

Oracle8i

バックアップおよびリカバリ・ガイド

リリース 8.1

Oracle8i バックアップおよびリカバリ・ガイド リリース 8.1

部品番号 : A62764-1

第 1 版 1999 年 5 月 (第 1 刷)

原本名 : Oracle8i Backup and Recovery Guide, Release 8.1.5

原本部品番号 : A67773-01

原本著者 : Connie Dialeris, Joyce Fee, Lance Ashdown

原本協力者 : Greg Pongracz, Francisco Sanchez, Steve Wertheimer

Copyright © 1999, Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

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

当ソフトウェア (プログラム) のリバース・エンジニアリングは禁止されています。

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

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

危険な用途への使用について

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

Restricted Rights Legend

Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication and disclosure of the Programs shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-14, Rights in Data -- General, including Alternate III (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

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

ドラフトのアルファ版およびベータ版ドキュメント

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

目次

はじめに	xix
------------	-----

第 I 部 バックアップおよび回復計画の立案

1 バックアップおよび回復

バックアップおよび回復.....	1-2
Oracle におけるデータベース・トランザクションの記録保持の仕組み	1-4
バックアップおよび回復操作.....	1-4
バックアップおよび回復の基本計画.....	1-5
バックアップおよび回復の実行時に重要となるデータ構造.....	1-6
データ・ファイル.....	1-6
制御ファイル.....	1-7
ロールバック・セグメント.....	1-8
オンライン REDO ログ・ファイル	1-8
アーカイブ REDO ログ・ファイル	1-10
バックアップ計画の基本の理解.....	1-11
バックアップの重要性.....	1-11
発生しうる障害のタイプ.....	1-11
バックアップ対象について.....	1-13
バックアップ方法の選択.....	1-13
一貫性バックアップと非一貫性バックアップの選択.....	1-16
バックアップの回数.....	1-17
回復計画の基本の理解.....	1-17
回復の仕組み.....	1-17
回復のタイプ.....	1-18

メディア回復.....	1-19
回復方法の選択.....	1-23

2 データ構造の管理

バックアップおよび回復のデータ構造の概要	2-2
制御ファイルの管理	2-3
制御ファイル情報の表示.....	2-4
構造変更後の制御ファイルのバックアップ.....	2-5
複数の制御ファイルのメンテナンス.....	2-7
制御ファイルの消失からの回復.....	2-9
オンライン REDO ログの管理	2-11
オンライン REDO ログ情報の表示	2-12
オンライン REDO ログ・ファイルの多重化	2-12
アーカイブ REDO ログの管理	2-14
NOARCHIVELOG モードおよび ARCHIVELOG モードの選択.....	2-15
アーカイブ REDO ログ情報の表示	2-18
複数の場所への REDO ログのアーカイブ	2-19

3 バックアップおよび回復計画の立案

バックアップ・タイプの選択	3-2
全体データベース・バックアップ.....	3-2
表領域のバックアップ.....	3-6
データ・ファイルのバックアップ.....	3-8
制御ファイルのバックアップ.....	3-8
バックアップ方法の選択	3-8
Recovery Manager	3-9
オペレーティング・システム (O/S).....	3-10
Export.....	3-11
Enterprise Backup Utility.....	3-11
各バックアップ方法の機能の比較.....	3-11
バックアップ形式の選択	3-12
バックアップ・セット.....	3-13
イメージ・コピー	3-14
オペレーティング・システムによるバックアップ.....	3-14
論理バックアップ.....	3-14

バックアップ計画の立案	3-15
ARCHIVELOG または NOARCHIVELOG の運用モードの決定.....	3-15
多重化された制御ファイル、オンライン REDO ログおよびアーカイブ REDO ログ.....	3-18
頻繁で定期的なバックアップの実行.....	3-18
構造変更時のバックアップの実行.....	3-19
使用回数の多い表領域の頻繁なバックアップ.....	3-19
回復不能な操作の後のバックアップ.....	3-20
RESETLOGS オプション使用後のデータベース全体のバックアップ.....	3-20
古いバックアップの保管.....	3-21
分散データベースにおけるバックアップの制約.....	3-22
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.....	3-23
オンライン REDO ログのバックアップ.....	3-23
回復計画の立案	3-24
バックアップおよび回復計画のテスト.....	3-24
非メディア障害への対処.....	3-25
メディア障害からの回復.....	3-27

第 II 部 Recovery Manager の使用方法

4 Recovery Manager の概要

Recovery Manager の概要	4-2
Recovery Manager の機能.....	4-4
Recovery Manager の機能の確認.....	4-4
Recovery Manager のコマンド	4-5
Recovery Manager PL/SQL パッケージ.....	4-5
Recovery Manager におけるコマンドのコンパイルおよび実行.....	4-6
Recovery Manager コマンドのタイプ.....	4-7
Recovery Manager コマンドのユーザー例外.....	4-9
Recovery Manager のコマンド・エラー.....	4-12
Recovery Manager のメタデータ	4-12
メタデータのリカバリ・カタログへの格納.....	4-13
メタデータの制御ファイルへの格納.....	4-15
メディア管理	4-16
メディア・マネージャを使用したバックアップおよび復元処理.....	4-17
メディア・マネージャのクロスチェック.....	4-17

プロキシ・コピー	4-18
メディア・マネージャのテスト	4-19
Backup Solutions Program	4-19
リストとレポート	4-20
バックアップおよびコピーのリスト	4-21
バックアップ、コピーおよびデータベース・スキーマに関するレポート	4-21
チャンネルの割当て	4-24
チャンネル制御オプション	4-25
チャンネルの平行化	4-26
バックアップ・セット	4-28
バックアップ・セットの格納	4-29
バックアップ・セットの圧縮	4-30
バックアップ・ピースのファイル名	4-30
バックアップ・ピースのサイズ	4-31
バックアップ・セットの数	4-31
多重バックアップ・セット	4-32
二重バックアップ・セット	4-33
バックアップの平行化	4-34
バックアップ・エラー	4-36
バックアップ・タイプ	4-36
全体バックアップ	4-38
増分バックアップ	4-38
バックアップの制約	4-44
イメージ・コピー	4-45
RMAN イメージ・コピー	4-45
O/S イメージ・コピー	4-45
バックアップおよびイメージ・コピーのタグ	4-46
ファイルの復元	4-47
復元処理時のファイル選択	4-48
復元の制約	4-48
メディア回復	4-49
増分バックアップおよび REDO レコードの適用	4-51
不完全回復	4-51
表領域の Point-in-Time 回復	4-51
データベースの複製	4-52
整合性チェック	4-53

物理ブロックの破損の検出.....	4-54
論理ブロックの破損の検出.....	4-54
オープン・バックアップ時の分裂ブロックの検出.....	4-55

5 Recovery Manager のスタート・ガイド

Recovery Manager のセットアップ	5-2
パスワード・ファイルの使用.....	5-2
NLS 環境変数の設定.....	5-2
スナップショットの制御ファイル位置の決定.....	5-3
リカバリ・カタログの使用・不使用の決定	5-4
リカバリ・カタログを RMAN メタデータに使用した場合.....	5-4
制御ファイルを RMAN メタデータに使用した場合.....	5-6
RMAN への接続	5-8
リカバリ・カタログを使用せずに RMAN に接続する場合.....	5-9
リカバリ・カタログを使用して RMAN に接続する場合.....	5-10
補助データベースへの接続.....	5-11
RMAN からの切断.....	5-12
基本的な RMAN コマンドの使用法	5-12
RMAN への接続.....	5-13
データベースのマウント.....	5-14
現行スキーマのレポート.....	5-14
データ・ファイルのコピー.....	5-15
表領域のバックアップ.....	5-16
バックアップおよびコピーのリスト表示.....	5-17
復元の妥当性チェック.....	5-18
メディア・マネージャの構成	5-19
メディア・マネージャとのリンク.....	5-19
一意的なファイル名の生成.....	5-19
ファイル・サイズの制限設定.....	5-20
デバイス固有文字列のメディア・マネージャへの送信.....	5-20
メディア・マネージャのトラブルシューティング.....	5-20
サンプル・スクリプトの使用法および使用例	5-21

6 Recovery Manager のメタデータの管理

リカバリ・カタログの作成.....	6-2
RMAN メタデータのメンテナンス	6-4
リカバリ・カタログへのターゲット・データベースの登録.....	6-4
リカバリ・カタログからのデータベースの登録解除.....	6-6
リカバリ・カタログのリセット.....	6-7
バックアップまたはファイル・コピーの可用性の変更.....	6-8
RMAN メタデータのクロスチェック.....	6-9
バックアップおよびコピーの削除と、RMAN メタデータにおけるそれらの状態の更新.....	6-13
バックアップおよびコピーの復元の妥当性チェック.....	6-18
リカバリ・カタログへのスクリプトの格納	6-21
リカバリ・カタログの再同期化.....	6-23
制御ファイル内のレコードの管理.....	6-27
O/S バックアップのカタログ化.....	6-28
リカバリ・カタログのバックアップと回復	6-30
リカバリ・カタログのバックアップ.....	6-31
リカバリ・カタログの回復.....	6-32
リカバリ・カタログの再作成.....	6-34
リカバリ・カタログのアップグレード	6-34
リカバリ・カタログの削除	6-35
リカバリ・カタログを使用しない RMAN の管理方法	6-36
制御ファイルのメタデータのメンテナンス.....	6-36
制御ファイルのバックアップ.....	6-37

