前に切替えを実行して、復元したデータ・ファイル

を現行のデータ・ファイルにする必要があります。制御ファイルを復元するときは、*to_specifier* 句が必要です。制御ファイルを復元するデフォルトの位置はありません。

データ・ファイルを復元するときに **set newname** を指定しない場合は、データベースがクローズされているか、データ・ファイルがオフラインになっている必要があります。データベース全体を復元する場合は、そのデータベースがクローズされていなければなりません。

ターゲット・データベースがマウントされている場合は、適用可能なデータ・ファイル・コピーと、復元されたファイルを記述するアーカイブ・ログ・レコードにより、そのデータベースの制御ファイルが更新されます。

復元の前には、**チャンネル割当て**を少なくとも 1 回実行する必要があります。**from copy** オペランドを使う場合は、割り当てるチャンネルのタイプが **type disk** でなければなりません。**from backup** オペランドを使う場合は、復元が必要となるバックアップ・セット用に、適切なタイプの順次 I/O デバイスを割り当てる必要があります。適切なタイプのデバイスを割り当てておかないと、復元候補のバックアップ・セットまたはコピーが見つからず、**restore** コマンドが正常に実行されません。

複数のチャンネルが割り当てられている場合は、1 回の復元の対象となるファイルは、**割り当てられているチャンネル数**に応じた並列度で、パラレルで復元されます。復元の対象として最善の候補が同じバックアップ・セット上にあり、しかもどの *restore_specification* の **PARM** オペランドや **CHANNEL** オペランドにも矛盾がない場合は、同じバックアップ・セットからそれぞれ別の *restore_specification* によってファイルを復元できます。

restore コマンドのオペランド・リスト

restore コマンドのオペランドには次のものがあります。これらのオペランドの中には、*restore_specification* レベルで指定できるものもあります。

from backupset/datafilecopy

これには、ディスク内のファイル・コピー、またはバックアップ・セットのどちらから復元するかを指定します。このオペランドを指定しなかった場合は、**restore** コマンドによって最新のバックアップ・セットまたはファイル・コピーが選択されます。「最新」とは、メディア回復の必要度が最も低いファイル・コピーまたはバックアップ・セットを意味します。

until

他の基準がない場合は、**Recovery Manager** は、復元する最新のファイル・コピーまたはバックアップ・セットを選択します。これが望ましくない場合は、**until** 句を指定して、指定した時点での **Point-in-Time** 回復の実行に適したバックアップ・セットまたはファイル・コピーが選択されるように制限できます。

tag

デフォルトの `restore` では、使用可能な最新のバックアップ・セットまたはファイル・コピーが選択されます。この自動選択は、`from_tag` を指定することにより上書きできます。`from_tag` は、自動選択の対象を、指定したタグを持つバックアップ・セットまたはファイル・コピーに制限するために使用します。一致するタグを持つ使用可能なバックアップ・セットまたはファイル・コピーが複数ある場合は、最新のものが選択されます。

channel

この復元に使うチャンネルの名前。チャンネルを指定しなかった場合は、必要なファイルを復元するための正しいデバイス・タイプが割り当てられている任意の使用可能なチャンネルが使われます。

parms

これは、オペレーティング・システムに固有の情報を示す引用符付き文字列です。バックアップ・ピースが1つ復元されるたびに、この文字列が OSD レイヤーに渡されます。

復元指定リスト

`restore_specification_list` は1つ以上の `restore_specifications` からなります。各 `restore_specification` には、復元するオブジェクトのリストの他に、必要に応じて、`restore_command_operand_list` のオプションを上書きする復元オプションが含まれています。

復元指定

各 `restore_specification` には、`restore_object_list` が1つずつ入っています。**restore** コマンドは `restore_object` のリストに対して実行されます。

database

データベース内のすべてのデータ・ファイルが復元されます。バックアップの場合と違って、これには制御ファイルは含まれません。オプションの **skip** 引数を使うと、特定の表領域の復元をスキップできます。これは、一時データだけが格納された表領域の復元を回避する場合に便利です。

tablespace

指定した表領域内のすべてのデータ・ファイルが復元されます。

datafile

指定したデータ・ファイルが復元されます。

controlfile

制御ファイルが指定の位置に復元されます。

archivelog

指定したアーカイブ・ログが復元されます。

restore_object の表領域およびデータベース・フォームは、操作を簡単にするために提供されているだけです。これらは、該当するデータ・ファイル番号のリストに変換されます。

1 つの復元ジョブで同じデータ・ファイルを複数回指定するのは無効です。たとえば、次の例は *datafile 1* が明示的に指定されていると同時に、*system* 表領域内で暗黙的に指定されているため、無効とみなされます。

```
restore (tablespace system) (datafile 1);
```

データ・ファイルの切替え

データ・ファイルを切り替えると、データ・ファイル・コピーが現行のデータ・ファイルに変換されます。「現行のデータ・ファイル」とは、制御ファイルが指し示しているファイルを意味します（たとえば、データ・ファイル・コピーのファイル名が、制御ファイル内に記録されているデータ・ファイルの新しいファイル名になります）。データ・ファイルの場合はメディア回復が必要になります。

データ・ファイルを切り替えるのは、あるデータ・ファイル・コピーをデータ・ファイルの現行のバージョンにする必要がある場合です。これは、**ALTER DATABASE RENAME DATAFILE** コマンドを使うのと同じです。これは、現行のデータ・ファイルの位置を変更するための効率的な方法です。また、切替え処理によりコピーが「消費」されます。つまり、リカバリ・カタログおよび制御ファイル内の対応するレコードが削除されます。

切替え先のデータ・ファイルは、ファイル名またはファイル番号で指定できます。使用するデータ・ファイル・コピーは、ファイル名かタグのどちらかで指定できます。タグが不明確である場合は、最新のコピー（メディア回復の必要度が最も低いもの）が使用されます。

切替え先を指定しなかった場合は、このファイル番号に対する前の **set newname** で指定されたファイル名が、切替え先として使用されます。**switch datafile all** を指定した場合は、このジョブで **set newname** が発行されているすべてのデータ・ファイルが、それぞれの新しい名前に切り替えられます。

データ・ファイルの回復

メディアの回復を実行し、増分バックアップを適用するために、**Recovery Manager** の **recover** コマンドを使用できます。回復するか増分バックアップを適用できるのは、現行のデータ・ファイルだけです。必要に応じて、メディア回復を行うため、アーカイブ・ログのバックアップ・セットが復元されます。デフォルトでは、ログは *init.ora* ファイルの中で指

定された現行のログ・アーカイブの宛先に復元されます。別の位置を指定するためのオペランドがあります。

増分バックアップを適用するか、REDO を適用するかを選択が必要になったときは、Recovery Manager は常に増分バックアップの使用を選択します。複数のオーバーラップしたレベルの増分バックアップが使用可能な場合は、最低レベルの増分バックアップ（最も長い時間にわたっているもの）が自動的に選択されます。

データ・ファイルの回復のガイドライン

リカバリ・カタログは、できるだけ回復に使用できるようにしておいてください。使用できない場合、Recovery Manager は、可能であれば、ターゲット・データベース制御ファイルの情報を使って回復を実行します。制御ファイルの回復が必要な場合は、リカバリ・カタログが使用可能な状態になればなりません。Recovery Manager もターゲット・データベース制御ファイルも使用できない場合は、Recovery Manager は動作しません。

アーカイブ・ログまたは増分データ・ファイルのバックアップ・セットの復元が必要ないことがわかっている場合以外は、**recover** コマンドの前に **allocate channel** コマンドを少なくとも 1 回実行しなければなりません。さらに、復元が必要なバックアップ・セット用として、適切なタイプのデバイス（複数も可）を割り当てる必要があります。適切なタイプのデバイスが使用できない場合は、**recover** コマンドは正常に実行されません。複数のチャンネルが割り当てられている場合は、データ・ファイルの増分バックアップ・セットは、**allocate channel** コマンドの数に応じた並列度で、パラレルで復元されます。

データベースの Point-in-Time 回復

時刻を指定する最も簡単な方法は、**set until** を使うことです。このコマンドは、同じ **run** コマンド内の後続の **restore** コマンドおよび **switch** コマンド、**recover** コマンドに影響を及ぼします。**set until** コマンドを **restore** の後でしか **recover** の前に指定した場合は、必要な時点までにデータベースを回復できない場合があります。復元されたファイルに、すでにその時点より新しいタイムスタンプが付いている場合があるからです。このため、**set until** コマンドは通常、**restore** コマンドまたは **switch** コマンドの前に指定します。

set until

これには、後続の **restore** コマンドまたは **recover** コマンドに使用する時点を指定します。次のいずれかのキーワードを使用して指定できます。

- *time string*
- *logseq integer thread integer*
- *scn integer*

time キーワードを指定する場合、*string* には、現在有効な NLS 日付書式の文字列を指定してください。ほとんどのプラットフォームでは、この書式は **NLS_DATE_FORMAT** 環境変数で設定されます。

回復コマンド

Recovery Manager の **recover** コマンドには次の 3 つの形式があります。

- **recover database**
- **recover tablespace**
- **recover datafile**

データ・ファイルおよび表領域の回復の場合は、ターゲット・データベースが起動され、かつマウントされていなければなりません。そのデータベースがオープン状態の場合は、回復するデータ・ファイルまたは表領域はオフラインでなければなりません。

データベースの回復の場合は、そのデータベースは起動されていなければなりません、オープンされてはいけません。ターゲット・データベースは、できればマウントされている方が効率的です。ターゲット・データベースがマウントされていない場合、Recovery Manager は、init.ora の **db_name** パラメータ値および **db_domain** パラメータ値から、操作の対象にするターゲット・データベースを特定します。ターゲット・データベースの init.ora ファイル内に **db_name** パラメータが指定されていない場合、**recover** コマンドは正常に実行されません。リカバリ・カタログ内に、名前とドメインが同一のターゲット・データベースが複数ある場合も、**recover** コマンドは正常に実行されません。

データ・ファイルの回復

この形式の回復を実行するのは、メディア障害によって 1 つ以上のデータ・ファイルが破損した場合です。

表領域の回復

この形式の回復を実行するのは、メディア障害によって 1 つの表領域のすべてのデータ・ファイルが破損した場合、または表領域を以前の時点に回復する必要がある場合です。

表領域の回復は、回復するファイルに命名するための簡単な手段でもあります。回復を必要としないファイルは単に無視されます。

データベースの回復

データベースを回復するのは次の場合です。

- メディア障害によりデータベース全体が破損した。
- データベース全体を過去のある時点に回復する必要がある。
- メディア障害により制御ファイルが破損した。
- **create controlfile** が実行された。

RECOVER コマンドのオブジェクト・リスト

datafile

回復する 1 つ以上のデータ・ファイルを列記して指定します。データ・ファイルは、ファイル名またはファイル番号で指定できます。名前は、リカバリ・カタログ内で認識されているデータ・ファイルの現行の名前でなければなりません。前回のバックアップ、または前回の **resync catalog** の実行時以降に、制御ファイル内でデータ・ファイルが改名されている場合は、そのデータ・ファイルの古い名前を使う必要があります。

tablespace

回復する 1 つ以上の表領域を列記して指定します。

database

データベース全体を回復することを指定します。オプションの *until_clause* を指定すると、指定した *until* 条件に達したときに回復が停止します。

オプションの **skip** 句も指定できます。**skip** 句には、回復しない表領域のリストを指定します。これは、一時データしか設定されていない表領域の回復を回避するため、または一部の表領域の回復を延期する場合に便利です。

skip 句を使うと、Recovery Manager は、メディア回復を始める前に、指定された表領域に属するデータ・ファイルをオフラインにします。これらのファイルは、メディア回復が完了した後もオフラインのままになります。不完全回復を実行している場合は、**skip** は使用できません。かわりに、RESETLOGS オプションによってデータベースをオープンした後で、スキップした表領域を **skip forever** を使って削除する必要があります。

skip forever 句を使うと、Recovery Manager は、**drop** オプションを使ってデータ・ファイルをオフラインにします。**skip forever** は、指定した表領域がデータベースのオープン後に削除される場合に限り使用してください。

バックアップと復元の監視

バックアップまたは復元、コピーを実行するサーバー・プロセスが何をしているかを識別するには、複数の方法があります。この項では、2 つのオプションについて説明します。

チャネルへのセッションの接続

どのサーバー・プロセスがどの Recovery Manager チャネルに対応するかを識別するには、**set command id** を使い、V\$SESSION.CLIENT_INFO 列に対して問合せを実行します。

set command id

指定された *string* をすべてのチャネルの V\$SESSION.CLIENT_INFO 列に入れます。これを使うと、どの Oracle サーバー・プロセスがどのチャネルに対応するかを識別できます。

V\$SESSION.CLIENT_INFO 列には、個々の Recovery Manager サーバー・プロセスについての情報が次のいずれかの形式で入ります。

- **id=string**

この形式は、Recovery Manager によって確立されたターゲット・データベースへの最初の接続について使われます。

- **id=string, ch=channel_id**

この形式は、割り当てられたすべてのチャネルについて使われます。

例：set command id の使用方法

```
run {
  set command id to 'rman';
  allocate channel t1 type 'SBT_TAPE';
  allocate channel t2 type 'SBT_TAPE';
  backup
    incremental level 0
    format 'df_%t_%s_%p'
    filesperset 5
    (tablespace data_1);
  sql 'alter system archive log all';
}
```

進行状況の監視

ビュー V\$SESSION_LONGOPS に問い合わせることにより、バックアップおよびコピー、復元の進行状況を監視できます。

バックアップまたは復元、コピーを実行する個々のサーバー・プロセスでは、復元の特定部分について、必要な作業の全体量と比較した進行状況が報告されます。

たとえば、2つのチャネルを使って復元を行い、チャネルごとにバックアップ・セットを2つずつ（合計4セット）復元する場合は、それぞれのサーバー・プロセスによって、1つのセットについての進行状況のレポートが更新されます。そのセットが完全に復元された後、次に復元するセットについての進行状況のレポートが開始されます。

この情報は、次の SQL 文を使って問い合わせることができます。

```
SELECT sid, serial#, context, sofar, totalwork
       round(sofar/totalwork*100,2) "% Complete",
```

```
FROM v$session_longops  
WHERE compnam = 'dbms_backup_restore';
```

Recovery Manager の使用例

この章では、Recovery Manager の使用例を示します。取り挙げる項目は次のとおりです。

- NOARCHIVELOG モードでのバックアップ
- データベースと表領域のバックアップ
- Parallel Server 環境でのバックアップ
- データ・ファイルのコピー
- 増分バックアップ
- エラー処理
- コピーのための O/S ユーティリティの使用方法
- バックアップの保存
- 復元と回復
- データベースの Point-in-Time 回復
- リカバリ・カタログへの問合せ
- リカバリ・カタログへの複雑な問合せのための report コマンドの使用方法

NOARCHIVELOG モードでのバックアップ

ここでは、データベースが NOARCHIVELOG モードで運用されている場合の使用例を紹介します。管理者はデータベースを正しく停止してバックアップをとろうとしています。

表領域はスキップできます。ただし、前回のバックアップの後でオフラインにも読取り専用になっていない表領域をスキップすると、バックアップからデータベースを復元したときに、その表領域は失われます。

データベースを停止するには、**Server Manager/LineMode** を使う必要があります。

```
SVRMGR> shutdown;  
SVRMGR> startup mount
```

次に **Recovery Manager** を起動し、次のように入力します。

```
run {  
# backup the database to disk  
allocate channel dev1 type disk;  
backup (database format '/oracle/backups/bp_%s_%p'); }
```

バックアップ・ピースには、**format** に指定された文字列を使用してファイル名が自動的に付きます。バックアップのディスクへの書込み前に、バックアップ先（ファイル・システムまたはロー・デバイス）に十分な空き領域があることを必ず確認してください。

データベースと表領域のバックアップ

この項で紹介する例では、データベースは ARCHIVELOG モードで運用されています。

データベースのバックアップ

データベースのバックアップをテープにとるには、最初にテープ・デバイスを割り当てる必要があります。使用可能なデバイスを調べるには、**V\$BACKUP_DEVICE** に問い合わせます。

データベース（オフラインの表領域を除く）のバックアップをとるには次の例のように指定します。

```
run {  
allocate channel dev1 type 'sbt_tape';  
backup skip offline (database format '%d_%u'); }
```

読取り専用表領域のバックアップ作成が必要となるのは、読取り専用にした後で1度か2度だけです。読取り専用のデータ・ファイルをスキップするには、**skip read only** オプションを使用します。**skip offline** オプションを指定すると、**backup** コマンドはオフラインのデータ・ファイルにアクセスしません。オフラインのデータ・ファイルが使用できないときは、このオプションを指定してください。

表領域のバックアップ

個々の表領域のバックアップを作成するには次の例のように指定します。重要なデータが格納されている表領域（システム・データ、およびロールバック・セグメントが入っている表領域）は頻繁にバックアップをとる必要があります。一時セグメントだけしか格納されていない表領域はバックアップをとる必要はありません。この例ではディスクにバックアップをとるので、`format` に指定した文字列に従ってバックアップ・ファイルに名前が付きます。

```
run {
  allocate channel dev1 type disk;
  backup
    (tablespace system,tbs_1,tbs_2,tbs_3,tbs_4,tbs_5
    format '/oracle/backups/bp_%s_%p');
}
```

個々のデータ・ファイルのバックアップ

Recovery Manager を使って個々のデータ・ファイルのバックアップをとることもできます。この場合は次のように指定します。

```
run {
  allocate channel dev1 type 'sbt_tape';
  backup
    (datafile '?/dbs/t_dbs1.f'
    format '%d_%u');
}
```

制御ファイルのバックアップ

システム表領域の最初のデータ・ファイルのバックアップが作成されるときに、現行の制御ファイルのバックアップも自動的に作成されます。現行の制御ファイルは、明示的にバックアップに組み込んだり、個別にバックアップを作成したりできます。

```
run {
  allocate channel dev1 type 'sbt_tape';
  backup
    (tablespace tbs_5 include current controlfile
    format '%d_%u');
}
```

アーカイブ・ログのバックアップ

アーカイブ・ログのバックアップをテープに作成することもできます。アーカイブ・ログの範囲は、時刻またはログ順序により指定できます。アーカイブ・ログの範囲を指定しても、その範囲内のすべての REDO のバックアップがとられるというわけではありません。たとえば、最後のアーカイブ・ログが範囲の終わりより前に終わっていることもあり、範囲内の

アーカイブ・ログの1つが欠けていることもあります。Recovery Manager は、検出したログのバックアップを作成するだけで、警告は出しません。オンライン・ログのバックアップはとることができません。オンライン・ログは、最初にアーカイブする必要があります。

ヒント： Recovery Manager を起動する前に、NLS_LANG 環境変数および NLS_DATE_FORMAT 環境変数を設定してください。

```
NLS_LANG=american
NLS_DATE_FORMAT='Mon DD YYYY HH24:MI:SS'

run {
  allocate channel dev1 type 'sbt_tape';
  backup
    (archivelog from time 'Nov 13 1996 20:57:13'
      until time 'Nov 13 1996 21:06:05'
    all
    format '%d_%u');
}
```

次の例では、順序番号 288 から順序番号 301 の範囲のアーカイブ・ログすべてについてバックアップを作成し、バックアップの完了後にアーカイブ・ログを削除します。バックアップが失敗した場合、ログは削除されません。

```
run {
  allocate channel dev1 type 'sbt_tape';
  backup
    (archivelog low logseq 288 high logseq 301 thread 1
    all delete input
    format '%d_%u');
}
```

直前の 24 時間以内に生成されたすべてのアーカイブ・ログのバックアップを作成するには、次のコマンドを使用します。現時点までに生成されたすべての REDO のバックアップを確実に作成するために、最初に現行のログをアーカイブします。

```
run {
  allocate channel dev1 type 'sbt_tape';
  sql "alter system archive log current";
  backup
    (archivelog from time 'SYSDATE-1' all
    format '%d_%u');
}
```

関連項目： 環境変数の詳細は、それぞれのオペレーティング・システム用のマニュアルを参照してください。

Parallel Server 環境でのバックアップ

次のスクリプトでは、データ・ファイルのバックアップ作業とアーカイブ・ログのバックアップ作業をパラレル・サーバー環境の2つのインスタンスに分散します。

```
run {

allocate channel node_1 type disk connect 'internal/kernel@node_1';
allocate channel node_2 type disk connect 'internal/kernel@node_2';

  backup
    filesperset 1
    format 'df_%s_%p'
    (tablespace system, rbs, data1, data2
     channel node_1)
    (tablespace temp, reccat, data3, data4
     channel node_2);
  backup
    filesperset 20
    format 'al_%s_%p'
    (archivelog
     until time 'SYSDATE'
     thread 1
     delete input
     channel node_1);
    (archivelog
     until time 'SYSDATE'
     thread 2
     delete input
     channel node_2);
  release channel node_1;
  release channel node_2;
}
```

データ・ファイルのコピー

ここでは、Recovery Managerを使ってデータ・ファイルのコピーをディスクに作成します。データ・ファイルのコピーとは、データ・ファイルのイメージ・コピーです。

```
run {
allocate channel dev1 type disk;
copy datafile '?/dbs/tbs_01.f/dbs/tbs_01.f' to '?/copy/temp3.f';
}
```

増分バックアップ

増分バックアップには、前回のバックアップ以降に変更されたブロックだけが含まれます。最初の増分バックアップは、使用されたブロックすべてを含むレベル0のバックアップでなければなりません。

```
run {
  allocate channel dev1 type 'sbt_tape';
  backup incremental level 0
  (database
   format '%d_%u');
}
```

次の増分バックアップ（レベル1以上）には、前回のレベル0またはレベル1のバックアップ以降に変更されたブロックがすべて含まれます。

```
run {
  allocate channel dev1 type 'sbt_tape';
  backup incremental level 1
  (database
   format '&d_%u');
}
```

新しいデータ・ファイルまたは表領域をデータベースに追加したときは、増分バックアップをとる前に、レベル0のバックアップをとる必要があります。そうしないと、**Recovery Manager** が新規データ・ファイル用の親バックアップを検出できないので、表領域またはデータベースの増分バックアップは失敗します。

```
run {
  allocate channel dev1 type 'sbt_tape';
  backup incremental level 0
  (tablespace new_tbs
   format '%d_%u');
}
```

増分バックアップはNOARCHIVELOGモードで実行できます。

エラー処理

デフォルトでは、データ・ファイルから読み取られる各ブロックについてチェックサムが計算され、バックアップに格納されます。**nochecksum** オプションを使うと、チェックサムは計算されません。ただし、ブロックにすでにチェックサムが含まれている場合は、そのチェックサムの妥当性が検査され、バックアップに格納されます。妥当性検査が失敗すると、そのブロックには、バックアップ内で破損のマークが付きます。

set maxcorrupt 句により、**backup** で許容される破損ブロックの数を指定できます。**maxcorrupt** に指定した値より多くの破損ブロックを持つデータ・ファイルがあると、バックアップは終了します。

デフォルトでは、データ・ファイルにアクセスできないときは **backup** が終了します。**skip inaccessible** オプションを指定すると、**backup** の実行時にアクセスできないデータ・ファイルがスキップされ、他のファイルのバックアップが続けられます。**skip offline** を指定すると、**backup** の実行時にオフライン・ファイルにはアクセスしません。

```
run {
  allocate channel dev1 type 'sbt_tape';
  set maxcorrupt 5;
  backup (database format 'bp_%s_%p'); }
```

コピーのための O/S ユーティリティの使用方法

O/S ユーティリティを使って作成するデータ・ファイルのコピーを、リカバリ・カタログにカタログ化できます。カタログ化できるのは、ディスクに作成されるコピーだけです。バックアップ・ピースは Oracle 独自の形式なので、O/S ユーティリティは、Recovery Manager に読取り可能な形式ではバックアップを書き込むことができません。

データ・ファイルのコピーは、Oracle7 の手法を使って作成しなければなりません。データベースがオープンされていてデータ・ファイルがオンラインになっている場合は、ALTER TABLESPACE BEGIN BACKUP を発行する必要があります。その結果生成されるバックアップから、次のようにしてカタログを作成できます。

```
catalog datafilecopy '?/dbs/tbs_33.f';
```

バックアップの保存

バックアップおよびコピーを保存しなければならない期間は、次の要因によって異なります。

- バックアップをとる頻度
- 過去の Point-in-Time 回復が必要な期間

たとえば、すべてのデータ・ファイルのバックアップを毎日とる場合は、Point-in-Time 回復は必要なく、各データ・ファイルについて1つだけバックアップがあれば十分で、新しいバックアップを完了すると同時に前のバックアップは削除できます。

```
# list all files####
# delete a specific datafilecopy
change datafilecopy '?/dbs/tbs_35.f' delete;

# delete archivelogs older than 31 days
change archivelog until time 'SYSDATE-31' delete';
```

マルチピース・バックアップ・セットのすべてのピースを1つのコマンドで削除する方法は、現在のところありません。バックアップ・ピースはすべて個別に削除する必要があります。バックアップ・セットの中のピースを1つでも削除すると、そのバックアップ・セットは自動的に復元および回復の対象から外されます。

バックアップ・ピースを削除する前に、チャンネルを割り当てておく必要があります。指定のバックアップ・ピースは、同じタイプのデバイスで作成されたものでなければなりません。

allocate channel for delete コマンドは **run** コマンドの内部では発行できません。

```
# delete a backup piece
allocate channel for delete type 'sbt_tape';
change backuppiece 'testdb_87fa39e0' delete;
release channel;
```

復元と回復

情報をいつ、どのような方法で復元、回復するかは、データベースの状態と、そのデータ・ファイルの位置によって決まります。

復元、回復のための最初のステップは、次の問合せを実行してデータベースの状態を判断することです。

```
select status, parallel from v$instance;
```

オープン状態のデータベースの復元と回復

status(状態) がオープンであれば、データベースはオープン状態にあるということですが、一部の表領域とそのデータ・ファイルには復元または回復が必要な場合があります。

回復または復元するには、次の問合せを実行します。

```
select file#, status, error, recover, tablespace_name, name
from v$datafile_header;
```

エラー列が NULL でなければ、データ・ファイルにアクセスできないか、データ・ファイルのヘッダーが妥当性検査に合格しませんでした。エラーの原因がハードウェアまたはオペレーティング・システムの一時的な問題でない限り、データ・ファイルを復元するか、そのデータ・ファイルのコピーと入れ替える必要があります。

```
run {
# recover tablespace tbs_1 while the database is open
allocate channel dev1 type 'sbt_tape';
sql "alter tablespace tbs_1 offline immediate" ;
restore tablespace tbs_1 ;
recover tablespace tbs_1 ;
sql "alter tablespace tbs_1 online" ;
release channel dev1;
}
```

ディスク障害が原因でデータ・ファイルにアクセスできない場合は、そのデータ・ファイルを新しい位置に復元するか、既存のデータ・ファイルのコピーと入れ替えなければならない可能性があります。次の復元の例では、ディスク・チャンネルおよび 'sbt_tape' チャンネルを1つずつ割り当て、restore がディスク上のデータ・ファイルのコピーと 'sbt_tape' 上のバックアップとを使用できるようにします。この例では、Recovery Manager がディスクとテープの両方

から復元できるように、ディスク・デバイスとテープ・デバイスを1つずつ割り当てています。

```
run {
  allocate channel dev1 type disk;
  allocate channel dev2 type 'sbt_tape';
  sql "alter tablespace tbs_1 offline immediate" ;
  set newname for datafile 'disk7/oracle/tbs11.f'
  to 'disk9/oracle/tbs11.f' ;
  restore (tablespace tbs_1) ;
  switch datafile all ;
  recover tablespace tbs_1 ;
  sql "alter tablespace tbs_1 online" ;
  release channel dev1;
  release channel dev2;
}
```

V\$DATAFILE_HEADER では、各データ・ファイルのヘッダー・ブロックが読み取られるだけなので、データ・ファイルの復元が必要になるような問題をすべて検出できるわけではありません。たとえば、データ・ファイルに読取り不能なデータ・ブロックがあっても、そのヘッダー・ブロックに障害がなければ、エラーはレポートされません。

ファイルにはアクセスできても（エラーが NULL）、recover 列に "yes" が戻されるときは、そのファイルを回復する必要があります。Recovery Manager の **recover** コマンドでは、最初に適切な増分バックアップがすべて適用され、次にアーカイブ・ログまたはオンライン・ログ（あるいはその両方）が適用されます。増分バックアップとアーカイブ・ログは、必要に応じて復元されます。

```
run {
  # recover tablespace tbs_1 while the database is open
  allocate channel dev1 type disk;
  sql "alter tablespace tbs_1 offline" ;
  recover tablespace tbs_1 ;
  sql "alter tablespace tbs_1 online" ;
  release channel dev1;
}
```

復元

データベースの状態が「マウント済み」の場合は、制御ファイルのタイプをチェックしてください。

```
select LOG_MODE, CONTROLFILE_TYPE, OPEN_RESETLOGS from v$database;
```

データベースの状態が「現行」の場合は、データベースをオープンしてみてください。

```
alter database open;
```

正常にオープンできたら、上記のプロシージャを使って、回復が必要な表領域およびデータ・ファイルがあるかどうかを調べます。

オープンできない場合は、最初に、そのデータベースを復元または回復する（あるいはその両方）必要があります。制御ファイル・タイプが "backup" なら、おそらくデータベースの復元および回復処理は実行中です。

データベースの Point-in-Time 回復

データベースの Point-in-Time 回復を実行するには、データベースをクローズしておく必要があります。データベースをクローズするには、Server Manager を使用できます。

```
SVRMGR> shutdown abort;  
SVRMGR> startup nomount;
```

この例では、午後 3 時の時点の状態にデータベースを回復します。表領域 TBS_1 のデータ・ファイルが格納されているディスクがクラッシュし、表領域 TEMP1 には一時セグメントがあるだけでユーザー・データは格納されていないので、TEMP1 のバックアップがないとします。TBS_1 データ・ファイル用の新しい位置を指定します。デフォルトでは、Recovery Manager がファイルを回復するのは、午後 3 時の時点でのファイル位置でなく、そのファイルの最新位置です。

ヒント： Recovery Manager を起動する前に、NLS_LANG 環境変数および NLS_DATE_FORMAT 環境変数を設定してください。

```
NLS_LANG=american  
NLS_DATE_FORMAT='Mon DD YYYY HH24:MI:SS'  
  
run {  
# recover database until 3pm after restoring tbs_1 to a new location  
allocate channel dev1 type disk;  
allocate channel dev2 type 'sbt_tape';  
set until time 'Nov 15 1996 15:00:00'  
set newname for datafile '/vobs/oracle/dbs/tbs_11.f'  
to '?/dbs/templ.f' ;  
set newname for datafile '?/dbs/tbs_12.f'  
to '?/dbs/temp2.f' ;  
restore controlfile to '/vobs/oracle/dbs/cf1.f' ;  
replicate controlfile from '/vobs/oracle/dbs/cf1.f';  
sql "alter database mount" ;  
restore database skip tablespace temp1;  
switch datafile all ;  
recover database skip tablespace temp1;  
sql "alter database open resetlogs";  
sql "drop tablespace temp1";  
sql "create tablespace temp1 datafile '/vobs/oracle/dbs/templ.f'size 10M";  
release channel dev1;
```

```
release channel dev2;
}
```

ログ順序 1234 がディスク・クラッシュのために失われ、使用可能なアーカイブ・ログを使ってデータベースを回復する必要があるとします。ログ順序 1234 の後では表領域 READONLY1 は変更されていないことがわかっているので、この表領域は復元しません。

```
run {
# recover database until log sequence 1234
allocate channel dev1 type disk;
allocate channel dev2 type 'sbt_tape';
set until logseq 1234 thread 1;
restore controlfile to '/vobs/oracle/dbs/cf1.f' ;
replicate controlfile from '/vobs/oracle/dbs/cf1.f';
sql "alter database mount" ;
restore database skip tablespace templ, readonly1;
recover database skip tablespace templ;
sql "alter database open resetlogs";
sql "drop tablespace templ";
sql "create tablespace templ datafile '/vobs/oracle/dbs/templ.f' size 10M";
release channel dev1;
release channel dev2; }
```

リカバリ・カタログへの問合せ

リカバリ・カタログの内容について問い合わせるには、**list** コマンドを使用します。リカバリ・カタログを使用していない場合は、ターゲット・データベースの制御ファイルを使用できません。

```
# list all backups of files in tablespace tbs_1 that were made since
# November first.
list until time 'Nov 1 1996 00:00:00' backupset of tablespace tbs_1;
# list all backups on device type 'sbt_tape'
list device type 'sbt_tape' backupset of database;
# list all copies of a datafile, qualified by tag and directory.
list tag foo like '/somedir/%' copy of datafile 21;
# list all database incarnations registered in the recovery catalog
list incarnation of database;
```

どのコピーおよびバックアップを削除できるかを判断する場合にも **list** コマンドを使用します。たとえば、データベースの全体バックアップが 11 月 2 日に作成されていて、それ以前の時点でデータベースを回復する必要がない場合は、次のレポートに記載されているバックアップ・セットを削除できます。

```
list until time 'Nov 1 1996 00:00:00' backupset of database;
```

リカバリ・カタログへの複雑な問合せのための report コマンドの使用法

リカバリ・カタログに対してさらに複雑な問合せを発行するには、**report** コマンドを使用します。リカバリ・カタログを使っていない場合は、**report** コマンドを使ってターゲット・データベースの制御ファイルを使用できます。

```
# report on all datafiles which need a new backup because they contain
# unlogged changes that were made after the last full or incremental backup.
report unrecoverable database;

# report on all datafiles which need a new backup because 3 or more incremental
# backups have been taken since the last full backup.
report need backup incremental 3 database;

# report on all datafiles in tablespace tbs_1 which need a new backup because
# the last full or incremental backup was taken more than 5 days ago.
report need backup days 5 database;
```

Recovery Manager による表領域の Point-in-Time 回復

注意： 表領域の Point-in-Time 回復の作業は複雑なので、ここで説明する手順を使用する前に Oracle の技術サポートに連絡し、指示に従って作業することをお勧めします。

この章では、Recovery Manager (RMAN) を使用した表領域の Point-in-Time 回復 (TSPITR) の実行方法を、次の項目順で説明します。

- Recovery Manager による表領域の Point-in-Time 回復の概要
- Recovery Manager による表領域の Point-in-Time 回復
- Recovery Manager による表領域の Point-in-Time 回復の実行
- チューニングに関する考慮事項

Recovery Manager による表領域の Point-in-Time 回復の概要

Recovery Manager(RMAN) による自動化された表領域の Point-in-Time 回復 (TSPITR) 機能を使用することで、1 つ以上の表領域をデータベースの残りの表領域とは異なる時点にすばやく回復させることができます。TSPITR は、次のような状況に最も有効です。

- 誤って実行した、表の削除または切捨て操作から回復する場合。
- 論理的に壊れた表を回復させる場合。
- 不正確なバッチ・ジョブ、またはデータベースのサブセットだけに影響を与えたその他の DML 文から回復する場合。
- 1 つの物理データベースの別々の表領域に複数の論理データベースがあり、そのうちの 1 つの論理データベースを残りの論理データベースとは異なる時点に回復させる場合。
- VLDB(大規模データベース)において、データベース全体の Point-in-Time 回復が可能であるが、バックアップからデータベース全体を復元しロールフォワードを実行するのではなく、表領域の Point-in-Time 回復だけを実行する場合。(事前に 10-3 ページの「Recovery Manager による表領域の Point-in-Time 回復」を参照してください)。

表のエクスポートと同様に、RMAN の TSPITR によって一貫性のあるデータ・セットを回復できます。ただし、データ・セットは 1 つのオブジェクトではなく表領域全体になります。

Oracle8 より前のバージョンでは、Point-in-Time 回復は、次の手順でデータベースのサブセットに対してだけ使用できました。

1. データベースのコピーを作成する
2. データベースのコピーを必要な時点までロールフォワードする
3. データベースのコピーから必要なオブジェクトをエクスポートする
4. 稼働データベースから関連するオブジェクトを削除する
5. オブジェクトを稼働データベースにインポートする

この方法では、ラージ・オブジェクトのエクスポートとインポートによるパフォーマンスのオーバーヘッドがありました。

次の用語と略称はこの章で頻繁に使用されるので、覚えておいてください。

TSPITR

表領域の Point-in-Time 回復

クローン・データベース

Oracle8 の TSPITR で回復に使用されるデータベースのコピーを、「クローン・データベース」といいます。クローン・データベースは、正規のデータベースとは様々な点で本質的に異なります。

回復セット

Point-in-Time 回復を必要とする表領域です。

補助セット

TSPITR に必要な要素で、次のものが含まれます。

- バックアップ制御ファイル
- システム表領域
- ロールバック・セグメントが入っているデータ・ファイル
- 一時表領域（オプション）

ソート操作のためのエクスポート用に小さい領域が必要です。一時表領域のコピーが補助セットに含まれていない場合、クローン・データベースを起動した後に新しい一時表領域を作成するか、またはシステム表領域ファイルに対して `autoextend` を ON に設定することによって、ソート領域を作成する必要があります。

Recovery Manager による表領域の Point-in-Time 回復

Recovery Manager による TSPITR の手順は複雑なので、慎重に計画する必要があります。作業を進める前に、この章全体をよく読んでください。

警告： RMAN の TSPITR を初めて実行する場合は、稼働システムでは実行しないでください。また、時間的な制約がある状況においても実行しないでください。

注意： この章で説明している制限事項や計画手順の多くは、Chapter 13, “表領域の Point-in-Time 回復の実行” にも出てきますが、異なる点があります。それらの違いについては、この章で説明します。

制限事項

ここでは、RMAN の TSPITR の実行に関連した制限事項について説明します。計画を開始するときには、これらの制限事項をよく理解してください。

RMAN の TSPITR を実行するかどうかを判断するときに、第一に考慮しなくてはならないのは、（明示的ではなく暗黙的な参照依存性が原因で）回復済み表領域内の表と未回復の表領域内の表の間に存在するアプリケーション・レベルの矛盾の問題です。これらの依存性を理解し、矛盾がある場合には、RMAN の TSPITR を実行する前にすべての矛盾を解決するための手段を用意しておく必要があります。

Recovery Manager は、すべての表が入っている回復セットだけをサポートする

注意： これは、RMAN の TSPITR に固有の制限事項です。

Recovery Manager TSPITR は、すべての表が入っている回復セットだけをサポートします。たとえば、パーティション表に対して RMAN の TSPITR を実行し、パーティションが複数の表にまたがっている場合、TSPITR のエクスポートのフェーズで RMAN はエラー・メッセージを戻します。回復セットに制約を伴わない表が入っている場合、または表がなく制約だけが入っている場合もエラーになります。

Recovery Manager はロールバック・セグメントが入っている表をサポートしない

注意： これは、RMAN の TSPITR に固有の制限事項です。

手動で TSPITR を実行する場合、回復セット内のロールバック・セグメントをオフラインにすることができます。これによって、回復が完了する前に回復セットに変更が加えられるのを防げます。しかし、RMAN の TSPITR の場合は、ロールバック・セグメントが入っている表領域の回復をサポートしません。TSPITR とロールバック・セグメントの詳細は、13-12 ページの「ステップ 3: TSPITR 用にプライマリ・データベースを準備する」を参照してください。

TS_PITR_CHECK は SYS が所有するオブジェクトをチェックしない

TS_PITR_CHECK ビューは、TSPITR 作業の障害となる依存性や制限に関する情報を提供します。ただし、TS_PITR_CHECK は、SYS が所有するオブジェクトについては依存性や制限の情報を提供しません。

回復セット内に、取消しセグメントなど SYS 所有のオブジェクトがある場合、これらのオブジェクトの回復に成功するという保証はありません (TSPITR が使用する Export ユーティリティおよび Import ユーティリティは、SYS 所有のオブジェクトに対しては実行できないからです)。SYS が所有している回復セット・オブジェクトを調べるには、次の文を発行してください。

```
SELECT OBJECT_NAME, OBJECT_TYPE
FROM SYS.DBA_OBJECTS
WHERE TABLESPACE_NAME IN ('<tablespacename1>','<tablespacename>','
<tablespace name N') and owner = 'SYS';
```

関連項目： TS_PITR_CHECK ビューの詳細は、13-8 ページの「ステップ 2: プライマリ・データベースにおける依存性の調査と解決」を参照してください。

TS_PITR_CHECK はスナップショット表を検出しない

TS_PITR_CHECK ビューは、スナップショット表を検出しません（スナップショット・ログは検出します）。スナップショット表はスタンダアロン表としてエクスポートされます。スナップショットが時刻 3 に削除され、時刻 1 のバックアップを使用して時刻 2 にロールフォワードした場合、TSPITR 完了後にスナップショット表はスタンダアロン表として作成されています。ただし、このスタンダアロン表に関連のスナップショット・ビューは付随しません。

パーティション表と TS_PITR_CHECK

述語に与えられた表領域にパーティション表の最初のセグメントが入っている場合、TS_PITR_CHECK ビューの結果セットが反転されます。述語に与えられた表領域にパーティション表の最初のセグメントが入っていない場合には、そのパーティションについて 1 行が戻ります。述語に与えられた表領域にパーティション表の最初のセグメントが入っている場合には、結果が反転されます（たとえば、そのパーティション表のパーティションを含む各表領域について 1 行ずつ戻りますが、述語に与えられた表領域については行は戻りません）。行が戻るということは、解決しなくてはならない矛盾が存在することを示します。この矛盾は、そのパーティションをスタンダアロン表と入れ替えることで解決します。

ビットマップ索引

TSPITR が終了した後に、ビットマップ索引を削除して再度作成する必要があります。これを行わないと、ビットマップ索引を使用できません。表にビットマップ索引が存在すると、たとえビットマップ索引がプライマリ・データベースから削除されていても、インポートは失敗します。このような場合、不正確な索引セグメントが作成されることもありますが、その索引は削除して再作成する必要があります。

パーティション化されていないグローバル索引

TS_PITR_CHECK ビューは、回復セットに含まれていないパーティション表のパーティション化されていないグローバル索引を検出しません。これは、ビューを手動で問い合わせた場合、および TSPITR のエクスポートとインポートのフェーズで明確にわかります。TSPITR の終了後、エラーは戻りませんが、古い索引が回復した表に残っています。この場合、索引を削除して作成し直す必要があります。

注意： 索引は依然として有効なので、その索引を使用する問合せでは不正確な行が戻ります。

一般的な制約事項

前述の制限事項の他に、RMAN の TSPITR には次のような制約があります。

- 削除した表領域を、RMAN の TSPITR を使って回復することはできない。
- 削除後に同じ名前で再作成した表領域を、RMAN の TSPITR を使って回復することはできない。

- 間違った表領域に追加されたデータ・ファイルを、RMAN の TSPITR を使って削除することはできない。そのファイルが、RMAN の TSPITR の実行対象となる時点より後に追加された場合、RMAN の TSPITR の実行後も、そのファイルは空のまま依然として表領域の一部として存在します。
- クローン・データベースに対して DML 文は使用できない。クローン・データベースは回復処理だけに使用できます。
- RMAN の TSPITR の終了後は、回復セットのデータ・ファイルの既存のバックアップを回復処理に使用できない。このため、回復したファイルのバックアップを新たに作成する必要があります。TSPITR を実行する前に作成したバックアップを使って回復処理を行うと、その回復処理は失敗します（エラー・メッセージ ORA-1247, 00000 「表領域: %s の TSPITR を使用してデータベースを回復します。」が戻されます）。
- RMAN の TSPITR は、オブジェクトについて計算された統計情報が入っているオブティマイザ統計を回復しない。TSPITR の実行後、統計を計算し直す必要があります。
- 次のオブジェクト型は、RMAN の TSPITR 回復セット内では使用できません。
 - レプリケートされたマスター表
 - varray 列が入っている表
 - ネストした表が入っている表
 - 外部 Bfile を伴う表
 - スナップショット・ログ
 - スナップショット表
 - SYS が所有するオブジェクト（ロールバック・セグメントを含む）

データ整合性と Recovery Manager による TSPITR

TSPITR は、回復する表領域内のオブジェクトとデータベースの残りの部分にあるオブジェクトとの間のデータの関連を検出できるビューを提供します。この関連を削除または中断するか、もしくは関連オブジェクトを回復セットに組み込むことによりこのデータの関連を管理しない限りは、TSPITR は正常に実行されません。

関連項目：詳細は、13-8 ページの「ステップ 2: プライマリ・データベースにおける依存性の調査と解決」および 10-4 ページの「TS_PITR_CHECK は SYS が所有するオブジェクトをチェックしない」の説明を参照してください。

Recovery Manager による TSPITR の計画の必要条件

RMAN の TSPITR を実行するには、事前に次の必要条件を満たしている必要があります。

クローン・インスタンスの Oracle パスワード・ファイルを作成する

Oracle パスワード・ファイルの作成とメンテナンスについては、『Oracle8 Server 管理者ガイド』の 1-9 ページ「パスワード・ファイルの管理」を参照してください。

クローン・インスタンスの init.ora パラメータを設定する

クローン・インスタンスが使用する *init.ora* ファイルにある次のパラメータを設定する必要があります。

- CONTROL_FILES
- DB_FILE_NAME_CONVERT
- LOG_FILE_NAME_CONVERT

これらの設定値は、クローン・データベースにおける制御ファイル名および変換されたデータ・ファイル名、変換されたログ・ファイル名に適用されます。

クローン・インスタンスの init.ora パラメータの設定値の例 クローン・インスタンスの *init.ora* パラメータの設定値の例を次にあげます。

```
control_files=/oracle/clone/cf/clon_prod_cf.f
db_file_name_convert=("/oracle/prod/datafile", "/oracle/clone/datafile")
log_file_name_convert=("/oracle/prod/redo_log", "/oracle/clone/redo_log")
```

関連項目： DB_FILE_NAME_CONVERT の詳細は、10-9 ページの「チューニングに関する考慮事項」を参照してください。

注意： これらのパラメータを設定した後に、ターゲット・データベースの稼働ファイルの *init.ora* の設定値を上書きしないように気を付けてください。

クローン・データベースを作成する

RMAN の TSPITR を開始する前に、クローン・データベースを起動し (NOMOUNT 起動で)、稼働させてください。

Recovery Manager のコマンド行インタフェースを起動する

次のいずれかの方法で、RMAN のコマンド行インタフェースを起動してください。

クローン・インスタンスおよびターゲット・インスタンス、リカバリ・カタログに接続する クローン・インスタンスおよびターゲット・インスタンス、リカバリ・カタログに接続するには、Recovery Manager の起動時に次の書式で情報を渡します。

```
rman target sys/<tsyspwd>@<target_str>
  rcvcat rman /<rmanpwd>@<rcvcat_str>
  clone sys/<csyspwd>@<clone_str>
```

このとき、

<tsyspwd>	ターゲット・データベースの orapwd ファイルに指定されている "connect sys" パスワード
<target_str>	ターゲット・データベースの TNS 別名
<rmanpwd>	リカバリ・カタログの orapwd ファイルに指定されている "connect rman" パスワード
<rcvcat_str>	リカバリ・カタログ・データベースの TNS 別名
<csyspwd>	クローン・データベースの orapwd ファイルに指定されている "connect sys" パスワード
<clone_str>	クローン・データベースの TNS 別名

クローン・インスタンスに接続しないで Recovery Manager を起動する クローン・インスタンスへ接続しないで Recovery Manager のコマンド行インタフェースを起動し、その後で次のように Recovery Manager コマンドの **connect clone** を使ってクローン接続を確立できます。

```
RMAN> connect clone sys/<syspwd>@<clone_str>;
```

Recovery Manager による表領域の Point-in-Time 回復の実行

計画に必要な条件をすべて満たした後、RMAN の TSPITR を実行できます。

RMAN の TSPITR を実行するには、次のコマンドを発行します (**<tbslist>** は、回復セット内の表領域名のリストです)。

```
'allocate clone channel'
'recover tablespace <tbslist> until'
```

allocate clone channel コマンドで、少なくとも 1 つのクローン・チャネルを割り当てなくてはなりません。

注意： 表領域の回復セットに、システム表領域やロールバック・セグメントが入った表領域を指定することはできません。

次の例に示す文は RMAN の TSPITR を実行します。

```
RMAN> run {
  allocate clone channel dev1 type 'SBT_TAPE';
  recover tablespace tbs_2, tbs_3 until time 'Jan 10 1998 20:00:00';
}
```


Recovery Manager による TSPITR 実行後の表領域のバックアップ

回復セット内の表領域は、RMAN の TSPITR が正常に実行されるまでオフラインのままです。

これらの表領域をオンラインにする前に、表領域のバックアップを作成しておく必要があります。回復セット内のデータ・ファイルの以前のバックアップはすべて無効です。

RMAN の TSPITR が正常に実行された後で、繰り返し実行することはできません。この時点で、回復セット内のすべての表領域とデータ・ファイルには、新しい SCN が割り当てられます。ただし、RMAN の TSPITR の実行後にバックアップを作成すると、このバックアップ・セットの作成時刻後の **until_time** に対して RMAN の TSPITR を再度実行できます。

RMAN の TSPITR の実行後、クローン・データベースは使用できません。したがって、SHUTDOWN ABORT 文を使ってメモリーを解放する必要があります。

RMAN の TSPITR の実行に失敗した場合は、SHUTDOWN ABORT を使ってクローン・インスタンスを停止し、エラーを見つけて修正した後に (NOMOUNT を使って) 再起動してください。たとえば、ターゲット・データベースと変換されたファイル名の間に矛盾がある場合、クローン・インスタンスを停止し、変換されたデータ・ファイル名を修正し、(NOMOUNT を使って) 再起動し、RMAN の TSPITR を再実行します。

また、RMAN の TSPITR を正常に実行した後、次のファイルを削除できます。

- RMAN の TSPITR の実行時に一時的な格納場所に復元されたクローン補助セット・データ・ファイル
- クローン・データベースの制御ファイル
- クローン・データベースの REDO ログ・ファイル

チューニングに関する考慮事項

ここでは、RMAN の TSPITR のパフォーマンスに影響を与える事柄について説明します。

補助セットの表領域内のデータ・ファイルへの新しい名前の指定

Recovery Manager は、クローン・インスタンスで回復セットおよび補助セット内の表領域に入っているすべてのデータ・ファイルを復元し、回復します。(補助セットには、システム表領域とロールバック・セグメントが入ったすべての表領域が含まれます)。Recovery Manager コマンドの **set newname** を使って、補助セット表領域内のデータ・ファイルに新しい名前を指定できます。Recovery Manager は、この新しいファイル名を、そのデータ・ファイルを復元し回復するための一時的な格納場所として利用します。この新しいファイル名によって、init.ora パラメータの **DB_FILE_NAME_CONVERT** の設定値が上書きされます。回復セット表領域内のあらゆるデータ・ファイルに新しい名前を指定できます。

回復セット表領域内のデータ・ファイルに新しい名前を指定すると、そのデータ・ファイルによって、ターゲット・データベースの制御ファイル内の元のデータ・ファイルが上書きされます。このように既存のファイル名が新しいファイル名に置き換えられます。

Recovery Manager は、クローン・データベースとターゲット・データベースでのデータ・ファイル名の矛盾をチェックしません。なんらかの矛盾があった場合、RMAN エラーになります。

Recovery Manager による TSPITR 用のクローン名の設定とデータ・ファイル・コピーの使用

ディスク上のデータ・ファイルのコピーを使用した方が、データ・ファイルを復元するよりも高速です。このため、データ・ファイルを復元し回復するかわりに、回復セットまたは補助セット内の該当するデータ・ファイルのコピーを使用してください。

次の 2 つの条件が満たされている場合、Recovery Manager はデータ・ファイルのコピーを使用します。

1. データ・ファイル・コピーの名前が、次のコマンドによって対応するデータ・ファイルのクローン名としてリカバリ・カタログに登録されている (**<datafile>** はデータ・ファイルの名前または番号、**<clone_datafilename>** はデータ・ファイルのクローン名)。

```
RMAN> set clone_name for datafile <datafile> to <clone_datafilename>
```

2. データ・ファイル・コピーが、次の RMAN コマンドによって (**<datafile>** はデータ・ファイル名) 回復の "until_time" の前に作成されている。

```
RMAN> run {  
  copy datafile <datafile> to clone_name;  
  ...}
```

例

次のコマンドは、Recovery Manager に必要な条件の例を示しています。

```
RMAN> set clone_name for datafile '/oracle/prod/datafile_1_1.dbf'  
  to '/oracle/prod_copy/datafile_1_1.dbf';  
  
RMAN> run {  
  allocate channel dev1 type disk;  
  copy datafile '/oracle/prod/datafile_1_1.dbf'  
  to clone_name;  
}
```

同じデータ・ファイルについて **set newname** を使用した場合、Recovery Manager はデータ・ファイルのコピーを使用しません。

Recovery Manager がデータ・ファイルのコピーを使用し、TSPITR が正常に終了した場合、リカバリ・カタログ内の **clone_datafilename** には、"deleted" のマークが設定されます。

RMAN の TSPITR の終了後、ターゲット・データベースの元のデータ・ファイルは、このデータ・ファイルのコピーに置き換えられます。

変換されたデータ・ファイル名の使用

補助セットの表領域内のデータ・ファイルに対して新規名もクローン名も設定されていない場合、**Recovery Manager** は、クローン・データベースの制御ファイル内に設定されている変換されたファイル名を使って、復元と回復を実行できます。**Recovery Manager** は、クローン・データベースでのデータ・ファイル名とターゲット・データベースでのデータ・ファイル名との間に矛盾がないかチェックします。矛盾があると、エラーが戻ります。

回復セットの表領域内のデータ・ファイルに対して新規名もクローン名も設定されていない場合、もしくはクローン名のファイルが使用不可能な場合、**Recovery Manager** は元のデータ・ファイルを使用します。

第 III 部

バックアップおよびリカバリ
—オペレーティング・システム

オペレーティング・システムのバックアップ の実行

この章では、Oracle データベース内のデータのバックアップ方法について説明します。主な項目は次のとおりです。

- バックアップの実行
- 非一貫性バックアップからの回復
- データベース保護を強化するための Export ユーティリティと Import ユーティリティの使用方法

関連項目： この章では、Oracle Enterprise Manager について記述している箇所があります。Enterprise Manager の GUI または Server Manager の LineMode を使用した作業の実行方法については、『Oracle Enterprise Manager 管理者ガイド』を参照してください。

バックアップの実行

ここでは、データベースのバックアップをとる場合のさまざまな局面について説明します。取り挙げる項目は次のとおりです。

- バックアップ前のデータベース・ファイルのリスト表示
- データベース全体のバックアップの実行
- 表領域およびデータ・ファイル、制御ファイル、アーカイブ・ログのバックアップの実行
- 制御ファイルのバックアップの実行

バックアップ前のデータベース・ファイルのリスト表示

データベース全体（または表領域、データ・ファイル、制御ファイル、アーカイブ・ログ）のバックアップをとる前に、バックアップの対象になるファイルを指定します。

V\$DATAFILE ビューに問い合わせ、データ・ファイルのリストを取得してください。

```
SELECT name FROM v$datafile;
```

次に示す問合せを使って、データベースのオンライン REDO ログ・ファイルのリストを取得します。

```
SELECT member FROM v$logfile;
```

これらの問合せにより、データベースの現行の制御ファイルに格納されている情報に従って、データベースのデータ・ファイルおよびオンライン REDO ログ・ファイルがそれぞれリスト表示されます。

最後に、Enterprise Manager で次の文を発行して、データベースの現行制御ファイルの名前を取得します。

```
SHOW PARAMETER control_files;
```

制御ファイルのバックアップをとる (BACKUP CONTROLFILE TO 'filename' オプションを指定して ALTER DATABASE コマンドを実行) ときは、必ず、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制御ファイルのバックアップとともに保存してください。このリストを取得するには、BACKUP CONTROLFILE TO TRACE オプションを指定して ALTERDATABASE コマンドを実行します。制御ファイルのバックアップを TO TRACE 起動時に出力された情報とともに保存しておけば、制御ファイルのバックアップ時点でのデータベースの物理構造を明確に文書化できます。

データベース全体のバックアップの実行

データベースを通常優先度で停止してシステム全体で使用できないようにしてから、データベースを構成する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とります。データベースがオープンしているときや、インスタンスのクラッシュ後または異常終了停止後に作成したデータベースの全体バックアップは、整合性がありません。このようなバックアップは、現時点からみてファイルの整合がとれておらず一貫性のあるデータベース全体のバックアップではありません。

データベース全体のバックアップでは、データベースが特定のアーカイブ・モードで運用されている必要はありません。データベースが ARCHIVELOG モードまたは NOARCHIVELOG モードのどちらで運用されていても、データベース全体のバックアップをとることができます。データベースが NOARCHIVELOG モードであれば、バックアップには整合性がなければなりません（つまり、データベースはバックアップ前に正しく停止されていなければなりません）。

すべてのファイルに同じ時点の情報が格納されているため、一貫性のあるデータベース全体のバックアップで作成したバックアップ・ファイルのセットは整合がとれています。データベースの復元が必要な場合は、これらのファイルを使うと、データベースを特定の時点の状態に完全に復元できます。データベースが ARCHIVELOG モードで運用されている場合は、バックアップ・ファイルを復元した後で、データベースをより最近の状態に回復させるための追加の回復操作を実行できます。また、データベースが ARCHIVELOG モードであれば、一貫性のないデータベース全体のバックアップをとることもできます。

警告： データベース全体のバックアップ時に作成されたバックアップ制御ファイルは、バックアップを復元するときには、必ずそのバックアップに含まれる他のファイルとともに使用してください。完全データベース回復や不完全データベース回復には使用しないでください。一貫性のあるデータベース全体のバックアップをとる場合以外は、**BACKUPCONTROLFILE** オプションを指定して **ALTER DATABASE** コマンドを実行することにより、制御ファイルのバックアップをとるようにしてください。

関連項目：制御ファイルのバックアップ方法の詳細は、11-10 ページの「制御ファイルのバックアップの実行」を参照してください。

一貫性のあるデータベース全体のバックアップをとる準備

データベースのデータ・ファイルを確実に整合のとれた状態にするには、データベース全体のバックアップをとる前に、必ず通常優先度または即時優先度でデータベースを停止する必要があります。インスタンス障害の後、または打ち切り優先度で（つまり **SHUTDOWN ABORT** 文を使って）データベースを停止した後は、データベースが ARCHIVELOG モードで運用されていない限り、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。このような場合、データ・ファイルは特定の時点での整合性がとれていないおそれがあります。

データベース全体のバックアップの準備

1. 通常優先度または即時優先度で、データベースを停止します。

データベース全体のバックアップをとるには、データベースを停止してすべてのデータ・ファイルをクローズする必要があります。インスタンスが異常終了した場合または障害のために停止した場合は、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。この場合は、データベースを再オープンして正常に停止してから、バックアップをとってください。

2. データベースを構成するすべてのファイルのバックアップをとります。

オペレーティング・システムのコマンドまたはバックアップ・ユーティリティを使って、そのデータベースのすべてのデータ・ファイルおよび1つの制御ファイルのバックアップをとります。データベースに関連するパラメータ・ファイルのバックアップもとってください。

OS バックアップは次の方法で実行できます。

- HOST コマンドを使って、Enterprise Manager の内部で行う。
- オペレーティング・システムのコマンドまたはバックアップ・ユーティリティを使って、Enterprise Manager の外部で行う。

3. データベースを再起動します。

データベースのすべてのデータ・ファイルおよび1つの制御ファイルのバックアップをとり終わったら、データベースを再起動できます。

バックアップの検証

DB_VERIFY は、オフライン・データベースについてデータの物理構造の整合性チェックを実行する外部コマンド行ユーティリティです。バックアップ・データベース（またはデータ・ファイル）を復元する前にそれが有効かどうかを確認する場合や、データ破壊の問題が生じたときの診断基準の1つとして、DB_VERIFY を使ってください。

関連項目：DB_VERIFY の名前と格納場所は、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 システムでの名前は dbv）。オペレーティング・システム・ファイルのバックアップの作成方法については、使用しているオペレーティング・システム用の Oracle マニュアルを参照してください。

DB_VERIFY の詳細は、『Oracle8 Server ユーティリティ』を参照してください。

表領域、データ・ファイル、制御ファイル、アーカイブ・ログのバックアップの実行

表領域およびデータ・ファイル、制御ファイル、アーカイブ・ログのバックアップをとるのは、データベースを ARCHIVELOG モードで運用しているときだけにしてください（場合によっては、この状態でしかバックアップがとれないこともあります）。この状態でとったバックアップは、NOARCHIVELOG モードで運用中のデータベースの復元には使用できません。

オンライン表領域およびデータ・ファイルのバックアップの作成

データベースをオープンしている間は、個々のオンライン表領域のすべてのデータ・ファイル、または 1 つのオンライン表領域の特定のデータ・ファイルのバックアップを作成できます。

個々のデータ・ファイルまたはオンライン表領域のバックアップを開始すると、Oracle は、バックアップ中のオンライン・データ・ファイルのヘッダー内にあるチェックポイントのオカレンスを記録しなくなります。Oracle は、ALTER TABLESPACE BEGIN BACKUP 文を発行した直接の結果としてチェックポイントの記録を停止します。つまり、データベースがホット・バックアップ・モードであり、データ・ファイルを復元するときに、そのデータ・ファイルは、オンライン表領域のバックアップの最中に発生した最新のデータ・ファイルのチェックポイントではなく、バックアップの前に発生した最新のデータ・ファイルのチェックポイントの情報をもっています。この結果、回復が必要な場合には、適用すべき適切な REDO ログ・ファイルのセットを求めるメッセージが表示されることになります。データ・ファイルのコピーが完了すると、Oracle はファイル・ヘッダーを現行のデータベース・チェックポイントまで進めます。これは、ALTER TABLESPACE END BACKUP 文の実行後に行われます。

オープン状態のデータベース内にオンライン表領域のバックアップをとる方法

1. データ・ファイルを指定します。

特定のデータ・ファイルのバックアップを作成する場合は、そのデータ・ファイルの完全修飾ファイル名を使います。

表領域全体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・ビューを使って、その表領域のすべてのデータ・ファイルを指定します。たとえば、USERS 表領域のバックアップをとる場合、USERS 表領域のデータ・ファイルを指定するには、DBA_DATA_FILES ビューに問い合わせます。

```
SELECT tablespace_name, file_name
FROM sys.dba_data_files
WHERE tablespace_name = 'USERS';
```

TABLESPACE_NAME	FILE_NAME
USERS	filename1
USERS	filename2

ここで、filename1 および filename2 は、USERS 表領域のデータ・ファイルに対応する完全修飾ファイル名です。

2. オンライン表領域のバックアップの開始点にマークを付けます。

オンライン表領域のデータ・ファイルをバックアップできる状態にする（また、それらのデータ・ファイルをホット・バックアップ・モードにする）には、Enterprise Manager の「オンライン・バックアップ開始」メニュー項目を使うか、BEGIN BACKUP オプションを指定して SQL コマンドの ALTER TABLESPACE を実行します。

表領域 USERS のオンライン・バックアップの開始点にマークを付けるには次の文を使用します。

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

警告： オンライン表領域バックアップの開始点にマークを付けるのを忘れた場合や、オンライン表領域のバックアップの前に BEGIN BACKUP コマンドが完了したかどうかを確認しなかった場合は、バックアップ・データ・ファイルをそれ以降の回復操作に使うことはできなくなります。このようなバックアップを回復するのは危険です。エラーが発生して結果的には後でデータに矛盾が出るおそれもあります。たとえば、回復操作をしようとすると「ファジー・ファイル」という警告が出て、データベースに矛盾が発生し、そのデータベースはオープンできなくなります。

3. オンラインのデータ・ファイルのバックアップをとります。

この時点で、Enterprise Manager を終了してオペレーティング・システムのコマンドを実行するか、Backup ユーティリティを起動することによって、HOST コマンドを使って Enterprise Manager 内部からオンライン表領域のオンラインのデータ・ファイルのバックアップをとることができます。

4. オンライン表領域のバックアップの終了点にマークを付けます。

オンライン表領域のデータ・ファイルのバックアップをとり終わったら、EnterpriseManager の「オンライン表領域のバックアップ終了」ダイアログ・ボックスを使うか、END BACKUP オプションを指定して SQL コマンドの ALTER TABLESPACE を実行し、オンライン・バックアップの終了点を指示します。

表領域 USERS のオンライン・バックアップの終了点にマークを付けるには次の文を使用します。

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

オンライン表領域のバックアップの終了点を指示するのを忘れた場合に、インスタンス障害または SHUTDOWN ABORT が起こると、次のインスタンスの起動時にメディア回復が必要になります（多くの場合はアーカイブ REDO ログが必要になります）。その場合にメディア

回復の実行を回避するには、`ALTER DATABASE datafile filename END BACKUP` 文を使います。

関連項目：DBA_DATA_FILES データ・ディクショナリ・ビューの詳細は、『Oracle8 Server リファレンス・マニュアル』を参照してください。

ファイルのオペレーティング・システムのバックアップの作成方法については、使用しているオペレーティング・システムに固有の Oracle マニュアルを参照してください。

データ・ファイルのバックアップ・ステータスの判別 データ・ファイルのバックアップ・ステータスを表示するには、データ・ディクショナリ表 `V$BACKUP` を使用します。この表はすべてのオンライン・ファイルのリストで、それらのファイルのバックアップ・ステータスがわかります。この表は、データベースをオープンしているときに最も役立ちます。また、クラッシュの直後にも、クラッシュ時のファイルのバックアップ・ステータスがわかるので便利です。この情報は、表領域をバックアップ・モードにしてあったかどうかを判断するために使用できます。

注意： 現在使用中の制御ファイルが、メディア障害が起こった後で復元されたバックアップまたはメディア障害後に新規作成された制御ファイルである場合、`V$BACKUP` は有効ではありません。復元または再作成された制御ファイルには、`V$BACKUP` に正しい値を入れるために Oracle が必要とする情報が格納されていないからです。また、ファイルのバックアップを復元した場合、`V$BACKUP` でそのファイルの `STATUS` 列に表示されるのは、そのファイルの最新バージョンではなく旧バージョンのバックアップ・ステータスです。したがって、このビューには、復元されたファイルについての誤った情報が格納されている可能性があります。

たとえば、次の問合せではデータ・ファイルの現行のバックアップ・ステータスが表示されます。

```
SELECT file#, status
FROM v$backup;
```

FILE#	STATUS
0011	INACTIVE
0012	INACTIVE
0013	ACTIVE
...	

`STATUS` 列の "INACTIVE" は、そのファイルが現在バックアップをとられていないことを示します。"ACTIVE" は、ファイルに現在バックアップ中のマークが付いていることを示します。

複数のオンライン表領域のバックアップ 複数のオンライン表領域のバックアップを作成する必要がある場合は、次のいずれかの方法を使います。

- オンライン表領域のバックアップをパラレル方式で作成する。たとえば、すべてのオンライン表領域のバックアップを作成できる状態にします。

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

次に、オンライン表領域のすべてのファイルのバックアップをとった後、オンライン・バックアップの完了を指示します。

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

- オンライン表領域のバックアップを順次方式で作成する。たとえば、個々のオンライン表領域を個別に準備し、バックアップを作成し、オンライン表領域ごとにバックアップを終了します。

```
ALTER TABLESPACE ts1 BEGIN BACKUP;  
backup files  
ALTER TABLESPACE ts1 END BACKUP;  
ALTER TABLESPACE ts2 BEGIN BACKUP;  
backup files  
ALTER TABLESPACE ts2 END BACKUP;
```

2 番目のオプションを使用すると ALTER TABLESPACE...BEGIN BACKUP コマンドから ALTER TABLESPACE...END BACKUP コマンドまでの処理時間を最短にできるので、このオプションを使用することをお勧めします。オンライン・バックアップの間に、その表領域についてさらに多くの REDO 情報が生成されます。

オフライン表領域およびデータ・ファイルのバックアップ

表領域がオフラインになっている間に、個々の表領域内の一部またはすべてのデータ・ファイルのバックアップを作成できます。データベースのその他の表領域はすべて、オープンしたまま、システム全体で使用できます。

注意： SYSTEM 表領域、またはアクティブなロールバック・セグメントのある表領域はオフラインにはできません。このような表領域については、次に示す手順は実行できません。

表領域をオフラインおよびオンラインにするには、MANAGE TABLESPACE システム権限が必要です。

1. オフライン表領域のデータ・ファイルを指定します。

データ・ファイルの完全修飾ファイル名で指定してください。

表領域をオフラインにする前に、データ・ディクショナリ・ビュー DBA_DATA_FILES に問い合わせ、データ・ファイル名を取得し、指定します。(11-4 ページのステップ 1 を参照。)

2. 表領域を（できれば通常優先度で）オフラインにします。

なるべく通常優先度にした方がよいわけは、いったんオフラインにした後、表領域を回復しなくても表領域をオンラインにできるからです。

表領域とそれに対応付けられたすべてのデータ・ファイルを通常優先度でオフラインにするには、Enterprise Manager の「オフライン設定」メニュー項目を使うか、OFFLINE パラメータを指定して SQL コマンドの ALTER TABLESPACE を実行します。次の文は、USERS という名前の表領域を通常優先度でオフラインにします。

```
ALTER TABLESPACE users OFFLINE NORMAL;
```

表領域を通常優先度でオフラインにした後は、その表領域のすべてのデータ・ファイルがクローズされます。

3. オフラインのデータ・ファイルのバックアップをとります。

この時点で、Enterprise Manager を終了してオペレーティング・システムのコマンドを実行するか、Backup ユーティリティを起動することにより、HOST コマンドを使って Enterprise Manager 内部からオフライン表領域のデータ・ファイルのバックアップをとることができます。

4. 表領域をオンラインにします。（オプション）

Enterprise Manager の「オンライン設定」メニュー項目を使うか、SQL コマンドの ALTER TABLESPACE に ONLINE オプションを指定して実行し、表領域をオンラインにします。次の文は、USERS という名前のオフライン表領域をオンラインにします。

```
ALTER TABLESPACE users ONLINE;
```

注意： 一時優先度または即時優先度で表領域をオフラインにした場合は、表領域回復を実行しなければ、その表領域はオフラインにできません。

表領域がオンラインになった後は、その表領域のデータ・ファイルはオープンされ使用できる状態になっています。

関連項目： ファイルのオペレーティング・システムのバックアップの詳細は、使用しているオペレーティング・システムに固有の Oracle マニュアルを参照してください。

制御ファイルのバックアップの実行

ARCHIVELOG モードで運用しているデータベースの構造を変更した場合は、そのデータベースの制御ファイルのバックアップをとってください。

データベースの制御ファイルのバックアップをとるには、ALTER DATABASE システム権限が必要です。

データベースの制御ファイルのバックアップをとるには、BACKUP CONTROLFILE オプションを指定して SQL コマンドの ALTER DATABASE を実行します。データベースの制御ファイルのバックアップを作成するには次の文を使用します。

```
ALTER DATABASE BACKUP CONTROLFILE TO 'filename' REUSE;
```

ここで、*filename* は新しい制御ファイルのバックアップの名前を示す完全修飾ファイル名です。

REUSE オプションを指定すると、現行の制御ファイルを新しい制御ファイルで上書きできます。

トレース・ファイルへの制御ファイルのバックアップの作成

ALTER DATABASE BACKUP CONTROLFILE コマンドの TRACE オプションは、制御ファイルの管理と回復に使用します。TRACE を指定すると、制御ファイルの物理バックアップは作成されず、データベースのトレース・ファイルに SQL コマンドが書き込まれます。これらの SQL コマンドにより、現行の制御ファイルに基づいて、データベースが起動し、制御ファイルが再作成され、データベースが適正に回復しオープンします。各コマンドにはコメントがついています。したがって、制御ファイルのすべてのコピーが失われた場合（または制御ファイルのサイズを変更する場合）は、トレース・ファイルからスクリプト・ファイルにこれらのコマンドをコピーし、必要に応じて編集して、そのスクリプトを使ってデータベースを回復できます。

たとえば、SALES データベースに使用可能なスレッドが3つあり、そのうちのスレッド2がパブリックでスレッド3がプライベートだとします。また、REDO ログ・ファイルは多重化されていて、オフライン表領域とオンライン表領域が1つずつあるとします。次にこの場合の例を示します。

```
ALTER DATABASE
  BACKUP CONTROLFILE TO TRACE NORESETLOGS;

3-JUN-1992 17:54:47.27:
# The following commands will create a new control file and use it
# to open the database.
# No data other than log history will be lost. Additional logs may
# be required for media recovery of offline data files. 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/databasel.dbf',
  '/diskb/prod/sales/db/filea.dbf'
;
# Take files offline to match current control file.
ALTER DATABASE DATAFILE '/diska/prod/sales/db/filea.dbf' OFFLINE

# Recovery is required if any data files are restored backups,
# or if the last shutdown was not normal or immediate.
RECOVER DATABASE;

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

# Database can now be opened normally
ALTER DATABASE OPEN;

# Files in normal offline tablespaces are now named.
ALTER DATABASE RENAME FILE 'MISSING0002'
  TO '/diska/prod/sales/db/fileb.dbf';
```

NORESETLOGS を指定せずにコマンドを実行しても、同じ情報が出力されます。
RESETLOGS を指定してコマンドを実行すると、データベースを回復してオープンするコマンドが設定された、よく似たスクリプトが生成されますが、この場合には起動時に REDO ログがリセットされます。

オンライン表領域のバックアップの失敗からの回復

次のような場合は、表領域のバックアップが失敗して、不完全なバックアップ・データが生成されるおそれがあります。

- オンライン表領域のバックアップ操作の終了を (ALTER TABLESPACE コマンドに END BACKUP オプションを指定して) 指示せず、その後データベースを ABORT オプションにより停止した。
- システム障害またはインスタンス障害、SHUTDOWN...ABORT により、バックアップが中断された。

起動時にオンライン表領域のバックアップが不完全であることを検出すると、Oracle は、起動処理を続けるためにはメディア回復（多くの場合はアーカイブ REDO ログが必要）が必要であるとみなします。メディア回復を実行しないようにするには、ALTER DATABASE DATAFILE...END BACKUP コマンドを使います。データベースを再起動する前に、前回の失敗したバックアップの対象であった表領域のすべてのデータ・ファイルをリスト表示してください。データ・ファイルがバックアップ途中だったかどうかを判断するには、V\$BACKUP ビューへの問合せを行います。

警告： 失敗したバックアップで生成された不完全なファイルをどれか 1 つでも復元した場合は、ALTER DATABASE DATAFILE...END BACKUP を使用しないでください。

データベースを再起動した後で、次のどちらかの方法を使ってデータベースを回復できます。

- STARTUP RECOVER コマンドを使ってデータベースを自動的に起動し回復する。
- インスタンスを起動し、データベースをオープンしマウントした後で、RECOVER DATABASE 文を発行する。

最初の方法では、必要な場合にだけ回復が実行されるので、この方法の方が簡単です。

関連項目：データベースの回復方法については、Chapter 12, “データベースの回復” を参照してください。

データベース保護を強化するための Export ユーティリティと Import ユーティリティの使用方法

この項では Import ユーティリティと Export ユーティリティについて説明します。取り上げる項目は次のとおりです。

- Import の使用方法
- Export の使用方法

Export および Import は、Oracle データを Oracle データベースに出し入れするためのユーティリティです。Export では、Oracle データベースからオペレーティング・システム・ファイルに特別な形式でデータが書き込まれます。Import では、Export ファイルが読み取られ、対応する情報が既存のデータベースに復元されます。Export および Import は本来 Oracle データの移動用ですが、データのバックアップを支援するツールとしても使用できます。

関連項目： Export および Import の両ユーティリティの詳細は、『Oracle8 Server ユーティリティ』を参照してください。

Export の使用方法

Export ユーティリティを使うと、データベースをオープンして使用可能な状態にしたまま、データベースのバックアップをとることができます。このユーティリティは、データベースのオブジェクトについての読取り一貫性のとれたビューを、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。

警告： Export を使ってバックアップをとる場合は、バックアップ・データが特定の一時点からみて整合がとれるように、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。Export の実行中はデータベースを変更しないようにしてください。できれば、データのエクスポート中はデータベースを制限モードで実行して、一般ユーザーがデータにアクセスできないようにしておくといでしょう。

表 11-1 に、使用できるエクスポート・モードを示します。

表 11-1 エクスポート・モード

モード	説明
ユーザー	該当する 1 人のユーザーが所有しているすべてのオブジェクトがエクスポートされる。
表	該当する 1 人のユーザーが所有しているすべての表または特定の表がエクスポートされる。
全データベース	データベースのすべてのオブジェクトがエクスポートされる。

エクスポートには次の種類があります。

増分エクスポート	<p>前回の増分エクスポートまたは累積エクスポート、完全エクスポートの後に変更されたデータベース・データだけがエクスポートされます。増分エクスポートでは、オブジェクトの定義およびそのオブジェクトのすべてのデータがエクスポートされます。増分エクスポートの実行頻度は、通常、累積エクスポートや完全エクスポートより高くなります。</p> <p>たとえば、表 A および表 B、表 C があり、前回の増分エクスポートの後で変更されたのが表 A の情報だけの場合、表 A だけがエクスポートされます。</p>
累積エクスポート	<p>前回の累積エクスポートまたは完全エクスポートの後で変更されたデータベース・データだけがエクスポートされます。</p> <p>累積エクスポートは、多数の増分エクスポートに分散している情報を集約する目的で、週に一度などの割合で実行します。</p> <p>たとえば、表 A および表 B、表 C があり、前回の累積エクスポートの後で変更されたのが表 A および表 B の情報だけの場合、表 A および表 B に対する変更内容だけがエクスポートされます。</p>
完全エクスポート	<p>すべてのデータベース・データがエクスポートされます。</p> <p>完全エクスポートは、データベース内に格納されているすべてのデータをエクスポートする目的で、月に一度などの割合で実行します。</p>

Import の使用方法

Import ユーティリティを使うと、前に作成したエクスポート・ファイルに入っているデータベース情報を復元できます。Import ユーティリティは Export ユーティリティと対のユーティリティです。

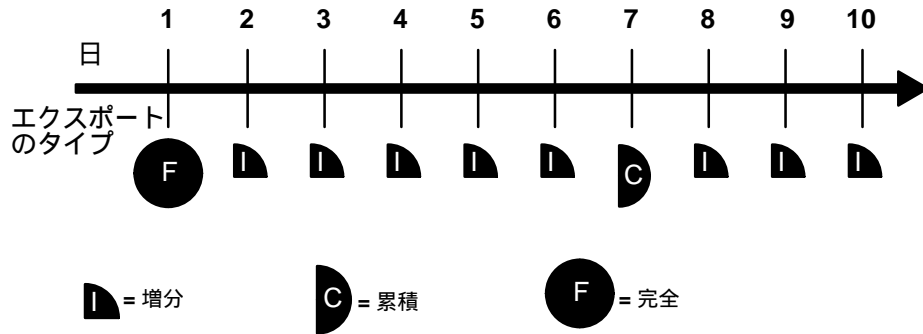
エクスポート・ファイルと Import ユーティリティを使ってデータベースを回復するには、次のようにします。

- すべての表領域およびユーザーも含めて、データベース構造を再作成する。

注意： これらの再作成した構造内にオブジェクトが入っていないはけません。

- 適切なエクスポート・ファイルをインポートして、データベースをできるだけ最新の状態に復元する。エクスポートの実行スケジュールによって、データベースの復元に必要なインポートの程度が異なります。
- 図 11-1 が示すスケジュールを使って、Oracle データベースからデータをエクスポートする。

図 11-1 典型的なエクスポート・スケジュール



完全エクスポートを毎月の1日にとり、累積エクスポートを毎週とり、増分エクスポートを毎日とっていました。この場合は次のようにします。

1. すべての表領域およびユーザーを含めたデータベースを再作成します。
2. 1日にとった完全データベース・エクスポートをインポートします。
3. 7日にとった累積データベース・エクスポートをインポートします。
4. 8日および9日、10日にとった増分データベース・エクスポートをインポートします。

データベースの回復

この章では、データベースの回復方法について説明します。取り挙げる項目は次のとおりです。

- 分散回復の調整
- 回復のシナリオ
- データベース全体のバックアップの復元、NOARCHIVELOG モード
- パラレル回復の指定
- メディア回復の準備
- 完全メディア回復の実行
- 不完全メディア回復の実行
- 災害時回復の準備
- 回復不能オブジェクトと回復
- 読取り専用表領域と回復
- 回復手順の例

関連項目： この章には、Oracle Enterprise Manager について言及している箇所があります。Enterprise Manager の GUI または Server Manager の LineMode の使用方法の詳細は、『Oracle Enterprise Manager User's Guide』を参照してください。

分散回復の調整

Oracle 分散データベース・アーキテクチャは、本来、それだけで独立して機能するアーキテクチャです。したがって、選択する回復操作のタイプによって、破損したデータベースが 1 つであっても分散データベース・システムのすべてのデータベース間で回復操作をグローバルに調整しなければならない場合と、その必要がない場合があります。表 12-1 には、様々なタイプの回復操作、および分散データベース・システムのノード間の調整が必要かどうか簡単に述べられています。

表 12-1 分散データベース環境の回復操作

回復操作のタイプ	分散データベース・システムの場合に必要な操作
リモート・ノードからはアクセス（更新または問合せ）されていなかったデータベースの全体バックアップの復元	無調整式自律データベース回復を使う。
リモート・ノードからアクセスされていたデータベース用の全体バックアップの復元	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ってすべてのデータベースを復元する。
分散データベースの 1 つ以上のデータベースの完全メディア回復	無調整式自律データベース回復を使う。
リモート・ノードからアクセスされていなかったデータベースの不完全メディア回復	無調整式自律データベース回復を使う。
リモート・ノードからアクセスされていたデータベースの不完全メディア回復	分散データベース内のすべてのデータベースについて、共通したある特定の時点の状態への、調整式不完全メディア回復を使う。

時間ベースおよび変更ベースの分散データベース回復の調整

特殊な環境では、分散データベース内の 1 つのノードを過去の時点まで回復しなければならない場合があります。その場合、グローバルなデータ整合性を保つために、システム内にある他のすべてのノードを同じ時点まで回復しなければならないことがよくあります。これを、「時間ベースの調整式分散データベース回復」と呼びます。この章で説明する時間ベースおよび変更ベースの回復の標準手順では、次の作業を実行しなければなりません。

1. 時間ベースの回復により、回復する必要があるデータベースを回復します。たとえば、ユーザー・エラー（間違っって表を削除したなど）が原因でデータベースの回復が必要になった場合は、時間ベースの回復を行って、まずそのデータベースを回復します。この時点では、他のデータベースの回復は行いません。
2. 問題のデータベースの回復が終わったら、RESETLOGS オプションを使ってそのデータベースをオープンし、そのデータベースの ALERT ファイルから RESETLOGS メッセージを探します。

メッセージが「RESETLOGS after complete recovery through changennnnnnnn」であった場合は、データベース内のすべての変更が適用され、完全回復が完了しています。分散システム内の他のデータベースは回復しないでください。それをする、と、削除する必要のない変更内容が削除されてしまいます。回復はこれで完了です。

リセット・メッセージが「RESETLOGS after incomplete recovery UNTILCHANGE nnnnnnnnn」であった場合は、不完全回復が正常に完了しています。メッセージに示された変更番号を書きとめておき、次のステップに進んでください。

3. 変更ベースの回復により、ステップ 2 で書きとめておいた変更番号 (SCN) を指定して、分散データベース・システム内の他のすべてのデータベースを回復します。

スナップショットによるデータベースの回復

マスター・データベースだけを独立させて過去の特定の時点の状態に回復した場合（つまり、時間ベースの調整式分散データベース回復を実行しなかった場合）は、その過去の時点から現在までの間にリフレッシュされた依存リモート・スナップショットについて、マスター表と整合がとれなくなります。この場合は、マスター・データベースの管理者は、整合性のないすべてのスナップショットを完全にリフレッシュするようにリモート・データベースの管理者に指示してください。

回復のシナリオ

次に、メディア回復を開始するさまざまな方法を、例をあげて説明します。

クローズ状態のデータベースの回復

データベースがマウント済みでもクローズされている場合は、Enterprise Manager の「回復アーカイブ適用」ダイアログ・ボックスを使うか、RECOVER コマンドに DATABASE パラメータを指定して、クローズ状態のデータベースの回復（完全または不完全）を開始します。

次の文を実行すると、制御ファイルのバックアップを使って、データベースが指定した時点の状態に回復されます。

```
RECOVER DATABASE
UNTIL TIME '1992-12-31:12:47:30' USING BACKUP CONTROLFILE;
```

オープン状態のデータベースでのオフライン表領域の回復

操作対象の表領域がオフラインになった後で、RECOVER コマンドに TABLESPACE パラメータを指定して、オープン状態のデータベースのオフライン表領域回復を開始できます。1 つ以上のオフライン表領域を回復できます。データベースのその他の表領域は、オープンかつオンラインの状態で、通常のデータベース運用に使用できます。

次の文では、2 つのオフライン表領域が回復されます。

```
RECOVER TABLESPACE ts1, ts2;
```

破損したファイルが格納されている表領域をオフラインにし、関連のデータ・ファイルもオフラインになっていることを確認 (V\$DATAFILE でファイルのステータスをチェック) した後で、RECOVER コマンドに DATAFILE パラメータを指定して、選択したデータ・ファイルを回復します。

```
RECOVER DATAFILE 'filename1', 'filename2';
```

通常、データベース回復には Enterprise Manager を使います。Enterprise Manager は、必要な情報の入力を求めるメッセージや、システムからのメッセージを表示します。Enterprise Manager のメディア回復オプションに相当する SQL コマンドは、RECOVER 句を指定した ALTER DATABASE コマンドです。SQL コマンドを使って独自の回復アプリケーションを設計する場合は、ALTER DATABASE コマンドを使用します。

インスタンス起動中の回復の開始

Enterprise Manager で、STARTUP コマンドに RECOVER オプションを指定して、完全メディア回復を開始できます。12-15 ページの「完全メディア回復の実行」で説明しているように、インスタンスが起動されデータベースがマウントされた後で、完全メディア回復が実行されます。

関連項目：表領域をオフラインにする方法については、11-8 ページの「オフライン表領域およびデータ・ファイルのバックアップ」を参照してください。

REDO ログ・ファイルの適用

完全または不完全メディア回復の実行時は、メディア回復のロールフォワード・フェーズで REDO ログ・ファイル（オンラインおよびアーカイブ）がデータ・ファイルに適用されます。REDO ログにはロールバック・データが記録されているので、ロールフォワードにより、対応するロールバック・セグメントが再生成されます。データベースを必要な時点まで先送りするために必要なだけの REDO ログ・ファイルのロールフォワードが行われます。ログ・ファイルが必要になると、Oracle からそのファイルの名前が表示されます。たとえば、Enterprise Manager を使っている場合は、次のメッセージとプロンプトが表示されます。

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

ALTER DATABASE... RECOVER 文を使っている場合も、同様のメッセージが表示されます。ただし、プロンプトは表示されません。

ログ・ファイルの適用

この項では、さまざまな環境でのログ・ファイルの適用方法について説明します。

ログ・ファイル名の提案

Oracle は、初期化パラメータ LOG_ARCHIVE_DEST と LOG_ARCHIVE_FORMAT の現在の設定値を連結し、さらに制御ファイルの情報を使って、ログ・ファイルの名前を提案します。したがって、必要なアーカイブ・ログ・ファイルがすべて LOG_ARCHIVE_DEST にマウントされていて、LOG_ARCHIVE_FORMAT の値が変更されていなければ、ユーザーが何もしなくても、Oracle が必要なログ・ファイルを提案して適用するので、メディア回復を自動的に実行できます。LOG_ARCHIVE_DEST に指定された位置が（メディア障害などで）使用できない場合は、このパラメータの値を変更し、ログ・ファイルを新しい位置に移して、メディア回復を開始する前に新しいインスタンスを起動できます。

場合によっては、ログ・ファイルのソースを示す LOG_ARCHIVE_DEST の現行の設定値の上書きが必要になることがあります。たとえば、データベースがオープン状態で、オフライン表領域の回復が必要なときに、LOG_ARCHIVE_DEST に指定された位置には必要なログ・ファイルをマウントするための領域が十分ないとします。この場合、ログ・ファイルを別の位置にマウントし、その別の位置を回復操作に使うように、Oracle に指示できます。必要なログ・ファイルを検出できる位置を指定するには、Enterprise Manager の SET コマンドの LOGSOURCE パラメータを使います。SQL の場合は、ALTER DATABASE コマンドの RECOVER...FROM パラメータを使います。

注意： ログ・ソースを上書きしても、満杯になってアーカイブされるオンライン・グループのアーカイブ・ログ宛先には影響ありません。

必要なログ・ファイルをすべて 1 箇所にマウントするための十分な領域がないため、LOG_ARCHIVE_DEST の現在の設定値を上書きする場合は、ログ・ファイル・ソースを、複数の位置への検索パスとして、オペレーティング・システム変数（論理変数または環境変数など）に設定できます。

関連項目： この機能はオペレーティング・システムにより異なります。詳細は、使用しているオペレーティング・システムに固有の Oracle マニュアルを参照してください。

Enterprise Manager を使用する場合のログ・ファイルの適用

提案されたアーカイブ REDO ログ・ファイルが適正であれば、それを適用してください。提案されたファイルが誤っていない限り、ファイル名を提案する必要はありません。ファイル名を指定すると、Oracle は、その REDO ログ・ファイルを適用して、復元されたデータ・ファイルをロールフォワードします。

Enterprise Manager では、次の 2 つの方法のどちらかを使います。提案された REDO ログ・ファイルが自動的に適用されるように設定できます。

- メディア回復を開始する前に、Enterprise Manager で次の文を発行して自動回復をオンにする。

```
SET AUTORECOVERY ON;
```

回復が開始されると、提案された REDO ログの自動適用が始まります。

- メディア回復が開始された後で、REDO ログ・ファイルの指定を求めるプロンプトが出たときに "auto" と入力する。この時点から、提案された REDO ログの自動適用が始まります。

提案された REDO ログ・ファイルは、それが不正確なファイル名が見つかるまでまたは回復が完了するまで、自動的に適用されます。取消しベースの回復または制御ファイルのバックアップを使っている場合は、オンライン REDO ログ・ファイルを手動で指定しなければならないことがあります。

関連項目： ログ・ファイルの適用の例については、使用しているオペレーティング・システムに固有の Oracle マニュアルを参照してください。

SQL コマンドを使用する場合のログ・ファイルの適用

REDO ログ・ファイルの適用は、ログ・ファイルの適用とほぼ同じです。ただし、メディア回復の開始後に、ログ・ファイルの指定を求めるプロンプトは表示されません。かわりに、ALTER DATABASE RECOVER LOGFILE 文を使って正しいログ・ファイルを指定する必要があります。たとえば、メッセージで LOG1.ARC が提案された場合、次の文を使ってそのログ・ファイルを適用できます。

```
ALTER DATABASE RECOVER LOGFILE 'log1.arc';
```

したがって、表領域の回復には、次の例に示すようにいくつかの文が必要です（次の例では、DBA 入力を太字で示し、変数情報をイタリックで示してあります）。

```
> ALTER DATABASE RECOVER TABLESPACE users;
ORA-00279: Change ##### generated at DD/MM/YY HH:MM:SS needed for thread #
ORA-00289: Suggestion : logfile1
ORA-00280: Change ##### for thread # is in sequence #
> ALTER DATABASE RECOVER LOGFILE 'logfile1';
ORA-00279: Change ##### generated at DD/MM/YY HH:MM:SS needed for thread # <D%0>
ORA-00289: Suggestion : logfile2
ORA-00280: Change ##### for thread # is in sequence #
> ALTER DATABASE RECOVER LOGFILE'logfile2';
(Repeat until all logs are applied.)
Statement processed.
> ALTER TABLESPACE users ONLINE;
Statement processed.
```

この例では、バックアップ・ファイルは復元済みで、ユーザーには管理者権限があることが前提になっています。

Enterprise Manager の場合と同様に、回復前および回復中にそれぞれ次の文を使って、REDO ログの自動適用を開始できます。

```
ALTER DATABASE RECOVER AUTOMATIC ...;  
ALTER DATABASE RECOVER AUTOMATIC LOGFILE suggested_log_file;
```

最初の文の例を次に示します。

```
> ALTER DATABASE RECOVER AUTOMATIC TABLESPACE users;  
Statement processed.  
> ALTER TABLESPACE users ONLINE;  
Statement processed.
```

この例では、バックアップ・ファイルは復元済みで、ユーザーには管理者権限があることが前提になっています。

ALTER DATABASE RECOVER AUTOMATIC LOGFILE 文の例を次に示します。

```
> ALTER DATABASE RECOVER TABLESPACE users;  
ORA-00279: Change ##### generated at DD/MM/YY HH:MM:SS needed for thread #  
ORA-00289: Suggestion : logfile1  
ORA-00280: Change ##### for thread # is in sequence #  
> ALTER DATABASE RECOVER AUTOMATIC LOGFILE 'logfile1';  
Statement processed.  
> ALTER TABLESPACE users ONLINE;  
Statement processed.
```

この例では、バックアップ・ファイルは復元済みで、ユーザーには管理者権限があることが前提になっています。

注意： ALTER DATABASE RECOVER コマンドを発行した後で、回復用に検討したすべてのファイルを V\$RECOVERY_FILE_STATUS ビューに表示できます。各ファイルのステータス情報には、V\$RECOVERY_STATUS ビューでアクセスできます。回復セッションを終了した後は、これらのビューにアクセスできません。

関連項目：回復に関するすべてのビューの詳細は、『Oracle8 Server リファレンス・マニュアル』を参照してください。

REDO ログの正常な適用

Enterprise Manager の回復オプション (SQL 文ではなく) を使っている場合は、Oracle が REDO ログ・ファイルを 1 つ適用し終わるたびに、次のメッセージが戻ります。

ログが適用されました。

REDO ログ・ファイルが1つ適用されるたびに、「ログが適用されました」というメッセージが出力されるのを確認してください。提案されたファイルが正しくなかった場合、または指定したファイル名が間違っていた場合は、このメッセージのかわりにエラー・メッセージが出力されます。「ログが適用されました」というメッセージが出力されずエラー・メッセージが表示された場合は、回復に必要な REDO ログ・ファイルは適用されていません。必要な REDO ログ・ファイルが適用されるまで、回復は継続できません。

REDO ログ・ファイル名を指定した後でエラー・メッセージが表示された場合は、次のいずれかのエラーが発生しています。

- ファイルが見つからないというエラー・メッセージが表示された場合は、間違ったファイル名を入力した可能性がある。正しいファイル名で入力し直してください。
- REDO ログ・ファイルは見つかったが、オープンできないというエラー・メッセージの場合は、そのファイルがロックされている可能性がある。REDO ログ・ファイルのロックを解除してから、ファイル名を入力し直してください。
- REDO ログ・ファイルが見つかってオープンできたが読み取ることができないというエラー・メッセージの場合は、I/O エラーが発生している。この場合には、REDO ログ・ファイルが部分的にしか書き込まれていないか、破損している可能性があります。そのログの破損していないコピーまたは完全なコピーの位置がわかっている場合は、そのコピーを適用するだけで済みます。回復操作を再度実行する必要はありません。それ以外の場合で、ログの他のコピーはないが最後の有効な REDO エントリの時点がわかっている場合は、時間ベースまたは変更ベースの回復を実行できます。この場合は、バックアップの復元も含めて、回復操作を最初から再度実行する必要があります。

メディア回復の中断

メディア回復操作を開始した後で中断しなければならなくなった場合（たとえば、夜になったために回復操作を終了して翌朝再開する場合など）は、次のどちらかの方法を使って、いつでも回復を中断できます。

- REDO ログ・ファイルの指定を求められたときに、"cancel" と入力する。
- 個々のデータ・ファイルの回復操作中、または自動回復の実行中に操作を中断しなければならない場合は、オペレーティング・システムの割込みシグナルを使う。

回復を取り消した場合、データベースをオープンして通常の運用を行うには、回復を完了させる必要があります。回復を再開するには、回復を再起動します。回復は、取り消された時点の状態ですべて再開されます。

警告： 回復を開始した後で再起動が必要になる場合もあります。たとえば、別のバックアップを使って再起動する場合や、同じバックアップを使用するけれども終了時刻を最初に指定した時点より前の時点に変更して再起動する必要がある場合などは、バックアップを復元して回復操作全体を再開する必要があります。これを行わないと、データベースをオープンしようとしたときに、「file inconsistent」というエラー・メッセージが表示されます。

データベース全体のバックアップの復元、NOARCHIVELOG モード

データベースが NOARCHIVELOG モードになっていて、メディア障害のためにデータ・ファイルの一部または全部が破損した場合は、通常、そのデータベースを回復するための唯一の方法は、最新の全体バックアップを復元することです。Export を使って定期バックアップを行っている場合は、復元操作のかわりに、データベースのエクスポート済みバックアップをインポートすることによりデータベースを復元できます。

NOARCHIVELOG モードの場合の欠点は、最新の全体バックアップの時点からメディア障害の発生時点までデータベースを回復するには、この期間に実行したすべての変更内容を手動で再入力しなければならない、ということです。一方、データベースが ARCHIVELOG モードになっていた場合は、アーカイブ・ログ・ファイルまたはオンライン・ログ・ファイルの形で、この期間についての REDO ログが保存されています。この場合は、完全回復または不完全回復を使ってデータベースを再構築でき、作業内容の消失も最小限に抑えることができます。

メディア障害により破損したデータベースが NOARCHIVELOG モードで運用されていた場合、それを最新の全体バックアップ（この時点での唯一のオプション）から復元するには、次の作業を行います。

最新のデータベース全体のバックアップを復元するには (NOARCHIVELOG モード)

1. データベースがオープン状態の場合は、Enterprise Manager の「データベース停止」ダイアログ・ボックスで「異常終了停止」モードを使うか、ABORT オプションを指定した SHUTDOWN コマンドを実行して、データベースを停止します。
2. メディア障害の原因になったハードウェア障害がすでに修復済みで、バックアップ・データベース・ファイルを元の場所に復元できる場合は、ステップ 2a だけを行い、ステップ 3 に進みます。ハードウェア障害がまだ修復されておらず、データベース・ファイルの一部またはすべてを別の場所に復元しなくてはならない場合は、ステップ 2a から 2d まですべて実行してください。

- a. 最新のデータベース全体のバックアップを復元します。破損したファイルだけでなく、データベース全体・バックアップのデータ・ファイルおよび制御ファイルをすべて復元する必要があります。これにより、ある 1 つの時点でのデータベース全体の整合性が保証されます。
 - a. 必要であれば、復元したパラメータ・ファイルを編集して、制御ファイルの新しい位置を指定します。
 - a. 復元し編集したパラメータ・ファイルを使っ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。ただし、データベースはまだオープンしないでください。
 - a. 復元したデータ・ファイルの再配置を記録するために必要な作業を実行します。オンライン REDO ログ・ファイルを使っている場合は、それらの REDO ログ・ファイルの再配置を記録するために必要な作業を実行します。
3. 不完全データベース回復を擬似実行する RECOVER DATABASE UNTIL CANCEL コマンドを発行します。
 4. ALTER DATABASE OPEN RESETLOGS コマンドを発行します。このコマンドにより、データベースがオープンされ、現行のログ順序番号が 1 にリセットされます。また、オンライン REDO ログ・ファイルの中のすべての REDO エントリが、このコマンドにより無効になります。データベース全体のバックアップから復元し、続いてログをリセットすることにより、バックアップがとられた時点からメディア障害の発生時点までにデータベースに対して行われたすべての変更が破棄されます。

関連項目：データ・ファイルの改名および再配置に関する詳細は、『Oracle8 Server 管理者ガイド』を参照してください。

パラレル回復の指定

RECOVERY_PARALLELISM 初期化パラメータには、回復操作で使用する同時実行回復プロセスの数を指定します。クラッシュ回復はインスタンスの起動時に行われるので、このパラメータは、クラッシュ回復に使用するプロセスの数を指定する場合に有効です。RECOVER コマンドに PARALLEL 句を指定しなかった場合は、このパラメータ値がメディア回復に使用するプロセス数のデフォルト値としても使われます。このパラメータの値は、1 より大きく、かつ PARALLEL_MAX_SERVERS パラメータの値以下でなければなりません。

一般に、パラレル回復の最大の効果は、複数のディスクにある複数のデータ・ファイルを同時に回復する場合に、回復時間を短縮できるということです。異なるディスク・ドライブ上に多数のデータ・ファイルがある場合、クラッシュ回復（インスタンス障害後の回復）やメディア回復にはパラレル回復を使うと効果的です。パラレル回復の方がシリアル回復より効率がよくなるのは、最低 8 つの回復プロセスがある場合です。

関連項目：パラレル回復の詳細は、『Oracle8 Parallel Server 概要および管理』を参照してください。

初期化パラメータの詳細は、『Oracle8 Server リファレンス・マニュアル』を参照してください。

メディア回復の準備

この項では、メディア回復の準備に関する事項について説明します。取り上げる項目は次のとおりです。

- メディア回復コマンド
- すべてのメディア回復操作に共通する事項
- データベース全体のバックアップの復元、NOARCHIVELOG モード
- パラレル回復の指定

関連項目：各種の問題に対する適切な回復方法については、12-44 ページの「回復手順の例」を参照してください。

メディア回復コマンド

基本的なメディア回復コマンドは 3 つありますが、それぞれ、回復対象のファイル・セットの判別方法が異なるだけです。どのコマンドでも、ファイルを回復できるかどうかを判別するために使う基準は同じです。メディア回復では、回復対象のファイルをロックできなければ、エラーが表示されます。これによって、2 つの回復セッションで同じファイルが回復される事態を防ぐことができます。また、使用中のファイルのメディア回復も、これで防止できます。メディア回復を実行する前に、すべてのメディア回復コマンドについてよく理解しておく必要があります。

RECOVER DATABASE コマンド

RECOVER DATABASE では、REDO を適用する必要があるすべてのオンライン・データ・ファイルに対して、メディア回復が実行されます。すべてのインスタンスが正しく停止されていて、バックアップがまったく復元されなかった場合は、回復が必要ないことを示すエラーが RECOVER DATABASE から表示されます。オープンされているデータベースのインスタンスが存在する場合にも（そのようなインスタンスはデータ・ファイル・ロックを持っているため）、このコマンドは失敗します。データベース全体（すべての表領域）に対してメディア回復を行うには、そのデータベースを EXCLUSIVE によりマウントし、クローズする必要があります。

RECOVER TABLESPACE コマンド

RECOVER TABLESPACE では、指定された表領域内のすべてのデータ・ファイルに対してメディア回復が実行されます。この回復を実行するには、表領域をオフラインにする必要があります。回復に必要なファイルがない場合は、エラーが表示されます。

RECOVER DATAFILE コマンド

RECOVER DATAFILE では、回復対象のデータ・ファイルをリスト表示します。メディア回復ロックを取得できさえすれば、データベースはオープンされていてもクローズされていてもかまいません。データベースをオープンしているインスタンスが 1 つでもあれば、データ・ファイル回復で回復できるのはオフライン・ファイルだけです。

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

すべてのメディア回復操作に共通する事項

この項では、すべての完全および不完全メディア回復操作に共通の事柄について説明します。回復プロセスを実行する前に、ここで説明する項目についてよく理解しておく必要があります。

回復するファイルの判別

表 V\$RECOVER_FILE を使って、回復するファイルを判別できることがよくあります。この表には、回復が必要なファイルがすべてリスト表示され、回復が必要な理由も示されています。

注意： 現在使用中の制御ファイルが、メディア障害が起こった後で復元されたバックアップである場合や、メディア障害後に新規作成された制御ファイルである場合には、この表は役に立ちません。復元または再作成された制御ファイルには、V\$RECOVER_FILE に正しい値を入れるために Oracle が必要とする情報が格納されていないからです。

次の問合せでは、回復が必要なデータ・ファイルのファイル ID 番号が表示されます。

```
SELECT file#, online, error
```

```
FROM v$recover_file;
```

FILE#	ONLINE	ERROR
0014	ONLINE	
0018	ONLINE	FILE NOT FOUND
0032	OFFLINE	OFFLINE NORMAL

...