7 Recovery Manager によるリストおよびレポートの生成

バックアップおよび回復計画におけるリストとレポートの使用.....	7-2
リストの生成.....	7-2
レポートの生成.....	7-5
リストとレポートの例	7-10
バックアップおよびコピーのリストの作成.....	7-10
リストを使用した不要なバックアップおよびコピーの判別.....	7-10
バックアップの必要なデータ・ファイルのレポート.....	7-11
回復不能データ・ファイルのレポート.....	7-11
不要なバックアップおよびコピーのレポート.....	7-11
不要なバックアップの削除.....	7-12

データベース・スキーマの履歴レポートの生成.....	7-13
----------------------------	------

8 Recovery Manager でのバックアップおよびコピーの作成

バックアップの作成.....	8-2
一貫性バックアップおよび非一貫性バックアップの作成.....	8-3
全体データベース・バックアップの作成.....	8-3
表領域およびデータ・ファイルのバックアップ.....	8-4
制御ファイルのバックアップ.....	8-6
アーカイブ REDO ログのバックアップ	8-8
増分バックアップの作成.....	8-9
イメージ・コピーの作成.....	8-10
バックアップおよびコピー処理の監視.....	8-12
サーバー・セッションとチャンネルの関連付け.....	8-12
進行状況の監視.....	8-14
パフォーマンスの監視.....	8-14
バックアップおよびコピーの例.....	8-14
バックアップに必要なデータ・ファイルのレポート.....	8-15
データベースのバックアップ時のファイルのスキップ.....	8-15
バックアップの複数ディスク・ドライブへの分散.....	8-16
大規模データベースの複数のファイル・システムへのバックアップ.....	8-16
バックアップ・セットのサイズ指定.....	8-17
バックアップにおけるデータ・ファイルの多重化.....	8-18
アーカイブ REDO ログのバックアップ	8-19
アーカイブ REDO ログの複数のコピーのバックアップおよび削除	8-20
差分増分バックアップの実行.....	8-21
累積増分バックアップの実行.....	8-22
バックアップ・セットの二重化.....	8-22
バックアップ・セットの平行化.....	8-23
NOARCHIVELOG モードでのバックアップ	8-24
平行・サーバー環境でのバックアップ.....	8-25
O/S コピーのカタログ化.....	8-25
バックアップおよびコピーのメンテナンス.....	8-26
バックアップおよびコピー時のエラー処理.....	8-26

9 Recovery Manager による復元と回復

データ・ファイル、制御ファイルおよびアーカイブ REDO ログの復元.....	9-2
データベースの復元.....	9-2
表領域およびデータ・ファイルの復元.....	9-9
制御ファイルの復元.....	9-11
アーカイブ REDO ログの復元	9-13
不完全回復準備のための復元.....	9-15
データ・ファイルの回復.....	9-15
メディア回復の準備.....	9-16
完全回復の実行.....	9-17
不完全回復の実行.....	9-23
復元および回復の使用例.....	9-29
データ・ファイル・コピーを使用した新しいホストへの復元.....	9-30
複数データベースが同じ名前を共有する場合の復元.....	9-30
バックアップ・セットからの制御ファイルの変則的な復元の実行.....	9-32
オープン状態のデータベースにおけるアクセス不可能なデータ・ファイルの回復.....	9-33
ディスクおよびテープのバックアップを使用してのアクセス不可能なデータ・ファイルの回復..	9-34
データベース全体のメディア障害に対する回復の実行.....	9-35
RESETLOGS 以前のバックアップの回復.....	9-37
NONARCHIVELOG モードでのデータベースの回復.....	9-38

10 Recovery Manager による複製データベースの作成

複製データベース作成の概要.....	10-2
制限事項.....	10-3
複製データベース用のファイルの作成.....	10-3
複製用の補助インスタンスの準備.....	10-7
複製データベースのローカルまたはリモート・ホストへの作成.....	10-10
同じディレクトリ構造を持つリモート・ホストへのデータベースの複製.....	10-10
異なるディレクトリ構造のリモート・ホストへのデータベースの複製.....	10-12
ローカル・ホストへの複製データベースの作成.....	10-16
複製データベースの使用例.....	10-17
新しいファイル名の手作業での設定.....	10-17
複製データベースとターゲット・データベースの再同期化.....	10-18
最新でない複製データベースの作成.....	10-19

第 III 部 Recovery Manager リファレンス

11 Recovery Manager のコマンド構文

このリファレンス・マニュアルでの表記法	11-2
コマンド入力	11-4
RMAN コマンドの概要	11-5
allocate	11-9
allocateForMaint	11-13
alterDatabase	11-15
archivelogRecordSpecifier	11-17
backup	11-21
catalog	11-32
change	11-35
cmdLine	11-39
completedTimeSpec	11-42
connect	11-44
connectStringSpec	11-46
copy	11-48
createCatalog	11-52
createScript	11-54
crosscheck	11-57
datafileSpec	11-60
debug	11-62
deleteExpired	11-63
deleteScript	11-65
deviceSpecifier	11-66
dropCatalog	11-68
duplicate	11-69
host	11-74
list	11-76
listObjList	11-85
printScript	11-87
recover	11-89
register	11-94
release	11-96
releaseForMaint	11-97
replaceScript	11-98

replicate	11-101
report	11-103
reset	11-111
restore	11-113
resync	11-119
rmanCmd	11-122
run	11-125
send	11-128
set	11-130
set_run_option	11-134
shutdown	11-138
sql	11-141
startup	11-143
switch	11-145
untilClause	11-147
upgradeCatalog	11-149
validate	11-151

12 リカバリ・カタログ・ビュー

RC_ARCHIVED_LOG	12-2
RC_BACKUP_CONTROLFILE	12-3
RC_BACKUP_CORRUPTION	12-4
RC_BACKUP_DATAFILE	12-6
RC_BACKUP_PIECE	12-8
RC_BACKUP_REDOLOG	12-9
RC_BACKUP_SET	12-11
RC_CHECKPOINT	12-12
RC_CONTROLFILE_COPY	12-12
RC_COPY_CORRUPTION	12-14
RC_DATABASE	12-15
RC_DATABASE_INCARNATION	12-15
RC_DATAFILE	12-16
RC_DATAFILE_COPY	12-17
RC_LOG_HISTORY	12-19
RC_OFFLINE_RANGE	12-19
RC_PROXY_CONTROLFILE	12-20
RC_PROXY_DATAFILE	12-22
RC_REDO_LOG	12-24

RC_REDO_THREAD	12-24
RC_RESYNC	12-25
RC_STORED_SCRIPT	12-26
RC_STORED_SCRIPT_LINE	12-26
RC_TABLESPACE	12-26

第 IV 部 オペレーティング・システム環境でのバックアップおよびリカバリの実行

13 オペレーティング・システム環境でのバックアップの実行

バックアップ実行前のデータベース・ファイルのリスト表示.....	13-2
O/S のバックアップの実行.....	13-3
全体データベース・バックアップの実行.....	13-3
表領域およびデータ・ファイルのバックアップの実行.....	13-6
制御ファイルのバックアップの実行.....	13-11
オンライン表領域のバックアップ失敗からの回復.....	13-14
保護強化のための Export と Import の使用方法.....	13-16
Export の使用方法.....	13-16
Import の使用方法	13-18

14 オペレーティング・システム環境での回復の実行

メディア回復.....	14-2
ファイルの復元.....	14-2
データ・ファイルの回復.....	14-2
回復するファイルの判別.....	14-3
ファイルの復元.....	14-5
バックアップ・データ・ファイルの復元.....	14-6
バックアップが使用できない場合のデータ・ファイルの再作成.....	14-6
必要なアーカイブ REDO ログ・ファイルの復元	14-7
基本的なメディア回復手順の理解.....	14-8
メディア回復文の使用法.....	14-8
アーカイブ REDO ログの適用	14-10
NOARCHIVELOG モードでのデータベースの回復.....	14-16
ARCHIVELOG モードでのデータベースの回復	14-18
パラレル方式でのメディア回復の実行.....	14-19
完全メディア回復の実行.....	14-19

クローズ状態のデータベースの回復.....	14-20
オープン状態のデータベースの回復.....	14-23
不完全メディア回復の実行	14-25
取消しベースの回復の実行.....	14-25
時間ベースの回復の実行.....	14-28
変更ベースの回復の実行.....	14-31
メディア回復後のデータベースのオープン	14-32
RESETLOGS または NORESETLOGS のいずれを指定するか決定.....	14-32
RESETLOGS モードでオープンする場合のガイドライン.....	14-33
RESETLOGS 以前のバックアップからの回復.....	14-34

15 オペレーティング・システム環境での回復シナリオ

メディア障害の種類の理解	15-2
データ・ファイルの消失後の回復	15-2
NOARCHIVELOG モードでのデータ・ファイルの消失.....	15-2
ARCHIVELOG モードでのデータ・ファイルの消失.....	15-3
ADD DATAFILE 操作を使用した回復方法	15-3
移動された表領域の回復	15-4
オンライン REDO ログ・ファイルの消失後の回復	15-5
多重化されたオンライン REDO ログ・グループのメンバーの消失後の回復.....	15-5
オンライン REDO ログ・グループの全メンバーの消失後の回復.....	15-6
アーカイブ REDO ログ・ファイルの消失後の回復	15-10
制御ファイルの消失後の回復	15-11
多重化された制御ファイルのメンバーの消失.....	15-11
現行の制御ファイルのすべてのコピーの消失.....	15-12
ユーザー・エラーからの回復	15-13
分散環境でのメディア回復の実行	15-14
Time-Based および Change-Based での分散データベース・リカバリの調整.....	15-15
スナップショットによるデータベースの回復.....	15-16

16 スタンバイ・データベースの管理

スタンバイ・データベースの計画	16-2
スタンバイ・データベースの利点.....	16-2
スタンバイ・データベースの要件.....	16-3
スタンバイ・データベースの作成	16-3

スタンバイ・データベース・モードの選択	16-5
スタンバイ・データベースの回復モードのメンテナンス	16-6
スタンバイ・データベースを手動回復モードにする.....	16-6
スタンバイ・データベースを管理回復モードにする.....	16-7
スタンバイ・データベースへのアーカイブ REDO ログの転送	16-9
スタンバイ・データベースの回復モードのメンテナンス.....	16-15
スタンバイ・データベースの読取り専用モードでのオープン	16-19
スタンバイ・データベースのアクティブ化	16-21
プライマリ・データベースの物理構造の変更	16-22
データ・ファイルの追加.....	16-23
データ・ファイルの改名.....	16-24
REDO ログの変更	16-24
制御ファイルの変更.....	16-25
初期化パラメータの設定.....	16-25
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.....	16-27
ダイレクト・パス操作の実行.....	16-27
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.....	16-29
OPS 構成でのスタンバイ・データベースの使用方法	16-29

A Recovery Manager による表領域の Point-in-Time 回復の実行

RMAN の TSPITR の概要	A-2
TSPITR の計画	A-4
リカバリ・カタログを使用しない TSPITR の実行	A-4
一般的な制約事項の理解.....	A-5
非一貫性の調査と解決.....	A-6
データの関連の管理.....	A-7
TSPITR のための補助インスタンスの準備	A-7
TSPITR の実行	A-10
TSPITR 後に使用するためのターゲット・データベースの準備	A-11
失敗した TSPITR への対応	A-12
TSPITR のパフォーマンスのチューニング	A-13
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.....	A-13
Recovery Manager による TSPITR 用の補助名の設定とデータ・ファイル・コピーの使用.....	A-13
補助制御ファイル内での変換されたファイル名の使用.....	A-14
まとめ：データ・ファイルの命名方法.....	A-15

B オペレーティング・システム環境での表領域の Point-in-Time 回復の実行

O/S 環境での表領域の Point-in-Time 回復の概要	B-2
TSPITR の利点	B-2
TSPITR の方法	B-3
TSPITR の用語	B-4
表領域の Point-in-Time 回復の計画	B-4
TSPITR の制限事項	B-5
TSPITR の必要条件	B-6
データベースでの TSPITR の準備	B-6
ステップ 1: オブジェクトが消失するかどうかを確認する	B-7
ステップ 2: プライマリ・データベースに対する依存性を調査および解決する	B-7
ステップ 3: プライマリ・データベースを準備する	B-9
ステップ 4: クローン・パラメータ・ファイルを準備する	B-10
ステップ 5: クローン・データベースを準備する	B-11
TSPITR の実行	B-12
ステップ 1: クローン・データベースを回復する	B-12
ステップ 2: クローン・データベースをオープンする	B-13
ステップ 3: クローン・データベースでエクスポートの準備を行う	B-13
ステップ 4: メタデータをエクスポートする	B-13
ステップ 5: 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	B-13
ステップ 6: プライマリ・データベースにメタデータをインポートする	B-14
ステップ 7: プライマリ・データベースを使用する準備をする	B-14
ステップ 8: プライマリ・データベース内の回復した表領域をバックアップする	B-14
パーティション表の部分的な TSPITR の実行	B-15
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	B-16
ステップ 2: 回復するパーティションの索引を削除する	B-16
ステップ 3: パーティションとスタンドアロン表を入れ替える	B-16
ステップ 4: 回復セットの表領域をオフラインにする	B-16
ステップ 5: クローン・データベースに表を作成する	B-16
ステップ 6: 回復するパーティションの索引を削除する	B-17
ステップ 7: パーティションとスタンドアロン表を入れ替える	B-17
ステップ 8: クローン・データベースをエクスポートする	B-17
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	B-17
ステップ 10: プライマリ・データベースにインポートする	B-17
ステップ 11: 回復セット表領域をオンラインにする	B-18

ステップ 12: パーティションとスタンドアロン表を入れ替える.....	B-18
ステップ 13: プライマリ・データベース内の回復した表領域をバックアップする.....	B-18
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.....	B-18
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.....	B-19
ステップ 2: 一時表を作成する.....	B-19
ステップ 3: パーティション表からレコードを削除する.....	B-19
ステップ 4: 回復セットの表領域をオフラインにする.....	B-19
ステップ 5: クローン・データベースに表を作成する.....	B-19
ステップ 6: 回復するパーティションの索引を削除する.....	B-20
ステップ 7: パーティションとスタンドアロン表を入れ替える.....	B-20
ステップ 8: クローン・データベースをエクスポートする.....	B-20
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.....	B-20
ステップ 10: プライマリ・データベースにインポートする.....	B-20
ステップ 11: 回復セットの表領域をオンラインにする.....	B-21
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.....	B-21
ステップ 13: プライマリ・データベース内の回復した表領域をバックアップする.....	B-21
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.....	B-22
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.....	B-22
ステップ 2-13: パーティション表領域の部分的 TSPITR と同じ手順を使用する.....	B-23
TSPITR のチューニングに関する考慮事項.....	B-23
回復セットの位置に関する考慮事項.....	B-23
バックアップ制御ファイルに関する考慮事項.....	B-24
トランспортаブル表領域を使用した TSPITR の実行.....	B-25

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。このマニュアルでは、Recovery Manager ユーティリティまたはオペレーティング・システムのコマンドを使用してバックアップ、復元および回復の作業を行うために必要な概念を説明し、個々の作業についても詳しく説明します。

注意： この『Oracle8i バックアップおよびリカバリ・ガイド』では、Oracle8i Standard Edition および Oracle8i Enterprise Edition の 2 つの製品の特徴と機能について説明します。Standard Edition と Enterprise Edition の基本機能は同じですが、Enterprise Edition でのみ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。

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	バックアップおよび回復計画の立案
第 1 章「バックアップおよび回復」	バックアップおよび回復の概念、方法について、全般的な概要を説明します。
第 2 章「データ構造の管理」	バックアップおよび回復を実行する際の、制御ファイル、オンライン REDO ログおよびアーカイブ REDO ログの管理方法を説明します。
第 3 章「バックアップおよび回復計画の立案」	バックアップおよび回復計画を立案する際のガイドラインを説明します。
第 2 部	Recovery Manager の使用方法
第 4 章「Recovery Manager の概要」	Recovery Manager ユーティリティの特徴と機能を説明します。
第 5 章「Recovery Manager のスタート・ガイド」	Recovery Manager を使用した起動方法と、リカバリ・カタログを使用すべきケースを説明します。さらに、各コマンドの実行方法、ターゲット・データベースへの接続方法についても説明します。
第 6 章「Recovery Manager のメタデータの管理」	リカバリ・カタログまたは制御ファイルを使用した、RMAN メタデータの管理方法を説明します。
第 7 章「Recovery Manager によるリストおよびレポートの生成」	バックアップ、コピーおよびレポートのリストの生成方法を説明します。このレポートはどのデータ・ファイルをバックアップすべきか、どのバックアップが古くなっているか、さらに、特定の時点のデータベース・スキーマの構造についても説明します。
第 8 章「Recovery Manager でのバックアップおよびコピーの作成」	Recovery Manager を使用した、データ・ファイル、制御ファイル、アーカイブ REDO ログのバックアップおよびイメージ・コピーの作成方法を説明します。
第 9 章「Recovery Manager による復元と回復」	データ・ファイル、制御ファイル、アーカイブ REDO ログのバックアップおよびコピーを復元し、データ・ファイルに対してメディア回復を実行する方法を説明します。
第 10 章「Recovery Manager による複製データベースの作成」	ターゲット・データベースのデータ・ファイルのバックアップを使用し、このデータベースの複製を、ローカル・ホストまたはリモート・ホスト上に作成する方法を説明します。