ファイル番号に基づいてファイルの名前を検索するには、データ・ディクショナリ・ビュー **V\$DATAFILE** を使用します。このビューは、ファイルの **NAME** と **FILE#** を表示します。

破損したデータ・ファイルの復元

メディア障害によりデータベースの 1 つ以上のデータ・ファイルが永続的に破損した場合、破損したファイルを回復するには、破損したデータ・ファイルのバックアップを復元する必要があります。

破損したファイルの再配置

破損したデータ・ファイルを元の位置に復元できない場合（たとえば、ディスクの交換が必要になり、ファイルを代替ディスクに復元する場合）は、そのようなファイルの新しい位置に関連のデータベースの制御ファイルに指示する必要があります。

バックアップがないデータ・ファイルの回復

データ・ファイルが破損し、そのファイルについて使用できるバックアップがない場合でも、次の場合にはデータ・ファイルを回復できます。

- 元のデータ・ファイルの作成時以降に書き込まれたすべてのログ・ファイルが使用できる。
- 制御ファイルに破損したファイルの名前が格納されている（つまり、制御ファイルが、現行のものであるか、破損したデータ・ファイルをデータベースに追加した後でとられたバックアップである）。

対応するバックアップがない破損したデータ・ファイルと置き換えるための空のデータ・ファイルを新規作成するには、**ALTER DATABASE** コマンドの **CREATE DATAFILE** 句を使用します。ただし、**SYSTEM** 表領域の最初のデータ・ファイルには **REDO** ログに記録されていない情報が含まれているので、最初のデータ・ファイルに基づいて新規ファイルを作成できません。たとえば、データ・ファイル "disk1:users1" が破損し、バックアップが使用できないとします。次の文を実行すると、元のデータ・ファイル（同じサイズ）がディスク 2 に再作成されます。

```
ALTER DATABASE CREATE DATAFILE 'disk1:users1' AS 'disk2:users1';
```

注意： **ALTER DATABASE CREATE DATAFILE** 文を実行すると、旧データ・ファイルが改名されて新しいデータ・ファイルとして保存されます。

この文を使って、失われたファイルに代わるファイルを空のファイルとして作成できます。Oracle は、制御ファイルとデータ・ディクショナリの中の情報を調べて、サイズ情報を取得します。この後で、空のデータ・ファイルに対してメディア回復を実行する必要があります。回復時に、元のデータ・ファイルを作成し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、失われたデータ・ファイルに代わる空のファイルに再適用しなければなりません。データベースが NOARCHIVELOG モードで作成されていた場合は、必要なアーカイブ REDO ログが使用できないので、SYSTEM 表領域の元のデータ・ファイルを、ALTER DATABASE CREATE DATAFILE 文を使って復元することはできません。

必要なアーカイブ REDO ログ・ファイルの復元

保留中のメディア回復に必要なアーカイブ REDO ログ・ファイルはすべて、最終的にはディスク上に置いて Oracle がいつでも使用できる状態にしておく必要があります。

どのアーカイブ REDO ログ・ファイルが必要かを判別するには、表 V\$LOG_HISTORY および表 V\$RECOVERY_LOG を使用できます。V\$LOG_HISTORY には、すべてのアーカイブ・ログがリスト表示され、現行のアーカイブ・ログ・ファイル命名計画（パラメータ LOG_ARCHIVE_FORMAT により設定されたもの）の形式で表したログの推定名称も示されます。V\$RECOVERY_LOG には、Oracle が回復を実行するために必要なアーカイブ REDO ログだけがリスト表示されます。これには、ファイルの推定名称も、LOG_ARCHIVE_FORMAT の形式で示されています。データ・ファイルがデータベースに追加された時点から後のすべての REDO 情報が必要です。

領域に余裕があれば、必要なすべてのアーカイブ REDO ログ・ファイルを、初期化パラメータ LOG_ARCHIVE_DEST により現在指定されている位置に復元してください。これにより、メディア回復時に必要な時点で、Oracle が正しいアーカイブ REDO ログ・ファイルを自動的に検出します。LOG_ARCHIVE_DEST により指示されている位置に使用できる十分な領域がない場合は、必要なアーカイブ REDO ログ・ファイルの一部または全部を、Oracle がアクセスできる任意のディスクに復元しても構いません。この場合は、メディア回復の前または途中で、アーカイブ REDO ログ・ファイルの位置を指定できます。

アーカイブ・ログを適用した後で、ディスク領域を解放するには、アーカイブ REDO ログ・ファイルの復元済みコピーを削除します。ただし、各アーカイブ REDO ログ・グループのコピーがオフラインの記憶域にまだ残っていることを確認してください。

関連項目：表に関する詳細は、『Oracle8 Server リファレンス・マニュアル』を参照してください。

メディア回復の開始

破損したデータベースが ARCHIVELOG モードになっている場合は、完全メディア回復操作、不完全メディア回復操作のどちらでも実行できます。メディア回復操作を開始するには、次に示す Enterprise Manager のオプションのいずれかを使用します。

- 「回復アーカイブ適用」ダイアログ・ボックス
- Enterprise Manager の RECOVER コマンド
- SQL コマンドの ALTER DATABASE

どのタイプのメディア回復を開始する場合も、管理者権限が必要です。すべての回復セッションに互換性がなければなりません。あるセッションで不完全メディア回復を実行している間は、別のセッションで完全メディア回復を実行できません。また、マルチスレッド・サーバー・プロセスを通じてデータベースに接続している場合は、メディア回復は開始できません。

完全メディア回復の実行

この項では、完全メディア回復操作に必要な作業について説明します。取り上げる項目は次のとおりです。

- クローズ状態のデータベースの回復
- オープン状態のデータベースのオフライン表領域の回復の実行
- オープン状態のデータベースのオフライン表領域の個別回復の実行

メディア障害からの回復に必要なすべての作業を理解するには、次の手順説明を読むだけでは不十分です。知識が不十分だと思われる場合は、第4章で説明している回復についての基本的な概念と計画をよく読んで理解してください。

クローズ状態のデータベースの回復

この項では、クローズ状態のデータベースを回復するための手順について説明します。クローズ状態のデータベースの回復には、破損したすべてのデータ・ファイルを1回の操作で回復する場合と、破損した各データ・ファイルを別々の操作で個別に回復場合があります。

クローズ状態のデータベースを回復するには

1. データベースがオープン状態の場合は、Enterprise Manager の「データベース停止」ダイアログ・ボックスで「異常終了停止」モードを使うか、ABORT オプションを指定した SHUTDOWN コマンドを実行して、データベースを停止します。
2. メディア障害からの回復の場合は、できればそのメディア障害を修復します。

注意： メディア障害の原因になったハードウェア上の問題が一時的なものでデータが破壊されていない場合（たとえばディスクやコントローラでの停電の場合）は、ここで作業を終了してください。

3. ファイルの破損が修復できない場合は、メディア障害が原因で破損したデータ・ファイルの最新のバックアップ・ファイル（全体バックアップまたは部分バックアップの一部としてとったもの）だけを復元します。破損していないデータ・ファイルやオンライン REDO ログ・ファイルを復元しないでください。ハードウェア障害が解消されていて、破損したデータ・ファイルを元の位置に復元できる場合は、それらのデータ・ファイルを元の位置に復元して、この手順のステップ 6 を省略してください。ハードウェア障害がまだ解決されていない場合は、データベース・サーバーの代替記憶デバイスにデータ・ファイルを復元し、次のステップに進んでください。

注意： 特定のデータ・ファイルのバックアップがない場合は、空の置換ファイルを作成し、それを回復に使用できます。

4. Enterprise Manager を起動し、管理者権限により Oracle に接続します。
5. 新しいインスタンスを起動し、Enterprise Manager の「データベース起動」ダイアログ・ボックスを使って「起動してマウント」ラジオ・ボタンを選択するか、MOUNT オプションを指定した STARTUP コマンドを実行して、データベースをマウントします（ただし、オープンはしないでください）。
6. ステップ 3 で 1 つ以上の破損データ・ファイルを別の位置に復元した場合は、これらのファイルの新しい位置を、関連のデータベースの制御ファイルに指定する必要があります。ここでは、必要に応じて『Oracle8 Server 管理者ガイド』の「データ・ファイルを改名、再配置する」で説明されている操作を行ってください。
7. 完全メディア回復の実行中は、回復するすべてのデータ・ファイルがオンラインになっていなければなりません。データ・ファイル名を調べるには、通常は現行の制御ファイルとともに格納されているデータ・ファイルのリストで確認するか、V\$DATAFILE ビューに問い合わせます。その後で、DATAFILE ONLINE オプションを指定した ALTER DATABASE コマンドを発行し、データベースのすべてのデータ・ファイルがオンラインになっていることを確認します。たとえば、USERS1 という名前（完全修飾ファイル名）のデータ・ファイルを確実にオンラインにするには、次の文を入力します。

```
ALTER DATABASE DATAFILE 'users1' ONLINE;
```

指定のデータ・ファイルがすでにオンラインになっていれば、この文は無視されます。

8. すべての破損データ・ファイルについてクローズ状態のデータベースの回復を 1 つのステップで開始するには、Enterprise Manager の「回復アーカイブ適用」ダイアログ・ボックスまたはこれに相当する RECOVER DATABASE 文を使います。

個々の破損データ・ファイルのクローズ状態のデータベースの回復を開始するには、Enterprise Manager の RECOVER DATAFILE 文を使います。

注意： パフォーマンスを最大にするには、パラレル回復を使ってデータ・ファイルを回復します。

9. これで、Oracle は、必要な REDO ログ・ファイル（アーカイブ済みでオンライン状態）を適用してメディア回復のロールフォワード・フェーズを開始し、復元済みのデータ・ファイルを再構築します。ファイルの適用が自動化されている場合を除き、必要な個々の REDO ログ・ファイルについてプロンプトが表示されます。

必要なアーカイブ REDO ログ・ファイルをすべて復元済みデータ・ファイルに適用し終わるまで、処理が続行されます。次に、オンライン REDO ログ・ファイルが復元済みデータ・ファイルに自動的に適用されると、メディア回復の完了を示すメッセージが表示されます。完全メディア回復にアーカイブ REDO ログ・ファイルが必要でない場合には、それに関するプロンプトは表示されません。かわりに、すべての必要なオンライン REDO ログ・ファイルが適用され、メディア回復は完了します。

クローズ状態のデータベースの回復が実行されると、データベースはメディア障害の発生時点の状態に回復されます。その後、SQL コマンドの ALTER DATABASE に OPEN オプションを指定して実行すると、データベースをオープンできます。

関連項目： REDO ログ・ファイルの適用方法については、12-4 ページの「REDO ログ・ファイルの適用」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個別回復の実行

この時点では、オープン状態のデータベースにメディア障害が発生しており、このデータベースはオープン状態のままで、破損していないデータ・ファイルはオンラインで使用可能な状態になっています。破損したデータ・ファイルは、Oracle が自動的にオフラインにします。

この手順は、SYSTEM 表領域のデータ・ファイルの完全メディア回復には使用できません。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、Oracle はそのデータベースを自動的に停止します。

関連項目： 完全メディア回復を行うには、12-15 ページの「クローズ状態のデータベースの回復」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個別回復を実行するには

1. この回復操作の開始点は、メディア障害の発生後にデータベースがオープン状態かどうかによって異なります。
 - データベースが停止されている場合は、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。この操作を実行するには、Enterprise Manager の「データベース起動」ダイアログ・ボックスを使って「起動してオープン」ラジオ・ボタンを選択するか、OPEN オプションを指定した STARTUP コマンドを実行します。データベースをオープンした後で、破損したデータ・ファイルがある表領域をすべてオフラインにしてください。
 - データベースがまだオープン状態で、そのデータベースの破損したデータ・ファイルだけがオフラインになっている場合は、破損したデータ・ファイルが含まれている表領域をすべてオフラインにします。破損したデータ・ファイルは、エラー・メッセージにより識別できます。表領域をオフラインにするには、Enterprise Manager の「オフライン設定」メニュー項目を使うか、SQL コマンドの ALTER TABLESPACE に OFFLINE オプションを指定して実行します (11-8 ページの「オフライン表領域およびデータ・ファイルのバックアップ」を参照してください)。できれば、回復の量を最小限にするために) 一時優先度を使って、破損した表領域をオフラインにします。
2. メディア障害の原因になったハードウェア障害を修復します。ハードウェア障害をすぐに修復できない場合は、破損したファイルを代替記憶デバイスに復元することにより、データベース回復の次のステップに進むことができます。
3. ファイルの破損が修復できない場合は、メディア障害が原因で破損したデータ・ファイルの最新のバックアップ・ファイルだけを復元します。破損していないデータ・ファイルやオンライン REDO ログ・ファイル、制御ファイルは復元しないでください。ハードウェア障害が解消されていて、破損したデータ・ファイルを元の位置に復元できる場合は、それらのデータ・ファイルを元の位置に復元します。ハードウェア障害がまだ解消されていない場合は、破損したデータ・ファイルを、データベース・サーバーの代替記憶デバイスに復元します。

注意： 特定のデータ・ファイルのバックアップがない場合は、空の置換ファイルを作成して、それを回復に使用できます。

4. 1 つ以上の破損したデータ・ファイルを代替位置に復元した場合 (ステップ 3) は、これらのファイルの新しい位置に関連するデータベースの制御ファイルに指定してください。指定するときに、必要に応じて『Oracle8 Server 管理者ガイド』の「データ・ファイルを改名、再配置する」で説明されている手順を使用してください。

5. 1 つ以上のオフライン表領域内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回復を 1 つのステップで開始するには、管理者権限で接続した後で、Enterprise Manager の RECOVER TABLESPACE 文を使います。

注意： パフォーマンスを最大にするには、パラレル回復を使ってデータ・ファイルを回復します。

6. Oracle が、必要な REDO ログ・ファイル（アーカイブおよびオンライン）を適用してメディア回復のロールフォワード・フェーズを開始し、復元済みのデータ・ファイルを再構築します。ファイルの適用が自動化されている場合以外は、個々の必要な REDO ログ・ファイルについてプロンプトが表示されます。

必要なアーカイブ REDO ログ・ファイルをすべて復元済みデータ・ファイルに適用し終わるまで、処理が続行されます。次に、メディア回復を完了するために、オンライン REDO ログ・ファイルが復元済みデータ・ファイルに自動的に適用されます。

完全メディア回復にアーカイブ REDO ログ・ファイルが必要でない場合には、それに關するプロンプトは表示されません。かわりに、すべての必要なオンライン REDO ログ・ファイルが適用され、メディア回復は完了します。

7. これで、オープン状態のデータベースの破損した表領域は、メディア障害の発生時点の状態に回復されました。Enterprise Manager の「オンライン設定」メニュー項目を使うか、SQL コマンドの ALTER TABLESPACE に ONLINE オプションを指定して実行すれば、オフラインの表領域をオンラインにできます。

関連項目： REDO ログの適用に関する詳細は、12-4 ページの「REDO ログ・ファイルの適用」を参照してください。

データ・ファイルの作成方法については、『Oracle8 Server 管理者ガイド』を参照してください。

オープン状態のデータベースのオフライン表領域の個別回復の実行

上記の操作例と同じく、ここでも、オープンしているデータベースにメディア障害が発生し、このデータベースはオープンされたままで、破損していないデータ・ファイルはオンラインで使用可能な状態になっています。破損したデータ・ファイルは、Oracle が自動的にオフラインにします。

注意： この手順は、SYSTEM 表領域のデータ・ファイルの完全メディア回復には使用できません。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、Oracle はそのデータベースを自動的に停止します。

オープン状態のデータベースのオフライン表領域の個別回復を実行するには

1. この回復操作の開始点は、メディア障害の発生後にデータベースがオープン状態かどうかによって異なります。
 - データベースが停止されている場合は、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。この操作を実行するには、Enterprise Manager の「データベース起動」ダイアログ・ボックスを使って「起動してオープン」ラジオ・ボタンを選択するか、OPEN オプションを指定した STARTUP コマンドを実行します。データベースをオープンした後で、破損したデータ・ファイルがある表領域をすべてオフラインにしてください。
 - データベースがまだオープン状態で、そのデータベースの破損したデータ・ファイルだけがオフラインになっている場合は、破損したデータ・ファイルが含まれている表領域をすべてオフラインにします。破損したデータ・ファイルは、エラー・メッセージにより識別できます。表領域をオフラインにするには、Enterprise Manager の「オフライン設定」メニュー項目を使うか、SQL コマンドの ALTER TABLESPACE で OFFLINE オプションを指定します。できれば、(回復の量を最小限にするために) 一時優先度を使って、破損した表領域をオフラインにします。
2. メディア障害の原因になったハードウェア障害を修復します。ハードウェア障害をすぐに修復できない場合は、破損したファイルを代替記憶デバイスに復元することにより、データベース回復の次のステップに進むことができます。
3. ファイルの破損が修復できない場合は、メディア障害が原因で破損したデータ・ファイルの最新のバックアップ・ファイル（全体バックアップまたは部分バックアップの一部としてとったもの）だけを復元します。破損していないデータ・ファイルやオンライン REDO ログ・ファイル、制御ファイルは復元しないでください。ハードウェア障害が解消されていて、破損したデータ・ファイルを元の位置に復元できる場合は、それらのデータ・ファイルを元の位置に復元します。ハードウェア障害がまだ解消されていない場合は、破損したデータ・ファイルを、データベース・サーバーの代替記憶デバイスに復元します。

注意： 特定のデータ・ファイルのバックアップがない場合は、空の置換ファイルを作成して、それを回復に使用できます。

4. 破損した 1 つ以上のデータ・ファイルを代替位置に復元した場合 (ステップ 3) は、それらのファイルの新しい位置を、対応するデータベースの制御ファイルに指示してください。
5. オフライン表領域内の破損した個々のデータ・ファイルの回復を開始するには、管理者権限で接続した後で、Enterprise Manager の RECOVER DATAFILE 文を使います。

注意： パフォーマンスを最大にするには、パラレル回復を使ってデータ・ファイルを回復します。

6. Oracle が、必要な REDO ログ・ファイル（アーカイブおよびオンライン）を適用してメディア回復のロールフォワード・フェーズを開始し、復元済みのデータ・ファイルを再構築します。ファイルの適用が自動化されている場合を除き、必要な個々の REDO ログ・ファイルについてプロンプトが表示されます。

必要なアーカイブ REDO ログ・ファイルをすべて復元済みデータ・ファイルに適用し終わるまで、処理が続行されます。次に、メディア回復を完了するために、オンライン REDO ログ・ファイルが復元済みデータ・ファイルに自動的に適用されます。

完全メディア回復にアーカイブ REDO ログ・ファイルが必要でない場合には、それに關するプロンプトは表示されません。かわりに、すべての必要なオンライン REDO ログ・ファイルが適用され、メディア回復は完了します。

7. これで、オープン状態のデータベースの破損した表領域は、メディア障害の発生時点の状態に回復されました。Enterprise Manager の「オンライン設定」メニュー項目を使うか、SQL コマンドの ALTER TABLESPACE に ONLINE オプションを指定して実行すれば、オフラインの表領域をオンラインにできます。

関連項目：完全メディア回復の実行方法は、12-15 ページの「クローズ状態のデータベースの回復」を参照してください。

データ・ファイルの作成方法については、『Oracle8 Server 管理者ガイド』を参照してください。

不完全メディア回復の実行

この項では、各種の不完全メディア回復操作を実行するために必要な作業について説明します。次の項目を取り挙げます。

- 取消しベースの回復の実行
- 時間ベースの回復の実行
- 変更ベースの回復の実行

関連項目：メディア障害からの回復に必要な手順を理解するには、この項の説明だけでは十分ではありません。

稼働中のデータベースのシステム時間の変更

季節的な時間変更（たとえば、夏時間）の影響を受けるデータベースの場合は、同じ時刻が REDO ログに 2 回設定されていて、2 番目の（つまり後に設定された）時刻の状態への回復が必要になったときに、問題の発生を招くことがあります。時間の変更に対処するには、時計を戻した時点の状態への取消しベース回復または変更ベース回復を実行し、その後で該当する時刻の状態への時間ベース回復を実行します。

取消しベースの回復の実行

この項では、取消しベースの回復の実行方法について説明します。

取消しベースの回復を実行するには

1. データベースがまだオープン状態で不完全メディア回復が必要な場合は、Enterprise Manager の「データベース停止」ダイアログ・ボックスの「異常終了停止」モードを使うか、ABORT オプションを指定した SHUTDOWN コマンドを実行して、データベースを停止します。
2. 回復作業の途中でエラーが発生した場合の対策として、データベース全体（データベースのすべてのデータ・ファイルおよび制御ファイル、パラメータ・ファイル）のバックアップをとります。
3. メディア障害が発生している場合は、そのメディア障害の原因になったハードウェア障害を修復します。
4. 回復の目標にしている時刻におけるデータベースの物理構造に現行の制御ファイルが一致しない場合（たとえば、回復の目標にしている時点の後で追加されたデータ・ファイルがある場合など）は、不完全メディア回復の終了予定時点におけるデータベースの物理ファイル構造を反映する制御ファイル（データ・ファイルおよびオンライン REDO ログ・ファイルの名前が設定されている）のバックアップを復元します。現行の制御ファイルに対応するファイルのリストと、各制御ファイル・バックアップを検討して、使用する正しい制御ファイルを選んでください。必要に応じて、データベースの現行の制御ファイルをすべて、正しい制御ファイル・バックアップで置換してください。また、新しい制御ファイルを作成して、欠落している制御ファイルと置換してもかまいません。

注意： データベース制御ファイルが正常に機能しないか、または制御ファイルのバックアップと置換できない場合は、データベースに対応付けられているパラメータ・ファイルを編集して、CONTROL_FILES パラメータを変更する必要があります。

5. 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・ファイル（全体バックアップまたは部分バックアップの一部としてとったもの）を復元します。既存のデータ・ファイルと置換するために使用するすべてのバックアップ・ファイルは、回復の目標時点より前にとったものでなければなりません。たとえば、REDO ログ順序番号 38 まで回復する場合は、REDO ログ順序番号 38 より前に完了しているバックアップのすべてのデータ・ファイルを復元します。

特定のデータ・ファイルのバックアップがない場合は、空の置換ファイルを作成して、それを回復に使用できます。

回復の目標時刻より後に追加されたデータ・ファイルがある場合、回復が完了した後、このファイルはデータベース用には使用されないため、このファイルのバックアップを復元する必要はありません。

メディア障害の原因になったハードウェア障害が解消していて、すべてのデータ・ファイルを元の位置に復元できる場合は、元の位置に復元して、この手順のステップ 8 を省

略してください。ハードウェア障害がまだ存在している場合は、破損したデータ・ファイルを代替記憶デバイスに復元します。

注意： 制御ファイルのバックアップを使っている場合は、読取り専用表領域内のファイルをオフラインにしてください。オフラインにしておかないと、回復により読取り専用ファイルのヘッダーが更新されることがあります。

6. Enterprise Manager を起動し、管理者権限により Oracle に接続します。
7. 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。この操作を実行するには、Enterprise Manager の「データベース起動」ダイアログ・ボックスを使って「起動してマウント」ラジオ・ボタンを選択するか、MOUNT オプションを指定した STARTUP コマンドを実行します。
8. ステップ 5 で 1 つ以上の破損データ・ファイルを別の位置に復元した場合は、これらのファイルの新しい位置を、関連のデータベースの制御ファイルに指定する必要があります。
9. この不完全回復で制御ファイルのバックアップを使っている場合（つまり、ステップ 4 で制御ファイルのバックアップまたは再作成した制御ファイルを復元した場合）は、回復を開始するために使うダイアログ・ボックスまたはコマンドの中で、そのことを示します（つまり、USING BACKUP CONTROLFILE パラメータを指定します）。
10. Enterprise Manager の「回復アーカイブ適用」ダイアログ・ボックスか、それに相当する RECOVER DATABASE UNTIL CANCEL 文を使って、取消しベースの回復を開始します。
11. Oracle が、必要な REDO ログ・ファイル（アーカイブおよびオンライン）を適用してメディア回復のロールフォワード・フェーズを開始し、復元済みのデータ・ファイルを再構築します。ファイルの適用が自動化されてる場合以外は、Oracle は、LOG_ARCHIVE_DEST から候補となるファイル名を表示し、そのログ・ファイルの適用を実行するか中止するかの指示を求めます。制御ファイルがバックアップ・ファイルの場合は、オンライン・ログの名前を指定する必要があります。
Oracle は、REDO ログ・ファイルの適用を続けます。
12. 最新の、破損していない REDO ログ・ファイルを復元済みのデータ・ファイルに適用し終わるまで、REDO ログ・ファイルの適用を続けます。
13. 破損したファイルの直前の REDO ログ・ファイルが適用された後で、"CANCEL" を入力して、回復を取り消します。取消しベースの回復はこれで完了です。
回復が成功したかどうかを示すメッセージが表示されます。

取消しベースの回復成功後のデータベースのオープン

不完全メディア回復の後で最初にデータベースをオープンするときは、**RESETLOGS** または **NORESETLOGS** のどちらかのオプションを指定することによって、ログ順序番号をリセットするかどうかを明示的に指定する必要があります。**REDO** ログをリセットするには、次のようにします。

- 回復時に適用されなかった **REDO** 情報をすべて破棄して、今後それが適用されないようにする。
- オンライン **REDO** ログおよび **REDO** スレッドについての制御ファイル情報を初期化しなおす。
- オンライン **REDO** ログの内容を消去する。
- オンライン **REDO** ログ・ファイルが現在ない場合は、オンライン **REDO** ログ・ファイルを作成する。
- ログ順序番号を 1 にリセットする。

警告： **REDO** ログをリセットすると、**REDO** 情報を最初に破棄した後でデータベースに対して行った変更内容はすべて破棄されます。その時点以降に入力された更新は、手動で再入力する必要があります。

RESETLOGS または **NORESETLOGS** のどちらにするかを決めるには、次のルールに従います。

- 実行した回復のタイプ（完全か不完全か）に関係なく、制御ファイルのバックアップを使って回復した場合は、ログ順序番号をリセットする。
- 回復が実際に不完全な場合は、ログ順序番号をリセットする。たとえば、先の時刻や **SCN**（システム変更番号）でなく、前の時刻や **SCN** を指定した場合などです。
- 回復が完了した場合は（バックアップ制御ファイルを使っていない限り）、ログをリセットしない。これに当てはまるのは、意図的に完全回復を実行した場合、および不完全回復を実行したが実際にはなんらかの方法で **REDO** ログ内のすべての変更内容を回復した場合です。**ALERT** ファイルを見て不完全回復が実際に完了したかどうかを調べる方法については、ステップ 12 の説明を参照してください。
- このデータベースのアーカイブ・ログをスタンバイ・データベース用に使う場合は、ログをリセットしない。ログをリセットする必要がある場合は、スタンバイ・データベースを再作成しなければなりません。

回復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ないためには、**SQL** コマンドの **ALTER DATABASE** に **OPEN NORESETLOGS** オプションを指定して実行します。回復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するには、**SQL** コマンドの **ALTER DATABASE** に **OPEN RESETLOGS** オプションを指定して実行します。（リセットしてはならないのにログをリセットしようとしたり、リセッ

としなければならないのにリセットしなかった場合は、エラーとなり、データベースはオープンしません。エラーを訂正して再試行してください。)

データベースのオープン時にログ順序番号がリセットされる場合は、回復が完全か不完全かによって、異なるメッセージが出力されます。完全回復の場合は、**ALERT** ファイルに次のメッセージが出力されます。

変更 *scn* を過ぎた完全回復後の **RESETLOGS**

不完全回復の場合は、**ALERT** ファイルに次のメッセージが出力されます。

変更 *scn* までの不完全回復後の **RESETLOGS**

データベースのオープン時に **REDO** ログ順序をリセットする場合は、ただちにデータベースを通常の方法で停止し、データベース全体のバックアップをとってください。そうしておかないと、ログをリセットした後で変更した内容を回復できなくなります。データベース全体のバックアップをとるまでは、上記の手順（ログをリセットするまでの手順）を繰り返す以外に回復する方法はありません。（ログ順序をリセットしなかった場合は、データベースのバックアップをとる必要はありません。）

RESETLOGS オプションを使ってデータベースをオープンした後で、**ALERT** ログをチェックして、**Oracle** がデータ・ディクショナリと制御ファイルとの間の矛盾（たとえば、データ・ディクショナリに格納されているデータ・ファイルが新しい制御ファイルに設定されていないなど）を検出したかどうかを調べます。

データ・ファイルがデータ・ディクショナリ内にはあるが新しい制御ファイル内にはない場合は、制御ファイル内に **MISSINGnnnnn** (nnnnn は 10 進数のファイル番号) と示されたプレースホルダ・エントリが作成されます。制御ファイル内で、**MISSINGnnnnn** には、オフラインでありメディア回復が必要なことを示すフラグが設定されます。**MISSINGnnnnn** に対応する実際のデータ・ファイルをアクセス可能にするには、**MISSINGnnnnn** を改名し、読取り専用または通常オフラインであったデータ・ファイルだけ指すようにします。一方、**MISSINGnnnnn** が読取り専用でも通常オフラインでもないデータ・ファイルに対応している場合は、改名してもデータ・ファイルをアクセス可能にすることはできません。なぜなら、データ・ファイルに必要なメディア回復が **RESETLOGS** の結果により不可能だからです。この場合には、そのデータ・ファイルが格納されている表領域を削除する必要があります。

これに対して、制御ファイルに示されているデータ・ファイルがデータ・ディクショナリ内にはない場合は、そのデータ・ファイルを参照している部分は新しい制御ファイルからは自動的に削除されます。どちらの場合も、検出された内容がわかるように、状況を説明するメッセージが **ALERT** ファイルに出力されます。

関連項目 : データ・ファイルの作成方法については、12-13 ページの「破損したデータ・ファイルの復元」を参照してください。

データ・ファイルの再配置または改名方法は、『**Oracle8 Server 管理者ガイド**』を参照してください。

REDO ログの適用に関する詳細は、12-4 ページの「**REDO** ログ・ファイルの適用」を参照してください。

時間ベースの回復の実行

時間ベースの不完全メディア回復を実行するときに、回復に使用するバックアップ制御ファイルに読取り専用表領域がある場合は、次の回復手順を実行する前に Oracle の技術サポートに連絡してください。

時間ベースの回復を実行するには

1. データベースがまだオープン状態で不完全メディア回復が必要な場合は、**Enterprise Manager** の「データベース停止」ダイアログ・ボックスの「異常終了停止」モードを使うか、**ABORT** オプションを指定した **SHUTDOWN** コマンドを実行して、データベースを停止します。
2. 回復作業の途中でエラーが発生した場合の対策として、データベース全体（データベースのすべてのデータ・ファイルおよび制御ファイル、パラメータ・ファイル）のバックアップをとります。
3. メディア障害が発生している場合は、そのメディア障害の原因になったハードウェア障害を修復します。
4. 回復の目標にしている時刻におけるデータベースの物理構造に現行の制御ファイルが一致しない場合（たとえば、回復の目標にしている時点の後で追加されたデータ・ファイルがある場合など）は、不完全メディア回復の終了予定時点におけるデータベースの物理ファイル構造を反映する制御ファイル（データ・ファイルおよびオンライン **REDO** ログ・ファイルの名前が設定されている）のバックアップを復元します。現行の制御ファイルに対応するファイルのリストと、各制御ファイル・バックアップを検討して、使用する正しい制御ファイルを選んでください。必要に応じて、データベースの現行の制御ファイルをすべて、正しい制御ファイル・バックアップで置換してください。また、新しい制御ファイルを作成して、欠落している制御ファイルと置換しても構いません。

注意： メディア障害の原因になっているハードウェア障害が解決されていないために、データベース制御ファイルが正常に機能しないか制御ファイル・バックアップと置換できない場合は、データベースに対応付けられているパラメータ・ファイルを編集して、**CONTROL_FILES** パラメータを変更する必要があります。

5. 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・ファイルを復元します。既存のデータ・ファイルと置換するために使用するすべてのバックアップ・ファイルは、回復の目標時点より前にとったものでなければなりません。たとえば、**REDO** ログ順序番号 38 まで回復する場合は、**REDO** ログ順序番号 38 より前に完了しているバックアップのすべてのデータ・ファイルを復元します。

特定のデータ・ファイルのバックアップがない場合は、空の置換ファイルを作成して、それを回復に使用できます。

回復の目標時刻より後に追加されたデータ・ファイルがある場合、回復が完了した後、このファイルはデータベース用には使用されないため、このファイルのバックアップを復元する必要はありません。

メディア障害の原因になったハードウェア障害が解消していて、すべてのデータ・ファイルを元の位置に復元できる場合は、元の位置に復元して、この手順のステップ 8 を省略してください。ハードウェア障害がまだ存在している場合は、破損したデータ・ファイルを代替記憶デバイスに復元します。

注意： 制御ファイルのバックアップを使っている場合は、読取り専用表領域内のファイルをオフラインにしてください。オフラインにしておかないと、回復により読取り専用ファイルのヘッダーが更新されることがあります。

6. Enterprise Manager を起動し、管理者権限により Oracle に接続します。
7. 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。この操作を実行するには、Enterprise Manager の「データベース起動」ダイアログ・ボックスで「起動してマウント」ラジオ・ボタンを選択するか、MOUNT オプションを指定した STARTUP コマンドを実行します。
8. ステップ 5 で 1 つ以上の破損データ・ファイルを別の位置に復元した場合は、これらのファイルの新しい位置を、関連のデータベースの制御ファイルに指定する必要があります。
9. オフライン表領域が通常の方法でオフラインにされた場合以外は、データベースのすべてのデータ・ファイルをオンラインにする必要があります。回復するすべてのデータ・ファイルの名前を調べるには、使用中の制御ファイルに通常は付随しているデータ・ファイルのリストをチェックするか、V\$DATAFILE ビューに問い合わせます。その後で、DATAFILE ONLINE オプションを指定した ALTER DATABASE コマンドを実行し、データベースのすべてのデータ・ファイルがオンラインになっていることを確認してください。たとえば、USERS1 という名前（完全修飾ファイル名）のデータ・ファイルを確実にオンラインにするには、次の文を入力します。

```
ALTER DATABASE DATAFILE 'users1' ONLINE;
```

この不完全回復で制御ファイルのバックアップを使っている場合（つまり、制御ファイルのバックアップまたは再作成した制御ファイルを復元した場合）は、そのことを、回復を開始するために使うダイアログ・ボックスまたはコマンドの中で指示します。指定のデータ・ファイルがすでにオンラインになっていれば、この文は無視されます。

10. RECOVER DATABASE UNTIL TIME 文を発行して、時間ベースの回復を開始します。時刻は、必ず次の書式を使って、引用符で区切って指定します。'YYYY-MM-DD:HH24:MI:SS'
11. Oracle が、必要な REDO ログ・ファイル（アーカイブおよびオンライン）を適用してメディア回復のロールフォワード・フェーズを開始し、復元済みのデータ・ファイルを再

構築します。ファイルの適用が自動化されてる場合を除き、Oracle は、**LOG_ARCHIVE_DEST** から候補となるファイル名を表示し、そのログ・ファイルの適用を実行するか中止するかを指示を求めます。制御ファイルがバックアップ・ファイルの場合は、オンライン・ログの名前を指定する必要があります。Oracle は、REDO ログ・ファイルの適用を続けます。

12. 必要な最後の REDO ログ・ファイルを復元済みのデータ・ファイルに適用し終わるまで、REDO ログ・ファイルの適用を続けてください。設定した時刻になると、Oracle は回復を自動的に終了し、回復が成功したかどうかを示すメッセージを戻します。

時間ベースの回復成功後のデータベースのオープン

不完全メディア回復の後で最初にデータベースをオープンするときは、**RESETLOGS** または **NORESETLOGS** のどちらかのオプションを指定することによって、ログ順序番号をリセットするかどうかを明示的に指定する必要があります。REDO ログをリセットするには、次のようにします。

- 回復時に適用されなかった REDO 情報をすべて破棄して、今後それが適用されないようにします。
- オンライン REDO ログおよび REDO スレッドについての制御ファイル情報を初期化し直します。
- オンライン REDO ログの内容を消去します。
- オンライン REDO ログ・ファイルが現在ない場合は、オンライン REDO ログ・ファイルを作成します。
- ログ順序番号を 1 にリセットします。

警告： REDO ログをリセットすると、REDO 情報を最初に破棄した後でデータベースに対して行った変更内容はすべて破棄されます。その時点以降に入力された更新は、手動で再入力する必要があります。

RESETLOGS または **NORESETLOGS** のどちらにするかを決めるには、次のルールに従います。

- 実行した回復のタイプ（完全か不完全か）に関係なく、制御ファイルのバックアップを使って回復した場合は、ログ順序番号をリセットする。
- 回復が実際に不完全な場合は、ログ順序番号をリセットする。たとえば、先の時刻や SCN（システム変更番号）でなく、前の時刻や SCN を指定した場合などです。
- 回復が完了した場合は（バックアップ制御ファイルを使っていない限り）、ログをリセットしない。これに当てはまるのは、意図的に完全回復を実行した場合、および不完全回復を実行したが実際にはなんらかの方法で REDO ログ内のすべての変更内容を回復した場合です。ALERT ファイルを見て不完全回復が実際に完了したかどうかを調べる方法については、ステップ 12 の説明を参照してください。

- このデータベースのアーカイブ・ログをスタンバイ・データベース用に使う場合は、ログをリセットしない。ログをリセットする必要がある場合は、スタンバイ・データベースを再作成しなければなりません。

回復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ないためには、SQL コマンドの ALTER DATABASE に OPEN NORESETLOGS オプションを指定して実行します。回復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するには、SQL コマンドの ALTER DATABASE に OPEN RESETLOGS オプションを指定して実行します。(リセットしてはならないのにログをリセットしようとしたり、リセットしなければならないのにリセットしなかった場合は、エラーとなり、データベースはオープンしません。エラーを訂正して再試行してください。)

データベースのオープン時にログ順序番号がリセットされる場合は、回復が完全か不完全かによって、異なるメッセージが出力されます。完全回復の場合は、ALERT ファイルに次のメッセージが出力されます。

変更 scn を過ぎた完全回復後の RESETLOGS

不完全回復の場合は、ALERT ファイルに次のメッセージが出力されます。

変更 scn までの不完全回復後の RESETLOGS

データベースのオープン時に REDO ログ順序をリセットする場合は、ただちにデータベースを通常の方法で停止し、データベース全体のバックアップをとってください。そうしておかないと、ログをリセットした後で変更した内容を回復できなくなります。データベース全体のバックアップをとるまでは、上記の手順（ログをリセットするまでの手順）を繰り返す以外に回復する方法はありません。(ログ順序をリセットしなかった場合は、データベースのバックアップをとる必要はありません。)

RESETLOGS オプションを使ってデータベースをオープンした後で、ALERT ログをチェックして、Oracle がデータ・ディクショナリと制御ファイルとの間の矛盾（たとえば、データ・ディクショナリに格納されているデータ・ファイルが新しい制御ファイルに設定されていないなど）を検出したかどうかを調べます。

データ・ファイルがデータ・ディクショナリ内にはあるが新しい制御ファイル内にはない場合は、制御ファイル内に MISSINGnnnn(nnnn は 10 進数のファイル番号) と示されたブレースホルダ・エントリが作成されます。制御ファイル内で、MISSINGnnnn には、オフラインでありメディア回復が必要なことを示すフラグが設定されます。MISSINGnnnn に対応する実際のデータ・ファイルをアクセス可能にするには、MISSINGnnnn を改名し、読取り専用または通常オフラインであったデータ・ファイルだけ指すようにします。一方、MISSINGnnnn が読取り専用でも通常オフラインでもないデータ・ファイルに対応している場合は、改名してもデータ・ファイルをアクセス可能にすることはできません。なぜなら、データ・ファイルに必要なメディア回復が RESETLOGS の結果により不可能だからです。この場合には、そのデータ・ファイルが格納されている表領域を削除する必要があります。

これに対して、制御ファイルに示されているデータ・ファイルがデータ・ディクショナリ内にはない場合は、そのデータ・ファイルを参照している部分は新しい制御ファイルからは自動

的に削除されます。どちらの場合も、検出された内容がわかるように、状況を説明するメッセージが **ALERT** ファイルに出力されます。

関連項目：詳細は、『Oracle8 Server 管理者ガイド』を参照してください。

REDO ログの適用に関する詳細は、12-4 ページの「REDO ログ・ファイルの適用」を参照してください。

変更ベースの回復の実行

この項では、変更ベースの回復の実行方法について説明します。

変更ベースの回復を実行するには

1. データベースがまだオープン状態で不完全メディア回復が必要な場合は、**Enterprise Manager** の「データベース停止」ダイアログ・ボックスの「異常終了停止」モードを使うか、**ABORT** オプションを指定した **SHUTDOWN** コマンドを実行して、データベースを停止します。
2. 回復作業の途中でエラーが発生した場合の対策として、データベース全体（データベースのすべてのデータ・ファイルおよび制御ファイル、パラメータ・ファイル）のバックアップをとります。
3. メディア障害が発生している場合は、そのメディア障害の原因になったハードウェア障害を修復します。
4. 回復の目標にしている時刻におけるデータベースの物理構造に現行の制御ファイルが一致しない場合（たとえば、回復の目標にしている時点の後に追加されたデータ・ファイルがある場合など）は、不完全メディア回復の終了予定時点におけるデータベースの物理ファイル構造を反映する制御ファイル（データ・ファイルおよびオンライン **REDO** ログ・ファイルの名前が設定されている）のバックアップを復元します。現行の制御ファイルに対応するファイルのリストと、各制御ファイル・バックアップを検討して、使用する正しい制御ファイルを選んでください。必要に応じて、データベースの現行の制御ファイルをすべて、正しい制御ファイル・バックアップで置換してください。また、新しい制御ファイルを作成して、欠落している制御ファイルと置換しても構いません。