部 / 章	内容
第 3 部	Recovery Manager リファレンス
第 11 章「Recovery Manager のコマンド構文」	すべての RMAN コマンドの、構文図、パラメータの説明およびコードのサンプルが記載されています。
第 12 章「リカバリ・カタログ・ビュー」	リカバリ・カタログで利用できるビューについて説明します。
第 4 部	バックアップおよび回復における O/S コマンドの使用方法
第 13 章「オペレーティング・システム環境で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説明します。
第 14 章「オペレーティング・システム環境での回復の実行」	オペレーティング・システムのコマンドを使用してメディア回復を実行するための手順を説明します。
第 15 章「オペレーティング・システム環境での回復シナリオ」	メディア回復実行の手順を、いくつかの実例を挙げて説明します。
第 16 章「スタンバイ・データベースの管理」	スタンバイ・データベースの計画、作成およびメンテナンス方法を説明します。
付録 A「Recovery Manager による表領域の Point-in-Time 回復の実行」	表領域の Point-in-Time 回復を、Recovery Manager を使用して実行する場合の計画作成のガイドラインを示し、実際の手順を説明します。
付録 B「オペレーティング・システム環境での表領域の 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場合の計画作成のガイドラインを示し、実際の手順を説明します。

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 DBA 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およびデータベースの基本的な管理方法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・システム環境をよく理解していることも前提になっています。

表記規則

次の項目について、このマニュアルで使用する表記規則について説明します。

- テキスト

- 構文図および表記法
- サンプル・コード

テキスト

この項では、テキスト中で使用される表記規則を説明します。

英大文字

英文大文字は表領域名、初期化パラメータおよび SQL キーワードなど、注意が必要なものに使用します。

たとえば、次のようになります。" プライベート・ロールバック・セグメントを作成する場合は、その名前を `init.ora` ファイルの `ROLLBACK_SEGMENTS` パラメータに記述する必要があります。この情報を確認するには、SQL*Plus で `SHOW PARAMETER` 文を発行します。"

イタリック文字

イタリック体の語句は SQL の変数または Recovery Manager の構文の変数を表します。

たとえば、次のようになります。" アーカイブ REDO ログとは、オフラインでコピーされた、オンライン REDO ログのことです。この機能を使用可能にするには、データベースを `ARCHIVELOG` モードで実行する必要があります。Recovery Manager を使用している場合は、`archive log like '/oracle/archive/arc_*` サブ句を使えば、`backup` コマンドにアーカイブ REDO ログを指定できます。"

太字

テキスト内の太字の語句は、Recovery Manager のキーワードまたはオペレーティング・システム固有のコマンドを表します。

たとえば、次のようになります。" データベースをバックアップするには、Recovery Manager の **backup** コマンドを使用します。同様にして、ファイルをコピーする場合は UNIX の **cp** コマンドを使用します。"

クーリエ文字

ファイル名およびディレクトリはクーリエ・フォントで示します。コード例の前のテキスト中のクーリエ文字は、そのコード例で使用されるファイル名またはキーワードを示します。

たとえば、次のようになります。" このコマンドで表領域 `tbs_1` がバックアップされます。

```
run {
    allocate channel c1 type disk;
    backup tablespace tbs_1;
}
```

Recovery Manager の構文図および表記法

Recovery Manager の構文規則は、11-2 ページの「[このリファレンス・マニュアルでの表記法](#)」を参照してください。

コード例

SQL、SQL*Plus および Recovery Manager のコマンドと構文は、クーリエ・フォントで示し、各段落の本文からは間隔を置いて記述します。たとえば、次のようになります。

```
INSERT INTO emp (empno, ename) VALUES (1000, 'SMITH');
ALTER TABLESPACE users ADD DATAFILE 'users2.ora' SIZE 50K;
run {
    allocate channel chl type disk;
    backup database;
}
```

コマンド行から RMAN を実行する場合のコマンド・プロンプトは `RMAN>` となります。SQL*Plus のコマンド行からコマンドを実行する場合のプロンプトは `SQL>` となります。これらのプロンプトは、混乱の防止に必要な場合のみコード例に示します。

SQL、SQL*Plus および RMAN のコマンドは、異なったプラットフォームのさまざまな環境で実行できます。このマニュアルは、できるだけ一般的なドキュメンテーションの提供を指向したものとなっています。つまり、特定のオペレーティング・システムまたはインタフェースでの使用を想定したものではありません。ただし、構文が O/S レベルでどのように作用するかを示して説明することが必要な場合もあります。そのような場合、このマニュアルでは、UNIX のコマンド行インタフェースを実例に使用し、O/S のプロンプトを表す場合は % を使用します。たとえば、次のようになります。

```
% rman target / rcvcat rman/rman@inst2
RMAN> startup
```

このマニュアルの使用法

このマニュアルを読む前に、必ず次のマニュアルを読んでください。

- 『Oracle8i 概要』の初めの部分。Oracle の概念および用語など、このマニュアルの内容の基礎となる情報が記載されています。『Oracle8i 概要』のその他の章には、Oracle のアーキテクチャおよび機能についてのさらに詳しい説明があります。
- 『Oracle8i 管理者ガイド』の、制御ファイル、オンライン REDO ログおよびアーカイブ REDO ログの管理に関する章。

参照する機会の多いリファレンス・ガイドは次のとおりです。

- 『Oracle8i SQL リファレンス』
- 『Oracle8i リファレンス・マニュアル』

第Ⅰ部

バックアップおよび回復計画の立案

バックアップおよび回復

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ます。この章の目的は、全般的な概要を説明することです。バックアップおよび回復の概念の詳細は、後の各章で説明します。

取り上げる項目は次のとおりです。

- バックアップおよび回復
- バックアップおよび回復の実行時に重要となるデータ構造
- バックアップ計画の基本の理解
- 回復計画の基本の理解

バックアップおよび回復

簡単にいえば、**バックアップ (backup)** とは、データのコピーのことです。このコピーには、制御ファイルやデータ・ファイルなどの、データベースの重要な部分が含まれています。バックアップは、予期しないデータの損害およびアプリケーション・エラーに対処するものです。オリジナルのデータが失われても、バックアップを使用すると元通りにできます。

バックアップは、**物理バックアップ (physical backups)** と **論理バックアップ (logical backups)** に分類されます。物理データベース・ファイルのコピーを物理バックアップと呼びますが、このマニュアルはその説明を第一の目的としています。これに対し、Oracle Export ユーティリティを使用して抽出し、バイナリ・ファイルに格納したデータで構成されるものを論理バックアップと呼びます。論理バックアップは物理バックアップの補完に使用します。物理バックアップを作成するには、Oracle8i Recovery Manager ユーティリティまたは O/S のユーティリティを使用します。