注意： データベースの制御ファイルが正常に機能しない場合や、制御ファイルのバックアップと置換できない場合は、データベースに対応付けられているパラメータ・ファイルを編集して、**CONTROL_FILES** パラメータを変更する必要があります。

5. 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・ファイルを復元します。既存のデータ・ファイルと置換するために使用するすべてのバックアップ・ファイルは、回復の目標時点より前にとったものでなければなりません。たとえば、**REDO** ログ順序番号 38 まで回復する場合は、**REDO** ログ順序番号 38 より前に完了しているバックアップのすべてのデータ・ファイルを復元します。

特定のデータ・ファイルのバックアップがない場合は、空の置換ファイルを作成して、それを回復に使用できます。

回復の目標時刻より後に追加されたデータ・ファイルがある場合、回復が完了した後、このファイルはデータベース用には使用されないため、このファイルのバックアップを復元する必要はありません。

メディア障害の原因になったハードウェア障害が解消していて、すべてのデータ・ファイルを元の位置に復元できる場合は、元の位置に復元して、この手順のステップ 8 を省略してください。ハードウェア障害がまだ存在している場合は、破損したデータ・ファイルを代替記憶デバイスに復元します。

注意： 制御ファイルのバックアップを使っている場合は、読取り専用表領域内のファイルをオフラインにしてください。オフラインにしておかないと、回復により読取り専用ファイルのヘッダーが更新されることがあります。

6. Enterprise Manager を起動し、管理者権限により Oracle に接続します。
7. 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。この操作を実行するには、Enterprise Manager の「データベース起動」ダイアログ・ボックスを使って「起動してマウント」ラジオ・ボタンを選択するか、MOUNT オプションを指定した STARTUP コマンドを実行します。
8. ステップ 5 で 1 つ以上の破損データ・ファイルを別の位置に復元した場合は、これらのファイルの新しい位置を、関連のデータベースの制御ファイルに指定する必要があります。
9. 回復するすべてのデータ・ファイルの名前を調べるには、使用中の制御ファイルに通常は付随しているデータ・ファイルのリストをチェックするか、V\$DATAFILE ビューに問い合わせます。その後で、DATAFILE ONLINE オプションを指定した ALTER DATABASE コマンドを実行し、データベースのすべてのデータ・ファイルがオンラインになっていることを確認してください。たとえば、USERS1 という名前（完全修飾ファイル名）のデータ・ファイルを確実にオンラインにするには、次の文を入力します。

```
ALTER DATABASE DATAFILE 'users1' ONLINE;
```

指定のデータ・ファイルがすでにオンラインになっていれば、この文は無視されます。

この不完全回復で制御ファイルのバックアップを使っている場合（つまり、制御ファイルのバックアップまたは再作成した制御ファイルを復元した場合）は、回復を開始するために使うダイアログ・ボックスまたはコマンドの中で、USING BACKUP CONTROLFILE パラメータを指定します。

10. RECOVER DATABASE UNTIL CHANGE 文を発行して、変更ベースの回復を開始します。SCN は、引用符なしの 10 進数を指定します。
11. Oracle が、必要な REDO ログ・ファイル（アーカイブおよびオンライン）を適用してメディア回復のロールフォワード・フェーズを開始し、復元済みのデータ・ファイルを再構築します。ファイルの適用が自動化されている場合を除き、Oracle は、LOG_ARCHIVE_DEST から候補となるファイル名を表示し、そのログ・ファイルの適用を実行するか中止するかを指示を求めます。制御ファイルがバックアップ・ファイルの

場合は、オンライン・ログの名前を指定する必要があります。Oracle は、REDO ログ・ファイルの適用を続けます。

12. 必要な最後の REDO ログ・ファイルを復元済みのデータ・ファイルに適用し終わるまで、REDO ログ・ファイルの適用を続けます。設定した時刻になると、Oracle は回復を自動的に終了し、回復が成功したかどうかを示すメッセージを戻します。

変更ベースの回復成功後のデータベースのオープン

不完全メディア回復の後で最初にデータベースをオープンするときは、RESETLOGS または NORESETLOGS のどちらかのオプションを指定することによって、ログ順序番号をリセットするかどうかを明示的に指定する必要があります。REDO ログをリセットするには、次のようにします。

- 回復時に適用されなかった REDO 情報をすべて破棄して、今後それが適用されないようにします。
- オンライン REDO ログおよび REDO スレッドについての制御ファイル情報を初期化し直します。
- オンライン REDO ログの内容を消去します。
- オンライン REDO ログ・ファイルが現在ない場合は、オンライン REDO ログ・ファイルを作成します。
- ログ順序番号を 1 にリセットします。

警告： REDO ログをリセットすると、REDO 情報を最初に破棄した後でデータベースに対して行った変更内容はすべて破棄されます。その時点以降に入力された更新は、手動で再入力する必要があります。

RESETLOGS または NORESETLOGS のどちらにするかを決めるには、次のルールに従います。

- 実行した回復のタイプ（完全か不完全か）に関係なく、制御ファイルのバックアップを使って回復した場合は、ログ順序番号をリセットする。
- 回復が実際に不完全な場合は、ログ順序番号をリセットする。たとえば、先の時刻や SCN（システム変更番号）でなく、前の時刻や SCN を指定した場合などです。
- 回復が完了した場合は（バックアップ制御ファイルを使っていない限り）、ログをリセットしない。これに当てはまるのは、意図的に完全回復を実行した場合、および不完全回復を実行したが実際にはなんらかの方法で REDO ログ内のすべての変更内容を回復した場合です。ALERT ファイルを見て不完全回復が実際に完了したかどうかを調べる方法については、ステップ 12 の説明を参照してください。
- このデータベースのアーカイブ・ログをスタンバイ・データベース用に使う場合は、ログをリセットしない。ログをリセットする必要がある場合は、スタンバイ・データベースを再作成しなければなりません。

回復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ないためには、SQL コマンドの **ALTER DATABASE** に **OPEN NORESETLOGS** オプションを指定して実行します。回復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するには、SQL コマンドの **ALTER DATABASE** に **OPEN RESETLOGS** オプションを指定して実行します。（リセットしてはならないのにログをリセットしようとしたり、リセットしなければならないのにリセットしなかった場合は、エラーとなり、データベースはオープンしません。エラーを訂正して再試行してください。）

データベースのオープン時にログ順序番号がリセットされる場合は、回復が完全か不完全かによって、異なるメッセージが出力されます。完全回復の場合は、**ALERT** ファイルに次のメッセージが出力されます。

変更 *scn* を過ぎた完全回復後の **RESETLOGS**

不完全回復の場合は、**ALERT** ファイルに次のメッセージが出力されます。

変更 *scn* までの不完全回復後の **RESETLOGS**

データベースのオープン時に **REDO** ログ順序をリセットする場合は、ただちにデータベースを通常の方法で停止し、データベース全体のバックアップをとってください。そうしておかないと、ログをリセットした後で変更した内容を回復できなくなります。全体バックアップをとるまでは、上記の手順（ログをリセットするまでの手順）を繰り返す以外に回復する方法はありません。（ログ順序をリセットしなかった場合は、データベースのバックアップをとる必要はありません。）

RESETLOGS オプションを使ってデータベースをオープンした後で、**ALERT** ログをチェックして、**Oracle** がデータ・ディクショナリと制御ファイルとの間の矛盾（たとえば、データ・ディクショナリに格納されているデータ・ファイルが新しい制御ファイルに設定されていないなど）を検出したかどうかを調べます。

データ・ファイルがデータ・ディクショナリ内にはあるが新しい制御ファイル内にはない場合は、制御ファイル内に **MISSINGnnnnn** (*nnnnn* は 10 進数のファイル番号) と示されたブレースホルダ・エントリが作成されます。制御ファイル内で、**MISSINGnnnnn** には、オフラインでありメディア回復が必要なことを示すフラグが設定されます。**MISSINGnnnnn** に対応する実際のデータ・ファイルをアクセス可能にするには、**MISSINGnnnnn** を改名し、読取り専用または通常オフラインであったデータ・ファイルだけ指すようにします。一方、**MISSINGnnnnn** が読取り専用でも通常オフラインでもないデータ・ファイルに対応している場合は、改名してもデータ・ファイルをアクセス可能にすることはできません。なぜなら、データ・ファイルに必要なメディア回復が **RESETLOGS** の結果により不可能だからです。この場合には、そのデータ・ファイルが格納されている表領域を削除する必要があります。

これに対して、制御ファイルに示されているデータ・ファイルがデータ・ディクショナリ内にはない場合は、そのデータ・ファイルを参照している部分は新しい制御ファイルからは自動的に削除されます。どちらの場合も、検出された内容がわかるように、状況を説明するメッセージが **ALERT** ファイルに出力されます。

関連項目 : **REDO** ログの適用に関する詳細は、12-4 ページの「**REDO** ログ・ファイルの適用」を参照してください。

災害時回復の準備

この項では、プライマリ・データベースに関する災害時回復手順の計画と実行の方法について説明します。取り挙げる項目は次のとおりです。

- スタンバイ・データベースの計画と作成
- プライマリ・データベースの物理構造の変更

スタンバイ・データベースの計画と作成

スタンバイ・データベースは、プライマリ・データベース（稼働データベースともいいます）の複製です。ここには予備のコピーが保持されているので、災害時（稼働システムのすべてのメディアが損傷を受けた場合）にもプライマリ・データベースを継続して利用できます。スタンバイ・データベースは常に回復モードになっています。災害発生時には、スタンバイ・データベースを回復モードから外し、アクティブにしてオンラインで使用できます。スタンバイ・データベースは、プライマリ・データベースの回復のためだけに使います。災害時回復をアクティブにする以外の目的で、問い合わせたりオープンしたりすることはできません。スタンバイ・データベースをアクティブにした後は、別のスタンバイ・データベースとして再作成しない限り、そのデータベースをスタンバイ回復モードに戻すことはできません。

警告： スタンバイ・データベースをアクティブにすると、スタンバイ・データベースのオンライン・ログがリセットされます。したがって、アクティブにした後は、スタンバイ・データベースのログと稼働データベースのログには互換性がありません。

プライマリ・データベースのデータ・ファイルおよびログ・ファイル、制御ファイルと、スタンバイ・データベースのこれらのファイルは、別個の物理メディアに配置する必要があります。したがって、プライマリ・データベースとスタンバイ・データベースの両方に同じ制御ファイルを使うことはできません。

スタンバイ・データベースの作成

この項では、スタンバイ・データベースを作成する場合の手順とルールを示します。

スタンバイ・データベースを作成するには

1. プライマリ・データベースから、データ・ファイルのバックアップを（オンラインまたはオフラインで）とります。
2. `ALTER DATABASE CREATE STANDBY CONTROLFILE AS 'filename'` コマンドを発行して、スタンバイ・データベース用の制御ファイルを作成します。このコマンドにより、プライマリ・データベースの制御ファイルの変更済みコピーが作成されます。

3. **ALTER SYSTEM ARCHIVE LOG CURRENT** コマンドを発行して、プライマリ・データベースの現行のオンライン・ログをアーカイブします。このコマンドを発行することで、ステップ 1 のデータ・ファイルおよびステップ 2 の制御ファイル、ログ・ファイルの間の整合性も確保できます。
4. スタンバイ・データベースの制御ファイルおよびアーカイブ・ログ・ファイル、バックアップ・データ・ファイルを、オペレーティング・システムのコマンドまたはユーティリティを使ってリモート（スタンバイ）サイトに転送します。バイナリ・ファイルを転送する場合は、それに適した方法を使ってください。

警告： プライマリ・データベースとスタンバイ・データベースのデータ・ファイルの名前が同じになるようなデータ・ファイル命名計画を使うことをお勧めします。それができない場合は、データ・ファイル名変換パラメータを使用できます。これらのデータ・ファイル命名計画のどちらかを使用しないと、スタンバイ・データベースがクラッシュするおそれがあります。

関連項目： スタンバイ・データベース作成時の名前変換パラメータの設定方法については、12-37 ページの「データ・ファイル名とログ・ファイル名の変換」を参照してください。

スタンバイ・データベースのメンテナンス

この項では、スタンバイ・ログ・ファイルの消去方法も含めて、スタンバイ・データベースのメンテナンス方法について説明します。

スタンバイ・データベースを回復モードでメンテナンスするには

1. NO MOUNT 句を使って、スタンバイ・データベースで Oracle インスタンスを起動します。
2. ALTER DATABASE MOUNT STANDBY DATABASE [EXCLUSIVE / PARALLEL] コマンドを発行します。
3. アーカイブ REDO ログを、プライマリ・データベースからリモート（スタンバイ）サイトに転送します。バイナリ・データの転送には、適正なオペレーティング・システム・ユーティリティを使ってください。
4. RECOVER [FROM 'location'] STANDBY DATABASE コマンドを発行して、スタンバイ・データベースを回復モードにします。

注意： アーカイブ・ログが生成されるたびに、継続的にそれをスタンバイ・データベースに転送し適用する必要があります。また、スタンバイ・データベースに適用できるのは、プライマリ・データベースでアーカイブされたログだけです。

オンライン・ログ・ファイルの消去 スタンバイ・データベースのオンライン・ログ・ファイルを消去することにより、スタンバイ・データベースのメンテナンス時のパフォーマンスを最適化できます。メンテナンス時にこの操作をしなくても、アクティブ化されるときにオンライン・ログ・ファイルは自動的に消去されます。ログ・ファイルを消去するには、次の文を使います。

```
ALTER DATABASE CLEAR LOGFILE GROUP integer;
```

データ・ファイル名とログ・ファイル名の変換

プライマリ・データベースの制御ファイルに設定されているすべてのファイル名をスタンバイ・データベースでできるように変換するには、次の初期化パラメータを設定します。

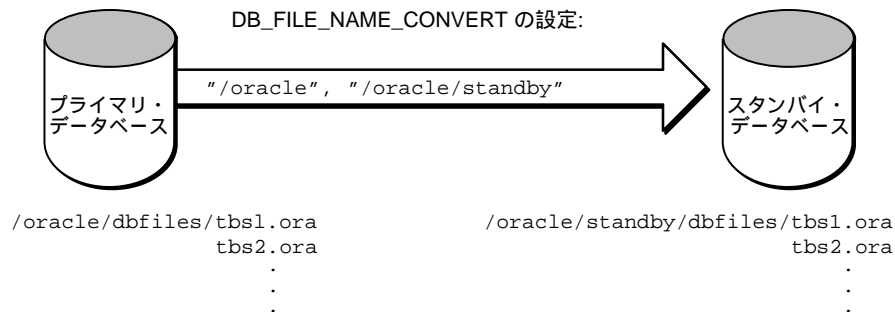
- DB_FILE_NAME_CONVERT
- LOG_FILE_NAME_CONVERT

プライマリ・データベースとスタンバイ・データベースが同じマシン上にある場合（本来は同じマシン上にあるべきではありません）は、上記のパラメータを設定しておくことをお勧めします。そうすることによって、スタンバイ・データベースのファイル名とプライマリ・データベースのファイル名が区別できるからです。

DB_FILE_NAME_CONVERT パラメータと LOG_FILE_NAME_CONVERT パラメータには 2 つの文字列が必要です。最初の文字列は、プライマリ・データベースのファイル名に設定されている文字列です。その文字列が一致したら、それが 2 番目の文字列で置換されて、スタンバイ・データベースのファイル名が作られます。

図 12-1 は、ファイル名変換パラメータがどのように機能するかを表しています。

図 12-1 ファイル名変換パラメータの設定



注意： スタンバイ・データベースでデータ・ファイル（またはログ・ファイル）の RENAME を実行した場合、または AS 句を指定して ALTER DATABASE CREATE FILE コマンドを使った場合は、変換パラメータはそのファイルには適用されません。

スタンバイ・データベースのアクティブ化

災害発生時には、（可能なら）プライマリ・データベースのログをアーカイブし (ALTER SYSTEM ARCHIVE LOG CURRENT)、アーカイブ・ログをスタンバイ・サイトに転送して、スタンバイ・データベースをアクティブにする前にそのログを適用してください。これで、スタンバイ・データベースをプライマリ・データベースと同じ時点（障害発生前）の状態にできます。現行のオンライン・ログをアーカイブできない場合は、プライマリ・データベースのアーカイブされていないログからトランザクションを回復せずに、スタンバイ・データベースをアクティブにする必要があります。

スタンバイ・データベースをアクティブにした後は、そのオンライン REDO ログはリセットされています。したがって、スタンバイ・データベースからのログとプライマリ・データベースからのログの間には互換性がありません。また、スタンバイ・データベースはアクティブ化のときにディスマウントされます。したがって、アクティブ化の直後に表およびビューを見ることはできません。

1. スタンバイ・データベースが EXCLUSIVE モードでマウントされていることを確認します。
2. ALTER DATABASE ACTIVATE STANDBY DATABASE コマンドを発行します。
3. スタンバイ・インスタンスを停止します。
4. 新しい稼働データベースのバックアップをできるだけ早くとってください。この時点では、前のスタンバイ・データベースが新しい稼働データベースになります。この作業は必須ではありませんが、バックアップがない状態でアクティブ化した後では変更内容を回復できないので、安全策として行うことをお勧めします。

5. 新しい稼働インスタンスを起動します。

注意： スタンバイ・データベースをアクティブにした後は、元の稼働データベースのアーカイブされていないログのトランザクションはすべて失われます。

プライマリ・データベースの物理構造の変更

プライマリ・データベースの物理構造を変更すると、スタンバイ・データベースにも影響が出ることがあります。次の項では、プライマリ・データベースの構造変更がスタンバイ・データベースに与える影響について説明します。

データ・ファイルの追加

プライマリ・データベースにデータ・ファイルを追加すると、REDO 情報が生成されます。この REDO 情報がスタンバイ・データベースに適用されるときに、そのデータ・ファイルの名前が自動的にスタンバイ制御ファイルに追加されます。スタンバイ・データベースにおいてその新しいファイルが新しいファイル名で検出されれば、回復プロセスが続行します。スタンバイ・データベースで新しいデータ・ファイルが検出されなければ、回復プロセスは停止します。

回復プロセスが停止した場合は、スタンバイ・データベースの回復プロセスを再開する前に、次の手順のどちらか一方を実行してください。

- 追加したデータ・ファイルのバックアップを、プライマリ・データベースからスタンバイ・データベースにコピーする。
- スタンバイ・データベースで `ALTER DATABASE CREATE DATAFILE` コマンドを発行する。

新しいデータ・ファイルがスタンバイ・データベースでは不要な場合は、`DROP` オプションを使ってそのデータ・ファイルをオフラインにできます。

関連項目： データ・ファイルをオフラインにする方法については、12-41 ページの「スタンバイ・データベースのデータ・ファイルのオフライン化」を参照してください。

ファイルの改名

プライマリ・データベース上で改名したデータ・ファイルは、スタンバイ・データベースの制御ファイルがリフレッシュされるまではスタンバイ・データベースに反映されません。プライマリ・データベースのデータ・ファイルを改名したときに、プライマリ・データベースとスタンバイ・データベースのデータ・ファイルの同期をとる場合は、スタンバイ・データベースに対して同様の改名操作を実行します。

ログ・ファイルの変更

スタンバイ・データベースに影響を与えずに、ログ・ファイルのグループまたはメンバーをプライマリ・データベースに追加できます。また、スタンバイ・データベースに影響を与えずに、ログ・ファイルのグループまたはメンバーをプライマリ・データベースから削除できます。同様に、プライマリ・データベースでスレッドを使用可能または使用禁止にしても、スタンバイ・データベースには影響しません。

オンライン・ログ・ファイルの構成を、プライマリ・データベースとスタンバイ・データベースで同じにしておくことができます。その場合は、プライマリ・データベースで **ALTER DATABASE ENABLE THREAD** を使ってログ・ファイル・スレッドを使用可能にしたら、スタンバイ・データベースをアクティブにする前に、スタンバイ・データベース用の新しい制御ファイルを作成してください。リフレッシュ方法については、12-42 ページの「スタンバイ・データベースの制御ファイルのリフレッシュ」を参照してください。

ALTER DATABASE CLEAR UNARCHIVED LOGFILE コマンドを発行してプライマリ・データベースのログ・ファイルを消去した場合や、**RESETLOGS** オプションを使ってプライマリ・データベースをオープンした場合は、スタンバイ・データベースが無効になります。スタンバイ・データベースの回復プロセスの続行に必要なアーカイブ・ログがなくなるので、スタンバイ・データベースを再作成する必要があります。

制御ファイルの変更

プライマリ・データベースで **CREATE CONTROLFILE** コマンドを使って次のいずれかの操作をする場合は、スタンバイ・データベースの制御ファイルが無効になることがあります。

- **REDO** ログ・ファイルのグループまたはメンバーの最大数を変更する。
- データベースを同時にマウントしオープンできるインスタンスの最大数を変更する。

スタンバイ・データベースの制御ファイルが無効にしたときは、制御ファイルの再作成が必要になります。再作成の手順については、12-42 ページの「スタンバイ・データベースの制御ファイルのリフレッシュ」を参照してください。

プライマリ・データベース上で **CREATE CONTROLFILE** コマンドに **RESETLOGS** オプションを指定すると、プライマリ・データベースを次にオープンしたときにオンライン・ログがリセットされ、それによってスタンバイ・データベースが無効になります。

初期化パラメータの設定

ほとんどの初期化パラメータは、プライマリ・データベースとスタンバイ・データベースで同じにしておきます。しかし、**CONTROL_FILES** および **DB_FILE_STANDBY_NAME_CONVERT** など、特定の初期化パラメータは変更してください。その他の初期化パラメータに相違があると、スタンバイ・データベースでパフォーマンスが低下し、場合によってはスタンバイ・データベースの動作が停止することもあります。

次の初期化パラメータは、スタンバイ・データベース回復プロセスで重要な役割を果たします。

- **COMPATIBLE**

COMPATIBLE パラメータは、プライマリ・データベースとスタンバイ・データベースで同じでなければなりません。違っていると、プライマリ・データベースからスタンバイ・データベースにログを適用できないことがあります。

- **DB_FILES**

プライマリ・データベースで使用可能なファイルと同じ数のファイルをスタンバイ・データベースでも使用できるように、MAXDATAFILES は、両方のデータベースで同じでなければなりません。

- **CONTROL_FILES**

CONTROL_FILES は、プライマリ・データベースとスタンバイ・データベースで違っていなければなりません。スタンバイ・データベースについてこのパラメータに示す制御ファイルの名前は、スタンバイ・データベースに存在するものでなければなりません。

- **DB_FILE_NAME_CONVERT (または LOG_FILE_NAME_CONVERT)**

スタンバイ・データベース・ファイル名をプライマリ・データベース・ファイル名と区別できるようにするには、DB_FILE_NAME_CONVERT(または LOG_FILE_NAME_CONVERT) パラメータを設定します。このパラメータの詳細は、12-37 ページの「データ・ファイル名とログ・ファイル名の変換」を参照してください。

関連項目：初期化パラメータの詳細は、『Oracle8 Server リファレンス・マニュアル』を参照してください。

スタンバイ・データベースのデータ・ファイルのオフライン化

プライマリ・データベースのデータ・ファイルのサブセットをサポートする手段として、スタンバイ・データベースのデータ・ファイルをオフラインにできます。たとえば、プライマリ・データベースの一時表領域をスタンバイ・データベースに回復するのが望ましくない場合があります。そのようなときは、ALTER DATABASE DATAFILE 'fn' OFFLINE DROP コマンドをスタンバイ・データベース上で使って、このデータ・ファイルをオフラインにします。この場合、スタンバイ・データベースをオープンした後で、そのオフライン・ファイルが格納されている表領域を削除する必要があります。

ダイレクト・パス操作の実行

ダイレクト・パス・ロード、副問合せによる表作成、プライマリ・データベースでの索引作成のいずれかに伴ってダイレクト・ロードを実行した場合、パフォーマンスが向上するのはプライマリ・データベースだけです。スタンバイ・データベースでは、対応する回復プロセスのパフォーマンスは向上しません。スタンバイ・データベース回復プロセスでは、回復不能なダイレクト・ロードにより生成された REDO 情報の読取りと適用は、それまでと同じく順次方式で行われます。

UNRECOVERABLE オプションを使うプライマリ・データベース・プロセスは、スタンバイ・データベースには反映されません。その理由は、これらのプロセスがアーカイブ REDO ログには記録されないからです。このようなプロセスをスタンバイ・データベースに反映させるには、次の作業のうち、どれか 1 つを実行してください。

UNRECOVERABLE プロセスをスタンバイ・データベースに反映させるには

1. スタンバイ・データベースの中の影響を受けるデータ・ファイルをオフラインにし、アクティブにした後で表領域を削除します。
2. 新しいデータベースのバックアップから、スタンバイ・データベースを再作成します。
3. 影響を受ける表領域のバックアップをとり、プライマリ・データベースの現行のログをアーカイブします。データ・ファイルをスタンバイ・データベースに転送します。その後、スタンバイ回復を再開します。これは、UNRECOVERABLE 操作の後で通常のデータベース回復可能性を確保する手順と同じです。

プライマリ・データベースで UNRECOVERABLE 操作を実行し、スタンバイ・データベースで回復しようとしても、回復操作中にエラー・メッセージを受け取ることはありません。このようなエラー・メッセージは、スタンバイ・データベースの警告ログに表示されます。したがって、スタンバイ・データベースの警告ログを定期的にチェックしてください。

関連項目：詳細については、12-41 ページの「スタンバイ・データベースのデータ・ファイルのオフライン化」を参照してください。

スタンバイ・データベースの制御ファイルのリフレッシュ

次の手順は、プライマリ・データベースの制御ファイルに対して行った変更をリフレッシュする方法、または変更のコピーを作成する方法を示しています。

スタンバイ・データ・ファイルの制御ファイルをリフレッシュするには

1. スタンバイ・データベースに対して **CANCEL** コマンドを発行し、その回復プロセスを停止します。
2. スタンバイ・インスタンスを停止します。
3. プライマリ・データベースに対して **ALTER DATABASE CREATE STANDBY CONTROLFILE AS 'filename'** 文を発行して、スタンバイ・データベース用の制御ファイルを作成します。
4. プライマリ・データベースに対して **ALTER SYSTEM ARCHIVE LOG CURRENT** 文を発行して、プライマリ・データベースの現行のオンライン・ログをアーカイブします。
5. スタンバイ制御ファイルおよびアーカイブ・ログ・ファイルをスタンバイ・サイトに転送します。
6. **ALTER DATABASE MOUNT STANDBY DATABASE [EXCLUSIVE/PARALLEL]** 文を発行して、スタンバイ・データベースを再起動しマウントします（ただし、オープンはいしないでください）。
7. **RECOVER [FROM 'location'] STANDBY DATABASE** 文を発行して、スタンバイ・データベースで回復プロセスを開始します。

回復不能オブジェクトと回復

CREATE TABLE AS SELECT コマンドを使って、表および索引を作成できます。また、それらの表および索引を回復不能として作成するように指定できます。表または索引を回復不能として作成すると、Oracle は、その操作についての **REDO** ログ・レコードを生成しません。したがって、**ARCHIVELOG** モードで実行している場合であっても、回復不能として作成されたオブジェクトは回復できません。

注意： 回復不能として作成した表または索引が消失すると困る場合は、作成した後でバックアップをとってください。

メディア回復を実行するとき、作成された表または索引に回復可能なものと回復不能なものがある場合は、回復不能オブジェクトには **RECOVER** 操作により、論理的に破損していることを示すマークが付きます。回復不能オブジェクトにアクセスしようとすると、**ORA-01578** のエラー・メッセージが戻されます。回復不能オブジェクトは、いったん削除し、必要であれば再作成するようにしてください。

表を回復不能として作成した後でその表についての索引を回復可能として作成することがあるので、メディア回復の実行後に、論理的に破損していることを示すマークは索引には付きません。ただし、表は回復不能なので（したがって、回復後に破損を示すマークが設定されるので）、索引は破損ブロックを示すことになります。この場合、索引をいったん削除し、必要であれば表と索引を再作成しなければなりません。

関連項目：スタンバイ・データベースへの UNRECOVERABLE 操作の影響については、12-14 ページを参照してください。

読取り専用表領域と回復

この項では、読取り専用表領域がインスタンス回復およびメディア回復にどのように影響するかについて説明します。

バックアップ制御ファイルの使用

USING BACKUP CONTROLFILE オプションを使うメディア回復では、読取り専用ファイルの有無がチェックされます。読取り専用ファイルを回復しようとする、エラーが発生します。このエラーを避けるには、バックアップ制御ファイルを使って回復を実行する前に、読取り専用表領域のすべてのデータ・ファイルをオフラインにします。したがって、回復用に正しいバージョンの制御ファイルを使うことが非常に重要です。回復の完了時に表領域を読取り専用にするには、表領域が読取り専用になった時点からの制御ファイルを使う必要があります。同様に、回復の完了時に表領域を読み書きモードにするには、制御ファイルでも読み書きモードになっていなければなりません。適切な制御ファイルが使用できない場合は、CREATE CONTROLFILE コマンドを使って新しい制御ファイルを作成してください。

制御ファイルの再作成

読取り専用表領域のあるデータベース用に制御ファイルを再作成する場合は、いくつか特別な手順に従う必要があります。必要な手順のリストを入手するには、ALTERDATABASE BACKUP CONTROLFILE TO TRACE コマンドを発行します。この手順は、データベースをオープンした後で表領域をオンラインにしなければならないという点を除けば、通常オフラインの表領域についての手順と似ています。

制御ファイルを再作成すると、ある時点で読取り専用になっていた読み書き両用モードの表領域の回復にも影響が出ます。表領域を書込み可能にした後で制御ファイルを再作成すると、Oracle では、表領域がいつ読取り専用から読み書き両用モードに変わったのかを判別できなくなります。したがって、それ以降は、表領域の読取り専用バージョンからの回復はできません。かわりに、最新のバックアップの時点から回復する必要があります。表領域を読み書き両用モードに変更した場合は、すぐにバックアップをとってください。

回復手順の例

この項では、一般的なメディア障害から回復する方法について説明します。取り挙げる項目は次のとおりです。

- メディア障害のタイプ
- データ・ファイルの消失
- オンライン REDO ログ・ファイルの消失
- アーカイブ REDO ログ・ファイルの消失

- 制御ファイルの消失
- ユーザー・エラーからの回復

メディア障害のタイプ

メディア障害は、一般に、永続障害と一時障害の2つのカテゴリに分類できます。永続メディア障害は、ディスク上のデータが永久に消失する原因になるような重大なハードウェア障害です。失われたデータを回復するには、障害が起きた記憶デバイスを修理または交換し、そこに格納されていたファイルのバックアップを復元する以外に方法はありません。一時メディア障害は、データが一時的にアクセス不能になるようなハードウェア障害で、それによってデータが破損することはありません。次に、一時メディア障害の例を2つ示します。

- ディスク・コントローラの故障。ディスク・コントローラを交換すると、ディスク上のデータにアクセスできるようになります。
- 記憶デバイスでの停電。電源が復旧すれば、記憶デバイスおよび関連のすべてのデータに再びアクセスできます。

データ・ファイルの消失

データベースのデータ・ファイルにメディア障害による影響が出た場合、データベースのアーカイブ・モードおよび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の種類によって、適切な回復手順が異なります。次の項で、各状況に適した回復計画について説明します。

データ・ファイルの消失 (NOARCHIVELOG モード)

NOARCHIVELOG モードで運用されているデータベースのデータ・ファイルのどれか1つでも永続または一時メディア障害の影響を受けると、Oracle はそのデータベースを自動的に停止します。メディア障害のタイプに応じて、次の2つの回復手順のどちらかを使用できます。

- 一時メディア障害の場合は、一時的なハードウェア障害を修復した後で、データベースを再起動する。通常はインスタンス回復ができ、オンライン REDO ログを使って、すべてのコミット済みトランザクションを回復できます。
- 永続メディア障害の場合は、page-16 ページのステップに従って、メディア障害から回復する。

データ・ファイルの消失 (ARCHIVELOG モード)

ARCHIVELOG モードで運用されているデータベースのデータ・ファイルが永続または一時メディア障害の影響を受けている状況として、次の場合が考えられます。

- SYSTEM 表領域のデータ・ファイルまたはアクティブ・ロールバック・セグメントを持つデータ・ファイルが一時または永続メディア障害の影響を受けた場合、データベース

は運用不能になるので、まだ自動的に停止されていなければ、ただちにデータベースを停止する。

ハードウェア障害が一時的なものの場合は、修復してからデータベースを再起動してください。通常はインスタンス回復ができ、オンライン REDO ログを使って、すべてのコミット済みトランザクションを回復できます。

永続的なハードウェア問題の場合は、12-15 ページの「クローズ状態のデータベースの回復」で説明している手順に従ってください。

- 一時または永続メディア障害の影響を受けたデータ・ファイルが、上記に示したもの以外のデータ・ファイルだけである場合、影響を受けたデータ・ファイルは使用不可であり、Oracle によって自動的にオフラインにされるが、データベースの運用は続行できる。

データベースの正常な部分をそのまま使用する場合は、データベースを停止しないでください。最初に、一時オプションを使って、問題のあるデータ・ファイルが格納されている表領域をすべてオフラインにしてください。次に、12-15 ページの「クローズ状態のデータベースの回復」で説明している手順に従ってください。

オンライン REDO ログ・ファイルの消失

メディア障害によりデータベースのオンライン REDO ログが影響を受けた場合、オンライン REDO ログの構成（ミラー化されているかいないか）、メディア障害のタイプ（一時か永続か）、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、アクティブでない、など）によって、適切な回復手順が異なります。次の項で、各状況に適した回復計画について説明します。

ミラー化したオンライン REDO ログのオンライン REDO ログ・メンバーの消失

データベースのオンライン REDO ログをミラー化してあり、各オンライン REDO ログ・グループの中にメディア障害の影響を受けていないメンバーが少なくとも 1 つある場合、Oracle では、そのデータベースはそのまま正常に利用できます (データベースの LGWR トレース・ファイルおよび ALERT ファイルにエラー・メッセージが書き込まれます)。ただし、次のどちらかのアクションをとって問題を処理してください。

- ハードウェア障害が一時的な場合は、修復する。修復が済んだら、LGWR は、障害がないものとして、障害発生前は使用不能だったオンライン REDO ログ・ファイルにアクセスします。
- ハードウェア障害が永続的なもの場合は、DROP コマンドを使って破損したメンバーを削除し、ADD コマンドを使って新しいメンバーを追加する。

注意： 新しく追加されたメンバーには、そのログ・グループが再使用されるまで、冗長性はありません。

オンライン REDO ログ・グループのすべてのオンライン REDO ログ・メンバーの消失

メディア障害により、あるオンライン REDO ログ・グループの中のすべてのメンバーが破損した場合、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、発生する状況が異なります。消失したファイルのステータスを検証 (そのファイルがアクティブでなかったことを確認) するには、ファイル名を V\$LOGFILE で見つけて、消失したファイル名に対応するグループ番号を探します。

```
SELECT *
FROM v$logfile
;
```

GROUP#	STATUS	MEMBER
0001		log1
0002		log2
0003		log3

```
SELECT *
FROM v$log
;
```

GROUP#	MEMBERS	STATUS	ARCHIVED
0001	1	INACTIVE	YES
0002	1	ACTIVE	YES
0003	1	CURRENT	NO

アクティブでないオンライン REDO ログ・グループの消失 アクティブでないオンライン REDO ログ・グループのすべてのメンバーが破損した場合は、次の状態になります。

- 一時メディア障害の影響を受けたのがアクティブでないオンライン REDO ログ・グループだけの場合は、修復する。LGWR はそのグループを必要ときに再使用できます。
- メディア障害により永続的にアクセス不能になったのがアクティブでないオンライン REDO ログ・グループだけの場合は、破損したアクティブでないオンライン REDO ログ・グループでは、結果として通常のデータベース運用が停止する。

データベースが停止する前に問題に気づいた場合は、**ALTER DATABASE CLEAR LOGFILE** コマンドを使ってください。

データベースがすでに停止している場合は、次の作業を行います。

アクティブでないオンライン REDO ログ・グループの消失から回復するには

1. Enterprise Manager の「データベース停止」ダイアログ・ボックスで「異常終了停止」ラジオ・ボタンを選択するか、**ABORT** オプションを指定した **SHUTDOWN** コマンドを実行して、現行のインスタンスをただちに異常終了させます。
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。ただし、データベースはオープンしないでください。この操作を実行するには、Enterprise Manager の「データベース起動」ダイアログ・ボックスで「起動してマウント」ラジオ・ボタンを選択するか、**MOUNT** オプションを指定した **STARTUP** コマンドを実行します。
3. 消失したログがアーカイブ済みのものなら、**ALTER DATABASE CLEARLOGFILE** コマンドを発行します。
4. 消失したログがアーカイブされていない場合は、**ALTER DATABASE CLEARUNARCHIVED LOGFILE** コマンドを発行し、その後ただちにデータベースのバックアップをとります。また、データベースの制御ファイルも (**BACKUP CONTROLFILE** オプションを指定した **ALTER DATABASE** コマンドを実行して) バックアップをとってください。

アーカイブされていないログは、消去後も、アーカイブしないで再使用できます。ただし、ログの最後の変更より前に作成されたバックアップは使用できなくなります。(ログの最初の変更の前にファイルをオフラインにした場合を除く)。したがって、消去されたログ・ファイルがバックアップの回復に必要な場合は、バックアップの回復ができなくなります。

あるオフライン・データ・ファイルが存在し、これをオンラインにするために消去済みのアーカイブされていないログが必要な場合は、キーワード **UNRECOVERABLE DATAFILE** を指定しなければなりません。このデータ・ファイルをオンラインにするた

めに必要な REDO は消去され、そのコピーもないため、このデータ・ファイルおよび関連の表領域全体をデータベースから削除しなければなりません。

注意： ALTER DATABASE CLEAR LOGFILE コマンドは、次の 2 つの場合には（メディア障害による I/O エラーのため）失敗することがあります。

- 現在構成されているログ・ファイル名をつけてログ・ファイルを再作成することにより、それを代替メディアに再配置できない場合。
- 現在構成されているログ・ファイル名自体が（たとえば、メディア障害のために）無効または使用できず、そのログ・ファイル名を再使用することによりログ・ファイルを再作成できない場合。

この 2 つの場合、CLEAR LOGFILE コマンドで（I/O エラーを受け取る前に）制御ファイルを正しく更新して、ログ・ファイルの状態を「消去中」および「アーカイブ不要」に変更している可能性があります。この I/O エラーは、CLEAR LOGFILE で新しいログ・ファイルが作成され、それにゼロが書き込まれる段階で発生したものです。

この時点で、次のコマンドを順番に実行することにより、回復を完了できます。

- ADD を実行して、新しい名前でログ・ファイルを追加する。
- DROP を実行して、古い名前のログ・ファイルを削除する。

これで、データベースをオープンできます。

アクティブなオンライン REDO ログ・グループの消失 データベースがまだ実行中であって、失われたアクティブ・ログが現行のログでない場合は、ALTER SYSTEM CHECKPOINT コマンドを使用できます。このコマンドが正常に実行できれば、アクティブだったログがアクティブでなくなり、12-48 ページの手順を実行できます。

コマンドの実行に失敗するか、データベースがすでに停止している場合は、12-48 ページの手順を実行できません。かわりに次の作業を行ってください。

アクティブなオンライン REDO ログ・グループの消失から回復するには

1. メディア障害が一時的なものであれば修復して、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. データベースが NOARCHIVELOG モードになっていて、永続メディア障害のためにアクティブなオンライン REDO ログ・グループにアクセスできなくなった場合は、全体バックアップからデータベースを復元します。

データベースを復元した後で、作業を再実行して、RESETLOGS オプションを使ってデータベースをオープンします。バックアップを失った後で行った更新は、再実行する必要があります。データベースを停止し、データベース全体のバックアップをとりま

3. データベースが ARCHIVELOG モードになっていた場合は、不完全メディア回復を実行する必要があります。12-15 ページの「完全メディア回復の実行」で説明した手順を使って、破損したログの直前のログまでを回復します。失われた REDO ログの現行の名前が、新しく作成したファイルで使用できることを確認してください。使用できない場合は、RENAME コマンドを発行して、破損したオンライン REDO ログ・グループを改名して新しい位置に入れます。
4. RESETLOGS オプションを使ってデータベースをオープンします。

注意： 不完全回復の終点から現在までに行った更新は、すべて再実行する必要があります。

複数の REDO ログ・グループの消失 オンライン REDO ログの複数のグループが消失した場合は、最も回復しにくいログに対応した回復方法を使用してください。回復の困難な方から順に示すと、次のとおりです。

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. アーカイブしていない REDO ログ
4. アクティブでないオンライン REDO ログ

アーカイブ REDO ログ・ファイルの消失

オンライン REDO ログ・グループが満杯になるとアーカイブされる方法でデータベースを運用していて、アーカイブ REDO ログ・ファイルのコピーが 1 つだけ破損した場合には、現在のデータベース運用に影響はありません。しかし、将来メディア回復が必要になったときに、次のような問題が起こることがあります。

- 満杯のオンライン REDO ログ・グループ（現在アーカイブされているもの）が書き込まれた後ですべてのデータ・ファイルのバックアップをとった場合は、アーカイブされた満杯のオンライン REDO ログ・グループは完全メディア回復操作には必要ありません。
- データ・ファイルの最新のバックアップ・ファイルが、満杯のオンライン REDO ログ・ファイルが書き込まれる前にとったものだとします。グループは、現在、破損したアーカイブ REDO ログ・ファイルに対応しています。この後のある時点で、対応するデータ・ファイルが永続メディア障害により破損したとします。破損したデータ・ファイルの最新バックアップを使う必要がありますが、不完全メディア回復では、破損したアーカイブ REDO ログ・ファイルの時点までしかデータベースを回復できません。
- 時間ベースの回復が必要であって、元のオンライン REDO ログ・グループが書き込まれる前にとられた古いデータ・ファイルのバックアップを使う場合に、破損したアーカイブ REDO ログ・ファイルが必要になることがあります。この場合、不完全メディア回復では、破損したアーカイブ REDO ログ・グループの時点までしか、データベースを回復できません。

警告： アーカイブ REDO ログ・グループが破損していることがわかったときは、すぐにすべてのデータ・ファイルのバックアップをとって、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っておいてください。

制御ファイルの消失

メディア障害によりデータベースの制御ファイルが影響を受けた場合は（制御ファイルがミラー化されているかいないかに関係なく）、その後最初に Oracle のバックグラウンド・プロセスから制御ファイルへのアクセスが必要になるまで、データベースはそのまま実行されます。制御ファイルへのアクセスが必要になった時点で、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、データベースがまだ停止していなければ、すぐにメディア障害を修復すると、データベースの自動停止を防止できます。ただし、一時メディア障害を修復する前にデータベースが停止した場合は、修復した（そして制御ファイルへのアクセスを復元した）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が永久にできなくなるようなメディア障害を回復するための適切な手順は、制御ファイルをミラー化しているかどうかによって異なります。次の項では、適切な回復手順について説明します。

ミラー化した制御ファイルのメンバーの消失

データベースの1つ以上のファイルが永続メディア障害により破損した後で、そのメディア障害により破損しなかった制御ファイルが少なくとも1つはある場合、データベースを回復するには、次の手順を実行します。

注意： ミラー化した制御ファイル構成のすべての制御ファイルが破損した場合は、ミラー化していない制御ファイルの消失からの回復手順に従ってください。

制御ファイルの破損後にデータベースを回復するには

1. インスタンスがまだ実行中の場合は、Enterprise Manager の「データベース停止」ダイアログ・ボックスの「異常終了停止」ラジオ・ボタンを選択するか、**ABORT** オプションを指定した **SHUTDOWN** コマンドを実行して、現行のインスタンスをただちに異常終了させます。
2. メディア障害の原因になったハードウェア障害を修復します。ハードウェア障害をすぐに修復できない場合は、破損した制御ファイルを代替記憶デバイスに復元することにより、データベース回復の次のステップに進むことができます。
3. データベースの制御ファイルの無傷のコピーを使って、破損した制御ファイルに上書きします。可能であれば、無傷の制御ファイルを、破損したすべての制御ファイルが入っていた元の位置にコピーしてください。ハードウェア障害がまだ存在している場合は、無傷の制御ファイルを代替位置にコピーします。破損した制御ファイルをすべて元の位置に復元した場合は、ステップ 5 に進んでください。破損した制御ファイルの一部を復元できなかった場合、または元の位置に復元しなかった場合には、ステップ 4 に進んでください。
4. ステップ 3 で、破損した制御ファイルの一部を復元できなかった場合、または元の位置に復元しなかった場合には、必ずデータベースのパラメータ・ファイルを編集してください。これにより、**CONTROL_FILES** パラメータですべての制御ファイルの現行位置にその編集内容が反映でき、復元されなかった制御ファイルはすべて除外されます。
5. 新しいインスタンスを起動します。そして、データベースをマウントし、オープンします。

現行の制御ファイルのすべてのコピーの消失

永続メディア障害のためにデータベースのすべての制御ファイルが失われたか破損した場合でも、オンライン REDO ログ・ファイルがすべて無傷であれば、(**CREATE CONTROLFILE** コマンドに **NORESETLOGS** オプションを指定して) 新しい制御ファイルを作成することにより回復できます。その後で、**RECOVER DATABASE** を実行し、さらに **ALTER DATABASE OPEN** を実行します。

制御ファイルのバックアップの存在と現行度に応じて、**CREATE CONTROLFILE** コマンドのテキストの生成時に次のオプションが使用できます。

- データベースの構造を最後に変更した後で、ALTER DATABASE BACKUP CONTROLFILE TO TRACE NORESETLOGS を実行していて、この SQL コマンドの出力を保存してある場合は、その出力をそのままの状態を使って、CREATE CONTROLFILE コマンドを実行できる。しかし、ALTER DATABASE BACKUP CONTROLFILE TO TRACE を最後に実行したのがデータベースの構造を変更する前であった場合は、ALTER DATABASE BACKUP CONTROLFILE TO TRACE の出力を編集して構造変更を反映させる必要があります。たとえば、データベースに最近データ・ファイルを追加した場合は、そのデータ・ファイルを CREATE CONTROLFILE コマンドの DATAFILE 句に追加しなければなりません。
- ALTER DATABASE BACKUP CONTROLFILE コマンドの TO TRACE オプションを使うかわりに、TO *filename* オプションを使って制御ファイルのバックアップをとった場合は、その制御ファイルのコピーを使って SQL コマンドの出力を取得できる。それには、バックアップ制御ファイルをコピーして、ALTER DATABASE BACKUP CONTROLFILE TO TRACE NORESETLOGS を実行する前に STARTUP MOUNT を実行します。制御ファイルのコピーの日付が最新の構造変更より前の場合は、TO TRACE 出力を編集して、その構造変更を反映させる必要があります。
- 制御ファイルのバックアップ (TO TRACE 形式または TO *filename* 形式) がない場合は、CREATE CONTROLFILE コマンドを手動で生成する必要がある。

ユーザー・エラーからの回復

データベースに対して、操作上あるいはプログラム上の変更が誤って行われた場合、それが原因でデータが失われたり破損したりすることがあります。回復するには、エラーの前の状態に戻すことが必要な場合があります。

注意： データベース管理者が、厳選した適切なユーザーだけに重要な権限 (DROP ANY TABLE など) を正しく付与していれば、データベースの回復が必要になるようなユーザー・エラーを最小限に抑えることができます。

1. 破損していない既存データベースのバックアップをとります。
2. 既存のデータベースはそのまま手をつけずにおいて、時間ベースの回復により、ユーザー・エラーの発生時点までのデータベースの一時コピーを再構築します。
3. データベースの一時コピー（再構築したもの）から、失われたまたは破損したデータをエクスポートします。
4. 消失または破損したデータを、永続データベースにインポートします。
5. ディスク領域の節約のため、データベースの一時コピー（再構築したもの）に対応付けられたファイルを削除します。

次の例では、誤って削除された表の回復方法を示します。

1. ユーザー・エラーがあったデータベースは、オンラインのままで通常に使用できる状態にしておくことができます。データベースは、オープンしたままにしても、停止してもかまいません。この手順の残りのステップの実行中にエラーが起きた場合に備えて、既存のデータベースのすべてのデータ・ファイルのバックアップをとります。
2. 時間ベースの回復を使って、過去の時点までのデータベースの一時コピーを作成します。永続データベースの既存の制御ファイルとの矛盾が起きないように注意してください。1つの制御ファイルのバックアップを別の位置に復元し(ステップ4)、必要に応じてパラメータ・ファイルを編集するか、新しい制御ファイルを別の位置に作成します。また、データベースの永続コピーに影響を与えないように、すべてのデータ・ファイルを別の位置に復元します(ステップ5)。
3. Oracle の Export ユーティリティを使って、復元したデータベースの一時コピーから、失われたデータをエクスポートします。この例では、誤って削除された表をエクスポートします。

注意： システム監査オプションはエクスポートされます。

4. Oracle Import ユーティリティを使って、エクスポートしたデータ(ステップ3)をデータベースの永続コピーにインポートします。
5. ディスク領域の節約のため、データベースの一時コピー(再構築したもの)のファイルを削除します。

関連項目： Import ユーティリティと Export ユーティリティの詳細は、『Oracle8 Server ユーティリティ』を参照してください。

表領域の Point-in-Time 回復の実行

注意： 表領域の Point-in-Time 回復の作業は複雑なので、ここで説明する手順を使用する前に Oracle の技術サポートに連絡し、指示に従って作業することをお勧めします。

この章では、表領域の Point-in-Time 回復 (TSPITR) の実行方法について説明します。取り挙げる項目は次のとおりです。

- 表領域の Point-in-Time 回復の概要
- 表領域の Point-in-Time 回復の計画
- 表領域の Point-in-Time 回復の実行
- パーティション表の部分的な TSPITR の実行
- パーティションを削除した場合のパーティション表の TSPITR の実行
- パーティションを分割した場合のパーティション表の TSPITR の実行
- TSPITR のチューニングに関する考慮事項

表領域の Point-in-Time 回復の概要

表領域の Point-in-Time 回復 (TSPITR) 機能を使用することで、1 つ以上の表領域をデータベースの残りの表領域とは異なる時点に素早く回復させることができます。TSPITR は、次のような状況に最も有効です。

- 誤って実行した、表の削除または切捨て操作から回復する場合。
- 論理的に壊れた表を回復させる場合。
- 不正確なバッチ・ジョブ、またはデータベースのサブセットだけに影響を与えたその他の DML 文から回復する場合。
- 1 つの物理データベースの別々の表領域に複数の論理データベースがあり、そのうちの 1 つの論理データベースを残りの論理データベースとは異なる時点に回復させる場合。
- VLDB(大規模データベース)において、データベース全体の Point-in-Time 回復が可能であるが、バックアップからデータベース全体を復元しロールフォワードを実行するのではなく、表領域の Point-in-Time 回復だけを実行する場合（事前に 13-3 ページの「表領域の Point-in-Time 回復の計画」を参照してください）。

表のエクスポートと同様に、TSPITR によって一貫性のあるデータ・セットを回復できます。ただし、データ・セットは 1 つのオブジェクトではなく表領域全体になります。

Oracle8 より前のバージョンでは、Point-in-Time 回復は、次の手順でデータベースのサブセットに対してだけ使用できました。

1. データベースのコピーを作成する
2. データベースのコピーを必要な時点までロールフォワードする
3. データベースのコピーから必要なオブジェクトをエクスポートする
4. 稼働データベースから関連するオブジェクトを削除する
5. オブジェクトを稼働データベースにインポートする

この方法では、大きいオブジェクトのエクスポートやインポートを行うとパフォーマンスに影響がありました。

TSPITR によってデータベースのサブセットを回復できます。また、ユーザーが（回復したデータベースの関連ファイルの）オペレーティング・システム・レベルのデータ・ファイル・コピーを稼働データベースに作成することで、TSPITR はエクスポート/インポートのフェーズを最適化します。ファイルの内容に関するデータ・ディクショナリ情報（ファイル内の回復されたセグメントの情報など）は、コピー・データベースからメタ・データのエクスポート/インポートという方法で稼働データベースに転送されます。ファイルのコピーも、このインポート処理によって稼働データベースに追加されます。

次の用語と略称はこの章で頻繁に使用されるので、覚えておいてください。

TSPITR

表領域の Point-in-Time 回復

クローン・データベース

Oracle8 の TSPITR で回復に使用されるデータベースのコピーを、「クローン・データベース」といいます。クローン・データベースは、正規のデータベースとは様々な点で本質的に異なります。

回復セット

Point-in-Time 回復を必要とする表領域です。

補助セット

TSPITR に必要な要素で、次のものが含まれます。

- バックアップ制御ファイル
- システム表領域
- ロールバック・セグメントが入っているデータ・ファイル
- 一時表領域（オプション）

ソート操作のためのエクスポート用に小さい領域が必要です。一時表領域のコピーが補助セットに含まれていない場合、クローン・データベースを起動した後に新しい一時表領域を作成するか、またはシステム表領域ファイルに対して自動拡張を ON に設定することによって、ソート領域を作成する必要があります。

表領域の Point-in-Time 回復の計画

TSPITR の手順は複雑なので、慎重に計画する必要があります。作業を進める前に、この章全体をよく読んでください。

警告： TSPITR を初めて実行する場合は、稼働システムでは実行しないでください。また、時間的な制約がある状況においても実行しないでください。

制限事項

ここでは、TSPITR に関する制限事項について説明します。

TSPITR を実行するかどうかを判断するときに、第一に考慮しなくてはならないのは、(明示的ではなく暗黙的な参照依存性が原因で) 回復済み表領域内の表と未回復の表領域内の表の間に存在するアプリケーション・レベルの矛盾の問題です。これらの依存性を理解し、矛盾がある場合には、RMAN の TSPITR を実行する前にすべての矛盾を解決するための手段を用意しておく必要があります。

TS_PITR_CHECK は SYS が所有するオブジェクトをチェックしない

TS_PITR_CHECK ビューは、TSPITR 作業の障害となる依存性や制限に関する情報を提供します。ただし、TS_PITR_CHECK は、SYS が所有するオブジェクトについては依存性や制限の情報を提供しません。

回復セット内に、取消しセグメントなど SYS 所有のオブジェクトがある場合、これらのオブジェクトの回復に成功するという保証はありません (TSPITR が使用する Export ユーティリティおよび Import ユーティリティは、SYS 所有のオブジェクトに対しては実行できないからです)。回復セット SYS が所有している内でオブジェクトを調べるには、次の文を発行してください。

```
SELECT OBJECT_NAME, OBJECT_TYPE
FROM SYS.DBA_OBJECTS
WHERE TABLESPACE_NAME IN ('<tablespacename>', '<tablespacename',
<tablespace name N') and owner = 'SYS';
```

関連項目：TS_PITR_CHECK ビューの詳細は、13-8 ページの「ステップ 2: プライマリ・データベースにおける依存性の調査と解決」を参照してください。

TS_PITR_CHECK はスナップショット表を検出しない

TS_PITR_CHECK ビューは、スナップショット表を検出しません (スナップショット・ログは検出します)。スナップショット表はスタンドアロン表としてエクスポートされます。スナップショットが時刻 3 に削除され、時刻 1 のバックアップを使用して時刻 2 にロールフォワードした場合、TSPITR 完了後にスナップショット表はスタンドアロン表として作成されています。ただし、このスタンドアロン表に関連のスナップショット・ビューは付随しません。

パーティション表と TS_PITR_CHECK

述語に与えられた表領域にパーティション表の最初のセグメントが入っている場合、TS_PITR_CHECK ビューの結果セットが反転されます。述語に与えられた表領域にパーティション表の最初のセグメントが入っていない場合には、そのパーティションについて 1 行が戻ります。述語に与えられた表領域にパーティション表の最初のセグメントが入っている場合には、結果は反転されます (たとえば、そのパーティション表のパーティションを含む各表領域について 1 行ずつ戻りますが、述語に与えられた表領域については行は戻りません)。

行が戻ったということは、解決しなくてはならない矛盾が存在することを示します。この矛盾は、そのパーティションをスタンドアロン表と入れ替えることで解決します。

関連項目：詳細は、13-16 ページの「パーティション表の部分的な TSPITR の実行」を参照してください。

ビットマップ索引

TSPITR が終了した後に、ビットマップ索引を削除して再度作成する必要があります。これを行わないと、ビットマップ索引を使用できません。表にビットマップ索引が存在すると、たとえビットマップ索引がプライマリ・データベースから削除されていても、インポートは失敗します。このような場合、不正確な索引セグメントが作成されることもあります。その索引は削除して再作成する必要があります。

パーティション化されていないグローバル索引

TS_PITR_CHECK ビューは、回復セットに含まれていないパーティション表のパーティション化されていないグローバル索引を検出しません。これは、ビューを手動で問い合わせた場合、および TSPITR のエクスポートとインポートのフェーズで明確にわかります。TSPITR の終了後、エラーは戻りませんが、古い索引が回復した表に残っています。この場合、索引を削除して作成し直す必要があります。

注意： 索引は依然として有効なので、その索引を使用する問合せでは不正確な行が戻されます。

一般的な制約事項

前述の制限事項の他に、TSPITR には次のような制約があります。

- 削除した表領域を、TSPITR を使って回復することはできない。
- 削除後に同じ名前で再作成された表領域を、TSPITR を使用して回復することはできない。
- 間違った表領域に追加されたデータ・ファイルを、TSPITR を使って削除することはできない。そのファイルが、TSPITR の実行対象となる時点より後に追加された場合、TSPITR の実行後もそのファイルは、空のまま依然として表領域の一部として保存されます。
- クローン・データベースに対して DML 文は使用できない。クローン・データベースは回復処理にだけ使用できます。
- TSPITR の終了後は、回復セットのデータ・ファイルの既存のバックアップを回復処理に使用できない。このため、回復したファイルのバックアップを新たに作成する必要があります。TSPITR を実行する前に作成されたバックアップを使って回復処理を行うと、その回復処理は失敗します（エラー・メッセージ ORA-1247, 00000 「表領域 : %s の TSPITR を使用してデータベースを回復します」が戻されます）。

- TSPITR は、オブジェクトについて計算された統計情報が入っているオブティマイザ統計を回復しない。TSPITR の実行後、統計を計算し直す必要があります。
- 次のオブジェクト型は、TSPITR 回復セット内では使用できません。
 - レプリケートされたマスター表
 - varray 列が入っている表
 - ネストした表が入っている表
 - 外部 B ファイルを伴う表
 - スナップショット・ログ
 - スナップショット表
 - SYS が所有するオブジェクト（ロールバック・セグメントを含む）

データ整合性と TSPITR

TSPITR は、回復する表領域内のオブジェクトとデータベースの残りの部分にあるオブジェクトと間のデータの関連を検出できるビューを提供します。この関連を削除または中断するか、もしくは関連オブジェクトを回復セットに組み込むことによりこのデータの関連を管理しない限りは、TSPITR は正常に実行されません。

関連項目：詳細は、13-8 ページの「ステップ 2: プライマリ・データベースにおける依存性の調査と解決」および 13-4 ページの「TS_PITR_CHECK は SYS が所有するオブジェクトをチェックしない」を参照してください。

TSPITR の必要条件

TSPITR を実行するには、事前に次の必要条件を満たしている必要があります。

- 回復セット表領域を構成するすべてのファイルは、回復セットに含まれていなくてはならない。そうしないと、TSPITR のエクスポート・フェーズは失敗します（エラー・メッセージ「1230: 読込み専用にできません-ファイル:%s がオフラインです」が戻されます）。
- 次の SQL 文を使って、補助セットに制御ファイルのバックアップを作成する必要がある。

```
ALTER DATABASE BACKUP CONTROLFILE TO '<controlfile_name>'
```

この制御ファイルのバックアップは、現在使用しているバックアップよりも後に作成されたものでなくてはなりません。そうしないと、エラー・メッセージ (ORA-01152 「ファイル:%s は十分に古いバックアップから復元されていません」) が戻されます。

- クローン・データベース（補助セットおよび回復セットなど）を格納するために十分なディスク領域。
- クローン・インスタンスを起動するために十分な実メモリー。

関連項目：詳細は、13-12 ページの「ステップ 4: クローン・データベースのパラメータ・ファイルを準備する」を参照してください。

表領域の Point-in-Time 回復の実行

ここでは、TSPITR の実行方法を次の順序で説明します。

- ステップ 1: TSPITR 実行時に消失するオブジェクトを調べる
- ステップ 2: プライマリ・データベースにおける依存性の調査と解決
- ステップ 3: TSPITR 用にプライマリ・データベースを準備する
- ステップ 4: クローン・データベースのパラメータ・ファイルを準備する
- ステップ 5: TSPITR 用にクローン・データベースを準備する
- ステップ 6: クローン・データベースを回復する
- ステップ 7: クローン・データベースをオープンする
- ステップ 8: エクスポート用にクローン・データベースを準備する
- ステップ 9: クローン・データベースをエクスポートする
- ステップ 10: 回復セットのクローン・ファイルをプライマリ・データベースにコピーする
- ステップ 11: プライマリ・データベースにインポートする
- ステップ 12: プライマリ・データベースを使用できるように準備する
- ステップ 13: プライマリ・データベースの回復した表領域のバックアップをとる

ステップ 1: TSPITR 実行時に消失するオブジェクトを調べる

表領域について TSPITR を実行すると、TSPITR の実行対象となる時点より後に作成されたオブジェクトは失われます。どのオブジェクトが消失するかを確認するには、プライマリ・データベースの TS_PITR_OBJECTS_TO_BE_DROPPED ビューを問い合わせます。ビューの内容を、表 13-1 に示します。

表 13-1 TS_PITR_OBJECTS_TO_BE_DROPPED ビュー

列名	NULL の場合 があるか	型
OWNER	NULL なし	VARCHAR2(30)
NAME	NULL なし	VARCHAR2(30)
CREATION_TIME	NULL なし	DATE
TABLESPACE_NAME		VARCHAR2(30)

TS_PITR_OBJECTS_TO_BE_DROPPED を問い合わせるときに、データ・フィールドのすべての要素を指定する必要があります。指定しないとデフォルトの設定値が使用されます。to_char 関数と to_date 関数も使用してください。たとえば、回復セットが TS1 と TS2 で構成され、回復時点が '1997-06-02:07:03:11' の場合、次のような問合せを発行します。

```
SVRMGR1> select owner, name, tablespace_name,
2> to_char(creation_time, 'YYYY-MM-DD:HH24:MI:SS'),
3> from ts_pitr_objects_to_be_dropped
4> where tablespace_name in ('TS1','TS2')
5> and
6> creation_time > to_date('97-JUN-02:07:03:11','YY-MON- DD:HH24:MI:SS')
7> order by tablespace_name, creation_time
8> / The information you find in TS_PITR_OBJECTS_TO_BE_DROPPED and TS_PITR_CHECK
can help you decide whether or not to perform TSPITR.
```

ステップ 2: プライマリ・データベースにおける依存性の調査と解決

TS_PITR_CHECK ビューを使って、回復セットの内部と外部で重複しているオブジェクト間の関係を明らかにすることができます。このビューが問合せに対して行を戻した場合、問題を調べて修正する必要があります。TS_PITR_CHECK ビューが行を戻さなかった場合にだけ、TSPITR を実行できます。このステップにおいて実行したすべてのアクションを記録してください。これによって、TSPITR の終了後に、オブジェクト間の関係をトレースし直すことができます。

次の条件が満たされていないと、TS_PITR_CHECK ビューは行を戻します。

- パーティション表のすべてのパーティションが回復セットに含まれている。あるパーティション表の一部のパーティションだけを回復する場合は、それらのパーティションをスタンダアロン表と入れ替えます (13-16 ページの「パーティション表の部分的な TSPITR の実行」を参照)。
- 次の要素がすべて回復セットに含まれている。
 - 表 (パーティション索引または非パーティション索引を含む)
 - クラスタ (パーティション索引または非パーティション索引を含む)
 - 主キーと外部キーの関係
 - LOB のすべての要素 (LOB セグメント、LOB 索引、LOB ロケータ)
- 次のオブジェクト型が回復セットにない。
 - レプロケートされたマスター表
 - varray 列が入っている表
 - ネストした表が入っている表
 - 外部 B ファイルを伴う表
 - スナップショット・ログ
 - スナップショット表

TS_PITR_CHECK ビューの問合せ

表 13-2 に、TS_PITR_CHECK ビューの内容を示します。

表 13-2 TS_PITR_CHECK ビュー

列	データ型	NULL	説明
OBJ1_OWNER	VARCHAR2(30)	NULL なし	表領域の Point-in-Time 回復の障害となるオブジェクトの所有者 (詳細は REASON 列を参照)
OBJ1_NAME	VARCHAR2(30)	NULL なし	表領域の Point-in-Time 回復の障害となるオブジェクトの名前
OBJ1_TYPE	VARCHAR2(15)		表領域の Point-in-Time 回復の障害となるオブジェクトのオブジェクト型
OBJ1_SUBNAME	VARCHAR2(30)		OBJ1_NAME に従属している
TS1_NAME	VARCHAR2(30)	NULL なし	表領域の Point-in-Time 回復の障害となるオブジェクトが入っている表領域名
OBJ2_NAME	VARCHAR2(30)		表領域の Point-in-Time 回復の障害になると考えられる第 2 のオブジェクトの名前 (NULL の場合は、オブジェクト 1 だけが原因)

表 13-2 TS_PITR_CHECK ビュー

列	データ型	NULL	説明
OBJ2_TYPE	VARCHAR2(15)		第2のオブジェクトのオブジェクト型 (OBJ2_NAME が NULL の場合は、これも NULL)
OBJ2_OWNER	VARCHAR2(30)		第2のオブジェクトの所有者 (OBJ2_NAME が NULL の場合は、これも NULL)
OBJ2_SUBNAME	VARCHAR2(30)		OBJ2_NAME に従属している
TS2_NAME	VARCHAR2(30)		表領域の Point-in-Time 回復の障害となる第2のオブジェクトが入っている表領域名 (-1 の場合は、該当なし)
CONSTRAINT_NAME	VARCHAR2(30)		制約の名前
REASON	VARCHAR2(78)		表領域の Point-in-Time 回復を実行できない理由

TS_PITR_CHECK ビューを問い合わせるには、回復セットの表領域の詳細を示す、4 行からなる述語を指定する必要があります。たとえば、回復セットが TS1 と TS2 で構成されている場合、TS_PITR_CHECK に対する SELECT 文は、次のとおりです。

```
SVRMGR 1> SELECT *
          2> FROM sys.ts_pitr_check
          3>   WHERE
          4>     (ts1_name in ('TS1','TS2'))
          5>   AND ts2_name not in ('TS1','TS2'))
          6>   OR (ts1_name not in ('TS1','TS2')
          7>   AND ts2_name in ('TS1','TS2'))
          8> /
```

TS_PITR_CHECK ビュー内の列の数と長さを考慮して、次のように列を書式設定できます。

```
column OBJ1_OWNER heading "owner1"
column OBJ1_OWNER format a6
column OBJ1_NAME heading "name1"
column OBJ1_NAME format a5
column OBJ1_SUBNAME heading "subname1"
column OBJ1_SUBNAME format a8
column OBJ1_TYPE heading "obj1type"
column OBJ1_TYPE format a8 word_wrapped
column TS1_NAME heading "ts1_name"
column TS1_NAME format a8
column OBJ2_NAME heading "name2"
column OBJ2_NAME format a5
column OBJ2_SUBNAME heading "subname2"
column OBJ2_SUBNAME format a8
column OBJ2_TYPE heading "obj2type"
column OBJ2_TYPE format a8 word_wrapped
column OBJ2_OWNER heading "owner2"
column OBJ2_OWNER format a6
column TS2_NAME heading "ts2_name"
```

```
column TS2_NAME format a8
column CONSTRAINT_NAME heading "cname"
column CONSTRAINT_NAME format a5
column REASON heading "reason"
column REASON format a57 word_wrapped
```

出力の例 パーティション表 TP には、P1 と P2 という 2つのパーティションがあり、それぞれ表領域 TS1 と TS2 に存在するとします。また、TP 上に定義されたパーティション索引 TPIND には、ID1 と ID2 という 2つのパーティションがあり、それぞれ表領域 ID1 と ID2 に存在するとします。この場合、表領域 TS1 と TS2 に対して TS_PITR_CHECK を問い合わせると、次のように出力されます（形式は、前もってこのように設定されているとします）。

```
owner1 name1 subname1 obj1type ts1_name name2 subname2 obj2type owner2
ts2_name cname reason
-----
SYSTEM TP P1 TABLE TS1 TPIND IP1 INDEX PARTITION PARTITION
SYS ID1 Partitioned Objects not fully contained in the recovery set

SYSTEM TP P1 TABLE TS1 TPIND IP2 INDEX PARTITION PARTITION SYS ID2
Partitioned Objects not fully contained in the recovery set
```

この出力から、表 SYSTEM.TP にはパーティション索引 TPIND があり、この TPIND は、表領域 ID1 内の IP1 と表領域 ID2 内の IP2 という 2つのパーティションで構成されていることがわかります。したがって、TPIND を削除するか、または ID1 と ID2 を回復セットに入れる必要があります。

ステップ 3: TSPITR 用にプライマリ・データベースを準備する

TSPITR 用にプライマリ・データベースを準備するには、次の作業を行ってください。

1. プライマリ・データベースに対して次の文を発行します。

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

2. 次の文を発行して、回復セット内のロールバック・セグメントをオフラインにします（補助セットのロールバック・セグメントはオフラインにする必要はありません）。

```
ALTER ROLLBACK SEGMENT <segment name> OFFLINE;
```

3. 次の文を発行して、プライマリ・データベース上の回復セット表領域を即時オフラインにします。

```
ALTER TABLESPACE <tablespace name> OFFLINE IMMEDIATE;
```

これにより、TSPITR の完了前に回復セットが変更されないようになります。

注意： 回復セットの表領域内に問合せを行うデータのサブセット（物理的にも論理的に破損していないものに限る）がある場合は、プライマリ・データベースの回復セット表領域を、クローンの回復セット中だけ **READ ONLY** に変更できます（これにより、回復セット表領域に対して問合せはできますが変更はできなくなります）。回復セット表領域は、クローン・ファイルをプライマリ・データベースに統合する前にオフラインにする必要があります（13-15 ページの「ステップ 10: 回復セットのクローン・ファイルをプライマリ・データベースにコピーする」を参照）。

ステップ 4: クローン・データベースのパラメータ・ファイルを準備する

新しい `init.ora` ファイルからパラメータ・ファイルを作成します（稼働インスタンスのファイルは使用しないでください）。`DB_BLOCK_BUFFERS` または `SHARED_POOL_SIZE`、`LARGE_POOL_SIZE` などのパラメータの設定値を「小さく」することで、メモリーを節約できます。ただし、クローン・データベースに稼働パラメータ・ファイルが使用されていると、他のパラメータ値（共有プール内からメモリーを割り当てる `ENQUEUE_RESOURCES` パラメータなど）が大きい場合に、これら 3 のパラメータ値を小さくすることで、クローン・データベースが起動するのを防ぐことができます。次のパラメータを変更する必要があります。

- `CONTROL_FILES` は、クローン制御ファイルの名前と格納場所を指し示す必要があります。

- LOCK_NAME_SPACE には、lock_name_space=CLONE など一意の値を設定する必要があります。

注意： LOCK_NAME_SPACE を設定することで、クローン・データベースがプライマリ・データベースと同じ名前であっても、クローン・データベースを起動できます (LOCK_NAME_SPACE を設定しないと、データベース名が不適正であるとしてマウント・ロックがかかり、起動は失敗します)。DB_NAME パラメータは変更しないでください。

必要であれば、次のパラメータを変更してください。

- DB_FILE_NAME_CONVERT
- LOG_FILE_NAME_CONVERT

これらのパラメータを使ってクローン・データベースのファイルの格納ディレクトリ名を指定し、クローン・データベースの制御ファイルを更新します。

たとえば、プライマリ・データベースのデータ・ファイルが /ora/primary ディレクトリにあり、クローンが /ora/clone ディレクトリにある場合、DB_FILE_NAME_CONVERT には "primary","clone" と設定します。

ステップ 5: TSPITR 用にクローン・データベースを準備する

TSPITR 用にクローン・データベースを準備するには、次の作業を行います。

1. 補助セットと回復セットを、プライマリ・データベースの格納場所とは異なる場所に復元します。

注意： 推奨はできませんが、回復セットのファイルを、プライマリ・データベース上の対応するファイルに上書きできます。詳細は、13-6 ページの「パーティション表の部分的な TSPITR の実行」を参照してください。

2. クローン・データベースを起動できるように環境を設定します (たとえば、UNIX では ORACLE_SID にクローンの名前を設定します)。
3. クローン・データベースを nomount で起動します。必要であれば pfile を指定します (指定例: 'STARTUP NOMOUNT PFILE=/path/initCLONE.ora')。
4. 次の文を発行して、クローン・データベースをマウントします。

```
ALTER DATABASE MOUNT CLONE DATABASE;
```

この時点で、データベースはクローンなので自動的にアーカイブ・ログ・モードではなくなります。同時に、すべてのファイルもこの時点でオフラインになります。ファイル

名変換パラメータの `DB_FILE_NAME_CONVERT` と `LOG_FILE_NAME_CONVERT` が設定されていても、クローン・データベースのすべてのファイルが、これらのパラメータに指定された場所に格納されていると想定することはできません。場合によっては、ディスク領域の制約のために、別の場所に復元されるクローン・データベースがあります。また、クローン・データベースに本当に必要なファイルは、回復セットと補助セットに含まれています。これらの2つのセットに含まれないその他のデータベース・ファイルは、オフラインのままです。

5. `DB_FILE_NAME_CONVERT` と `LOG_FILE_NAME_CONVERT` が設定されていない場合、次の文を発行して、新しい格納場所を反映するようにファイルを改名する必要があります。

```
ALTER DATABASE RENAME FILE '<name of file in primary location>'
TO '<name of corresponding file in clone location>';
```

6. `DB_FILE_NAME_CONVERT` と `LOG_FILE_NAME_CONVERT` が設定されていても、別の場所に復元されたファイルがある場合、この時点でそのファイル名を変更する必要があります。次の SQL 文を発行して、すべての回復セットと補助セットをオンラインにします。

```
ALTER DATABASE DATAFILE '<datafile name>' ONLINE;
```

注意： 各回復セット表領域のすべてのファイルがオンライン状態にならないと、TSPITR のエクスポート・フェーズは実行できません。

ステップ 6: クローン・データベースを回復する

`USING BACKUP CONTROLFILE` オプションを指定して、クローン・データベースに必要な時点まで回復します。次のように、時間ベースの回復や取消しベースの回復などあらゆる形態の割込み回復を実行できます。

```
RECOVER DATABASE USING BACKUP CONTROLFILE UNTIL TIME 'YYYY-MM-DD:HH24:MI:SS';
```

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

クローン・データベースのファイルがオンライン状態でないと、エラー・メッセージ（「ORA 264: 回復は必要ありません」）が戻されます。

ステップ 7: クローン・データベースをオープンする

次の文で、クローン・データベースのオープン・リセットログを変更します。

```
ALTER DATABASE OPEN RESETLOGS;
```

データベースはクローンなので、この時点では **SYSTEM** ロールバック・セグメントだけがオンラインになります。このため、ユーザー表領域に対しては **DML** 文を実行できません。ユーザー・ロールバック・セグメントをオンラインにしようとしても失敗します (ORA-1698 「クローン・データベースで **SYSTEM** ロールバック・セグメントをオンラインにしています」というエラー・メッセージが戻されます)。

ステップ 8: エクスポート用にクローン・データベースを準備する

プライマリ・データベースの場合と同様に (13-8 ページの「ステップ 2: プライマリ・データベースにおける依存性の調査と解決」を参照)、**TS_PITR_CHECK** ビューを使って依存性を解消し、クローン・データベースのエクスポートに備えてください。**TS_PITR_CHECK** が行を戻さない場合にだけ、**TSPITR** のエクスポート・フェーズを実行できます。

ステップ 9: クローン・データベースをエクスポートする

次の文を発行して、回復セット表領域のメタ・データをエクスポートします。

```
exp sys/<password> point_in_time_recover=y  
recovery_tablespace=<tablespace1>,<tablespace2>,<tablespaceN>
```

エクスポート・フェーズが失敗した場合 (エラー・メッセージ「ORA 29308 ビュー **TS_PITR_CHECK** で障害が発生しました。」) は、**TS_PITR_CHECK** を再度問い合わせ問題为解决し、エクスポートを再実行してください。**TSPITR** のエクスポート・フェーズは、ユーザー **SYS** で実行する必要があります。そうしないと、エクスポートは失敗します (エラー・メッセージ「ORA-29303: ユーザーが **SYS** でログインしていません」)。エクスポートが正常に終了したら、クローン・データベースを停止します。

ステップ 10: 回復セットのクローン・ファイルをプライマリ・データベースにコピーする

プライマリ・データベース上の回復セットの表領域が **READ ONLY** になっている場合は、**OFFLINE** に変更する必要があります。クローン・データベースから回復セットのファイルをプライマリ・データベースにコピーします。このとき、プライマリ・データベース上の補助セットのファイルを上書きしないように注意してください。

ステップ 11: プライマリ・データベースにインポートする

次のコマンドによって、回復セットのメタ・データをプライマリ・データベースにインポートします。

```
imp sys/<password> point_in_time_recover=true
```

このインポート処理によって、コピーされたファイルのファイル・ヘッダーが更新され、プライマリ・データベースに統合されます。

ステップ 12: プライマリ・データベースを使用できるように準備する

最初に、プライマリ・データベースの回復セットの表領域をオンライン状態にします。次に、回復セットの表領域を **READ WRITE** モードに変更します (**READ ONLY** に変更されていた場合は、13-12 ページの「ステップ 3: TSPITR 用にプライマリ・データベースを準備する」を参照してください)。

プライマリ・データベースを使用できるように準備するには、依存性の解消のために実行したすべてのステップを取り消します。たとえば、索引を再作成したり制約を使用可能状態に戻します (13-8 ページの「ステップ 2: プライマリ・データベースにおける依存性の調査と解決」を参照)。TSPITR を実行する前に回復セットのオブジェクトの統計が存在していた場合、その統計を再計算する必要があります。パーティション表の場合には、スタンドアロン表をパーティション表のパーティションと入れ替える必要があります (詳細は、13-16 ページの「パーティション表の部分的な TSPITR の実行」を参照してください)。

ステップ 13: プライマリ・データベースの回復した表領域のバックアップをとる

表領域に対する TSPITR が完了した後で、表領域のバックアップをとります。

警告： 表領域のバックアップは必ず必要です。バックアップをとらないと、表領域を失うことになる場合もあります。たとえば、データベースの最後のバックアップからのアーカイブ・ログが回復した表領域に論理的にリンクしていないために、メディア障害が発生した場合に表領域が消失するおそれがあります。TSPITR の前に作成されたバックアップから回復セット表領域を回復しようとする、失敗します (エラー・メッセージ「ORA-1246, 表領域: %s の TSPITR を使用してファイルを回復します」が戻されます)。

パーティション表の部分的な TSPITR の実行

ここでは、範囲が変更も拡張もされていないパーティション表の部分的な TSPITR の実行方法について説明します。

- ステップ 1: 回復する各パーティション用にプライマリ・データベースに表を作成する
- ステップ 2: 回復するパーティションの索引を削除する
- ステップ 3: パーティションをスタンドアロン表と入れ替える
- ステップ 4: 回復セット表領域をオフラインにする
- ステップ 5: クローン・データベースに表を作成する

- ステップ 6: 回復するパーティションの索引を削除する
- ステップ 7: パーティションをスタンダアロン表と入れ替える
- ステップ 8: クローン・データベースをエクスポートする
- ステップ 9: 回復セットのデータ・ファイルをプライマリ・データベースにコピーする
- ステップ 10: プライマリ・データベースにインポートする
- ステップ 11: 回復セットの表領域をオンラインにする
- ステップ 12: パーティションをスタンダアロン表と入れ替える
- ステップ 13: プライマリ・データベースの回復した表領域のバックアップをとる

注意： 範囲が拡張されたパーティションを回復する以外に、削除したパーティションを回復しなければならない場合がよくあります。このような場合には、13-21 ページの「パーティションを削除した場合のパーティション表の TSPITR の実行」を参照してください。

ステップ 1: 回復する各パーティション用にプライマリ・データベースに表を作成する

プライマリ・データベース上に、回復する各パーティション用の表を作成します。この表には、回復するパーティション表とまったく同じ列名と列データ型が存在していなくてはなりません。次の文で表を作成できます。

```
CREATE TABLE <new table> AS SELECT * FROM <partitioned table> where 1=2;
```

作成した表は、各回復セットのパーティションとの入替えに使われます（ステップ 3 を参照）。

ステップ 2: 回復するパーティションの索引を削除する

回復するパーティションの索引を削除するか、もしくは回復するパーティションにある索引と同一の非パーティション索引を（ステップ 1 で作成した表に）作成します。回復するパーティションの索引を削除した場合は、クローン・データベース上の対応する索引も削除する必要があります（ステップ 6 を参照）。また、TSPITR が完了した後に索引を再作成しなくてはなりません。

ステップ 3: パーティションをスタンドアロン表と入れ替える

次のコマンドを使って、回復セット内の各パーティションを、対応付けられたスタンドアロン表（ステップ 1 で作成した表）と入れ替えます。

```
ALTER TABLE <table name> EXCHANGE PARTITION <partition name> WITH TABLE <table name>;
```

ステップ 4: 回復セット表領域をオフラインにする

プライマリ・データベースで、次の文を発行して各回復セットの表領域をオフラインにします。

```
ALTER TABLESPACE <tablespace name> OFFLINE IMMEDIATE;
```

これにより、プライマリ・データベース上の回復セットの表領域は変更できなくなります。

ステップ 5: クローン・データベースに表を作成する

クローンを回復しリセットログをオープンした後で、回復するパーティション表と同一の列名および列データ型が存在する表を、各パーティションごとに作成します。作成された表は、後で各回復セットのパーティションとの入替えに使われます。

ステップ 6: 回復するパーティションの索引を削除する

回復するパーティションの索引を削除するか、もしくは（ステップ 1 で作成した表上に）回復するパーティションにある索引と同一の非パーティション索引を作成します。

ステップ 7: パーティションをスタンドアロン表と入れ替える

クローン・データベースの回復セット内の各パーティションについて、次の文を発行してパーティションをスタンドアロン表（ステップ 5 で作成した表）と入れ替えます。

```
ALTER TABLE <partitioned_table_name> EXCHANGE PARTITION <partition_name> WITH TABLE  
<table_name>;
```

ステップ 8: クローン・データベースをエクスポートする

次の文を発行して、回復セットの表領域用のクローン・データベースに対してエクスポートを実行します。

```
exp sys/<password>  
point_in_time_recover=yrecovery_tablespaces=<tablespace1>,<tablespace2>,<tablespaceN>
```

エクスポート・フェーズが失敗した場合（エラー・メッセージ「ORA-29308 ビュー TS_PITR_CHECK で障害が発生しました」が戻されます）は、TS_PITR_CHECK を再度問い合わせて問題を解決し、エクスポートを再実行してください。TSPITR のエクスポート・フェーズは、ユーザー SYS で実行する必要があります。そうしないと、エクスポートは失敗

します (エラー・メッセージ「ORA-29303 ユーザーが SYS でログインしていません」が戻されます)。エクスポートが正常に終了したら、クローン・データベースを停止します。

ステップ 9: 回復セットのデータ・ファイルをプライマリ・データベースにコピーする

プライマリ・データベース上の回復セットの表領域が **READ ONLY** になっている場合、それらを **OFFLINE** に変更する必要があります。クローン・データベースから回復セットのデータ・ファイルをプライマリ・データベースにコピーします。このとき、プライマリ・データベース上の補助セットのファイルを上書きしないように注意してください。

ステップ 10: プライマリ・データベースにインポートする

次のコマンドによって、回復セットのメタ・データをプライマリ・データベースにインポートします。

```
imp sys/<password> point_in_time_recover=true
```

このインポート処理によって、コピーされたファイルのファイル・ヘッダーが更新され、プライマリ・データベースに統合されます。

ステップ 11: 回復セットの表領域をオンラインにする

プライマリ・データベースで、次の文を発行して各回復セットの表領域をオンラインにします。

```
ALTER TABLESPACE <tablespace name> ONLINE;
```

ステップ 12: パーティションをスタンドアロン表と入れ替える

次の文を発行して、プライマリ・データベースの回復した各パーティションに対応するスタンドアロン表を元の場所に戻します。

```
ALTER TABLE <table name> EXCHANGE PARTITION <partition name> WITH TABLE <table name>;
```

対応付けられた索引を削除した場合、再作成する必要があります。

ステップ 13: プライマリ・データベースの回復した表領域のバックアップをとる

プライマリ・データベース上の回復した表領域のバックアップを作成します。バックアップを作成しておかないと、メディア障害が発生した場合にデータを失うことになります。

警告： 表領域のバックアップは必ず必要です。バックアップをとらないと、表領域を失うことになる場合もあります。たとえば、データベースの最後のバックアップからのアーカイブ・ログが回復した表領域に論理的にリンクしていないために、メディア障害が発生した場合に表領域が消失するおそれがあります。TSPITR の前に作成されたバックアップから回復セットの表領域を回復しようとする、失敗します (エラー・メッセージ「ORA-1246, 表領域: %s の TSPITR を使用してファイルを回復します」が戻されます)。

パーティションを削除した場合のパーティション表の TSPITR の実行

ここでは、パーティションを削除した場合にパーティション表の TSPITR を実行する方法を、次の順序で説明します。

- ステップ 1: 削除したパーティションの下限および上限の範囲を確認する
- ステップ 2: 一時表を作成する
- ステップ 3: パーティション表からレコードを削除する
- ステップ 4: 回復セットの表領域をオフラインにする
- ステップ 5: クローン・データベースに表を作成する
- ステップ 6: 回復するパーティションの索引を削除する
- ステップ 7: スタンドアロン表をパーティションと入れ替える
- ステップ 8: クローン・データベースをエクスポートする
- ステップ 9: 回復セットのデータ・ファイルをプライマリ・データベースにコピーする
- ステップ 10: プライマリ・データベースにインポートする
- ステップ 11: 回復セットの表領域をオンラインにする
- ステップ 12: スタンドアロン表をパーティション表に挿入する
- ステップ 13: プライマリ・データベースの回復した表領域のバックアップをとる

ステップ 1: 削除したパーティションの下限および上限の範囲を確認する

パーティションが削除されると、そのパーティションの上にあるパーティションの範囲が下方向に拡張されます。このため、パーティションを回復した後に、実際は削除されたパーティションにあったレコードがその上のパーティションに入っていることがあります。これを確認するには、プライマリ・データベースで次のコマンドを発行します。

```
SELECT * FROM <partitioned table> WHERE <relevant key> BETWEEN <low range of partition  
that was dropped> and <high range of partition that was dropped>;
```

ステップ 2: 一時表を作成する

レコードが戻った場合は、そのレコードを格納する一時表を作成します。そうすることで、(必要であれば) 後で回復したパーティションにそのレコードを挿入できます。

ステップ 3: パーティション表からレコードを削除する

一時表に格納したすべてのレコードをパーティション表から削除します。

ステップ 4: 回復セットの表領域をオフラインにする

プライマリ・データベースで、次の文を発行して各回復セットの表領域をオフラインにします。

```
ALTER TABLESPACE <tablespace name> OFFLINE IMMEDIATE;
```

ステップ 5: クローン・データベースに表を作成する

クローンを回復しリセットログをオープンした後で、回復するパーティション表と同一の列名および列データ型が存在する表を、各パーティションごとに作成します。作成した表は、後で各回復セットのパーティションとの交換に使われます。

ステップ 6: 回復するパーティションの索引を削除する

回復するパーティションの索引を削除するか、もしくは回復するパーティションに存在する索引と同一の非パーティション索引を作成します。

ステップ 7: スタンドアロン表をパーティションと入れ替える

クローン回復セット内の各パーティションについて、次の文を発行してパーティションをステップ 5 で作成したスタンドアロン表と入れ替えます。

```
ALTER TABLE <partitioned_table_name> EXCHANGE PARTITION <partition_name> WITH TABLE <table_name>;
```

ステップ 8: クローン・データベースをエクスポートする

次の文を発行して、回復セットの表領域用のクローン・データベースに対してエクスポートを実行します。

```
exp sys/<password>  
point_in_time_recover=yrecovery_tablespaces=<tablespace1>,<tablespace2>,<tablespaceN>
```

エクスポート・フェーズが失敗した場合 (エラー・メッセージ「ORA-29308 ビュー TS_PITR_CHECK で障害が発生しました」が戻されます) は、TS_PITR_CHECK を再度問い合わせて問題を解決し、エクスポートを再実行してください。TSPITR のエクスポート・フェーズは、ユーザー SYS で実行する必要があります。そうしないと、エクスポートは失敗

します (エラー・メッセージ「ORA-29303: ユーザーが SYS でログインしていません」が戻されます)。エクスポートが正常に終了したら、クローン・データベースを停止します。

ステップ 9: 回復セットのデータ・ファイルをプライマリ・データベースにコピーする

プライマリ・データベース上の回復セット表領域が **READ ONLY** になっている場合、それらを **OFFLINE** に変更する必要があります。クローン・データベースから回復セットのデータ・ファイルをプライマリ・データベースにコピーします。このとき、プライマリ・データベース上の補助セットのファイルを上書きしないように注意してください。

ステップ 10: プライマリ・データベースにインポートする

次のコマンドによって、回復セットのメタ・データをプライマリ・データベースにインポートします。

```
imp sys/<password> point_in_time_recover=true
```

このインポート処理によって、コピーされたファイルのファイル・ヘッダーが更新され、プライマリ・データベースに統合されます。

ステップ 11: 回復セットの表領域をオンラインにする

次の文を発行して、プライマリ・データベースの各回復セットの表領域をオンラインにします。

```
ALTER TABLESPACE <tablespace name> ONLINE;
```

ステップ 12: スタンドアロン表をパーティション表に挿入する

この時点で、スタンドアロン表をパーティション表に挿入しなくてはなりません。最初に、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、パーティション 2 の範囲内にあるキーは表からすでに削除されているので、パーティション 2 は空です。

スタンドアロン表をパーティションと入れ替えるために、次の文を発行します。

```
ALTER TABLE EXCHANGE PARTITION <partition name> WITH TABLE <table name>;
```

(必要であれば) ここで、ステップ 2 で保存したレコードを、回復したパーティションに挿入します。

注意： 削除したパーティションが表内の最終パーティションである場合は、次の文を使って追加できます。

```
ALTER TABLE ADD PARTITION;
```

ステップ 13: プライマリ・データベースの回復した表領域のバックアップをとる

プライマリ・データベース内の回復した表領域のバックアップを作成します。バックアップを作成しておかないと、メディア障害が発生した場合にデータを失うことになります。

警告： 表領域のバックアップは必ず必要です。バックアップをとらないと、表領域を失うことになる場合もあります。たとえば、データベースの最後のバックアップからのアーカイブ・ログが回復した表領域に論理的にリンクしていないために、メディア障害が発生した場合に表領域が消失するおそれがあります。TSPITR の前に作成されたバックアップから回復セット表領域を回復しようとすると、失敗します (エラー・メッセージ「ORA-1246- 表領域: %s の TSPITR を使用してファイルを回復します。」が戻されます)。

注意： 13-4 ページの「制限事項」で説明しているように、削除した表領域の回復には TSPITR を使用できません。このため、パーティションとともにパーティションの対応付けられた表領域も削除した場合、TSPITR ではそのパーティションを回復できません。通常のエクスポート / インポート回復を実行する必要があります。具体的な作業は次のとおりです。

- データベースのコピーを作成する
- そのコピーをロールフォワードする
- データベースをオープンする
- パーティションをスタンドアロン表と入れ替える
- スタンドアロン表の表レベルでのエクスポートを実行する
- 表をプライマリ・データベースにインポートし、次の文を使ってその表をパーティション表に挿入する。

```
ALTER TABLE SPLIT PARTITION or ALTER TABLE ADD  
PARTITION;
```

パーティションを分割した場合のパーティション表の TSPITR の実行

ここでは、パーティションを分割した場合のパーティション表の回復方法を、次の順序で説明します。

- ステップ 1: プライマリ・データベースの 2 つのパーティションのうち下位の方を削除する
- ステップ 2: 回復するパーティションの索引を削除する
- ステップ 3: スタンドアロン表とパーティションを入れ替える
- ステップ 4: 回復セットの表領域をオフラインにする
- ステップ 5: クローン・データベースに表を作成する
- ステップ 6: 回復するパーティションの索引を削除する
- ステップ 7: スタンドアロン表とパーティションを入れ替える
- ステップ 8: クローン・データベースをエクスポートする
- ステップ 9: 回復セットのデータ・ファイルをプライマリ・データベースにコピーする
- ステップ 10: プライマリ・データベースにインポートする
- ステップ 11: 回復セットの表領域をオンラインにする
- ステップ 12: パーティションをスタンドアロン表と入れ替える
- ステップ 13: プライマリ・データベースの回復した表領域のバックアップをとる

ステップ 1: プライマリ・データベースの 2 つのパーティションのうち下位の方を削除する

プライマリ・データベースにおいて、2 つに分割されたパーティションを回復する場合、回復する各パーティションごとに 2 つの分割パーティションのうち下位の方を削除します。これによって、上位の分割パーティションが範囲を下方に拡張されます（つまり、上位の分割パーティションは、分割する前のパーティションと同じ範囲を確保します）。たとえば、P1 が P1A と P1B という 2 つのパーティションに分割された場合、P1B を削除する必要があります。これにより P1A パーティションは P1 と同じ範囲を確保します。

範囲が分割されたパーティションを回復する場合、そのパーティションごとに次の文を発行することにより、回復するパーティション表と同一の列名と列データ型がある表を作成します。

```
CREATE TABLE <new table> AS SELECT * FROM <partitioned table> where 1=2;
```

作成された表は、ステップ 3 で各回復セットのパーティションとの入替えに使用されます。

ステップ 2: 回復するパーティションの索引を削除する

回復するパーティションの索引を削除するか、もしくは回復するパーティションにある索引と同一の非パーティション索引を、ステップ 1 で作成した表に作成します。回復するパーティションの索引を削除する場合、クローン・データベースの対応する索引も削除する必要があります（ステップ 6 を参照）。したがって、回復の完了後にそれらの索引を再作成しなければなりません。

ステップ 3: スタンドアロン表とパーティションを入れ替える

次のコマンドを使って、回復セット内の各パーティションを、対応付けられたスタンドアロン表（ステップ 1 で作成した表）と入れ替えます。

```
ALTER TABLE <table name> EXCHANGE PARTITION <partition name> WITH TABLE <table name>;
```

ステップ 4: 回復セットの表領域をオフラインにする

プライマリ・データベースで、次の文を発行して各回復セットの表領域をオフラインにします。

```
ALTER TABLESPACE <tablespace name> OFFLINE IMMEDIATE;
```

これにより、プライマリ・データベース上の回復セットの表領域は変更できなくなります。

ステップ 5: クローン・データベースに表を作成する

クローン・データベースでクローンを回復しリセットログをオープンした後で、回復する各パーティションについて、そのパーティション表と同一の列名と列データ型がある表を作成します。作成した表は、各回復セットのパーティションとの入替えに使われます（ステップ 7 を参照）。

ステップ 6: 回復するパーティションの索引を削除する

回復するパーティションの索引を削除するか、もしくは回復するパーティションにある索引と同一の非パーティション索引を、ステップ 1 で作成した表に作成します。

ステップ 7: スタンドアロン表とパーティションを入れ替える

クローン回復セット内の各パーティションについて、次の文を発行してパーティションをステップ 5 で作成したスタンドアロン表と入れ替えます。

```
ALTER TABLE <partitioned_table_name> EXCHANGE PARTITION <partition_name> WITH  
TABLE<table_name>;
```

ステップ 8: クローン・データベースをエクスポートする

次の文を発行して、回復セットの表領域用のクローン・データベースに対してエクスポートを実行します。

```
exp sys/<password>
point_in_time_recover=yrecovery_tablespace=<tablespace1>,<tablespace2>,<tablespaceN>
```

エクスポート・フェーズが失敗した場合（エラー・メッセージ「ORA-29308 ビュー TS_PITR_CHECK に障害が発生しました」が戻されます）は、TS_PITR_CHECK を再度問い合わせて問題を解決し、エクスポートを再実行してください。TSPITR のエクスポート・フェーズは、ユーザー **SYS** で実行する必要があります。そうしないと、エクスポートは失敗します（エラー・メッセージ「ORA-29303: ユーザーが **SYS** でログインしていません」が戻されます）。エクスポートが正常に終了したら、クローン・データベースを停止します。

ステップ 9: 回復セットのデータ・ファイルをプライマリ・データベースにコピーする

プライマリ・データベースの回復セット表領域が **READ ONLY** になっている場合、**OFFLINE** に変更する必要があります。クローン・データベースから回復セットのデータ・ファイルをプライマリ・データベースにコピーします。このとき、プライマリ・データベース上の補助セットのファイルを上書きしないように注意してください。