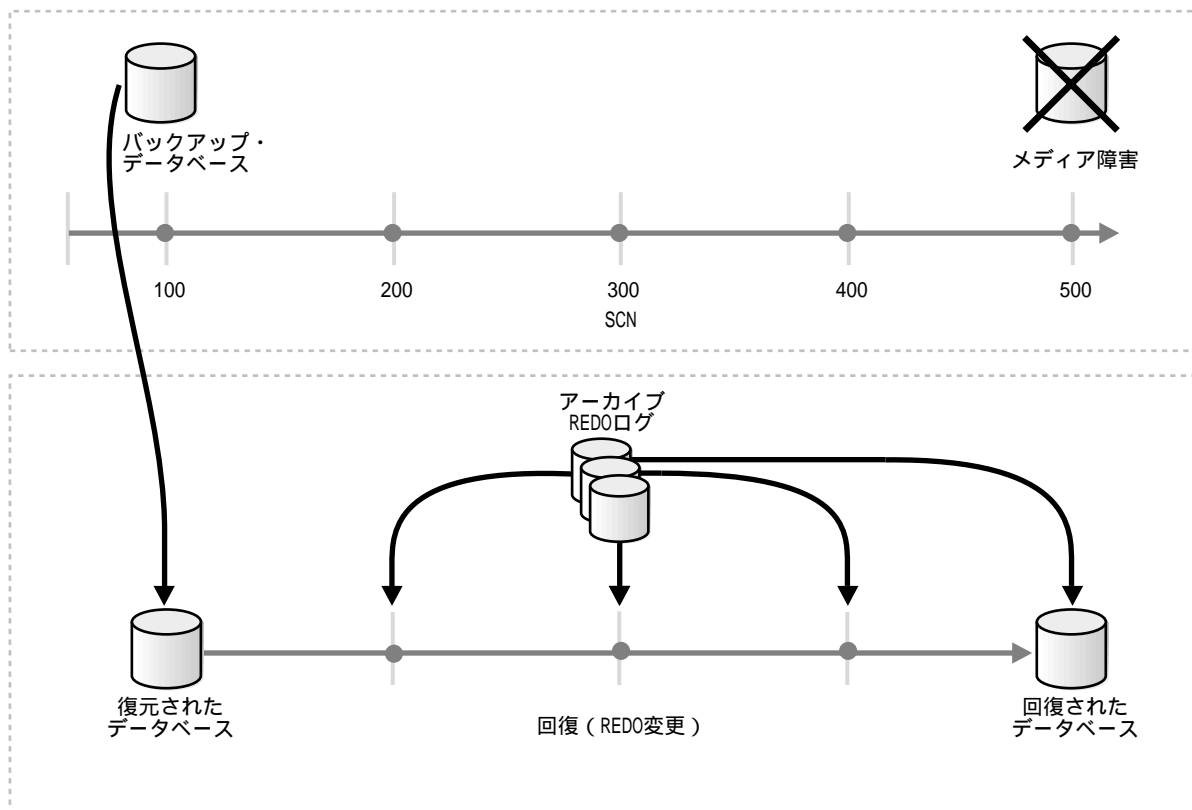
物理バックアップの**復元 (restore)** とは、これを再構築し、Oracle Server が使用できる状態にすることです。復元されたデータ・ファイルの**回復 (recovery)** とは、**REDO レコード (redo record)** (バックアップの実行後にデータベースに加えた変更の記録) を使用して、データ・ファイルを更新することです。Recovery Manager (RMAN) を使用している場合は、復元されたデータ・ファイルの回復に、**増分バックアップ (incremental backup)** を使用することもできます。増分バックアップとは、変更されたデータ・ブロックのみで構成されるデータ・ファイルのバックアップのことです。

インスタンス障害が発生した場合、Oracle は自動的に**クラッシュ回復 (crash recovery)** および**インスタンス回復 (instance recovery)** を実行します。インスタンス回復では、2 つの処理が自動的に実行されます。オンライン REDO レコードを適用して、できるだけ現行に近い時点までバックアップを**ロールフォワード (rolling forward)** し、コミットしていないトランザクションで行われた変更はすべて、オリジナルの状態に**ロールバック (rolling back)** します。

インスタンス回復に対して、**メディア回復 (media recovery)** では回復コマンドの発行が必要です。RMAN を使用している場合に、アーカイブ REDO ログまたは増分バックアップをデータ・ファイルに適用するには、**recover** コマンドを発行します。RMAN は、適切な増分バックアップまたは REDO ログを自動選択して適用します。SQL*Plus を使用している場合にアーカイブ・ログを適用するには、RECOVER 文または ALTER DATABASE RECOVER 文を発行します。

図 1-1 に、データベースにおける、バックアップ、復元およびメディア回復の基本原則を示します。

図 1-1 データベースの回復および復元



一般に、回復とは、バックアップの復元、ロールフォワードおよびロールバックに伴う操作のことを指します。バックアップと回復とは、一般に、データベースをデータ損失から保護し、万一データが失われたときに、そのデータを再構築するための、方針および操作のことです。

関連項目： Recovery Manager の機能の概要は、[第 4 章の「Recovery Manager の概要」](#)を参照してください。O/S バックアップおよび回復の実行方法は、[第 13 章の「オペレーティング・システム環境でのバックアップの実行」](#)および[第 14 章の「オペレーティング・システム環境での回復の実行」](#)を参照してください。

Oracle におけるデータベース・トランザクションの記録保持の仕組み

バックアップおよび回復の基礎を理解するには、Oracle がどのような仕組みで、データベースに加えられた変更を記録するかについて、理解する必要があります。データベースに変更が加えられるたびに、Oracle は、[REDO ログ・バッファ \(redo log buffer\)](#) の変更記録をメモリーに生成します。この記録を REDO レコードと呼びます。Oracle は、コミットされた変更も、コミットされていない変更も、REDO ログ・バッファに記録します。

この REDO ログ・バッファは、ディスク上に存在する[オンライン REDO ログ \(online redo log\)](#) に頻繁に書き込まれます。オンライン REDO ログは、最低 2 つのオンライン REDO ログ・ファイルで構成されます。これらのログに対し、Oracle は循環方式で書込みを実行します。すなわち、あるログ・ファイルに書込みを行い、現行ログが満杯になると、利用可能な次のファイルに切り替えます。

Oracle が ARCHIVELOG モード (NOARCHIVELOG モードではなく) で運用されているときのみ、システムはディスクの指定した場所にファイルをコピーし、非現行のオンライン REDO ログ・ファイルに REDO 情報を[アーカイブ \(archiving\)](#) する処理を開始します。オンライン REDO ログおよびアーカイブ REDO ログには、データベースへの変更の記録がすべて含まれているため回復にとって重要です。

関連項目：オンライン REDO ログおよびアーカイブ REDO ログの概要は、『Oracle8i 概要』を参照してください。前述のデータ構造の基礎的な管理方法は、『Oracle8i 管理者ガイド』を参照してください。バックアップおよび回復の実行時における、構造の管理方法は、[第 2 章の「データ構造の管理」](#)を参照してください。

バックアップおよび回復操作

物理バックアップとは、データ・ファイル、表領域またはデータベースの、ある時点でのスナップショットです。データベースのバックアップを定期的にとっておくと、元のデータベースのデータの一部が失われても、そのバックアップを使用してメディア回復が実行できます。データベースを現行の状態に戻すには、メディア回復の実行時に、REDO レコードまたは増分バックアップを最新のバックアップに適用します。このバックアップが[一貫性バックアップ \(consistent backup\)](#) であれば、回復を実行せずに復元することもできます。

Oracle では、古いバックアップを復元した後、一部の REDO データを適用するのみで、データベースを、指定した時点または SCN の状態まで回復することが可能です。このようなタイプの回復を、[不完全回復 \(incomplete recovery\)](#) と呼びます。不完全回復を実行した後は、RESETLOGS 操作でデータベースを開き、そのオンライン REDO ログをリセットしてください。

メディア回復の概念を、例を使用して説明します。データベースのバックアップを正午に作成したと仮定します。正午から、このデータベースに毎秒 1 回ずつ変更が加えられました。1 時までには REDO ログの切り替えが、3 回行われましたが、データベースが ARCHIVELOG モードで実行されていたため、これらの REDO ログはディスクにアーカイブされました。午後 1 時、ディスク・ドライブに障害が発生しました。しかし、正午の状態のデータベース・バックアップ全体を、機能しているディスク・ドライブに復元し、さらにアーカイブ

REDO ログを使用して、データベースを 1 時の状態にまで回復し、変更を再構築できました。

関連項目：一貫性バックアップを、O/S コマンドを使用して作成する方法は、13-4 ページの「[一貫性のある全体データベース・バックアップの作成](#)」を参照してください。メディア回復の実行方法は、第 14 章の「[オペレーティング・システム環境での回復の実行](#)」を参照してください。

バックアップおよび回復の基本計画

バックアップおよび回復の操作は、煩雑になることがありますが、効果的な計画を立案するための基本原則は、次に示すように単純です。

1. 同じ内容の複数のオンライン REDO ログを、別々のディスクに保存する。
2. REDO ログを複数の場所にアーカイブするか、またはアーカイブ REDO ログを頻繁にバックアップする。
3. 制御ファイルの同時コピーを複数作成して保管する。1-7 ページの「[制御ファイル](#)」を参照してください。
4. データ・ファイルおよび制御ファイルのバックアップを頻繁にとり、これらを安全な場所（できれば複数のメディア）に保存する。

これらはバックアップおよび回復方法の原理です。データ・ファイル、制御ファイルおよびアーカイブ REDO ログを、安全な場所に保管しておけば、万一火災でハードウェアが損傷を受けることがあっても、元のデータベースを再作成できます。

災害からの効果的かつ洗練された保護手法の一つとして、スタンバイ・データベースを保持しておくことが挙げられます。スタンバイ・データベースとは、稼動データベースの内容をそのまま複製したもので、Net8 接続を通じて配分される REDO ログにより、自動的な更新が可能です。

関連項目：オンライン REDO ログなど、重要なデータ構造の詳細は、第 2 章の「[データ構造の管理](#)」を参照してください。バックアップおよび回復計画立案の詳細は、第 3 章の「[バックアップおよび回復計画の立案](#)」を参照してください。スタンバイ・データベースの保持の詳細は、第 16 章の「[スタンバイ・データベースの管理](#)」を参照してください。

バックアップおよび回復の実行時に重要となるデータ構造

バックアップおよび回復計画を詳細に検討するにあたっては、バックアップおよび回復操作に関係する、物理データベースの構造を理解する必要があります。この項では、次の項目について簡単に説明します。