ステップ 10: プライマリ・データベースにインポートする

次のコマンドによって、回復セットのメタ・データをプライマリ・データベースにインポートします。

```
imp sys/<password> point_in_time_recover=true
```

このインポート処理によって、コピーされたファイルのファイル・ヘッダーが更新され、プライマリ・データベースに統合されます。

ステップ 11: 回復セットの表領域をオンラインにする

次の文を発行して、プライマリ・データベースの各回復セットの表領域をオンラインにします。

```
ALTER TABLESPACE <tablespace name> ONLINE;
```

ステップ 12: パーティションをスタンドアロン表と入れ替える

プライマリ・データベースの回復した各パーティションについて次の文を発行して、対応するスタンドアロン表を元の場所に戻します。

```
ALTER TABLE <table name> EXCHANGE PARTITION <partition name> WITH TABLE <table name>;
```

対応付けられた索引を削除した場合、再作成する必要があります。

ステップ 13: プライマリ・データベースの回復した表領域のバックアップをとる

プライマリ・データベース内の回復した表領域のバックアップを作成します。バックアップを作成しておかないと、メディア障害が発生した場合にデータが失うことになります。

関連項目：詳細は、13-21 ページの「パーティションを削除した場合のパーティション表の TSPITR の実行」を参照してください。

警告： 表領域のバックアップは必ず必要です。バックアップをとらないと、表領域を失うことになる場合もあります。たとえば、データベースの最後のバックアップからのアーカイブ・ログが回復した表領域に論理的にリンクしていないために、メディア障害が発生した場合に表領域が消失するおそれがあります。TSPITR の前に作成されたバックアップから回復セットの表領域を回復しようとする、失敗します (エラー・メッセージ「ORA-1246, 表領域: %s の TSPITR を使用してファイルを回復します」が戻されます)。

TSPITR のチューニングに関する考慮事項

ここでは、TSPITR のチューニングに関連する事柄を、次の順序で説明します。

- 回復セットの位置に関する考慮事項
- バックアップ制御ファイルに関する考慮事項

回復セットの位置に関する考慮事項

領域に余裕がない場合は、回復セットを「本来の場所」に回復できます。つまり、プライマリ・データベース上の対応するファイルを上書きする要領で回復できます。この方法は最良の方法ではありません。Oracle がお薦めする最良の方法は、ファイルを別の場所に格納しておき、TSPITR のインポート・フェーズが完了する前にコピーするという方法です (13-15 ページの「ステップ 11: プライマリ・データベースにインポートする」を参照)。

上記の 2 つの方法の長所と短所について、説明します。

別の場所へ回復することの長所と短所

別の場所へ回復することの長所は、基本的に高い可用性と柔軟性です。回復セットをプライマリ・データベースと統合する前の時点で回復を中止した場合、プライマリ・データベース上に回復セットのファイルを復元して通常の方法でそのファイルを回復する必要はありません。また、回復がクローンで行われている間に、プライマリ・データベースの回復セットの表領域にアクセスできます。たとえば、回復セットの表領域内にある破損していないデータのサブセットにアクセスする場合があります (13-12 ページの「ステップ 3: TSPITR 用にプライマリ・データベースを準備する」を参照)。このような場合、プライマリ・データベース上の回復セットの表領域を READ ONLY モードに変更することによって、この回復セットの

表領域を変更はできなくても問合せはできるようになります。回復セットのファイルをプライマリ・データベース上の元の場所で回復した場合、これは不可能です。

別の場所に回復することの短所は、クローン・データベース用としてより多くの領域が必要になることです。

元の場所に回復することの長所と短所

元の場所で回復することの長所は、回復セットのファイルに必要な領域を節約できることです。クローンの回復が完了した後で、プライマリ・データベースに回復セットのファイルをコピーする必要がありません。

回復セットをプライマリ・データベースに統合する前の時点で回復を中止した場合 (13-15 ページの「ステップ 11: プライマリ・データベースにインポートする」を参照)、プライマリ・データベースの上書きされた回復セット・ファイルをバックアップから復元し、通常の方法で回復する必要があります。この間データは使用できません。回復処理中は、回復セットの表領域内の破損していないデータを問い合わせることはできません。

バックアップ制御ファイルに関する考慮事項

プライマリにクローンを統合する前にそのクローンで回復が実行されていない場合、エラー・メッセージ「ORA-01152 ファイル %s は十分に古いバックアップから復元されていません」が戻されます。たとえば、時刻 A にバックアップをとり、時刻 B でデータベースの特定の表領域について TSPITR を実行し、その表領域を時刻 A の状態に戻す必要がある場合、実際には、回復が行われていない状態でクローン・データベースのリセットログをオープンしてしまいます。クローンを回復するためのコマンドは次のとおりです。

```
SVRMGRL> recover database using backup controlfile until cancel;  
SVRMGRL> cancel;  
SVRMGRL> open database resetlogs;
```

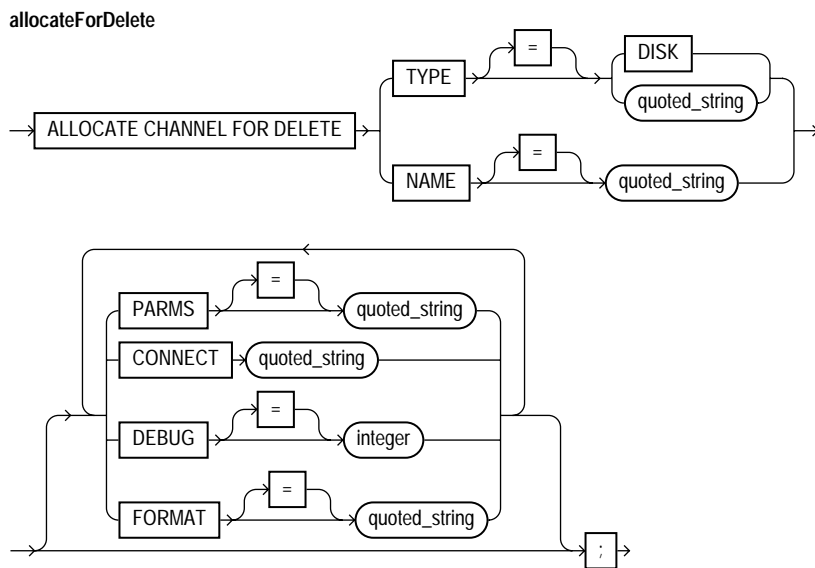
この時点ではログは適用されていませんが、データベースをオープンしようとしています。ところが、制御ファイルに対するチェックポイントは Oracle8 に保管されているため、クローン・データベースとスタンバイ・データベースでは、制御ファイル以外のファイルのバックアップを作成した後で、制御ファイルのバックアップを作成する必要があります。そうしないと、オープン時にエラー・メッセージ「ORA-01152 ファイル 1 は十分に古いバックアップから復元されていません」が戻されます。これは、ファイル 1 自体が（データベースの残りの部分と同期をとっているため）新しすぎるからではなく、制御ファイルよりも新しいからです。

注意： 問題箇所がなく一貫性のあるバックアップと古いバックアップ制御ファイルが使用されている場合、リセットログは正規のデータベースでも利用できます。それ以外の場合は、リセットログは既存のバックアップ・スクリプトと互換性がありません。

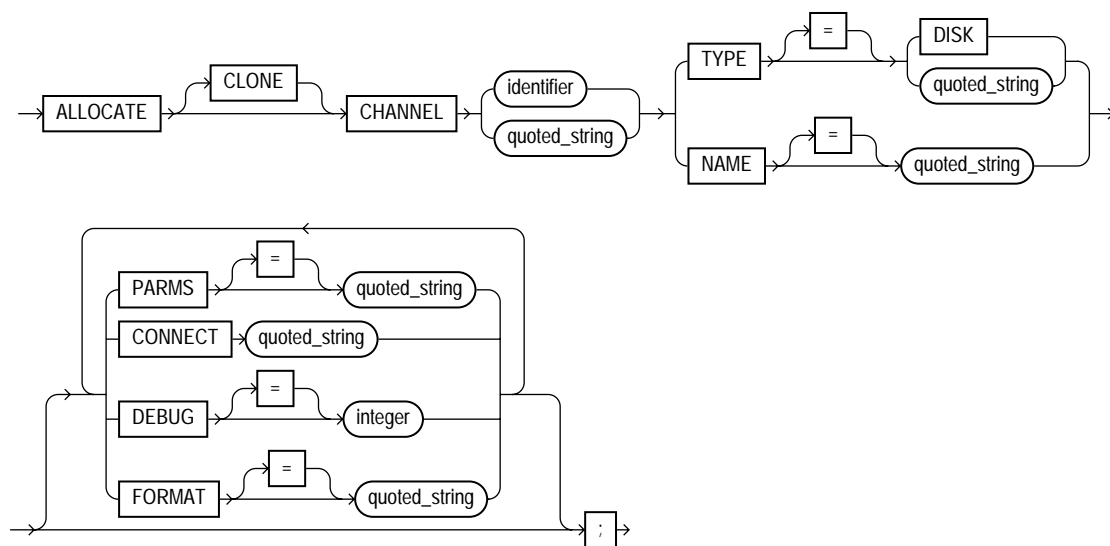
Recovery Manager のコマンド構文

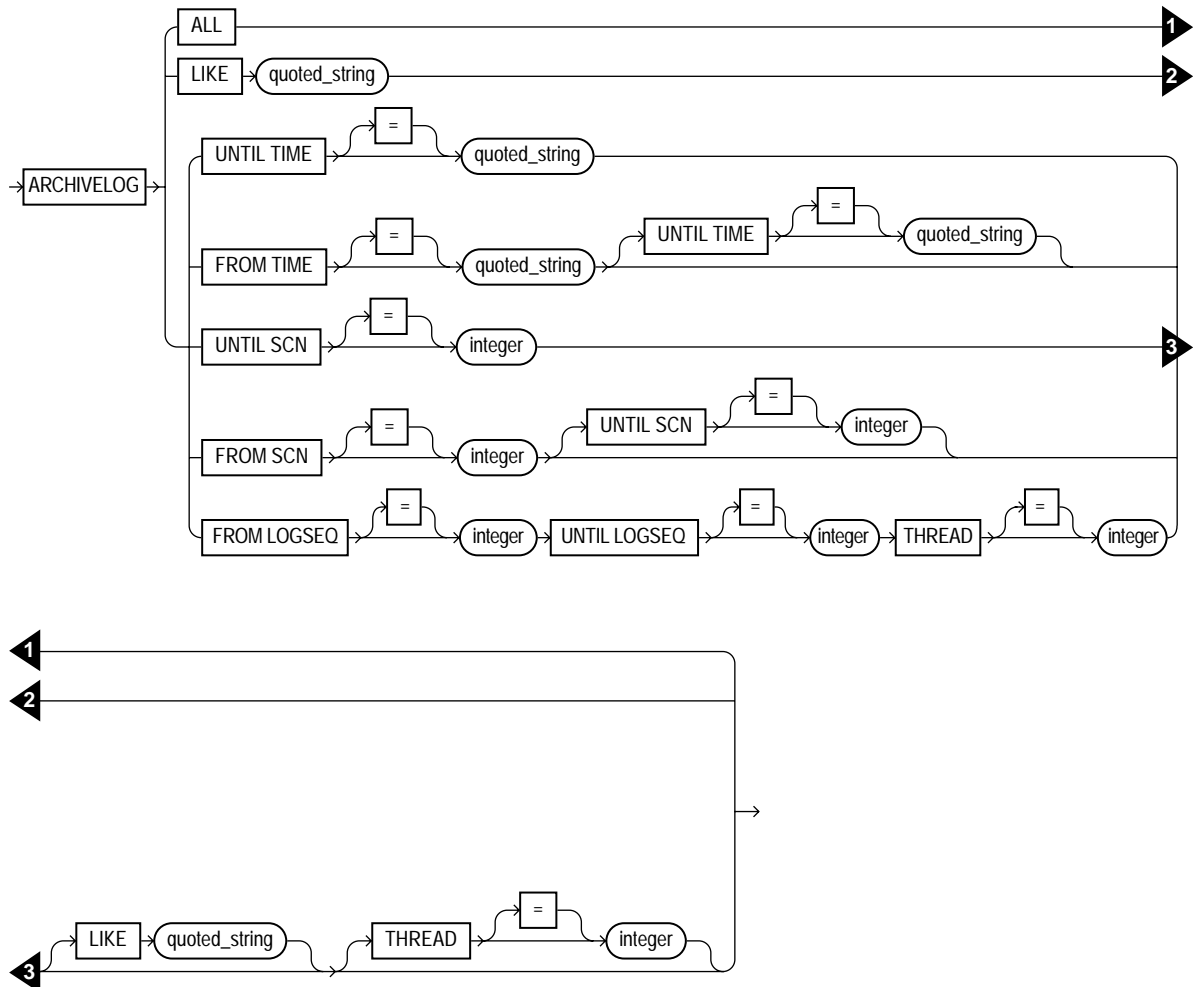
この付録では、Recovery Manager のコマンドを記載します。

allocateForDelete



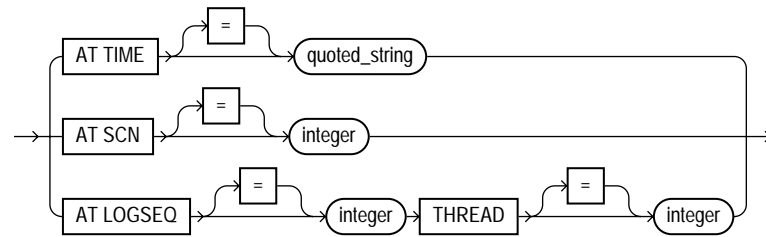
allocate



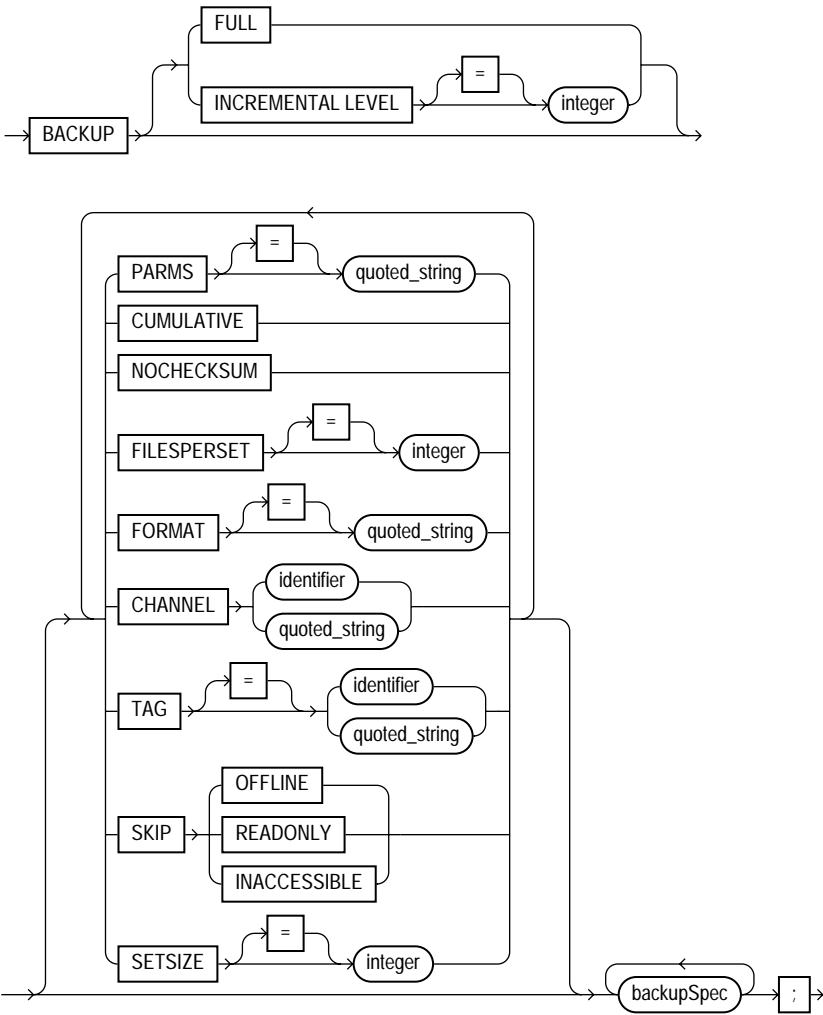


atClause

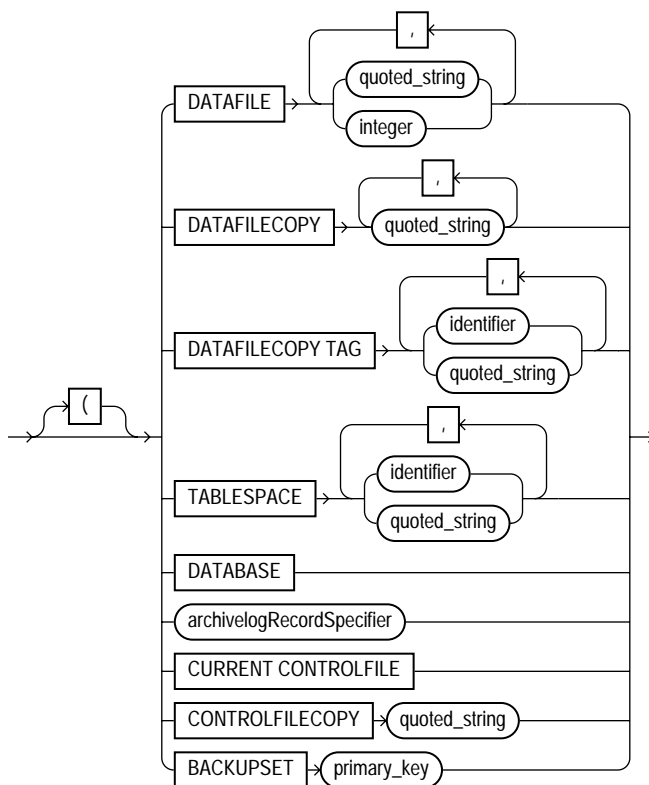
atClause

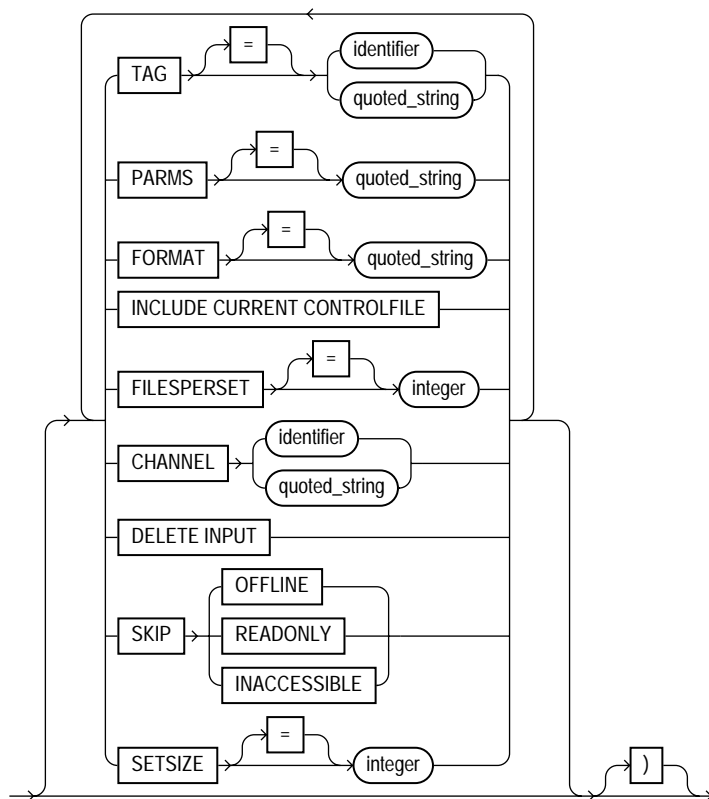


backup

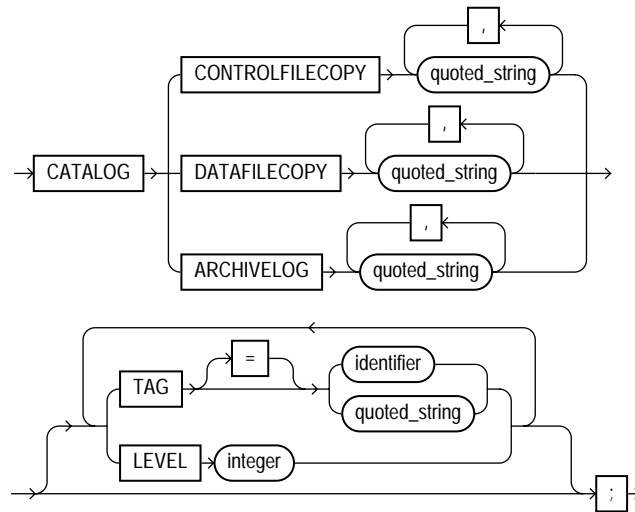


backupSpec

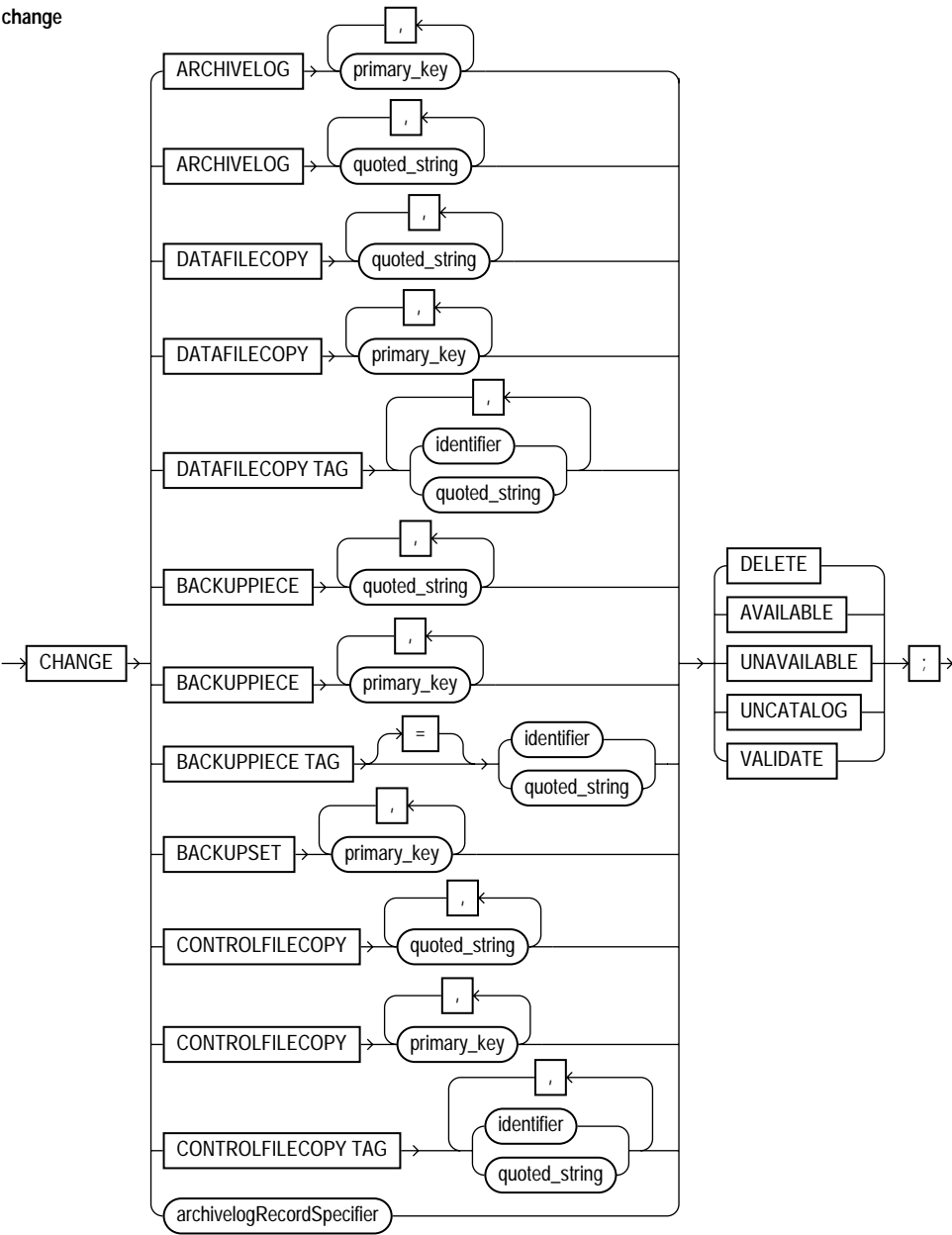




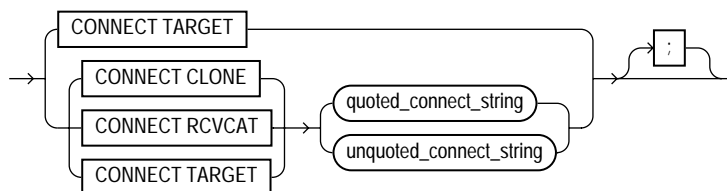
catalog

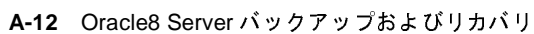


change

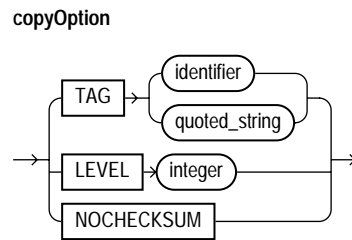


connect

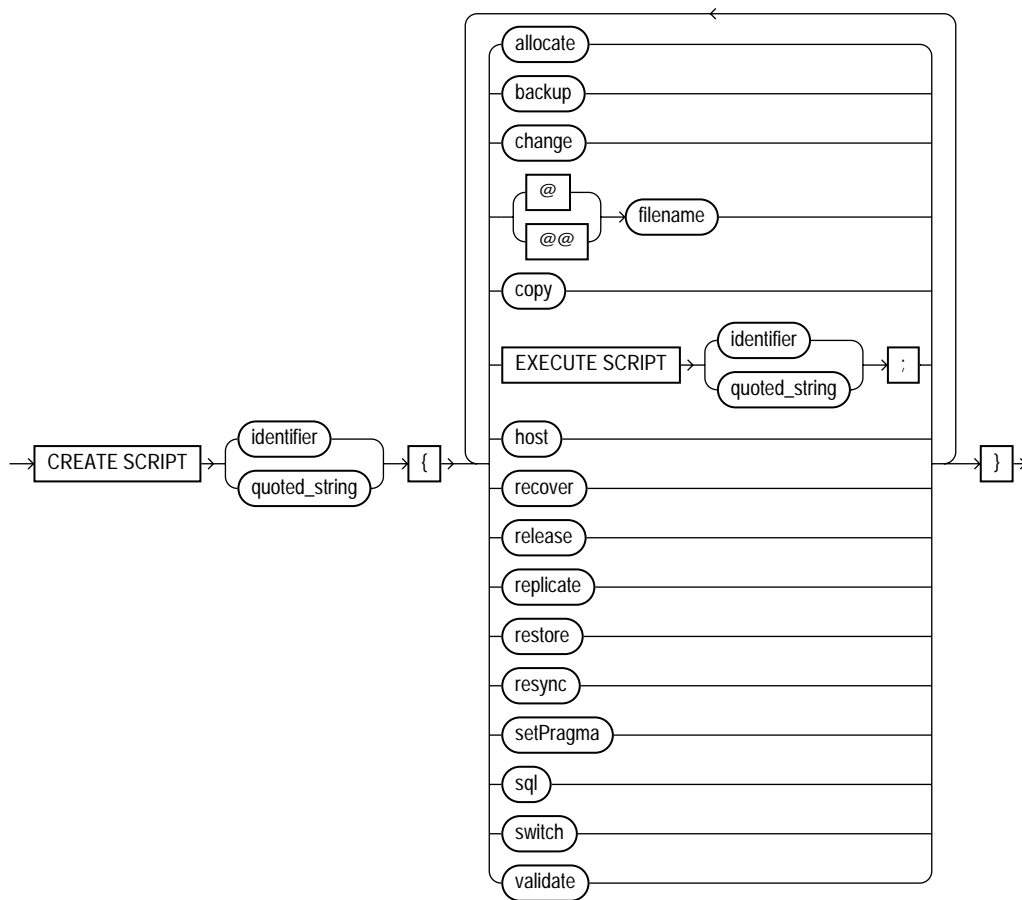




copyOption

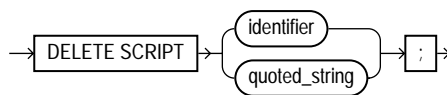


createScript

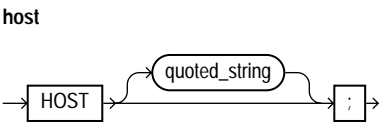


deleteScript

deleteScript

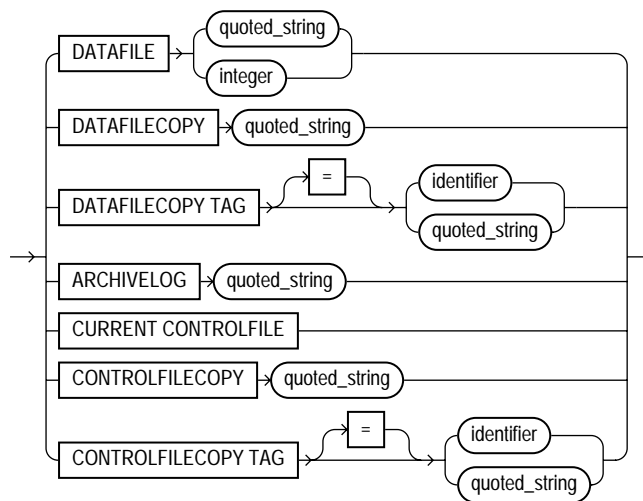


host

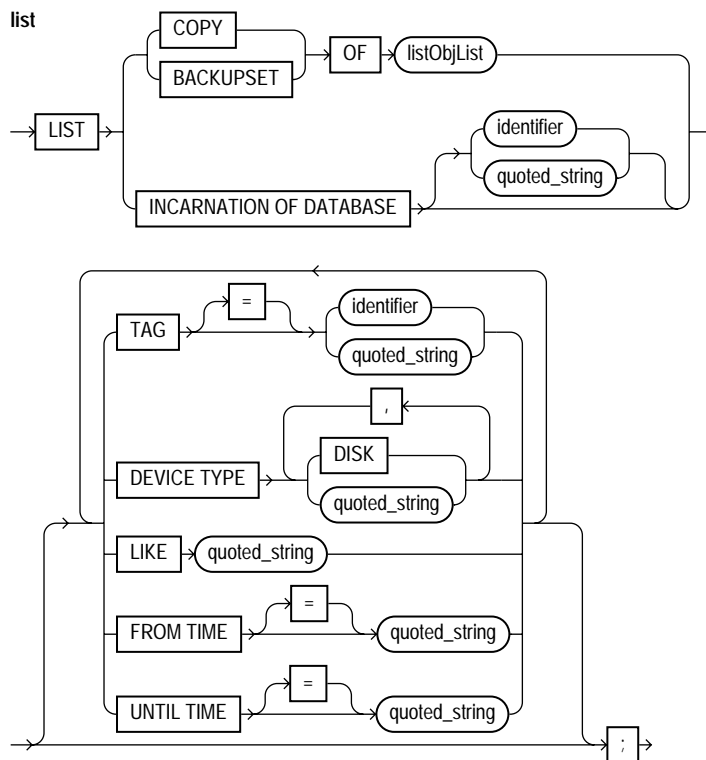


inputfile

inputfile

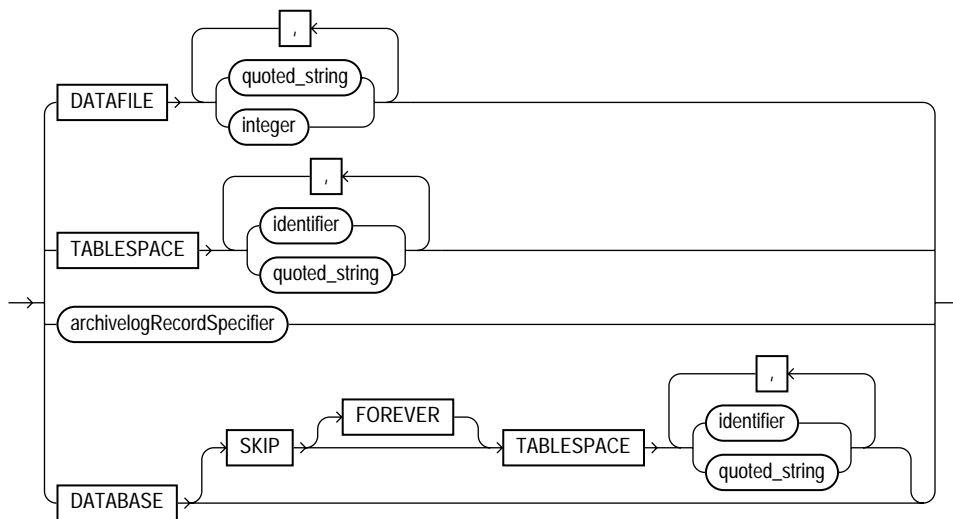


list

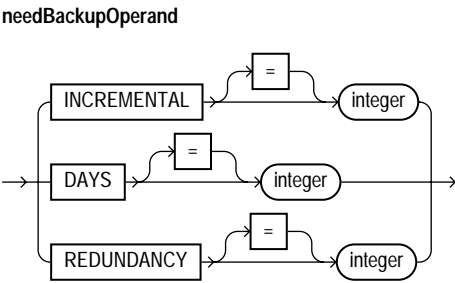


listObjList

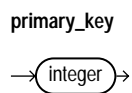
listObjList



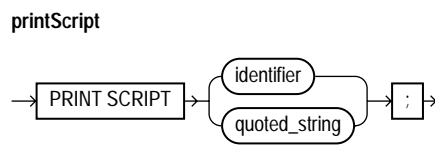
needBackupOperand



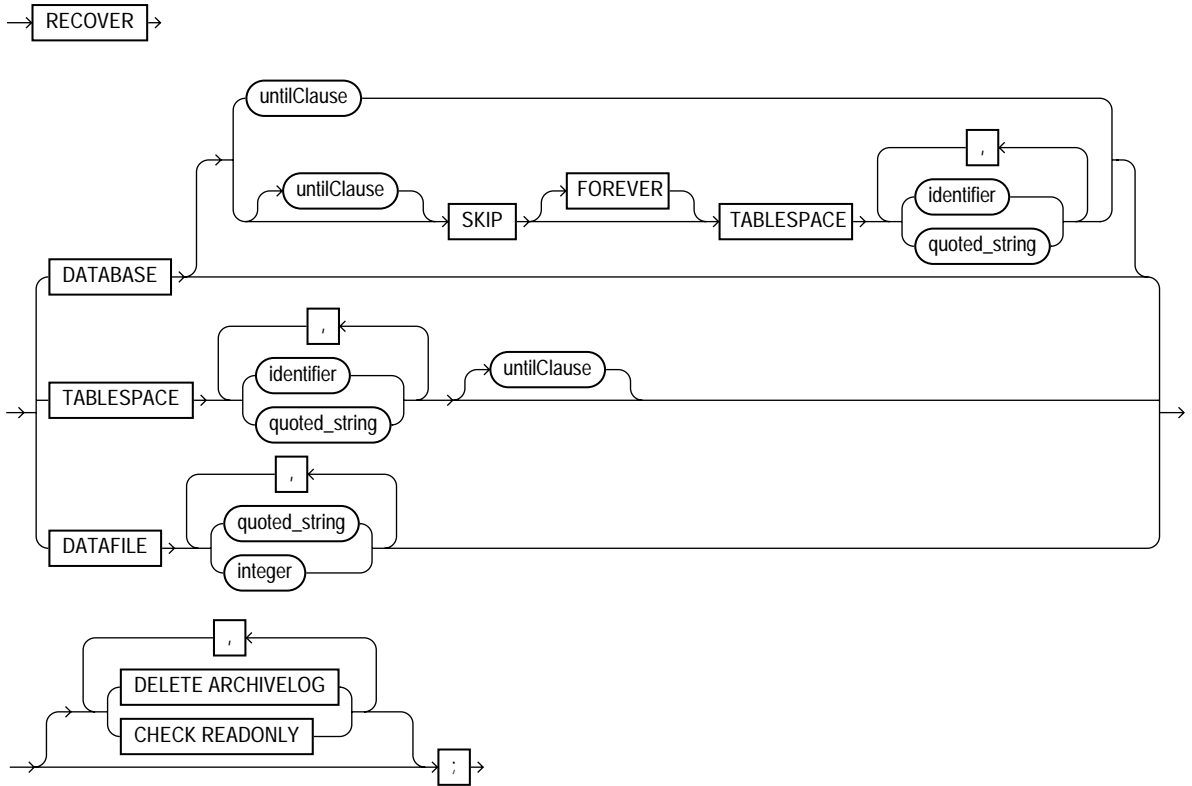
primary_key



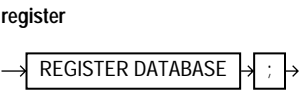
printScript



recover

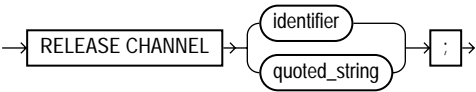


register

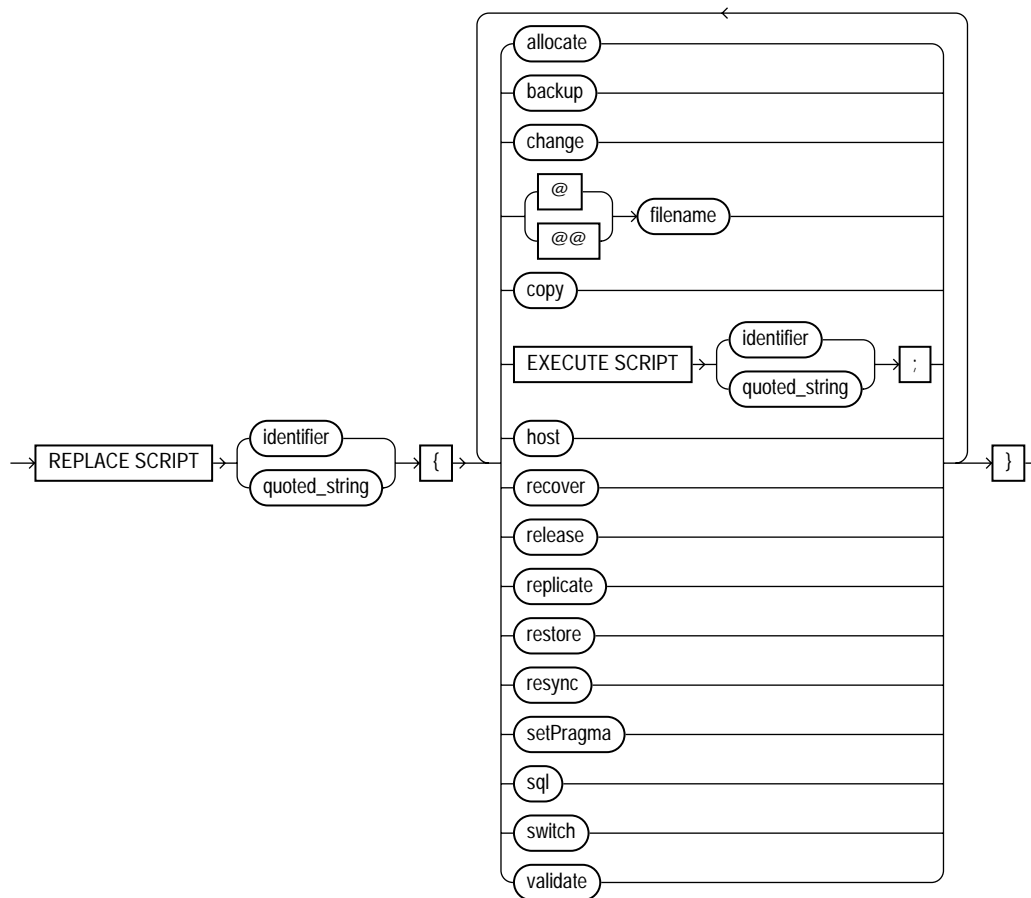


release

release



replaceScript

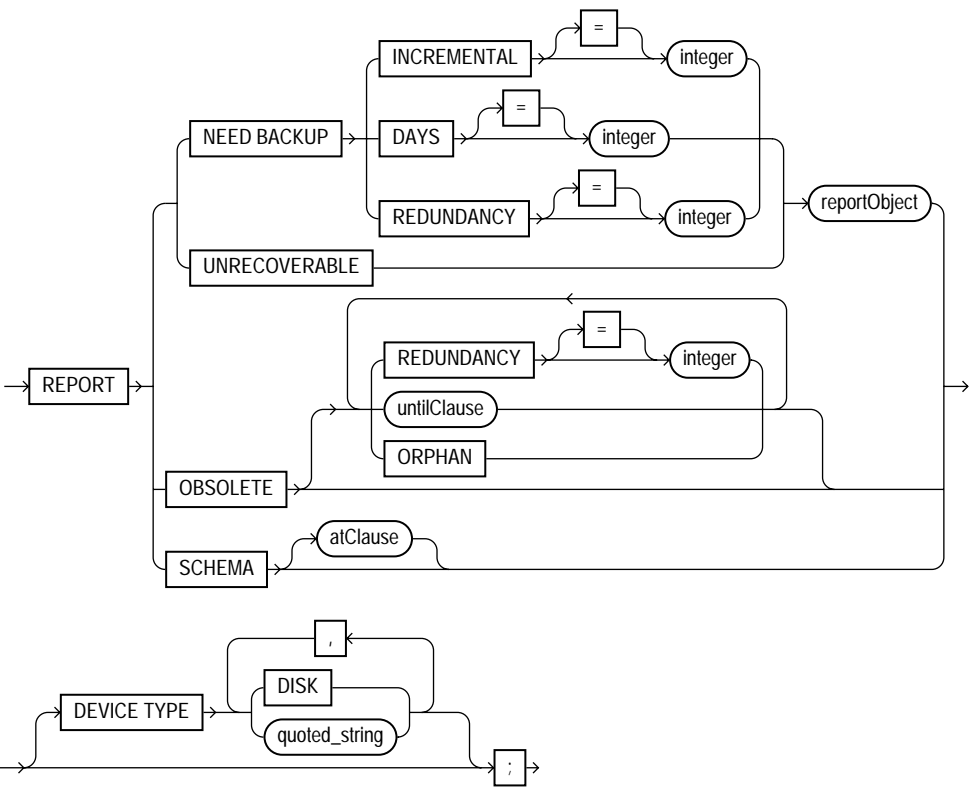


replicate

replicate

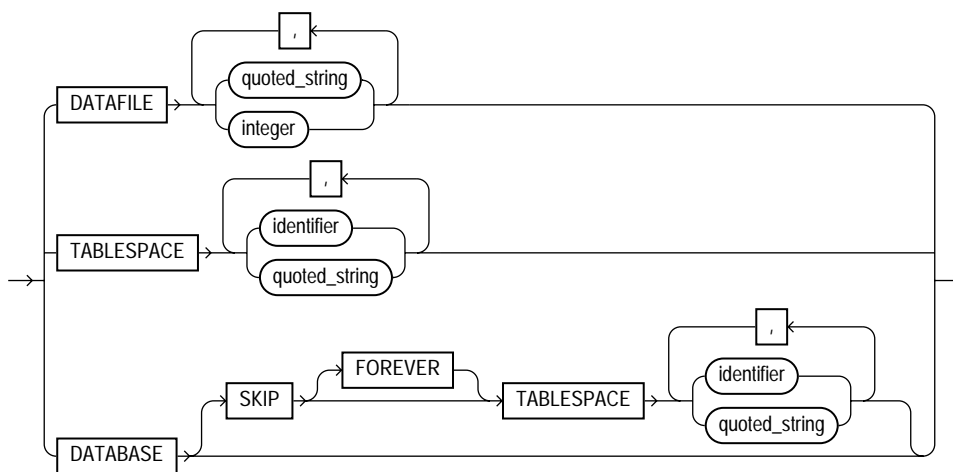


report

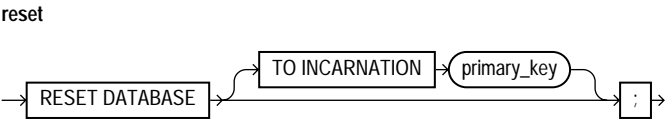


reportObject

reportObject

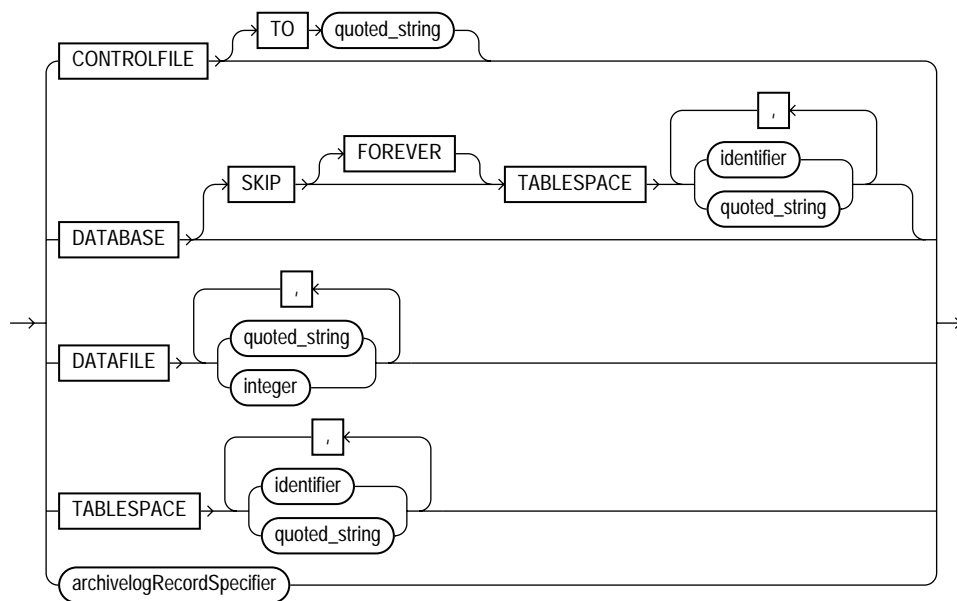


reset

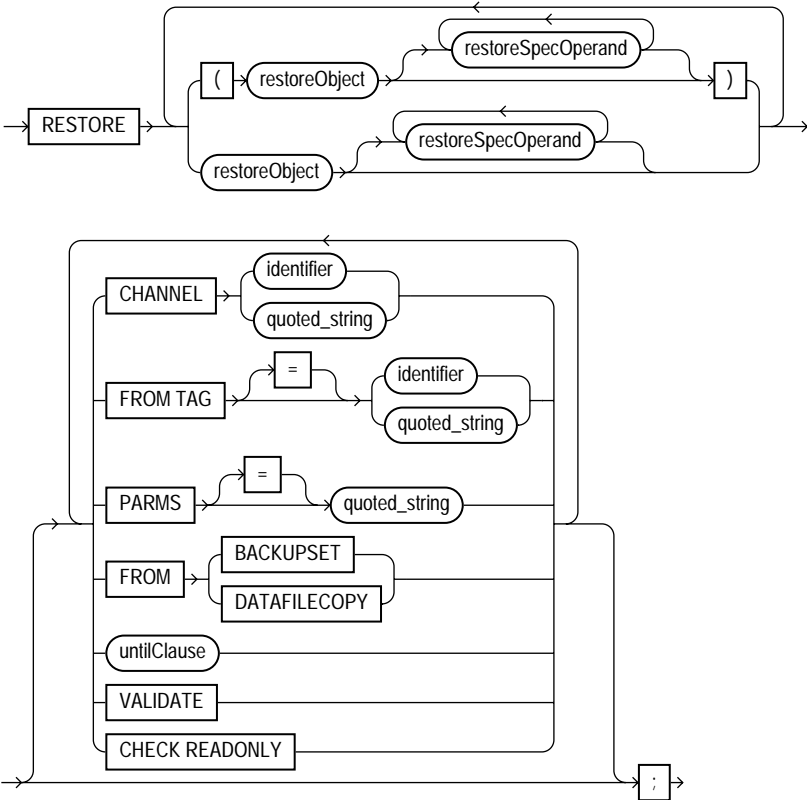


restoreObject

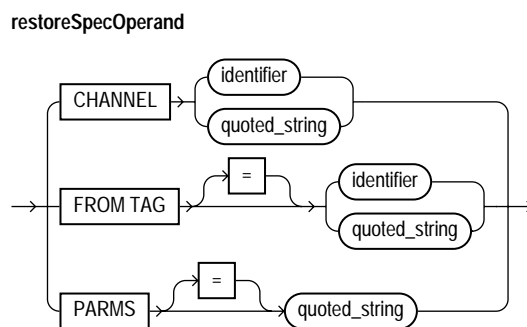
restoreObject



restore



restoreSpecOperand

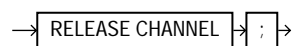


resync

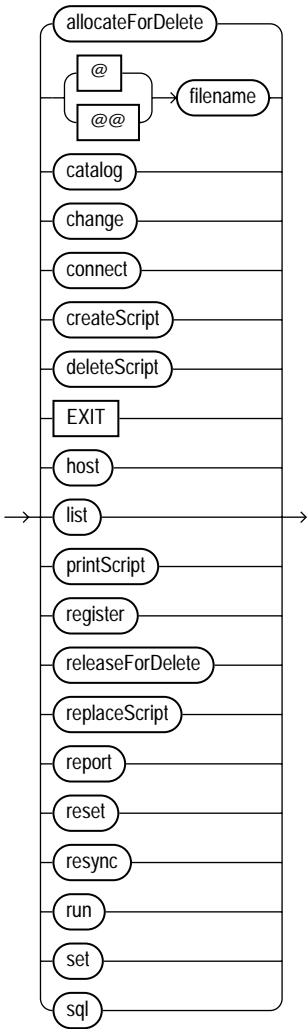


releaseForDelete

releaseForDelete

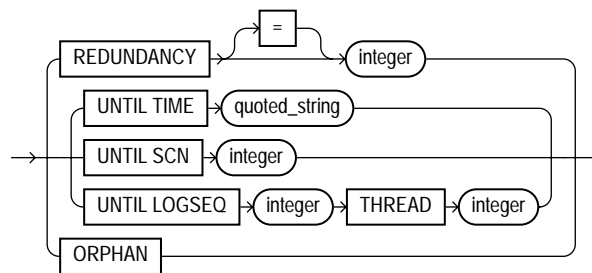


rmanCmd

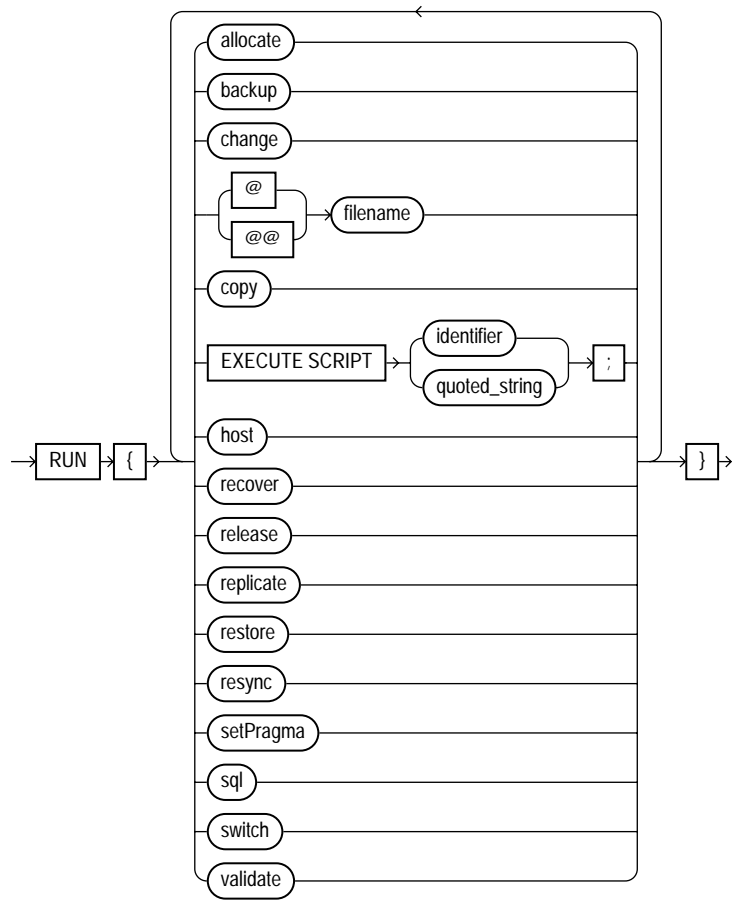


reportObsoleteOperand

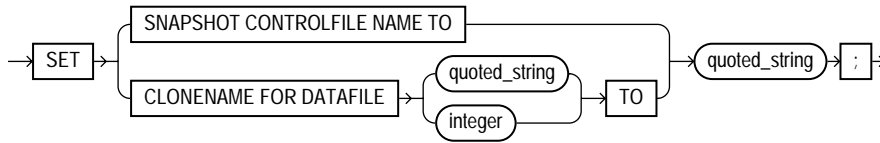
reportObsoleteOperand



run

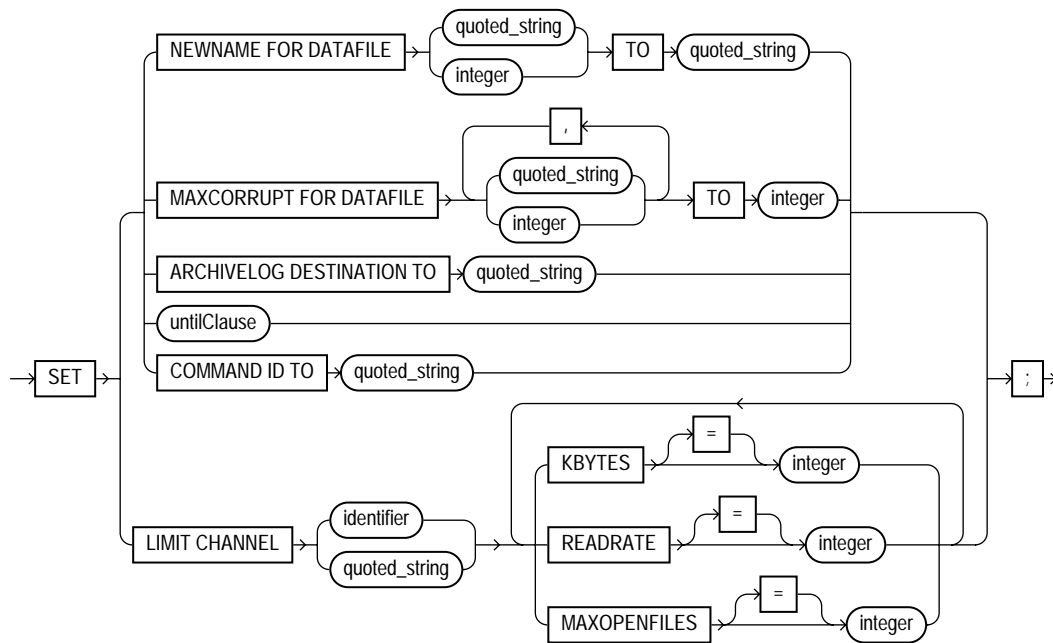


set

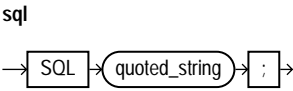


setPragma

setPragma

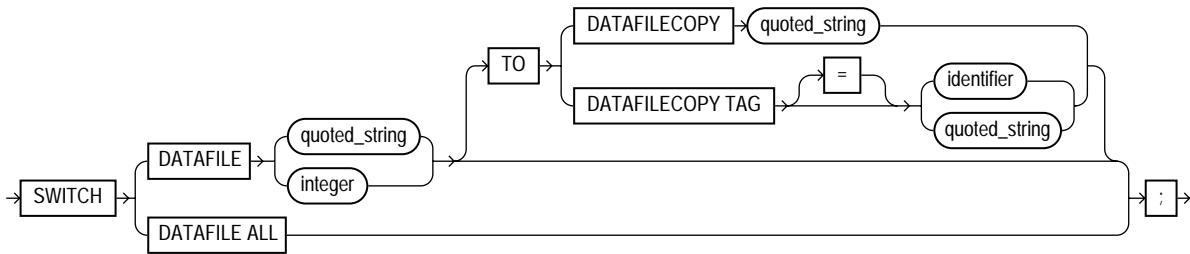


sql



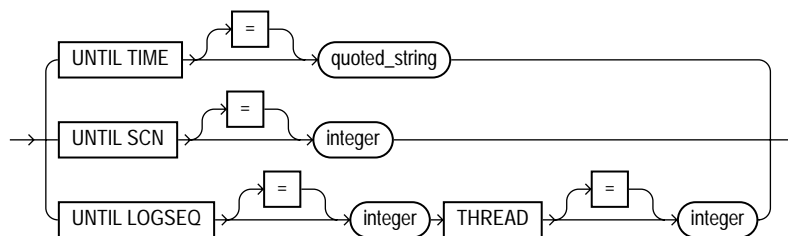
switch

switch

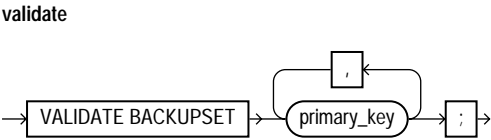


untilClause

untilClause



validate



用語集

ATL(自動テープ・ライブラリ)

1 つ以上のテープ・ドライブおよびロボット・アーム、テープ・シェルフが装備されている装置。ATL は、オペレータの操作なしにテープをシェルフからテープ・ドライブにロードおよびアンロードできます。さらに洗練されたテープ・ライブラリは、各テープを識別できます。たとえば、ロボット・アームには、バーコード読取り機が内蔵されている場合があります、それを使って各テープのバーコードを走査して識別できます。テープ **silo** とシノニムです。

補助セット

表領域の **Point-in-Time** 回復に関する用語。補助セットは、回復セットにはないファイルの集合です。**Point-in-Time** 回復セットを正しく回復するために、補助セットはクローン・データベースでも復元する必要があります。補助セットには、次のものが含まれます。

- バックアップ制御ファイル
- システム表領域
- ロールバック・セグメントが入っているデータ・ファイル
- 一時表領域 (オプション)

エクスポートするには、ソート操作用の小さい領域が必要です。また、一時表領域のコピーが補助セットに含まれていない場合は、複製データベースが始動した後に、一時表領域を新たに作成するか、または、システム表領域ファイルについて自動拡張を **ON** に設定することによって、ソート領域を確保することが必要です。「回復セット」も参照。

バックアップ

バックアップをとることによって作られる、データの代理コピー。Oracle では、バックアップは、次の方法で作成できます。

- データのファイル・コピーを (エクスポートを介して) とる
- **Recovery Manager** を使って、バックアップ (データ・ファイルのコピーまたはバックアップ・セット) を作成する

- オペレーティング・システム・ユーティリティ (cp、tar、dd など) を使用して、ディスクまたはテープにバックアップを作成する

データ・ファイル、制御ファイル、その他の Oracle データベース構造のイメージ・コピーをとること。

バックアップをとる: データの代理コピーを作成すること。元のデータが消失した場合、バックアップを使って、失われた情報を再構築できます。

backup – Recovery Manager のコマンド

Recovery Manager が認識するキーワード。Recovery Manager は、このキーワードによって、これからユーザーがデータのバックアップをとるので、そのデータをバックアップ・セット形式で書き出さなくてはならないこと認識します (「コピー」の (2) と比較してください)。

バックアップ・セット

バックアップ・セットは、(1 つ以上の) Oracle ファイルのバックアップであり、ファイルは、まとめて多重化されています。多重化には次のような利点があるので、パフォーマンスが上がります。

- 1 つのデータ・ファイルに読み込みが集中しない
- テープ・ドライブの高速ストリーミングを維持する

バックアップ・セットのファイルは、restore コマンドを使って抽出する必要があります。バックアップ・セットには、次の 2 つのタイプがあります。

- データファイルのバックアップ・セット

データ・ファイルまたは制御ファイルのバックアップをとったときに作成されます。このタイプのバックアップには、使用されているデータ・ファイル・ブロックだけが含まれます。未使用のブロックは省かれます (「未使用ブロックと圧縮」を参照)。

- アーカイブ・ログのバックアップ・セット

アーカイブ・ログのバックアップをとったときに作成されます。

バックアップ・セットは、バックアップ・ピースと呼ばれる 1 つ以上の物理ファイルで構成されています。

バックアップ・ピース

バックアップ・ピースは、1 つのバックアップ・セットだけに属します。通常、バックアップ・セットは、1 つのバックアップ・ピースだけで構成されます。Recovery Manager によって複数のバックアップ・ピースが作成されるのは、ユーザーが、setlimit コマンドでキーワード 'KBYTES' を指定して、ピース・サイズを明示的に制限した場合だけです。backup コマンドで KBYTES を指定するのは、バックアップの書込み先の記憶領域またはメディア・マネージャが、'KBYTES' より大きいサイズのファイルの書込みをサポートできない場合だけです。

バックアップデータベース全体

データベース全体のバックアップとは、制御ファイルおよびデータベースに属するすべてのデータ・ファイルのバックアップです。データベース全体のバックアップは、いつでも作成できます（データベースのマウント時、オープン時、クローズ時など）。

ブロックー Oracle

Oracle データベースの最小単位。ブロックのサイズは、データベースの作成時にパラメータ `DB_BLOCK_SIZE` に指定した値によって決まります。

変更ベクトル

単一データ・ブロックに対する単一の変更。変更ベクトルは、変更の最小単位です。「**REDO** レコード」も参照。

チャネル

チャネルは、**Recovery Manager** のリソース割当てです。割り当てられた各チャネルは、新しい Oracle サーバー・プロセスを開始します。このプロセスは、バックアップの復元および回復作業を行います。割り当てるチャネルの数が多いほど、バックアップおよび復元、回復の並列度が高まります。指定されたチャネルのタイプによって、ディスクに対する読取りと書込みを、Oracle サーバー・プロセスが行うか、またはメディア管理インタフェースなどを介して行うかが決まります。

- チャネルのタイプがディスクの場合、サーバー・プロセスはディスクに対してバックアップの読取りと書込みを行います。
- チャネルのタイプが '**SBT_TAPE**' の場合、サーバー・プロセスは、メディア・マネージャに介してバックアップの読取りと書込みを行います。

チャネルは、指定されたタイプに関係なく、ディスクに対しては、読取りと書込みを常時実行できます。

チェックポイント

チェックポイントが発生すると、**SGA** 内の変更されたデータベース・バッファはすべて、**DBWR** によってデータ・ファイルに書き込まれます。チェックポイントが完了すると、データ・ファイルと制御ファイルは、チェックポイントが正常に完了したことを反映するように変更されます。

チェックサム

Oracle ブロックに格納されている、Oracle 内部の計算済み数値。Oracle サーバーは、この数値を利用してブロックの一貫性を検査します。

クローン・データベース

新しい場所に復元され、新しいインスタンス名で起動するデータベースのサブセット。クローン・データベースの使用目的は、通常は表領域の **Point-in-Time** 回復です。クローン・データベースは、回復セットと補助セットで構成されます。クローン・データベースは、通常のデータベースのコピーとはさまざまな点で本質的に異なります。

クローズ状態のバックアップ

データベースがクローズしている時に作成される、1つ以上のデータベース・ファイルのバックアップ。通常、クローズ状態のバックアップは、データベース全体のバックアップです。データベースが正常にクローズするということは、バックアップ内のすべてのファイルに一貫性があるということです（詳細は、「データベースのバックアップ—一貫性のある」を参照）。異常終了停止またはインスタンス異常終了によってデータベースがクローズされた場合、バックアップを回復してからでないと、一貫性は持てません。

クローズ状態のデータベース

「データベース—クローズ状態」を参照。

コールド・バックアップ

「クローズ状態のバックアップ」を参照。

完全回復

「回復—完全」を参照。

制御ファイル

データベース内のすべてのファイルの物理構造とタイムスタンプをメンテナンスする、データベースに対応付けられたファイル。制御ファイルは、データベースの使用中断えず更新されます。したがって、データベースがマウントされている間は、制御ファイルは常に書込み可能になっていなければなりません。

制御ファイル—バックアップ

バックアップ制御ファイルとは、制御ファイルのバックアップです。バックアップ制御ファイルは、通常、現行の制御ファイルのコピーがすべて破損した場合に復元されますが、ある種の **Point-in-Time** 回復の実行前に復元されることもあります。

制御ファイル—現行の

現行の制御ファイルは、ディスクにある制御ファイルです。この制御ファイルは、そのインカーネーション・データベース用の、一番最近変更されたものです。バックアップから復元された制御ファイルは、回復時に現行とは認識されません。

一貫性のあるデータベースのバックアップ

「データベースのバックアップ—一貫性のある」を参照。

コピーー一般用語

コピーする：

元のものの複製を作ること。

Oracle のデータ・ファイルおよび制御ファイル、アーカイブ REDO ログのコピーをとる方法は、次の 2 通りです。(1) オペレーティング・システム・ユーティリティを使用する (Unix では、cp か dd を使用できます)。 (2) Recovery Manager の **copy** コマンドを使用する。

COPY — Recovery Manager のコマンド

Recovery Manager のキーワード。これを使用すると、データベースのデータ・ファイルまたは制御ファイル、アーカイブ REDO ログの複製が作られます。この複製は、ディスクに対して Oracle ファイルの読み込みや複製の書き込みを行う Recovery Manager チャンネルに割り当てられた、Oracle サーバー・プロセスによって作られます。Recovery Manager を使うと、データベースの表領域をホット・バックアップ・モードに変えなくても、オープン・データベースのデータ・ファイルをコピーできます。

破損ブロック

認識された Oracle 形式ではない Oracle ブロック、または内容に一貫性がない Oracle ブロック。Oracle は、破損ブロックを、次のいずれかのタイプに識別します。

- 論理破損：REDO の適用によって破損したブロック。
- メディア破損：ブロック形式が正しくない。ブロックが次の状態だと、メディア破損として識別されます。
 - 無効な形式
 - データ・ブロック・アドレスが間違っている
 - ブロック・タイプが無効

メディア破損は、次の方法でしか修復できません。

- ブロックを置き換えて、回復する。たとえば、データ・ファイルを復元すると、ブロックは置き換えられます。また、増分バックアップの適用によってもブロックは新しいブロックに置き換えられます。
- ブロックを再生する。これは、表を削除して、ブロックを別のオブジェクト用に再使用することをいいます。

メディア破損の原因がハードウェア故障の場合、上記のどちらの解決策も、そのハードウェアの故障を直すまでは効果がありません。

破損データ・ファイル

1 つ以上の破損ブロックを含むデータ・ファイル。

データベースのバックアップ—一貫性のある

一貫性バックアップは、データベース全体のバックアップ（すべてのデータ・ファイルと制御ファイル）です。このバックアップは、**RESETLOGS** オプションを使うことで回復処理なしでオープンできます（つまり、このバックアップの一貫性を保つために、**REDO** をこのバックアップ内のファイルに適用する必要はありません）。このようなバックアップは、データベースが正しく停止された後に実行する必要があります。バックアップが完了するまで、データベースはクローズされたままでなくてはなりません。

一貫性バックアップ内のすべてのファイル（制御ファイルを含む）は、同じシステム変更番号 (SCN) でチェックポイントされる必要があります。また、その SCN を過ぎた変更は含むことはできません。一貫性バックアップ内のファイルのうち、古い SCN を含むことができるファイルは、表領域内の読込み専用または通常オフラインのファイルだけです。

データベースのバックアップ—一貫性のない

一貫性のないデータベースのバックアップとは、バックアップ内の幾つかのファイルに、ファイルがチェックポイントされた後の変更が含まれる場合のバックアップのことです。このタイプのバックアップを一貫性のあるものにするには、回復処理が必要です。一貫性のないデータベースのバックアップは、通常、オープン状態のデータベースのバックアップ取得時に作成されます。つまり、ファイルのバックアップ中、データベースはオープンされているということです。非一貫性バックアップは、データベースがクローズしている間にも、次のような時に作成されます。

- Oracle インスタンスがクラッシュした直後 (Oracle Parallel Server クラスタ内のすべてのインスタンス)
- 異常終了停止によってデータベースを停止した後

データベース—クローズ状態の

ユーザーが問合せおよび更新できないデータベース。データベースがクローズされていても、次の作業は可能です。

- データベースをマウントする（ただし、オープンできません）
- インスタンスを開始する（制御ファイルはマウントできません）

データベース—マウントされた

開始され、データベースに対応付けられた制御ファイルがマウントされた（つまり、制御ファイルがオープンされた）インスタンス。データベースは、通常、メンテナンス、または復元および回復用にマウントされた（ただし、オープンされていない）状態になります。

データベース—オープン状態の

ユーザーが問合せおよび更新できるデータベース。データベースは、**startup** コマンドによって自動的にオープンします。または **alter database open** コマンドで明示的にオープンできます。

データ・ファイル

データ・ファイルは、Oracle で作成された、オペレーティング・システムの物理ファイルで、ディスク上にあります。データ・ファイルには、表や索引などのデータ構造が含まれます。データ・ファイルは、1 つのデータベースにだけ属することができます（実際には、1 つの表領域にだけ属することができます）。

データ・ファイルアクセス不可能

Oracle が読み込みを試行しても、検索できないデータ・ファイル。アクセス不可能なファイルにアクセスしようとすると、エラーが発生します。通常、ファイルがアクセス不可能になるのは、格納先のメディアに障害があるか、もしくは、ファイルが移動または削除された場合です。

データ・ファイル・コピー

次のいずれかの方法で、ディスク上に作成されたデータ・ファイルのコピー。

- Recovery Manager の COPY コマンド、または
- カタログ化された、データ・ファイルのオペレーティング・システムのバックアップ

データ・ファイル・ヘッダー

「ファイル・ヘッダー」を参照。

二重アーカイブ REDO ログ

Oracle アーカイバ・プロセス (ARCH) は、REDO ログのコピーを 2 つアーカイブできます。これを、アーカイブ REDO ログの二重化といいます。アーカイブ REDO ログを二重化するには、LOG_ARCHIVE_DUPLEX_DEST を init.ora ファイルに設定します。

ファイル・ヘッダー

Oracle データ・ファイルの最初の数ブロック。ファイル・ヘッダーには、このファイルに関連する読み取り / 書き込みの情報が入っています。

分裂ブロック

オープン・データベースのバックアップ時に発生する可能性があるメディア破損のタイプ。これは、オペレーティング・システム・ユーティリティがバックアップのためのブロックを読み込んでいる時、同時に DBWR がブロックを書き込んでいる場合に起こります。オペレーティング・システムが読み込んだブロックは、「分割されている」可能性があります。たとえば、ブロックの先頭がある時点に書き込まれ、ブロックの最後が別の時点に書き込まれるといったことが発生します。このバックアップが復元に使用され、Oracle がこのブロックを読み込んだ場合、ブロックは「破損している」と認識されます。

分裂ブロックおよびオペレーティング・システム・バックアップ

ブロックを書き込んでいる Oracle と、同じブロックを読み込んでいるオペレーティング・システムの間でタイミングの不一致が発生する可能性があります。このため、オペレーティング・システムのバックアップを開始する前に、表領域をホット・バックアップ・モードに変

える必要があります。ホット・バックアップ・モードのデータベースは、Oracle データ・ブロック全体を REDO ログに書き出します。これにより、ブロックがバックアップ内で分割されても、REDO ログを使ってロールフォワードすることにより修復できます。Recovery Manager には、この問題は影響しません。その理由は、バックアップ（バックアップ・セットまたはコピー）を実行しているサーバー・プロセスは、バックアップ用に読み込んでいるブロックが分割されると、それを検出し、一貫性のあるバージョンを得るために再度ブロックを読み込むからです。

全体バックアップ

Recovery Manager バックアップ・セットの 1 つで、増分ではないもの。この場合、全体とは、データベースのうちバックアップをとった割合を示すのではなく、そのバックアップが増分バックアップではなかったことを示します。したがって、1 つのデータ・ファイルの全体バックアップをとることが可能です。レベル > 0 の増分バックアップが、データ・ファイルを後の時点でロールフォワードする場合は、全体バックアップに加えて、増分バックアップを適用できます。全体バックアップとレベル 0 の増分バックアップの違いは、全体バックアップは、後続の増分バックアップでバックアップをとるブロック数に影響を与えないことだけです。「バックアップ—データベース全体」と比較してください。

全体エクスポート

データベース全体のエクスポート。

全体再同期化

「再同期化—全体」を参照。

ホット・バックアップ

「オープン・バックアップ」を参照。

ホット・バックアップ・モード

ALTER TABLESPACE <tablespace name> BEGIN BACKUP を実行すると、表領域は「ホット・バックアップ・モード」になります。このコマンドは、オープン・データベース（つまり、ホット）バックアップをとる前に実行します。

このコマンドは、バックアップをとる表領域を構成する (1 つ以上の) データ・ファイルのオペレーティング・システムのバックアップをとる前に、実行する必要があります。Recovery Manager を使ってバックアップをとる場合、ホット・バックアップ・モードは使用しないでください。「分裂ブロック」を参照。

アクセス不可能なデータ・ファイル

「データ・ファイル—アクセス不可能」を参照。

インカーネーション（データベース）

同じ物理データベースの異なる「バージョン」を指します。データベースのインカーネーションは、RESETLOGS オプションによってオープンされることによって変わり、以前の「バージョン」と区別できるようになります。

一貫性のないデータベースのバックアップ

「データベースのバックアップ—一貫性のない」を参照。

増分バックアップ

増分バックアップは、変更されたブロックだけのバックアップをとる方法です。レベル0の増分バックアップと全体バックアップは、使用済みのブロックすべてのバックアップをとるという点で、同じ機能を実行します。レベル0の増分バックアップと全体バックアップが異なる点は、全体バックアップの場合は、後続の増分バックアップによってバックアップをとられるブロックはどれかということに影響を与えませんが、増分バックアップの場合、後続の増分バックアップによってコピーをとられるブロックはどれかということに影響があります。

増分バックアップがレベル>0の場合、バックアップをとるブロックは、前回の増分バックアップ以降に変更されたブロックだけです。前回の増分バックアップ以降変更されていないブロックのバックアップはとられません。

インスタンス

メモリーに常駐している、一連のメモリー構造と Oracle コード (バックグラウンド・プロセスを含む)。startup コマンドを発行すると、インスタンスが作成され、メモリーに常駐します。インスタンスは、次のコマンドを使ってメモリーに常駐させることができます。

- **STARTUP NOMOUNT** —インスタンスが開始しますが、制御ファイルはマウントされず、データベースもオープンされません。
- **STARTUP MOUNT** —インスタンスが開始し、データベースの制御ファイルがマウントされます。データベースはオープンされません。「データベース—マウントされた」を参照。
- **STARTUP** —インスタンスが開始し、データベースの制御ファイルがマウントされ、データベースがオープンされます。「データベース—オープン」を参照。インスタンスは、shutdown コマンドによってメモリーから削除されます。

メディア・マネージャ

ソフトウェア・ベンダーから提供されるユーティリティ。テープ・ドライブなどの順次メディアをロードおよびラベル付け、アンロードする機能があります。メディア・マネージャによって、メディアの時間切れや再利用を構成できます。また、通常は自動テープ・ライブラリの制御機能も備えています。

メディア管理インターフェース

(次のように呼ばれることもあります。

- メディア管理レイヤー
- メディア管理ライブラリー (MML)
- SBT インタフェース)

Oracle が発表したインタフェース。メディア管理を扱うベンダーが、このインタフェースに対して互換性のあるソフトウェア・ライブラリを作成しています。このソフトウェアは Oracle に統合されます。これによって Oracle サーバー・プロセスは、メディア・マネージャにコマンドを発行して、順次記憶装置へバックアップ・ファイルを書き出したり、順次記憶装置からファイルを読み込むことができます。Oracle がファイルのバックアップまたは復元の要求を出すと、メディア・マネージャは、正しいテープのロードおよびラベル付け、アンロードに必要なアクションを処理します。

ミラー化

データの正確なコピーが複数あることを示す。一般的に、この用語は、ハードディスクをミラー化することによって、ディスクの 1 つが使用不可能になった場合に、中断せずにもう一方のディスクで要求に応じてサービスを継続できることを示すのに使われます。

ミラー化－解除

ミラー化の解除とは、ミラーを管理するオペレーティング・システムまたはハードウェアが、ミラー・イメージをこれ以上最新の状態に維持しなくなることです。これは、オペレーティング・システムのデータベースのバックアップを作成するための方法の 1 つです。データベース内のすべての表領域はホット・バックアップ・モードに変わり、ミラー化は解除されます。すべての表領域は、ホット・バックアップ・モードで取り出されます。そして、ミラー化を解除された方の情報から、テープへバックアップをとります。バックアップの完了後、ミラーを復元できます。

ミラー化－ミラー復元

ミラーの復元とは、ミラーを管理するオペレーティング・システムまたはハードウェアに、ミラー化を解除された古いミラーを、更新されているもう片方のミラーからリフレッシュさせ、ミラーの両側をメンテナンスさせることです。

ミラー化された制御ファイル

「多重制御ファイル」を参照。

ミラー化されたオンライン REDO ログ

「多重オンライン REDO ログ」を参照。

多重

- オンライン・ログ

Oracle は、オンライン・ログの複数のコピーを自動的にメンテナンスできます。これをオンライン・ログの多重化といいます。この多重化は、各 REDO ログ・グループに複数のメンバーを作成することによって制御されます。多重度は、各グループのメンバー数に直接関係します。ミラー化されたオンライン REDO ログともいいます。

- 制御ファイル

Oracle は、データベースの制御ファイルの複数コピーを自動的にメンテナンスできます。これは、多重制御ファイルと呼ばれます。多重制御ファイルは、`init.ora` ファイルの

CONTROL_FILES パラメータに複数の項目を指定することにより制御されます。ミラー化された制御ファイルともいいます。

- バックアップ・セット

同じバックアップ・セット内のデータ・ファイルのブロックは、まとめて多重化されます。つまり、バックアップ・セット内のファイルのブロックは、同じバックアップ・セット内の他のファイルのブロックとともにバックアップ・セット全体に分散されるということです。

- アーカイブ **REDO** ログ

「多重アーカイブ・ログ」を参照。

未使用ブロックと圧縮

未使用ブロックとは、今までにデータが書き込まれたことがない **Oracle** ブロックです。新たに作成されたデータ・ファイルには、多数の未使用ブロックが含まれます。**Oracle** がバックアップ・セットを書き出す場合、使用されているブロックだけを書き出します（未使用ブロックは書き出しません）。つまり、データ・ファイルは「圧縮」されます。

オフライン表領域

データベースがオープンしているときにユーザーが使用できない表領域。データベースがオープンしている場合、表領域をオフラインにすることができるのは **DBA** だけです。ある表領域がオフラインになると、その表領域を構成するすべての（まだオフラインになっていない）ファイルがオフラインになります。

表領域は、次の 3 通りの優先順位によってオフラインになります。

- 通常オフライン（またはクリーン）

表領域内のすべてのファイルは、チェックポイントされた後オフラインになります。表領域のファイルが 1 つでも使用できない場合、その表領域は通常オフラインにはなりません。表領域内のファイルが正常にオフラインになった場合、表領域をオンライン状態に戻すために、そのファイルを回復する必要はありません。

- 一時的なオフライン

Oracle がアクセスできる、表領域のすべてのファイルは、チェックポイントされた後オフラインになります。**offline-temporary** コマンドによってチェックポイントされたファイルは、回復は必要ありませんが、(**offline-immediate** コマンドの実行時にアクセス不可能だったために) チェックポイントされなかったファイルは、表領域をオンラインに戻す前に回復する必要があります。

- 即時オフライン

表領域内のすべてのファイルは、最初にファイルのチェックポイントを試行せずに、オフラインになります。表領域のすべてのファイルは、その表領域をオンラインに戻す前に、回復する必要があります。表領域をオフラインにするには、データベースをオープンする必要があります。

オフライン・バックアップ（「クローズ状態のバックアップ」を参照）

この用語は、オフライン表領域およびオフライン・データ・ファイルと間違いやすいので、クローズ状態のデータベースのバックアップを指す用語としては、現在は使用されていません。

オフライン・データベース（「データベースクローズ状態の」を参照）

この用語は、オフライン表領域およびオフライン・データ・ファイルと間違いやすいので、クローズ状態のデータベースを指す用語としては、現在は使用されていません。

オフライン・データ・ファイル

データベースがオープンしているときにユーザーが使用できないデータ・ファイル。データ・ファイルは、オフラインの表領域の一部として、または個別にオフラインにできます。例外的な状況では、Oracle は、必要であれば自動的にデータ・ファイルをオフラインにします。

データ・ファイルは、次のいずれかの場合にオフラインになります。

- 表領域をオフラインにしなければならない障害が発生し、Oracle が自動的にオフラインにする表領域に、そのデータ・ファイルが属している場合。このファイルをオンラインに戻すには、その前に回復処理が必要です。
- コマンド ALTER DATABASE DATAFILE <filename>OFFLINE が発行された場合。

alter database datafile コマンドによってオフラインになったデータ・ファイルは、オンラインに戻す前に回復する必要があります。このオフライン・コマンドは、データベースがマウントされているとき、またはオープンしているときに実行できます。

オンライン表領域

データベースがオープンしている場合に、ユーザーが使用できる表領域。コマンド ALTER TABLESPACE <name> ONLINE を発行することにより、表領域にユーザーがアクセスできるようになります。表領域をオンラインに変えるには、データベースをオープンし、その表領域内のすべてのファイルを、データベースの残りのファイルと一貫性のある状態にする必要があります。

オンライン・バックアップ（「オープン状態のバックアップ」を参照）

この用語は、オンライン表領域およびオンライン・データ・ファイルと間違いやすいので、オープン・データベースのバックアップを指す用語としては、現在は使用されていません。

オンライン・データベース（「データベースオープン状態の」を参照）

この用語は、オンライン表領域およびオンライン・データ・ファイルと間違いやすいので、オープン・データベースを指す用語としては、現在は使用されていません。

オンライン・データ・ファイル

ユーザーがアクセスできるようになったデータ・ファイル。コマンド ALTER DATABASE DATAFILE<filename> ONLINE は、データベースがオープンしているか、またはマウントさ

れているときに発行できます。データベースがオープンしている場合、ファイルをオンラインにするには、そのファイルを、データベースの残りのファイルと一貫性のある状態にする必要があります。データベースがマウントされている場合、ファイルは、他のデータ・ファイルとの一貫性がなくてもオンラインに変更できます（これは、通常、データベース回復を開始する前に行います）。

オープン・バックアップ（オープン状態のデータベースのバックアップ、またはホット・バックアップともいう）

データベースがオープンしているときに作成される、1 つ以上のデータ・ファイルのバックアップ。オープン・バックアップは、オペレーティング・システムのバックアップ（表領域がホット・バックアップ・モードでなくてはならない）の場合と、**Recovery Manager** のバックアップ（表領域はホット・バックアップ・モードでなくてもよい）場合があります。

オープン状態のデータベース

「データベースーオープン」を参照。

パラレル化

Recovery Manager

Recovery Manager を使ってバックアップおよび復元をパラレルで処理するには、複数のチャネルを割り当てる必要があります。

バックアップ・セットの作成

複数のチャネルを割り当てて、**filesperaset** を指定することで、パラレルで処理できます。

ファイル・コピーの作成

複数のチャネルを割り当てて、コピーする複数のファイルを単一の **copy** コマンドに指定することによって、パラレルで処理できます。

復元のパラレル化

Recovery Manager が復元をパラレルで処理します。並列度は、割り当てられたチャネル数、および読み込み可能なバックアップ・セットまたはファイル・コピーの数によって異なります。

回復のパラレル化

Recovery Manager は、増分バックアップを適用している場合は、回復もパラレルで処理します。並列度は、割り当てられたチャネル数、および読み込み可能なバックアップ・セットの数によって異なります。

回復の並列性

Recovery Manager または O/S のバックアップのどちらの場合も、回復の並列性を利用して REDO ログの適用をパラレルで処理できます。

部分的な再同期化

「再同期化－部分的」を参照。

パスワード・ファイル

orapwd コマンドで作成されたファイル。

ネットワークを介して内部接続する場合、データベースはパスワード・ファイルを使用する必要があります。詳細は、『Oracle8 Server 管理者ガイド』を参照してください。

読取り専用表領域

「表領域－読取り専用」を参照。

回復

復元されたファイルを現行ファイルにすること。

または

復元されたファイルを、ある特定の時点での現行ファイルにすること。ARCHIVELOG モードでは、完全回復または不完全回復のどちらかを選べます（「回復－完全」および「回復－不完全」を参照）。

NOARCHIVELOG モードのデータベースの場合、最新のバックアップからしか復元できません。例外的な状況では、データベースが ARCHIVELOG モードでなければ、データ・ファイルまたはデータベースを回復できる場合もあります。NOARCHIVELOG モードのデータベースを回復できるのは、バックアップ後に、オンライン・ログが 1 つも上書きされていない場合だけです。

たとえば、次のような場合です。

1. DBA が、10 組のオンライン・ログ・グループで構成される、データベースのクローズ状態の（一貫性のある）バックアップを作成した。
2. DBA がそのデータベースをオープンして、5 つのログ・スイッチが起こった（つまり、9 つ以下のログを、10 個のオンライン・ログに切り替えることができる）。
3. 1 つ以上のデータ・ファイルが破損するメディア故障が発生した。

このような場合、バックアップ以降作成された REDO ログはすべて有効なので、回復できます。

ただし、これは例外的なケースです。データベースを復元するだけでなく回復できるようにしたい場合は、データベースを ARCHIVELOG モードにする必要があります。オンライン・ログのバックアップはとらないでください。メディア障害時にデータベースを回復できるようにしたい場合、これは必ず守ってください。

回復－ Recovery Manager

Recovery Manager のコマンド。このコマンドは、最初に増分バックアップ（もしあれば）を適用し、次に REDO(アーカイブ・ログおよびオンライン REDO ログ)を適用して、復元されたデータ・ファイルを再生します。

RECOVER－SQL コマンド

REDO(アーカイブ・ログおよびオンライン REDO ログ)の適用により、復元されたファイルを再生するための SQL コマンド。

回復－完全

データベースを復元、回復し、バックアップ以降に生成された REDO(アーカイブ・ログとオンライン REDO ログの両方)をすべて適用します。このタイプの回復は、通常、メディア故障により 1 つ以上のデータ・ファイルまたは制御ファイルが破損した場合に実行されます。破損したファイルは、復元されたバックアップの最初の作成時以降に生成された、すべての REDO によって、完全に回復します。

回復－不完全

バックアップからデータベースを復元して回復するが、バックアップ以降に生成された REDO 情報をすべては適用しないこと。

不完全回復は、通常、次のような場合に行います。

- ハードウェア障害のため、オンライン・ログが消失した場合。
この場合、データベースは、障害前に生成された最新のアーカイブ・ログまで回復します。
- ユーザーのエラーが原因で、エラー発生直前まで回復する必要がある場合。
必要なことは、データベースで不当なアクションが発生する前のある時点まで回復することです。たとえば、給与支払いのトランザクションが支払代行機関に送られる前に、ユーザーが誤ってそのトランザクションを削除してしまったとします。このような場合、DBA は、データベース全体を復元して、ユーザーがトランザクションを削除する直前の時点まで不完全回復を実行する必要があります。
- 回復に必要なアーカイブ・ログが紛失している場合。
完全回復に必要なアーカイブ・ログのバックアップが作成されなかったか、アーカイブ・ログの内容が（メディア故障などが原因で）破損している場合。この場合、紛失しているログの時点まで回復するしか方法はありません。

上記のような場合、データベースは、RESETLOGS オプションでオープンする必要があります。

リカバリ・カタログ

リカバリ・カタログとは、Recovery Manager が Oracle データベースの情報を格納する、Oracle の表とビューのセットです。この情報は、Recovery Manager が、Oracle データベースのバックアップおよび復元、回復を管理するためのものです。

リカバリ・カタログ用のデータベース

リカバリ・カタログのスキーマが設定されている Oracle データベース。

Recovery Manager

DBA が、Oracle データベース（データ・ファイルおよび制御ファイル、アーカイブ・ログ）のバックアップおよび復元、回復に使用するツール。このツールは、リカバリ・カタログとともに、またはリカバリ・カタログなしで使用できます。リカバリ・カタログを使用しない場合、Recovery Manager は、データベースの制御ファイルを使って、データベースのバックアップおよび復元、回復方法を判断します。

回復セット

Point-in-Time 回復が必要な、データベースの表領域のサブセット。「補助セット」も参照。

回復－ Point-in-Time

データベース

すべてのデータ・ファイルおよび制御ファイルの、特定の時点までの回復。これは、完全回復ではありません。

表領域

表領域内のすべてのデータ・ファイルの、特定の時点までの回復。これは、完全回復ではありません。この回復は、クローン・データベースで行われます。その後、表領域は元のデータベースに再統合されます。これは、Oracle8 の新しい機能です。

REDO ログ

オンライン REDO ログ

オンライン REDO ログは、Oracle データ・ファイルと制御ファイルへのすべての変更を記録する、2つ以上のファイル・セットです。LGWR がログへ書き込みます。「多重オンライン REDO ログ」を参照。

現行のオンライン REDO ログ

現在 LGWR が書き込んでいる、オンライン REDO ログ・ファイル（多重ログの場合は、オンライン REDO ログ・グループ）。

アーカイブ REDO ログ

これは、オフライン REDO ログとも呼ばれます。データベースが ARCHIVELOG モードの場合、1つのオンライン REDO ログが満杯になるたびに、1つ以上のアーカイブ・ログにコピーされます。このコピーが、アーカイブ REDO ログです。

「二重アーカイブ REDO ログ」を参照。

REDO ログ・グループ

各オンライン REDO ログは、1つのグループに属しています。グループには、少なくとも1メンバーが必要ですが、複数のメンバーがある場合もあります。REDO ログ・グループに複数のメンバーがある場合、この状態を多重化されているといいます。

スレッド

各 Oracle データベース・インスタンスには、独自のオンライン REDO ログ・グループのセットがあります。このようなオンライン REDO ログ・グループをインスタンスの、オンライン REDO のスレッドといいます。Oracle Parallel Server 環境以外では、1つのデータベースにはスレッドが1つしかなく、これは、そのデータベースにアクセスしているインスタンスに属します。Oracle Parallel Server 環境では、インスタンスごとに別々のスレッドがあります（つまり、各インスタンスには、独自のオンライン REDO ログがあります）。各スレッドには、独自の現行ログ・メンバーがあります。

REDO レコード

REDO レコードは、変更ベクトルのグループで、データベースに対する単一の最小単位の変更が記述されています。すべてのデータ・ブロック変更について、REDO レコードが組み立てられ、ディスク上のオンライン REDO ログに保存されます。REDO レコードを使うと、障害があった場合でも、複数のデータベース・ブロックを変更して、すべての変更を有効にするか、変更がまったくなかった状態に戻すことができます。

register database

データベースをリカバリ・カタログに登録するための Recovery Manager のコマンド。どの resync catalog コマンドを使っても、リカバリ・カタログは、データベースの現行の構成で更新されます。

RESETLOGS オプション

データベースをオープンする方法の1つ。この方法を使用すると、データベースのインカーネーションが変更され、ログ順序番号が1にリセットされ、オンライン・ログが再フォーマット（なければ、作成）されます。通常、データベースは、次のようなときに、RESETLOGS キーワードでオープンされます。

- 不完全回復（任意の時点または、ログ順序番号、変更、取消しまで）の後
- バックアップ制御ファイルを使用した回復の後

このオプションでデータベースをオープンするのは、過去にそのデータベースで作成されたすべてのアーカイブ・ログが、誤ってこのインカーネーション・データベースに適用されないようにすることで、データベースの破損を防ぐためです。

復元

バックアップからファイルの元のコピーを再構築または復活させること。

再同期化－完全な

カタログの完全な再同期化は **Recovery Manager** が行う処理の 1 つで、データベースの制御ファイル内の変更された情報をすべて使って、リカバリ・カタログをリフレッシュします。完全な再同期化は、DBA が、**Recovery Manager** コマンド **resync catalog** を使って頻繁に実行する必要があります。また、**Recovery Manager** が、あるコマンドを実行する前に完全な同期化が必要と判断した場合は、**Recovery Manager** がこの操作を実行することもあります。

再同期化－部分的

部分的な再同期化とは、アーカイブ REDO ログおよびバックアップ・セット、データ・ファイル・コピーに関する情報を、リカバリ・カタログに転送することです。部分的な再同期化では、次のような情報は転送されません。

- 新規データ・ファイル
- 新規表領域または削除された表領域
- 新規または削除されたオンライン・ログ・グループおよびメンバー

部分的な再同期化は、次の処理の前に **Recovery Manager** によって実行されます。

- バックアップおよびコピー、復元、回復、リスト、レポート

また、次の処理の後に **Recovery Manager** が必要と判断した場合にも、再同期化が実行されます。

- バックアップおよびコピー、復元、切替え、登録、リセット、カタログ化

ロールフォワード

データ・ファイルおよび制御ファイルを回復する（つまり、これらのファイルに対する変更を回復する）ために、REDO ログをデータ・ファイルおよび制御ファイルに適用すること。

SBT

System Backup to Tape(テープへのシステム・バックアップ)の頭字語。

「メディア管理インタフェース」を参照。

正しく停止

データベースが、即時停止またはトランザクション停止、標準停止によって停止された状態を指す。正しく停止されたデータベースは、すでに一貫性のある状態なので、回復は必要ありません。

スナップショットの制御ファイル

Recovery Manager が作成した、データベースの制御ファイルのコピー。スナップショットの制御ファイルは、**Recovery Manager** が、制御ファイルの一貫性があるバージョンを読み込むために使います（読込みの目的は、リカバリ・カタログの再同期化のため、またはリカバリ・カタログを使用していない場合に **Recovery Manager** が実行すべき作業を判断するため）。スナップショット制御ファイルは、**Recovery Manager** が、バックアップ制御ファイルを作成

するのと同じ Oracle コード (ALTER DATABASE BACKUP CONTROL FILE TO'<バックアップ先>) を使って作成します。

分割ブロック（「分裂ブロック」を参照）

ステージング（アーカイブ REDO ログの、テープからディスクへの）

アーカイブ・ログのステージングとは、アーカイブ・ログを、回復処理を実行するために 3 次記憶装置（通常はテープ）からディスクへ復元することをいいます。

SWITCH

データ・ファイルのコピーを、Oracle データベース用のデータ・ファイルに変換するための Recovery Manager のコマンド。このコマンドは、SQL コマンドの ALTER DATABASE RENAME DATAFILE FROM'<元の名前>' TO'<変更後の名前>' と同じ機能を実行し、さらにそのデータ・ファイル・コピーに、これ以降使用不可能というマークを設定します。

表領域

データベースは、1 つ以上の、表領域と呼ばれる論理記憶単位に分割されます。各表領域には、その表領域に排他的に対応付けられたデータ・ファイルが多数格納されています。

表領域－読取り専用

更新禁止の状態になった表領域。表領域は、SQL 文の ALTER TABLESPACE <表領域> READ ONLY を実行することで読込み専用モードになります。通常、表領域を読込み専用モードにするのは、バックアップをとる回数を減らせるようにするためです。たとえば、表領域のバックアップを毎晩とるかわりに、バックアップをとる回数を毎月 1 回またはそれ以下に減らすことができます。

注意： 表領域のバックアップをとる間隔が長くなると（コピーを取る回数が減るので）、バックアップ・メディアを維持しなければならない時間も長くなり、バックアップ・メディアが破損するリスクも大きくなります。

表領域の Point-in-Time 回復

「回復－Point-in-Time 表領域」を参照。

テープのストリーミング

テープに連続的に書き込めるように、高速でテープ・ドライブに出力を書き込むこと。

テープ・ドライブ

磁気テープからの読込み、および磁気テープへの書込みを行うハードウェア。

テープ・ボリューム

ベンダーがよく使う用語で、1 巻の物理的なテープ・メディアを指す。

A

ABORT オプション
 SHUTDOWN コマンド, 12-9, 12-22, 12-26, 12-52
ACTIVATE STANDBY DATABASE オプション
 ALTER DATABASE コマンド, 12-38
ALERT ファイル, 2-18, 12-3, 12-25, 12-29, 12-34
allocate channel for delete コマンド, 8-7, 9-8
allocate channel コマンド
 type disk オペランド, 8-10
allocate channel コマンド, 8-10, 8-29, 8-36
ALTER DATABASE コマンド
 ACTIVATE STANDBY DATABASE オプション,
 12-38
 BACKUP CONTROLFILE TO TRACE オプション,
 11-10, 12-53
 BACKUP CONTROLFILE オプション, 2-23
 CLEAR LOGFILE GROUP オプション, 12-37
 CLEAR LOGFILE オプション, 12-48
 CREATE DATAFILE オプション, 12-13, 12-39
 CREATE STANDBY CONTROLFILE オプション,
 12-35
 DATAFILE ONLINE オプション, 12-27, 12-32
 MOUNT STANDBY DATABASE オプション, 12-37,
 12-43
 NORESET LOGS オプション, 11-11, 12-24, 12-29,
 12-34, 12-53
 OPEN NORESETLOGS オプション, 12-17
 OPEN RESETLOGS オプション, 8-4, 12-10, 12-24,
 12-29, 12-34
 OPEN オプション, 12-52
 RECOVER AUTOMATIC LOGFILE オプション, 12-7
 RECOVER LOGFILE オプション, 12-6
 RECOVER...FROM パラメータ, 12-5
 RECOVER 句, 12-4