- データ・ファイル
- 制御ファイル
- ロールバック・セグメント
- オンライン REDO ログ・ファイル
- アーカイブ REDO ログ・ファイル

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。

関連項目：Oracle8i アーキテクチャの概要は、『Oracle8i 概要』を参照してください。データ構造の管理方法は、『Oracle8i 管理者ガイド』を参照してください。

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・ファイルは、表領域と呼ばれる論理構造に属し、データベースのデータを含んでいます。データ・ファイルは、**データ・ブロック (data block)** と呼ばれるさらに小さい単位に分割されます。表および索引などの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・ファイルのブロックに物理的に存在します。

各データ・ファイルの最初のブロックをヘッダーと呼びます。ヘッダーには、ファイル・サイズ、ブロック・サイズ、作成タイムスタンプおよびチェックポイント SCN (1-9 ページの「**システム変更番号 (SCN)**」を参照) などの、重要な情報が含まれています。データベースをオープンするたびに、Oracle はデータ・ファイルのヘッダー情報を、制御ファイルに格納された情報と照合し、回復が必要かどうかを判断します。

データ・ファイルの使用方法

Oracle は、通常の操作中にデータ・ファイルのデータを読み取り、メモリー内のバッファ・キャッシュに格納します。たとえば、ユーザーがある表に入っているデータにアクセスすると想定します。要求された情報が既にバッファ・キャッシュに入っていないければ、Oracle は適当なデータ・ファイルから情報を読み取り、メモリーに格納します。

修正されたバッファは、バックグラウンド・プロセス DBWn (データベース・ライターまたは db ライター) によってディスクに書き込まれます。通常、Oracle Server プロセスがバッファ内のブロックを変更する時点と、DBWn がこれをディスクに書き込む時点は異なります。ディスクに書き込めずにメモリーに蓄積されるデータが多くなると、インスタンス回復の時間が長くなります。クラッシュまたはメディア障害が発生した場合、Oracle は現行のオンライン・ログの REDO ログを適用して変更を回復する必要があるためです。この時間、

つまり平均回復時間（MTTR）を最小限にすることは、バックアップおよび回復計画の重要なポイントです。

関連項目： MTTR 調整パラメータの詳細は、『Oracle8i チューニング』を参照してください。

制御ファイル

各 Oracle データベースには、**制御ファイル（control file）** が 1 つあります。制御ファイルは、データベースを構成する全ファイルの、オペレーティング・システムのファイル名を含んでいるため、特に重要なバイナリ・ファイルです。ここには、回復時に使用される一貫性情報も含まれています。それは次のようなものです。

- データベース名
- データベース作成時のタイムスタンプ
- データベースのデータ・ファイルの名前と、オンラインおよびアーカイブ REDO ログ・ファイルの名前
- **チェックポイント（checkpoint）** 情報。このポイント以前に加えられた変更は、すべてデータ・ファイルに保存されていることを示す、REDO ログ内のレコードです。
- バックアップ情報（Recovery Manager ユーティリティを使用する場合）

制御ファイルを**多重化（multiplexing）**する場合は、そのコピーを複数作成するよう Oracle を構成し、制御ファイルをデータ損失から保護します。ディスクのミラー化をサポートしているオペレーティング・システムであれば、制御ファイルの**ミラー化（mirroring）**、すなわち、制御ファイルのコピーを複数のディスクに書き込むことが可能です。

関連項目： 制御ファイルの多重化およびミラー化の詳細は、2-7 ページの「**複数の制御ファイルのメンテナンス**」を参照してください。制御ファイルの管理方法の概略は、『Oracle8i 管理者ガイド』を参照してください。

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、データベース操作の実行にオープンする必要があるデータ・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）、Oracle はそのデータベースの制御ファイルを修正し、変更を反映します。

データベースを構成するファイルのセットを変更したときは、必ず制御ファイルをバックアップしてください。構造の変更の例としては、データ・ファイルまたは表領域の追加、削除または変更や、オンライン REDO ログの追加および削除があります。

関連項目： 制御ファイルのバックアップ方法は、2-5 ページの「**構造変更後の制御ファイルのバックアップ**」を参照してください。

ロールバック・セグメント

データベースには、必ず 1 つ以上のロールバック・セグメントが含まれています。ロールバック・セグメントとは、データ・ファイルに含まれている論理構造です。トランザクションによってデータ・ブロックが修正されると、ロールバック・セグメントは変更前の情報を記録します。

ロールバック・セグメントの使用方法

Oracle は、ロールバック・セグメントを使用してさまざまな処理を実行します。通常、データベースのロールバック・セグメントには、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。データベース回復を行う場合、Oracle はロールバック・セグメントの情報をを使用して、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し、データの一貫性を保ちます。

たとえば、ユーザーが表の行値を 5 から 7 に変更したとします。REDO ログはこの変更のレコードを保持し、ロールバック・セグメントは古い値を格納します。しかし、ユーザーがこのトランザクションをコミットする前に、停電が発生したと仮定します。電源が回復すると、データベースはクラッシュ回復操作を実行します。この場合、REDO ログを使用して値を 5 から 7 にロールフォワードし、続いてロールバック・セグメントを使用してコミットされていない 7 から 5 への変更を取り消します。

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[オンライン REDO ログ \(online redo log\)](#) ファイルがあります。Oracle は、識別のため、各 REDO ログ・ファイルに一意の[ログ順序番号 \(log sequence number\)](#) を割り当てます。データベースの REDO ログ・ファイル集合が、そのデータベースの[REDO ログ \(redo log\)](#) になります。Oracle ではデータベースへのすべての変更を、REDO ログを使用して記録します。

Oracle は、[REDO レコード \(redo record\)](#) の変更をすべて記録します。これらの変更は、何が変わったかを示す、REDO ログへのエントリとなります。たとえば、ユーザーが給与計算表の列値を 5 から 7 に更新したとします。Oracle は古い値をロールバック・セグメントに、新しい値を REDO レコードに記録します。REDO ログにはデータベースへの変更がすべて格納されるため、このトランザクションの REDO レコードに含まれる要素は実際には次の 3 つとなります。

- ロールバック・セグメントのトランザクション表の変更
- ロールバック・セグメントのデータブロックの変更
- 給与計算表のデータブロックの変更

給与計算表の更新をコミットすると、Oracle は新しい REDO レコードを生成し、この変更 SCN を割り当てます。このように、データベースで行われた処理はすべて注意深く管理されます。

REDO ログ・ファイルの循環使用

オンライン REDO ログ・ファイルへの書き込みは、ログ・ライター・バックグラウンド・プロセス (LGWR) によって循環方式で行われます。すなわち、[現行のオンライン REDO ログ \(current online redo log\)](#) が満杯になると、LGWR は利用可能な次の[非アクティブ REDO ログ \(inactive redo log\)](#) に書き込みを行います。LGWR はデータベースの各オンライン REDO ログ・ファイルを巡回して、古い REDO データを上書きします。満杯になったオンライン REDO ログ・ファイルは再使用できますが、使用可能になる時期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、満杯のオンライン REDO ログは、そのログに記録された変更がデータ・ファイルに反映された後、使用可能になります。
- アーカイブが使用可能な場合、満杯のオンライン REDO ログは、変更がデータ・ファイルに反映され、さらに同ファイルがアーカイブされた後、使用可能になります。

システム変更番号 (SCN)

[システム変更番号 \(system change number: SCN\)](#) は、データベースのコミット済みバージョンを一意的に識別する、増加し続ける値です。ユーザーがトランザクションをコミットするたびに、Oracle は新しい SCN を記録します。SCN はアラート・ログ・ファイルからの取得など、多数の取得方法があります。SCN は回復時の識別子に使用できます。たとえば、データベースを SCN1030 の状態まで不完全回復するといったことが可能です。

SCN は制御ファイル、データ・ファイル・ヘッダーおよび REDO レコードで使用されます。REDO ログ・ファイルには、必ずログ順序番号と、低および高 SCN があります。低 SCN にはログ・ファイル内の最も小さい SCN、高 SCN にはログ・ファイル内の最も大きい SCN が記録されています。

オンライン REDO ログの使用方法

REDO ログは回復時に重要な意味を持ちます。たとえば、停電により、変更済みデータの永続的な書き込みが、データ・ファイルに行えなくなったとします。このような場合でも、データ・ファイルの古いバージョンと、オンライン REDO ログおよびアーカイブ REDO ログに記録された変更を組み合わせれば、失われたデータを再構築できます。

REDO ログ障害からデータベースを保護するため、Oracle では REDO ログの多重化が可能です。Oracle が REDO ログを多重化した場合、REDO ログの複数のコピーが異なるディスクに保持されます。オンライン REDO ログはバックアップしないでください。また、復元の必要も一切ありません。REDO ログはアーカイブして保管します。