 RESETLOGS オプション, 11-11, 12-3
ALTER SYSTEM コマンド
 ARCHIVE LOG CURRENT オプション, 12-36, 12-38
ALTER TABLESPACE コマンド
 BEGIN BACKUP オプション
 開始から終了までの時間を最小限にする, 11-8
 開始点のマーク, 11-6
 BEGIN/END BACKUP オプション, 8-8
 END BACKUP オプション
 開始から終了までの時間を最小限にする, 11-8
 終了点のマーク, 11-6
 OFFLINE オプション, 11-9
 ONLINE オプション
 表領域をオンラインに戻す, 11-9
ARCHIVE LOG CURRENT オプション
 ALTER SYSTEM コマンド, 12-36, 12-38
ARCHIVELOG モード
 概要, 1-7
 定義, 2-16
 データ・ファイルの消失, 12-45
 バックアップ・オプション, 3-5
 バックアップ計画, 3-8
 分散データベースのバックアップ, 3-6
available オペランド
 change コマンド, 8-7

B

BACKUP CONTROLFILE TO TRACE オプション
 ALTER DATABASE コマンド, 11-2, 11-10, 12-53
BACKUP CONTROLFILE TO オプション
 ALTER DATABASE コマンド, 11-2
BACKUP CONTROLFILE オプション
 ALTER DATABASE コマンド, 2-23
backup コマンド

- format オペランド, 8-20
- setmaxcorrupt 句, 9-6
- skip inaccessible オプション, 9-7
- skip offline オプション, 9-2, 9-7
- skip read only オプション, 9-2
- tag オペランド, 8-23
- backup コマンド, 8-21
- BEGIN/END オプション
 - ALTER TABLESPACE コマンド, 8-8

C

- catalog コマンド, 8-2, 8-8
- catrman.sql スクリプト, 6-5
- catrman.sql スクリプト, 8-3
- CE CodeExInd, xvii
- change...delete コマンド, 8-8
- change コマンド
 - available オペランド, 8-7
 - delete オペランド, 8-7
 - unavailable オペランド, 8-7
 - uncatalog オペランド, 8-7
- change コマンド, 8-6
- CLEAR LOGFILE GROUP オプション
 - ALTER DATABASE コマンド, 12-37
- CLEAR LOGFILE オプション
 - ALTER DATABASE コマンド, 12-48
- COMPATIBLE 初期化パラメータ, 12-41
- CONTROL_FILES 初期化パラメータ, 8-32, 12-52
- copy コマンド, 8-29
- CREATE CONTROLFILE コマンド
 - DATAFILE 句, 12-53
 - スタンバイ・データベースへの影響, 12-40
- CREATE DATAFILE オプション
 - ALTER DATABASE コマンド, 12-13, 12-39
- CREATE STANDBY CONTROLFILE オプション
 - ALTER DATABASE コマンド, 12-35

D

- database オプション
 - list コマンド, 8-17
- DATABASE パラメータ
 - RECOVER コマンド, 12-3
- DATAFILE ONLINE オプション
 - ALTER DATABASE コマンド, 12-27, 12-32
- datafile オプション

- list コマンド, 8-17
- DATAFILE 句
 - CREATE CONTROLFILE コマンド, 12-53
- DATAFILE パラメータ
 - RECOVER コマンド, 12-4
- days オプション
 - report コマンド, 8-14
- DB_BLOCK_MAX_DIRTY_TARGET, 2-7
- DB_FILE_STANDBY_NAME_CONVERT 初期化パラメータ, 12-37
- プライマリ・データベースおよびスタンバイ・データベースの場合, 12-40
- DB_NAME 初期化パラメータ, 2-20
- DB_VERIFY ユーティリティ, 11-4
- DBA_DATA_FILES ビュー, 11-5
- db 識別子, 8-3
- delete script コマンド, 8-19
- delete オペランド
 - change コマンド, 8-7
- device_type_list オプション
 - list backupset コマンド, 8-18
 - report コマンド, 8-13

E

- Enterprise Backup Utility, 5-3
- Enterprise Manager
 - RECOVER DATABASE UNTIL TIME 文, 12-27
 - SET AUTORECOVERY ON 文, 12-6
 - STARTUP コマンド
 - RECOVER オプション, 12-4
 - 「回復アーカイブ適用」ダイアログ・ボックス, 12-3, 12-23
 - 「データベース起動」ダイアログ・ボックス, 12-27
 - 「データベース停止」ダイアログ・ボックス, 12-22, 12-26, 12-52
 - 異常終了停止モード, 12-9
 - ログ・ファイルの適用, 12-5
- Enterprise Manager の「データベース起動」ダイアログボックス, 12-23
- execute script コマンド, 8-19
- Export ユーティリティ, 2-28, 5-3
 - エクスポートのタイプ, 11-14
 - バックアップ, 2-28, 3-6
 - バックアップのための使用, 11-13
 - 読取り一貫性, 11-13

F

file_name_pattern オプション
list コマンド, 8-18
format オペランド
backup コマンド, 8-20
from backup オペランド
restore コマンド, 8-33
from copy オペランド
restore コマンド, 8-33
from/until time オプション
list コマンド, 8-18

I

I/O エラー
バックアップへの影響, 8-20
I/O デバイス
Recovery Manager での解放, 8-11
Import ユーティリティ, 11-13
使用手順, 11-14
データベース回復および, 11-14
incremental オプション
report コマンド, 8-14
init.ora ファイル, 8-35

L

list backupset コマンド
device_type_list オプション, 8-18
list backupset コマンド, 8-18
list copy コマンド, 8-18
list_object_list オプション
list コマンド, 8-17
list_operand オプション
list コマンド, 8-17, 8-18
list_qualifier オプション
list コマンド, 8-18
LIST コマンド, 7-22
list コマンド
database オプション, 8-17
datafile オプション, 8-17
file_name_pattern オプション, 8-18
from/until time オプション, 8-18
list_object_list オプション, 8-17
list_operand オプション, 8-18
list_qualifier オプション, 8-18

redundancy オプション, 8-15, 8-16
tablespace オプション, 8-17
tag オプション, 8-18

list コマンド, 8-12, 9-11
LOG_ARCHIVE_DEST 初期化パラメータ, 12-5, 12-23, 12-27
LOG_ARCHIVE_FORMAT 初期化パラメータ, 12-5
LOG_ARCHIVE_START 初期化パラメータ, 2-17
LOG_CHECKPOINT_INTERVAL パラメータ, 2-6
LOG_CHECKPOINT_TIMEOUT パラメータ, 2-6
LOG_CHECKPOINTS_TO_ALERT 初期化パラメータ, 2-9
LOG_FILE_STANDBY_NAME_CONVERT 初期化パラメータ, 12-37
LOGSOURCE パラメータ
SET コマンド, 12-5

M

MOUNT STANDBY DATABASE オプション
ALTER DATABASE コマンド, 12-37, 12-43
MOUNT オプション
STARTUP コマンド, 12-23, 12-27

N

NLS_DATE_FORMAT 環境変数, 6-10, 9-4, 9-10
NLS_LANG 環境変数, 6-10, 9-4, 9-10
NOARCHIVELOG モード
回復、から, 12-9
概要, 1-7
定義, 2-16
データ・ファイルの消失, 12-45
バックアップ, 9-2
バックアップ・オプション, 3-5
バックアップ計画, 3-7
不利な点, 12-9
分散データベースのバックアップ, 3-6
NORESET LOGS オプション
ALTER DATABASE コマンド, 12-53
制御ファイルのバックアップ, 11-11
NORESETLOGS オプション, 12-24, ?? – 12-25

O

OFFLINE オプション
ALTER TABLESPACE コマンド, 11-9

ONLINE オプション
 ALTER TABLESPACE コマンド, 11-9
OPEN RESETLOGS オプション
 ALTER DATABASE コマンド, 8-4, 12-10, 12-24,
 12-29, 12-34
OPEN オプション
 ALTER DATABASE コマンド, 12-52
ORA-01578 のエラー・メッセージ, 12-43
Oracle Enterprise Manager, 5-3
Oracle8 のユーティリティ
 Recovery Manager, 7-2
OS ユーティリティ, 9-7

P

Parallel Server
 オンライン REDO ログのスレッド, 2-13
 バックアップ, 9-5
PARALLEL_MAX_SERVERS 初期化パラメータ, 12-11
PARALLEL 句
 RECOVER コマンド, 12-11
Point-in-Time 回復
 データベース, 4-6, 9-10
 表領域, 4-6
print script コマンド, 8-19

R

RECOVER AUTOMATIC LOGFILE オプション
 ALTER DATABASE コマンド, 12-7
RECOVER DATABASE UNTIL CANCEL 文, 12-23
RECOVER DATABASE UNTIL CHANGE 文, 12-32
RECOVER DATABASE UNTIL TIME 文, 12-27
RECOVER DATABASE コマンド, 12-52
RECOVER FROM...STANDBY DATABASE コマンド,
 12-37, 12-43
RECOVER...FROM パラメータ
 ALTER DATABASE コマンド, 12-5
, 8-7
catalog, 8-8
Recovery Manager, 5-2, 8-19
 recover コマンド, 4-2
 一般的な接続, 6-7
 回復コマンド, 8-37
 概要, 7-2
 コマンド
 allocate channel, 8-10, 8-29, 8-36

 allocate channel for delete, 8-7, 9-8
 backup, 8-21, 9-6
 catalog, 8-2
 change, 8-6, 8-7
 change...delete, 8-8
 copy, 8-29
 delete script, 8-19
 execute script, 8-19
 list, 8-12, 9-11
 list backupset, 8-18
 list copy, 8-18
 print script, 8-19
 recover, 8-35, 9-8
 register database, 8-3
 release channel, 8-11
 replace script, 8-19
 replicate, 8-32
 report deletable, 8-15
 report need backup, 8-13
 report need backup days, 8-14
 report schema, 8-16
 report unrecoverable, 8-14
 reset database, 8-4
 restore, 9-8
 resync catalog from backup controlfile, 8-9
 set newname, 8-32, 8-35
 setlimit channel, 8-11
 switch, 8-35
コマンドの実行, 6-8 – 6-11
コマンド・ファイル, 6-9
サンプル・スクリプト, 6-11
使用例, 9-1 – 9-12
ストアド・スクリプト, 6-10, 7-9
接続オプション, 6-7
対話形式の使用方法, 6-9
データ・ファイルのコピー, 9-5
データベース情報のリセット, 8-4
データベースの登録, 8-3
パスワード・ファイルを使用した接続, 6-7
パスワード・ファイルを使用しない接続, 6-7
バックアップ
 使用例, 8-28
 分裂ブロックの検出, 7-25
バックアップのタイプ, 7-9, 8-22
バックアップのためのユーザー・タグ, 7-23
パラメータ設定, 6-10
分裂ブロックの検出, 7-25

- 必要なりカバリ・カタログ, 8-2
- ファイルの削除, 8-7
- リカバリ・カタログ, 4-4
- リカバリ・カタログを使用した接続, 6-8
- リカバリ・カタログを使用しない接続, 6-7
- Recovery Manager のコマンド・ファイル, 6-9
- RECOVERY_PARALLELISM 初期化パラメータ, 12-11
- RECOVER オプション
 - STARTUP コマンド, 12-4
- RECOVER 句
 - ALTER DATABASE, 12-4
- RECOVER コマンド
 - DATABASE パラメータ, 12-3
 - DATAFILE パラメータ, 12-4
 - PARALLEL 句, 12-11
 - TABLESPACE パラメータ, 12-4
 - 回復不能オブジェクトおよびスタンバイ・データベース, 12-43
- recover コマンド
 - Recovery Manager, 4-2
- recover コマンド, 8-35, 8-38
- REDO エントリ
 - 内容, 2-2
- REDO レコード
 - 定義, 4-3
- REDO ログ
 - 定義, 4-3
 - バックアップ用のログ・ファイルのリスト表示, 11-2
- REDO ログ・バッファ
 - 書込み, 2-2
- REDO ログ・ファイル, 1-5
- REDO エントリ, 2-2
- アーカイブ済み
 - ログ・スイッチ, 2-5
- アーカイブ
 - アーカイブ時のエラー, 2-18
 - アーカイブの仕組み, 2-14
 - 自動, 2-17
 - 手動, 2-18
 - 制御ファイル, 2-14
 - 内容, 2-15
 - 利点, 2-13
 - アーカイブ・ログ, 1-7
 - アクティブでない, 2-4
 - アクティブ（現行）, 2-4
 - オンライン, 2-2
 - 回復時の使用方法, 2-2
 - スレッド, 2-13
 - チェックポイント障害後, 2-8
 - 複数の要件, 2-3
- 概要, 1-5
- グループ, 2-10
- 手動アーカイブ, 2-18
- 使用可能, 2-3
- 制御ファイルに指定されているファイル, 2-19
- 多重, 2-9
 - ダイアグラム, 2-10
- 目的, 1-5
- 内容, 2-2
- バックアップの中のファジー・データ, 3-9
- 分散トランザクション情報, 2-3
- ミラー化
 - アーカイバ・プロセス (ARCH), 2-14
 - 一部にアクセスできない場合, 2-10
 - すべてにアクセスできない場合, 2-11
 - ログ・スイッチ, 2-10
- 命名, 12-5
- メンバー, 2-10
- ログ順序番号, 2-5
- ログ・スイッチ, 2-5
- ログ・ライター・プロセス, 2-3
- redundancy オプション
 - list コマンド, 8-15, 8-16
- register database コマンド, 8-2, 8-3
- release channel コマンド, 8-11
- replace script コマンド, 8-19
- replicate コマンド, 8-32
- REPORT DELETABLE コマンド, 7-22
- report deletable コマンド, 8-15
- report need backup days コマンド, 8-14
- REPORT NEED BACKUP コマンド, 7-22
- report need backup コマンド, 8-13
- report schema コマンド, 8-16
- REPORT UNRECOVERABLE コマンド, 7-22
- report unrecoverable コマンド, 8-14
- report_object_list オプション
 - report コマンド, 8-13
- REPORT コマンド, 7-22
- report コマンド
 - days オプション, 8-14
 - device_type_list オプション, 8-13
 - incremental オプション, 8-14
 - report_object_list オプション, 8-13

- リカバリ・カタログへの複雑な問合せのための使用
方法, 9-12
- report コマンド, 9-12
- reset database コマンド, 8-4
- RESETLOGS オプション, 12-3, 12-24 – 12-25
 - ALTER DATABASE コマンド
 - 制御ファイルのバックアップ, 11-11
 - データベースのオープン, 3-10
 - データベースの回復, 3-10
 - データベースのバックアップ, 3-5
- restore コマンド
 - from backup オペランド, 8-33
 - from copy オペランド, 8-33
- restore コマンド, 9-8, 9-9, ?? – 9-11, ?? – 9-11
- resync catalog from backup controlfile コマンド, 8-9

S

- SET AUTORECOVERY ON 文, 12-6
- set newname コマンド, 8-30, 8-32, 8-35
- Set Tablespace Offline ダイアログ, 11-9
- Set Tablespace Online ダイアログ, 11-9
- set until 制限
 - setlimit channel コマンド, 8-30, 8-36
- setlimit channel コマンド
 - set until 制限, 8-30, 8-36
- setlimit channel コマンド, 8-11
- setmaxcorrupt 句
 - backup コマンド, 9-6
- setsize パラメータ, 8-24
- SET コマンド
 - LOGSOURCE パラメータ, 12-5
- SHUTDOWN コマンド
 - ABORT オプション, 12-9, 12-22, 12-26, 12-52
- skip inaccessible オプション
 - backup コマンド, 9-7
- skip offline オプション
 - backup コマンド, 9-2, 9-7
- skip read only オプション
 - backup コマンド, 9-2
- SQL コマンド
 - 回復, 4-2, 12-4
 - 表領域を回復するための, 12-6
 - ログ・ファイルの適用, 12-6
- STARTUP コマンド
 - MOUNT オプション, 12-23, 12-27
 - RECOVER オプション, 12-4

- switch コマンド, 8-35
- SYSTEM 表領域
 - ARCHIVELOG モードでのメディア障害, 12-45
 - 回復, 12-17
- SYS が所有するオブジェクト
 - TSPITR, 10-4, 13-4

T

- tablespace オプション
 - list コマンド, 8-17
- TABLESPACE パラメータ
 - RECOVER コマンド, 12-4
- tag オペランド
 - backup コマンド, 8-23
 - list コマンド, 8-18
- TS_PITR_CHECK ビュー, 10-4, 13-4
- TSPITR 「表領域の Point-in-Time 回復」を参照
- type disk オペランド
 - allocate channel コマンド, 8-10

U

- unavailable オペランド
 - change コマンド, 8-7
- uncatalog オペランド
 - change コマンド, 8-7
- USING BACKUP CONTROLFILE パラメータ, 12-23, 12-32

V

- V\$BACKUP, 11-7
- V\$BACKUP_CORRUPTION, 8-21
- V\$BACKUP_DEVICE, 9-2
- V\$DATAFILE, 12-4, 12-27, 12-32
 - バックアップ用のファイルのリスト表示, 11-2
- V\$DATAFILE_HEADER, 9-9
- V\$LOGFILE, 12-47
 - バックアップ用のファイルのリスト表示, 11-2
- V\$LONGOPS, 8-21
- V\$RECOVER_FILE, 12-12

あ

- アーカイバ・プロセス (ARCH), 2-17 – 2-18
- オンライン REDO ログ・ファイルのアーカイブ,

2-14
アーカイブ
 自動, 2-17
 手動, 2-18
 モード
 データベース, 2-16 – 2-19
アーカイブ REDO ログ, 1-7, 2-13 – 2-15
 回復後の削除, 12-14
 回復時の位置, 12-4
 回復時の適用, 12-4, 12-6
 回復中のエラー, 12-8
 時間ベースの回復, 12-51
 消失, 12-51
 多重化, 2-15
 追跡, 7-25
 ディスクへの復元, 12-14
 ログ順序番号の保存, 12-24, 12-28, 12-33
 ログ順序番号のリセット, 12-24, 12-28, 12-33
アーカイブ REDO ログの多重化, 2-15
アーカイブ・オンライン REDO ログ
 グループの消失, 12-50
アーカイブ・ログ
 コピー、リスト, 8-12
 削除, 8-6
 使用可能のマーク付け, 8-7
 使用不能のマーク付け, 8-6
 登録, 8-2
 バックアップ, 9-3
 レコードの削除, 8-6
アクティブでないオンライン REDO ログ
 消失, 12-48
アクティブでないログ・グループの消失, 12-48

い

一貫性のあるデータベース全体のバックアップ, 2-22
一貫性のないデータベース全体のバックアップ
 定義, 2-23
イメージ・コピー, 7-16
 カタログ化, 7-26
 ユーザー作成, 7-17
インスタンス
 起動中の回復, 12-4
 障害, 1-3
インストール
 リカバリ・カタログ, 8-2

え

エクスポート
 完全, 11-14
 増分, 11-14
 モード, 11-13
 累積, 11-14
エラー
 アーカイバ・プロセス, 2-18

お

オープン状態のデータベースのバックアップ
 分裂ブロックの検出, 7-25
オブジェクト・リスト
 recover コマンド, 8-38
オフライン・バックアップ, 3-2
オペレーティング・システムのバックアップ, 5-2
オンライン REDO ログ, 1-5, 2-2 – 2-13, 12-48
 アーカイブ済みグループ, 12-47
 アクティブでないグループ, 12-47
 アクティブ・グループ, 12-47
 アクティブ・グループの消失, 12-50
 回復時の適用, 12-4
 概要, 1-5
 グループの消失, 12-47
 現行のグループ, 12-47
 消去, 12-37
 消失
 回復, 12-46 – 12-50
 すべてのメンバーの消失, 12-47
 多重, 1-5
 バックアップ, 3-9
 バックアップ用のログ・ファイルのリスト表示,
 11-2
 複数のグループの消失, 12-50
 ミラー化したメンバーの消失, 12-47
 メンバーのステータス, 12-47
 リセット
 効果, 12-24
 手順, 12-24 – 12-25
 ログ順序番号の保存, 12-24, 12-33
 ログ順序番号のリセット, 12-24, 12-33
オンライン表領域のバックアップ開始ダイアログ, 11-6
オンライン表領域のバックアップ終了ダイアログ, 11-6
オンライン・バックアップ, 3-2

か

回復セット, 10-3
 すべての表が入っている, 10-4
 プライマリ・データベースへのインポート, 13-15
 プライマリ・データベースへのコピー, 13-15
回復, 9-8
 Import ユーティリティ, 11-14
 Import ユーティリティの使用, 11-14
 PARALLEL_MAX_SERVERS パラメータ, 12-11
 Point-in-Time, 4-6
 データベース, 9-10
 recover コマンド, 9-8
 REDO ログの適用, 12-4
 RESETLOGS オプション使用後, 3-10
 SQL コマンド, 12-4
 オープン状態のデータベース, 9-8
 オープン状態のデータベースでのオフライン表領域
 , 12-4
 回復不能オブジェクトと, 12-43 – 12-44
 完全な回復のための手順, 12-15 – 12-21
 クローズ状態のデータベース, 12-3
 計画, 4-1 – 4-7
 コマンド, 12-11 – 12-12
 Recovery Manager, 8-37
 最起動, 12-8
 削除された表, 12-54
 時間ベース, 4-7, 12-26 – 12-30
 消失した制御ファイル, 12-51
 使用するプロセス数の設定, 12-11
 制御ファイル破損後の, 12-52
 タイプ
 分散データベース・システムおよび, 12-2
 中断, 12-8
 データ・ファイル
 ガイドライン, 8-36
 バックアップがない場合, 12-13
 データベースの起動、起動中, 12-4
 テスト, 4-6
 取消しベース, 4-7
 手順, 12-21
 の準備, 12-11 – 12-15
 パラレル処理, 12-11
 必要な権限, 12-14
 不完全な回復のための手順, 12-21 – 12-34
 分散データベース, 12-2
 スナップショット, 12-3

 変更ベース, 4-7, 12-31 – 12-34
 方針, 4-1 – 4-7
 メディア, 4-7, 12-1
 完全, 4-7
 使用可能, 2-16
 使用禁止, 2-16
 制御ファイル, 2-14
 不完全, 4-7
 メディア障害
 例, 12-44 – 12-54
 読取り専用表領域, 12-44
 ロールフォワード・フェーズ, 12-4
「回復アーカイブ適用」ダイアログ・ボックス, 12-23
 Enterprise Manager, 12-3
回復した表
 バックアップ, 13-16
回復不能オブジェクト
 RECOVER 操作, 12-43
 回復, 12-43 – 12-44
環境変数
 NLS_DATE_FORMAT, 6-10
 NLS_LANG, 6-10
完全エクスポート, 11-14
完全回復
 手順, 12-15

く

グループ、REDO ログ
 アーカイブ REDO ログ, 12-47
 オンライン REDO ログ, 12-47
クローズ状態のデータベース
 回復, 12-3
クローン・データベース, 10-2
 TSPITR 用の準備, 13-13
 TSPITR 用のデータ・ファイルのコピーの使用,
 10-10
 エクスポート, 13-15
 オープン, 13-15
 回復, 13-14
 パラメータ・ファイルの準備, 13-12
 変換されたファイル名, 10-11

け

警告

一貫性と Export バックアップ, 11-13
最初のバックアップのアーカイブ・モード, 3-8
権限
バックアップ
制御ファイル, 11-10

こ

コールド・バックアップ
データベース全体のバックアップ, 11-3
コマンド
LIST, 7-22
REPORT, 7-22
REPORT DELETABLE, 7-22
REPORT NEED BACKUP, 7-22
REPORT UNRECOVERABLE, 7-22
種類, 7-3
ストアド・スクリプトのメンテナンス用, 7-3
バックアップ, 7-3
復元, 7-3
リカバリ・カタログのメンテナンス用, 7-3
コマンド、print script, 8-19
コマンド、Recovery Manager
allocate channel, 8-10, 8-29, 8-36
allocate channel for delete, 9-8
backup, 8-21
catalog, 8-2, 8-8
change, 8-6, 8-7
change...delete, 8-8
copy, 2-27, 8-29
delete script, 8-19
execute script, 8-19
list, 8-12, 9-11
list backupset, 8-18
list copy, 8-18
register database, 8-3
release channel, 8-11
replace script, 8-19
replicate, 8-32
report, 9-12
report deletable, 8-15
report need backup, 8-13
report need backup days, 8-14
report schema, 8-16
report unrecoverable, 8-14
reset database, 8-4
restore, 9-9

resync catalog from backup controlfile, 8-9
set newname, 8-32, 8-35
setlimit channel, 8-11
switch, 8-35

さ

災害時回復
計画, 12-35 — 12-44
削除
Recovery Manager を使うファイル・レコード, 8-7
Recovery Manager を使ったファイル, 8-7
ストアド・スクリプト, 8-19
リカバリ・カタログ・レコード, 8-7
作成
スタンバイ・データベース, 12-35

し

時間ベースの回復, 4-7, 12-26 — 12-30
アーカイブ REDO ログ, 12-51
分散データベース内の調整式の, 12-2
時間ベースの調整式分散データベース回復, 12-2
時刻書式
RECOVER DATABASE UNTIL TIME 文, 12-27
時刻パラメータを使用した Recovery Manager の設定,
6-10
システム時間、変更
回復への影響, 12-21
システム障害, 1-3
システム変更番号 (SCN)
決まる時点, 2-2
形式, 12-32
定義, 4-3
分散回復での使用方法, 12-3
自動アーカイブ, 2-17
手動アーカイブ, 2-18
順次メディア
バックアップ, 5-5
障害
REDO ログ・ファイルのアーカイブ, 2-18
インスタンス, 1-3
システム, 1-3
チェックポイント処理中, 2-8
ディスク, 1-4
ハードウェア, 1-3
プロセス, 1-3

- 文, 1-3
- メディア, 1-4, 12-44
 - 多重オンライン REDO ログ・ファイル, 2-9
- ユーザー・エラー, 1-3
- 使用例
 - Recovery Manager, 9-1 — 9-12
- 初期化パラメータ
 - COMPATIBLE, 12-41
 - CONTROL_FILES, 8-32, 12-52
 - プライマリ・データベースおよびスタンバイ・データベースの場合, 12-40
 - DB_FILE_STANDBY_NAME_CONVERT, 12-37
 - プライマリ・データベースおよびスタンバイ・データベースの場合, 12-40
 - DB_NAME, 2-20
 - LOG_ARCHIVE_DEST, 12-5, 12-23, 12-27
 - LOG_ARCHIVE_FORMAT, 12-5
 - LOG_ARCHIVE_START, 2-17
 - LOG_CHECKPOINT_INTERVAL, 2-6
 - LOG_CHECKPOINT_TIMEOUT, 2-6
 - LOG_CHECKPOINTS_TO_ALERT, 2-9
 - LOG_FILE_STANDBY_NAME_CONVERT, 12-37
 - PARALLEL_MAX_SERVERS, 12-11
 - RECOVERY_PARALLELISM, 12-11
 - プライマリ・データベースおよびスタンバイ・データベースの場合, 12-40

す

- スキーマ変更
 - リカバリ・カタログの更新, 8-5
- スクリプト
 - catrman.sql, 8-3
- スクリプト、格納された
 - 削除, 8-19
 - 作成, 8-19
 - 出力, 8-19
- スタンバイ・データベース, 12-35 — 12-44
 - アクティブ化, 12-38
 - 作成, 12-35
 - の制御ファイル, 12-35
 - プライマリ・データベースの変更の影響, 12-39
 - マウント, 12-38
 - メンテナンス, 12-36
- ストアド・スクリプト, 7-9, 8-19
 - Recovery Manager, 6-10
 - 削除, 8-19

- 作成, 8-19
- 出力, 8-19
- スナップショット
 - 分散データベース回復および, 12-3
 - メディア回復, 12-3
- スナップショット制御ファイル, 7-8
- スナップショット表
 - TSPITR, 10-5, 13-4
- スレッド
 - オンライン REDO ログ, 2-13

せ

- 制御ファイル, 1-4, 2-19 — 2-20
 - CONTROL_FILES パラメータ
 - プライマリ・データベースおよびスタンバイ・データベースの場合, 12-40
 - アーカイブ REDO ログ情報, 2-14
 - 記録される変更, 2-20
 - 時間ベースの回復のとき, 12-26
 - 消失, 12-51, 12-52
 - スタンバイ・データベース, 12-35
 - 多重, 2-20
 - チェックポイント情報, 2-20
 - 定義, 1-4, 4-3
 - データ・ディクショナリとの矛盾, 12-25, 12-29
 - 内容, 1-4, 2-19
 - バックアップ, 9-3, 11-2, 11-10
 - バックアップの権限, 11-10
 - ファイル名の検索, 11-2
 - 不完全回復時, 12-22, 12-31
 - 復元の宛先, 8-31
 - ミラー化した、消失, 12-52
 - メディア回復, 2-14
 - 目的, 1-4
 - ログ順序番号, 2-5
- 制御ファイルのバックアップ
 - 定義, 2-24
- 整合性チェック, 7-25
- 制約
 - バックアップ, 7-24
 - 復元, 7-24
- 全体バックアップ, 7-11

そ

- 増分エクスポート, 11-14

増分バックアップ, 7-12, 9-6
Recovery Manager, 8-22

た

多重

REDO ログ・ファイル, 2-9
制御ファイル, 2-20

多重化

アーカイブ REDO ログ, 2-15

多重バックアップ・セット, 7-21

ち

チェックポイント, 2-5 – 2-9

一定時間後, 2-6

インスタンスの停止, 2-6

オンライン REDO ログ・ファイル, 2-5

概要, 2-5

強制実行, 2-6

高速, 2-8

処理中のイベント, 2-8

制御ファイル, 2-20

増分, 2-7

タイプ, 2-6

チェックポイント・プロセス (CKPT), 2-8

通常, 2-8

定義, 4-4

データベース, 2-6

データ・ファイル, 2-6

パフォーマンスへの影響, 2-5

表領域をオフラインにした後, 2-6

ログ・スイッチ発生時, 2-6

チャンネル

割当て, 7-18

チャンネル制御, 7-18, 8-9

チャンネルの割当て, 8-9

中断後の回復の再開, 12-8

て

停止

チェックポイント, 2-6

ディスク障害, 1-4

データベース

db 識別子, 8-3

Point-in-Time 回復, 9-10

RESETLOGS オプション使用後の回復, 3-10

アーカイブ・モード, 2-16 – 2-19

インカーネーション

リスト, 8-13

回復

制御ファイル破損後の, 12-52

手順, 12-1 – 12-54

起動

回復, 12-4

クローズ状態

回復, 12-3

災害時回復の計画, 12-35 – 12-44

スタンバイ, 12-35 – 12-44

アクティブ化, 12-38

作成, 12-35

メンテナンス, 12-36

増分バックアップ

Recovery Manager, 8-22

データベース全体のバックアップ, 11-3 – 11-4

Recovery Manager, 8-22

バックアップ, 9-2

Recovery Manager, 8-20

バックアップのための停止, 9-2

ファイル

バックアップのためのリスト表示, 11-2

物理構造, 1-4 – 1-7, 4-5, 6-12

物理構造の変更

スタンバイ・データベースへの影響, 12-39

モード

NOARCHIVELOG, 2-16

リカバリ・カタログとの再同期化, 8-4

リカバリ・カタログへの登録, 8-2, 8-3

「データベース起動」ダイアログ・ボックス, 12-27

データベース構造

物理的, 6-12

データベース全体のバックアップ, 11-3 – 11-4

一貫性のある, 2-22

一貫性のない, 2-23

回復、データベース全体のバックアップから, 12-9

定義, 2-21

「データベース」停止ダイアログ・ボックス

異常終了停止モード, 12-9

「データベース停止」ダイアログ・ボックス

Enterprise Manager, 12-52

異常終了停止モード, 12-22, 12-26

データベースのオープン

RESETLOGS オプション, 3-10

- データベースの起動
 - 回復, 12-4
- データベースの構造
 - REDO ログ・ファイル, 1-5
 - アーカイブ REDO ログ, 1-7
 - オンライン REDO ログ, 1-5
 - 制御ファイル, 1-4
 - データ・ファイル, 1-5
 - 物理的, 1-4 – 1-7, 4-5
 - ロールバック・セグメント, 1-6
- データベースの物理構造, 1-4 – 1-7, 4-5, 6-12
- データベース・バッファ
 - チェックポイント, 2-5
- データベース・ライター・プロセス (DBWR)
 - チェックポイント, 2-5
- データ・ディクショナリ
 - 制御ファイルとの矛盾, 12-25, 12-29
 - ビュー, 11-5
- データ・ファイル, 1-5
 - 回復
 - ガイドライン, 8-36
 - バックアップがない場合, 12-13
 - 回復後の欠落, 12-25, 12-29
 - 回復のための作成, 12-13
 - 回復不能、リスト, 8-14
 - 概要, 1-5
 - 切替え, 8-35
 - コピー, 2-27
 - ディスク, 9-5
 - 利点, 8-28
 - コピーの削除, 8-6
 - コピーの使用可能のマーク付け, 8-7
 - コピーの使用不能のマーク付け, 8-6
 - コピー、リスト, 8-12, 8-18
 - 削除するデータ・ファイルのコピー、リスト表示, 8-15
 - 消失, 12-45
 - 使用方法, 1-6
 - スタンバイ使用の名前の変換, 12-37
 - 制御ファイルに指定されている, 2-19
 - 登録, 8-2
 - バックアップ, 2-24, 2-27, 9-3
 - Recovery Manager, 8-20
 - オフライン, 11-5, 11-8
 - バックアップが必要、リスト表示, 8-14, 8-15
 - バックアップ、リスト, 8-12
 - バックアップ・ステータスのチェック, 11-7

- バックアップ・セット
 - リスト, 8-17
- 必要なバックアップのリスト表示, 8-14
- 表示
 - 回復が必要なファイル, 12-12
 - バックアップ・ステータス, 11-7
- 復元, 8-30
 - オープン状態のデータベース, 9-8
 - ガイドライン, 8-32
- 復元の宛先, 8-31
- プライマリ・データベースへの追加
 - スタンバイ・データベースへの影響, 12-39
- 命名
 - スタンバイ・データベースへの影響, 12-39
- リスト, 8-16
 - バックアップ用, 11-2
 - レコードの削除, 8-6
- データ・ファイルの復元, 8-30
- デバイス
 - バックアップ、検索, 9-2

と

- 問合せ
 - リカバリ・カタログ, 9-12
- 登録
 - アーカイブ・ログ, 8-2
 - データ・ファイル, 8-2
- トランザクション
 - コミット
 - REDO ログ・バッファの書込み, 2-2
 - コミットされていない, 1-6
 - ロールバック, 1-6
 - トランザクションのコミット
 - REDO ログ・バッファの書込み, 2-2
- 取消しベースの回復, 4-7
 - 手順, 12-21
- トレース・ファイル
 - ARCH, 2-18
 - 制御ファイルのバックアップの作成, 11-10
 - ログ・ライター・プロセス, 2-11

は

- パーティション化されていないグローバル索引
 - TSPITR, 10-5, 13-5
- パーティション表

- TSPITR, 10-5, 13-4
- 削除されたパーティション, 13-21
- 部分的な TSPITR の実行, 13-16
- 分割パーティション, 13-25
- ハードウェア障害, 1-3
- パスワード・ファイル
 - Recovery Manager の使用方法, 6-6
 - 使用して Recovery Manager に接続する場合, 6-7
 - 使用せずに Recovery Manager に接続する場合, 6-7
- 破損データ・ファイル・ブロック
 - 制御ファイルのレコード, 8-21
 - バックアップの最大値の設定, 9-6
- 破損の検出, 7-17
- バックアップ
 - ARCHIVELOG モード, 3-8
 - DB_VERIFY ユーティリティ, 11-4
 - Export ユーティリティ, 2-28, 3-6
 - NOARCHIVELOG モード, 3-7, 9-2
 - Parallel Server 環境, 9-5
 - Recovery Manager によるバックアップのタイプ, 7-9
 - Recovery Manager によるバックアップのためのユーザー・タグ, 7-23
 - RESETLOGS オプション使用後, 3-5
 - アーカイブ・ログ, 9-3
 - 一貫性のあるデータベース全体のバックアップ, 2-22
 - 一貫性ないデータベース全体, 2-23
 - イメージ・コピー, 7-9, 7-16
 - オペレーティング・システム, 5-2
 - オンライン REDO ログ, 3-9
 - ガイドライン, 3-2
 - 回復、データベース全体のバックアップ, 12-9
 - 計画の作成, 3-7
 - 計画のテスト, 3-7
 - 個々のデータ・ファイル, 9-3
 - 指定リスト, 8-24
 - 重要性, 1-2
 - 終了点のマーク, 11-6
 - 順次メディア, 5-5
 - 使用される Export ユーティリティ, 5-3
 - 使用する Enterprise Backup Utility, 5-3
 - 使用例、Recovery Manager, 8-28
 - ストアド・スクリプトの使用, 8-19
 - 制御ファイル, 2-24, 9-3, 11-10
 - 制御ファイル用の権限, 11-10
 - 制約, 7-24
 - 全体, 7-11
 - 増分, 9-6
 - Recovery Manager, 8-22
 - 定義, 4-4
 - 増分, 7-12
 - タイプ, 2-21 – 2-25, 11-2 – 11-14
 - 定義, 1-2, 4-3
 - ディスク, 5-4
 - ディスク、Recovery Manager の使用, 8-10
 - データベース, 9-2
 - データベース作成前の計画, 3-2
 - データベース全体, 2-21, 11-3 – 11-4
 - Recovery Manager, 8-22
 - データベースの構造変更時, 3-3
 - データ・ファイル, 2-24
 - データ・ファイルのバックアップ・ステータスのチェック, 11-7
 - 手順, 11-2 – 11-14
 - オフラインのデータ・ファイル, 11-8
 - オフライン表領域, 11-8
 - オンラインのデータ・ファイル, 11-5
 - オンライン表領域, 11-5
 - デバイスの指定, 8-10
 - バックアップ・セット, 7-9
 - 必要なファイルのリスト, 11-2
 - 表領域, 3-3, 9-2, 9-3, 11-8
 - 非累積増分, 7-13
 - 頻度, 3-2
 - 分散データベース, 3-6
 - 方法の選択, 5-1 – 5-7
 - 保存, 3-4, 9-7
 - リカバリ・カタログ, 6-6
 - 累積増分, 7-12, 7-15
 - レポートの生成, 8-12
 - 論理
 - 定義, 2-28
 - バックアップ指定リスト, 8-24
 - バックアップ制御ファイル
 - スナップショット, 7-8
 - バックアップ中の表領域のスキップ, 9-2
 - バックアップの制約, 7-24
 - バックアップ方法, 5-1 – 5-7
 - バックアップ・オブジェクト・リスト, 8-24
 - バックアップ・セット, 7-9
 - 圧縮, 7-10
 - 全体, 7-11
 - 増分, 7-11
 - 多重, 7-21

- バックアップ・ピース, 7-10
- ファイルの数の制限, 8-20
- 編成, 8-21
- マルチピースの削除, 9-7
- リスト, 8-12, 8-17, 8-18
- バックアップ・ピース, 7-10
 - 削除, 8-6
 - 使用可能のマーク付け, 8-7
 - 使用不能のマーク付け, 8-6
 - 命名, 8-20
 - レコードの削除, 8-6
- バックアップ・ファイル
 - ユーザー作成
 - カタログ化, 8-8
- パフォーマンス
 - チェックポイントの効果, 2-5
- パラレル化, 7-19
 - 度合いを左右する要素, 7-19
- パラレル回復, 12-11

ひ

- ビットマップ索引
 - TSPITR, 10-5, 13-5
- ビュー
 - DBA_DATA_FILES, 11-5
 - V\$BACKUP, 11-7
 - V\$BACKUP_CORRUPTION, 8-21
 - V\$DATAFILE, 11-2, 12-4, 12-27, 12-32
 - V\$LOGFILE, 11-2, 12-47
 - V\$LONGOPS, 8-21
 - V\$RECOVERFILE, 12-12
 - データ・ディクショナリ, 11-5
- 表
 - 削除
 - 回復, 12-54
- 表領域
 - TSPITR 実行後のバックアップ, 10-9
 - オープン状態のデータベースでのオフラインの回復, 12-4
 - オンラインでのバックアップ, 11-5, 11-8
 - データ・ファイルを含むバックアップ・セットのリスト表示, 8-17
 - バックアップ, 9-2, 9-3
 - バックアップ中のスキップ, 9-2
 - バックアップとチェックポイント, 2-6
 - バックアップの頻度, 3-3

- 復元
 - オープン状態のデータベース, 9-8
 - 複数のオンライン表領域のバックアップ, 11-8
 - 読取り専用
 - 回復への影響, 12-44
 - バックアップ, 9-2
- 表領域の Point-in-Time 回復
 - 概要, 10-2, 13-2
 - 計画, 10-3, 10-6, 13-3
 - 実行, 10-8, 13-1 – 13-29
 - すべての表が入っている回復セット, 10-4
 - 制限事項, 10-3, 13-4
 - 制約事項, 10-5, 13-5
 - チューニングに関する考慮事項, 10-9, 13-28
 - 回復セットの位置, 13-28
 - バックアップ制御ファイル, 13-29
 - 要件, 13-6
- 非累積増分バックアップ, 7-13

ふ

- ファイル
 - 使用不能のマーク付け, 8-7
- ファイルの回復, 1-5
- ファイル名
 - バックアップのためのリスト表示, 11-2
- 不完全回復
 - 時間ベースの, 12-26 – 12-30
 - 手順, 12-21 – 12-34
 - 変更ベースの, 12-31 – 12-34
- 復元, 9-8
 - オープン状態のデータベース, 9-8
 - データベース全体のバックアップ, 12-9
 - データ・ファイル
 - ガイドライン, 8-32
 - バックアップ・ファイルの選択, 8-31
- 復元指定, 8-34
 - リスト, 8-34
- 復元の宛先
 - 制御ファイル, 8-31
 - データ・ファイル, 8-31
- 復元の制約, 7-24
- プライマリ・データベース
 - TSPITR 用に準備する, 13-12
 - 使用のための準備, 13-16
- プロセス障害, 1-3
- 分散データベース

- 回復, 12-2
- 時間ベースの調整式回復, 12-2
- バックアップ, 3-6
- 変更ベースの回復, 12-2
- メディア回復とスナップショット, 12-3
- 分散データベース・システム
- 回復, 12-2
- 文障害, 1-3
- 分裂ブロックの検出, 7-25

へ

- 変更ベースの回復, 4-7, 12-31 – 12-34
- 分散データベース・システム内の調整式, 12-2

ほ

- 補助セット, 10-3
- 表領域内のデータ・ファイルの命名, 10-9

み

- ミラー化したオンライン REDO ログ, 2-10
- 消失, 12-47
- メンバーの消失, 12-47
- ミラー化した制御ファイル
- 消失, 12-52

め

- メディア回復, 4-7
- NOARCHIVELOG モード, 12-9
- Recovery Manager の使用方法, 8-35
- REDO ログ・ファイルのエラー, 12-8
- SYSTEM 表領域, 12-17
- アーカイブ REDO ログの適用, 12-4
- インスタンス起動時, 12-4
- エラー・メッセージ, 12-8
- オープン状態のデータベースのオフライン表領域, 12-17
- オンライン REDO ログ・ファイル, 12-46
- 開始, 12-14
- 回復が必要なファイルの決定, 12-12
- 完全なメディア回復のための手順, 12-15 – 12-21
- 完了, 12-17, 12-19, 12-21
- コマンド, 12-11 – 12-12
- 時間ベース, 12-21

- 時間ベースの不完全な, 12-26 – 12-30
- 消失したファイル
- 消失したアーカイブ REDO ログ・ファイル, 12-51
- 消失した制御ファイル, 12-51
- 消失したデータ・ファイル, 12-45
- 消失したミラー化制御ファイル, 12-52
- スナップショットと, 12-3
- 正常に適用された REDO ログ, 12-7
- 中断, 12-8
- 中断後の再開, 12-8
- 取消しベース, 12-21
- の準備, 12-11 – 12-15
- 破損していない表領域のオンライン, 12-17
- 不完全なメディア回復のための手順, 12-21 – 12-34
- 不完全メディア回復の必要条件, 12-28
- 不完全メディア回復の要件, 12-24, 12-33
- 復元
- アーカイブ REDO ログ・ファイル, 12-14
- データベース全体のバックアップ, 12-9
- 破損したファイル, 12-13
- 分散データベース
- 時間ベースの調整式, 12-2
- 変更ベース, 12-21, 12-31 – 12-34
- メディア回復の割込み, 12-8
- メディア管理ライブラリ (MML), 5-6
- メディア障害, 1-4
- NOARCHIVELOG モード, 12-9
- アーカイブ REDO ログ・ファイルの消失, 12-51
- オンライン REDO ログの消失, 12-46
- オンライン REDO ログ・グループの消失, 12-47
- 回復
- 分散データベース, 12-2
- 回復手順
- 例, 12-44 – 12-54
- 完全な回復手順, 12-15 – 12-21
- 制御ファイルの消失, 12-51, 12-52
- 説明, 12-45
- データ・ファイルの消失, 12-45

も

- モード
- ARCHIVELOG, 1-7, 2-16
- NOARCHIVELOG, 1-7, 2-16
- 障害からの回復, 12-9

ゆ

- ユーザー作成バックアップ・ファイル
 - カタログ化, 8-8
- ユーザー・エラー
 - 回復、ユーザー・エラーから, 12-53
 - データベース障害, 1-3
- ユーザー・タグ, 7-23
- ユーティリティ
 - OS、コピー作成のための使用方法, 9-7
 - Recovery Manager, 7-2

よ

- 読取り一貫性
 - Export ユーティリティ, 11-13
- 読取り専用表領域
 - 回復, 12-44
 - 回復への影響, 12-44
 - バックアップ, 9-2

り

- リカバリ・カタログ, 6-2 – 6-6, 7-5, 8-2 – 8-9
 - インストール, 8-2
 - 更新, 8-4
 - スキーマの変更後, 8-5
 - 再同期化, 6-6, 8-4
 - 使用して Recovery Manager に接続する場合, 6-8
 - 使用せずに Recovery Manager に接続する場合, 6-7
 - 消失または破損したりカバリ・カタログの回復, 8-9
 - スキーマ, 6-2
 - 設定, 6-5
 - 定義, 4-4
 - データベースの登録, 8-2, 8-3
 - 問合せ, 8-12, 9-11
 - なしでの運用, 7-7
 - バックアップ, 6-6
 - 複雑な問合せのための report コマンドの使用法, 9-12
 - 領域要件, 6-3
 - レコードの削除, 8-7
 - ログ・スイッチのレコード, 8-6
- リカバリ・カタログの再同期化, 6-6

る

- 累積エクスポート, 11-14
- 累積増分バックアップ, 7-12

れ

- レポート
 - 生成, 8-12
- レポート生成, 7-22

ろ

- ロールバック・セグメント, 1-6
 - TSPITR, 10-4
 - 定義, 4-3
- ロールフォワード, 1-5
- ログ順序番号
 - 1 にリセット, 12-24, 12-28, 12-33
 - 回復後の保存, 12-24, 12-28, 12-29, 12-33
 - 回復後のリセット, 12-24, 12-29
 - 回復時に要求された, 12-4
 - 制御ファイル, 2-5
 - 多重 REDO ログ, 2-10
- ログ・スイッチ
 - 説明, 2-5
 - ミラー化した REDO ログ・ファイル, 2-10
 - ログ順序番号, 2-5
- ログ・スイッチのレコード
 - リカバリ・カタログ, 8-6
- ログ・ファイル
 - スタンバイ使用の名前の変換, 12-37
- ログ・ライター・プロセス (LGWR)
 - アーカイバ・プロセス (ARCH), 2-14
 - オンライン REDO ログ・ファイルへの書込み, 2-2, 2-3
 - 手動アーカイブ, 2-18
 - 使用可能なオンライン REDO ログ, 2-3
 - 多重 REDO ログ・ファイル, 2-10
 - トレース・ファイル, 2-11
- 論理バックアップ
 - 定義, 2-28