関連項目 : バックアップおよび回復時のオンライン REDO ログ管理の詳細は、2-11 ページの「[オンライン REDO ログの管理](#)」を参照してください。全般的な説明は、『Oracle8i 管理者ガイド』を参照してください。

アーカイブ REDO ログ・ファイル

アーカイブ REDO ログ (archived redo log) とは、REDO エントリが満杯になったため非アクティブとなり、パラメータ・ファイルで指定した 1 つ以上の宛先にコピーされたオンライン REDO ログ・ファイルです。Oracle は、次の 2 つのアーカイブ・モードのうち、いずれかを実行できます。

ARCHIVELOG	満杯になったオンライン REDO ログ・ファイルをアーカイブし、循環方式で再使用します。
NOARCHIVELOG	満杯になったオンライン REDO ログ・ファイルをアーカイブせず、循環方式で再使用します。

データベースを ARCHIVELOG モードで実行すると、次のような結果が得られます。

- インスタンス障害、メディア障害のいずれの場合でもデータベースを完全に回復できる。
- **ホット・バックアップ (hot backup)** が可能。すなわち、データベースをオープンして使用できる状態でバックアップできます。
- アーカイブ REDO ログの **スタンバイ・データベース (standby database)** への送信および適用が可能。スタンバイ・データベースとは、プライマリ・データベースの複製です。
- 回復方法の選択肢が増える。たとえば、不完全回復および表領域の Point-in-Time 回復 (TSPITR) が可能です。
- アーカイブ REDO ログを格納するには、追加の管理操作を実行する必要がある。
- アーカイブ・ログを格納するには、ディスク領域が余分に必要となる。

データベースを NOARCHIVELOG モードで実行すると、次のような結果が得られます。

- データベースのバックアップは、正しく停止し完全にクローズした場合に限り可能。
- 通常、メディア回復は、データベース全体の復元に限定される。最後の全体データベース・バックアップ以降の変更はすべて失われる。
- アーカイブ REDO ログが作成されないため、追加の管理作業が発生しない。

注意： NOARCHIVELOG モードで運用中の場合、現行のオンライン・ログ・ファイルが一番最後にバックアップをとってからまだ上書きされていない状態でないとデータベースを回復できません。

関連項目 : REDO ログのアーカイブの詳細は、2-14 ページの「[アーカイブ REDO ログの管理](#)」および『Oracle8i 管理者ガイド』を参照してください。スタンバイ・データベースの詳細は、[第 16 章の「スタンバイ・データベースの管理」](#)を参照してください。

バックアップ計画の基本的理解

物理バックアップとは、データ・ファイル、制御ファイルまたはアーカイブ REDO ログのコピーで、データ損失に備えて保存します。バックアップ計画は、次の事柄を考慮して検討します。

- [バックアップの重要性](#)
- [発生しうる障害のタイプ](#)
- [バックアップ対象について](#)
- [バックアップ方法の選択](#)
- [一貫性バックアップと非一貫性バックアップの選択](#)
- [バックアップの回数](#)

バックアップの重要性

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、特急配送サービス、銀行または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。また、メディア障害のために重要な給与計算表のデータ・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。失われた情報の量および価値によっては、重大な結果がもたらされる場合があります。

頻繁にバックアップを作成すれば、データが失われても少なくともその一部は復元することができます。データベースが ARCHIVELOG モードで運用されている場合は、復元されたバックアップに REDO を適用すれば、失われた変更をすべて再構築できます。

発生しうる障害のタイプ

バックアップは、データ損失または破損に対処します。残念なことに、データはさまざまな理由で失われたり破損したりします。したがって、バックアップ計画は、下記に示したさまざまな可能性を考慮して立案する必要があります。データ損失が発生する、最も一般的なタイプの障害は次のとおりです。

文障害

Oracle プログラムで文を処理する際に発生する論理障害。たとえば、ユーザーが無効な SQL 文を発行した場合などに発生します。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	不正な接続の切断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。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。ユーザー・プロセスがデータベースの修正に失敗した場合、Oracle バックグラウンド・プロセスは、コミットされていない実行結果を取り消します。
インスタンス障害	<p>Oracle インスタンス、すなわち SGA およびバックグラウンド・プロセスが機能しなくなる問題。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。インスタンスに障害が発生すると、Oracle は SGA のバッファ内のデータをデータ・ファイルに書き込みません。</p> <p>単一サーバーの環境では、次の起動でデータベースがオープンされたときに、Oracle が自動的にクラッシュ回復を実行します。Oracle Parallel Server 環境（OPS）では、別のインスタンスが障害発生後すぐにインスタンス回復を実行します。</p>
ユーザー・エラーまたはアプリケーション・エラー	<p>データの損失をもたらすユーザーまたはアプリケーションの問題。たとえば、ユーザーが給与計算表のデータを誤って削除してしまうことがあります。</p> <p>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。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 はデータベース Point-in-Time 回復（DBPITR）または表領域 Point-in-Time 回復（TSPITR）を提供します。たとえば、ユーザーが誤って給与計算のデータを削除した場合でも、そのデータが削除される前の状態にデータベースを回復できます。</p>
メディア障害	データベースの運用に必要なファイルの書き込みまたは読み取りを、Oracle が試行したときに発生する物理的な問題。一般的な例は、ディスク・ヘッドのクラッシュで、この場合はディスク・ドライブ上すべてのデータが失われます。ディスク障害により、データ・ファイル、REDO ログ・ファイルおよび制御ファイルなど複数の種類のファイルに影響が出ることがあります。それ以降データベース・インスタンスは正常に動作しないので、SGA 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・ファイルへの書き込みができません。

バックアップ対象について

データベースにはさまざまなタイプのデータが含まれています。バックアップ計画を立案する際には、どの情報をコピーするかを決定する必要があります。バックアップの基本的なタイプは次のとおりです。

- データベース全体
- 表領域
- データ・ファイル
- 制御ファイル
- アーカイブ REDO ログ

何をバックアップするかを決定する際の基本原則は、重要性および変更の度合いに基づいて、データに優先順位を付与することです。たとえば、アーカイブ REDO ログは、変更されることなくても、データベースの回復には重要な意義を持ちます。したがって、可能であれば複数のコピーを保持します。別のケースとしては、ユーザーが頻繁に更新する、重要な必要経費のデータを含む表領域が挙げられます。この表領域は、頻繁にバックアップをとると考えられます。したがって、回復時に多くの REDO データを適用する必要はありません。

バックアップのタイプは、さまざまに組み合わせることができます。たとえば、全体データベース・バックアップを週に 1 回とり、データベースのオリジナルの情報を、現行に近い状態でコピーしておくという慎重な方法をとることが可能です。頻繁にアクセスする表領域は、毎日バックアップできます。また、重要な制御ファイルおよびオンライン REDO ログはすべて多重化するという、追加の保護策も実行できます。

バックアップ方法の選択

バックアップ作成の基本的な方法は 3 つあります。次の中から選択できます。

- Recovery Manager (RMAN) を使用する。このユーティリティは、サーバー・セッションとの接続を確立し、バックアップおよび回復操作時のデータの移動を管理します。

注意： RMAN は、リリース 8.0 以上の Oracle データベースとのみ互換性があります。Oracle7 データベースの場合は、Enterprise Backup Manager (EBU) を使用してください。

- オペレーティング・システム固有のコマンドを実行し、データベースを手動でバックアップする。

- Oracle Export ユーティリティを使用し、論理バックアップを作成する。このユーティリティは、Oracle データベースのデータを、特殊な形式でオペレーティング・システム・ファイルに書き込みます。このデータは、後でデータベースにインポートできます。

注意： 論理バックアップは、データベース全体の物理バックアップの代替ではありません。論理バックアップは、バックアップおよび回復の計画全般における、追加ツールと考えてください。

関連項目： バックアップ方法の選択の詳細は、3-8 ページの「[バックアップ方法の選択](#)」を参照してください。Recovery Manager の機能の概要は、[第 4 章の「Recovery Manager の概要」](#)を参照してください。

Recovery Manager バックアップおよびイメージ・コピーの作成

RMAN は、データのバックアップまたはイメージ・コピーの作成が可能な、パワフルかつ汎用性の高いプログラムです。RMAN の **backup** コマンドを使用してファイルまたはアーカイブ・ログを指定すると、RMAN は **バックアップ・セット (backup set)** を作成して出力します。

バックアップ・セットは、1 つ以上のデータ・ファイル、制御ファイルまたはアーカイブ REDO ログから構成され、RMAN 固有の形式で書かれています。したがって、回復操作を実行するには、RMAN の **restore** コマンドを使用する必要があります。これに対し、**copy** コマンドを使用して作成した、ファイルのイメージ・コピーは、インスタンスが使用可能な形式になっています。同コピーを復元または回復する場合、RMAN を起動する必要はありません。

backup や **copy** などの RMAN コマンドを発行すると、RMAN は Oracle サーバー・セッションとの接続を確立します。サーバー・セッションは、ターゲット・データベースの指定されたデータ・ファイル、制御ファイルまたはアーカイブ・ログをバックアップします。

RMAN は、必要な情報を制御ファイルまたはオプションのリカバリ・カタログから取得します。リカバリ・カタログとは、バックアップおよび回復の、さまざまな情報を保管しているリポジトリです。RMAN は、バックアップの必要があるファイルすべてについて、自動的に名前と場所を識別するので便利です。

RMAN には複数の利点があります。RMAN を使用する重要な利点は、RMAN には **増分バックアップ (incremental backup)** 機能があることです。従来のバックアップ方法では、データ・ファイルで使用されたデータ・ブロックをすべてバックアップする、**全体バックアップ (full backup)** を実行する必要があります。増分バックアップ機能を使用すると、前回のバックアップ以降に変更されたデータ・ブロックのみをバックアップできます。

RMAN では、2 つのタイプの増分バックアップを実行できます。**差分バックアップ (differential backup)** または **累積バックアップ (cumulative backup)** です。差分レベル n の増分バックアップでは、レベル n 以下の最後のバックアップ以降に変更されたブロックがすべてバックアップされます。たとえば、レベル 2 の差分バックアップでは、前回のレベル

1 とレベル 2 のバックアップの、どちらが新しいかを RMAN が判断し、新しい方のバックアップ以後に修正された全ブロックをバックアップします。

累積レベル n のバックアップでは、レベル $n-1$ 以下の最新のバックアップ以後に使用されたブロックをすべてバックアップします。たとえば、レベル 3 の差分バックアップでは、前回のレベル 2 とレベル 1 のバックアップの、どちらが新しいかを RMAN が判断し、新しい方のバックアップ以後に修正された全ブロックをバックアップします。

メディア管理 RMAN は、[メディア・マネージャ \(media manager\)](#) と容易に統合できます。メディア・マネージャは、テープなどのアーカイブ・メディアにバックアップすることを可能にする、ベンダー供給のソフトウェア・パッケージです。RMAN は、このメディア・マネージャと協調して、ディスクと記憶装置の間のデータ移動を実行します。使用しているメディア・マネージャに[プロキシ・コピー \(proxy copy\)](#) を作成する機能がある場合は、メディア・マネージャによるバックアップおよび復元操作時のデータ転送が可能です。

関連項目 : RMAN の概要は、[第 4 章の「Recovery Manager の概要」](#)を参照してください。RMAN を使用したバックアップおよびコピーの作成方法は、[第 8 章の「Recovery Manager でのバックアップおよびコピーの作成」](#)を参照してください。増分バックアップの詳細は、4-38 ページの[「増分バックアップ」](#)を参照してください。

O/S バックアップの作成

RMAN を使用せずにバックアップを作成するには、UNIX の `cp` コマンドなど、オペレーティング・システムのコマンドを使います。スクリプトを作成し、バックアップ操作の自動化も可能です。

データベース全体のバックアップを一度にとることも、個々の表領域、データ・ファイル、制御ファイルおよびアーカイブ・ログのバックアップで、データベース全体のバックアップを補完することも可能です。O/S コマンドを使用すると、前述のようなバックアップが実行できます。

関連項目 : O/S バックアップの作成方法は、[第 13 章の「オペレーティング・システム環境でのバックアップの実行」](#)を参照してください。

バックアップによる保護を強化するための Export ユーティリティの使用 方法

Export ユーティリティを使用して、データの論理バックアップを作成し、物理バックアップを補足できます。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。Export ユーティリティは、Oracle データベースのデータを、専用の形式でオペレーティング・システム・ファイルに書き込むものです。このデータをデータベースにインポートするには、Import ユーティリティを使用します。

関連項目 : Export ユーティリティおよび Import ユーティリティの使用方法は、『Oracle8i ユーティリティ・ガイド』を参照してください。

一貫性バックアップと非一貫性バックアップの選択

RMAN および O/S コマンドを使用して、**非一貫性バックアップ (inconsistent backup)** または **一貫性バックアップ (consistent backup)** を作成できます。非一貫性バックアップとは、データベースがオープンしているとき、またはデータベースが異常終了した場合に作成する、1 つ以上のデータベース・ファイルのバックアップです。一貫性バックアップとは、データベースが正しくクローズされた場合に作成する、1 つ以上のデータベース・ファイルのバックアップです。非一貫性バックアップとは異なり、一貫性バックアップでは復元後にインスタンス回復を実行する必要はありません。

一貫性バックアップと非一貫性バックアップのどちらを作成するかは、さまざまな要因によって決まります。常時オープンしておく必要があり、かつ利用可能にしておく必要があるデータベースの場合は、非一貫性バックアップを作成します。使用頻度の少ない期間が定期的に存在する場合は、データベース全体の一貫性バックアップを行い、さらに使用回数の多い表領域をオンライン・バックアップして補完するという方法もあります。

一貫性バックアップ

データベース全体を正しく停止してからバックアップすることを、**全体データベース・バックアップ (whole database backup)** と呼びます。一貫性のある全体データベース・バックアップをとるための、基本的な手順は次のとおりです。

1. データベースを正しく停止する。
2. すべてのデータ・ファイル、制御ファイルおよびパラメータ・ファイルをバックアップする。
3. Oracle を NORMAL モードで再起動する。

NOARCHIVELOG モードで運用している場合は、データベースを正しく停止してからコールド・バックアップを作成します。このようにしないと、データベースが SCN に関して一貫性がなく、インスタンス回復を実行して一貫性のある状態にする必要が出てきます。また、作成されたバックアップも使用できない場合があります。

非一貫性バックアップ

表領域、データ・ファイルおよびアーカイブ REDO ログのバックアップは、データベースをオープンして実行します。可能であれば、指定した表領域をオフラインにしてデータ・ファイルをバックアップします。表領域をオンラインにして、O/S コマンドを使用してデータ・ファイルをバックアップする場合は、まず ALTER TABLESPACE ... BEGIN BACKUP コマンドを発行し、その表領域をホット・バックアップ・モードにします。ALTER TABLESPACE ...END BACKUP を発行し、ホット・バックアップ・モードを解除します。RMAN を使用している場合は、このコマンドを発行する必要はありません。

関連項目：一貫性バックアップを、O/S コマンドを使用して作成する方法は、13-6 ページの「**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**」を参照してください。RMAN を使用したバックアップの作成方法は、第 8 章の「**Recovery Manager でのバックアップおよびコピーの作成**」を参照してください。

バックアップの回数

バックアップ計画は、業務上のニーズに合わせて調整します。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。バックアップの回数が少なければ、その他の処理で Oracle のリソースを使用できるという利点があります。反面、データが消失したり回復時間が長くなる可能性も出てきます。

しかし、常時データベースを使用可能な状態にする必要がある場合は、データベースのオンライン・バックアップを頻繁にとってください。その場合は、毎日ホット・バックアップをとり、オンライン REDO ログを多重化（複数のコピーを作成する）し、さらに REDO ログを複数の場所にアーカイブしてもよいでしょう。オリジナルのデータベースを複製した、スタンバイ・データベースを別の都市に保持しておき、これを定期的に更新することも可能です。

関連項目：効果的なバックアップ計画を立案する上で重要な留意事項は、[第 3 章の「バックアップおよび回復計画の立案」](#)を参照してください。

回復計画の基本的理解

基本的なメディア回復には次の 2 つの段階があります。物理バックアップを復元し、これをデータベースの変更で更新することです。回復における最も重要な点は、同一 SCN では、すべてのデータ・ファイルが一貫性を持つことです。Oracle には整合性のチェック機能があり、すべてのデータ・ファイルが一貫性を持たなければデータベースをオープンできません。

回復計画を立案する場合は、次の事柄について理解しておく必要があります。

- [回復の仕組み](#)
- [回復のタイプ](#)
- [メディア回復](#)
- [回復方法の選択](#)

回復の仕組み

どのタイプのインスタンス回復、またはメディア回復でも（RMAN 増分バックアップを使用したメディア回復を除き）Oracle は REDO データのデータ・ブロックへの適用を順次行います。Oracle は制御ファイルおよびデータ・ファイル・ヘッダーの情報を使用して、回復が必要かどうかを判断します。

回復には、[ロールフォワード（rolling forward）](#)と[ロールバック（rolling back）](#)の、2 つの段階があります。Oracle はロールフォワードする際、REDO レコードを対応するデータ・ブロックに適用します。Oracle は体系的に REDO ログをチェックし、どの変更をどのブロックに適用する必要があるかを判断した後、該当するブロックを変更します。たとえば、ユーザーが行を表に追加し、サーバーがクラッシュしてこの変更がディスクに保存されな

かった場合、Oracle はこのトランザクションの REDO レコードを使用してデータ・ブロックを更新し、新しい行を反映します。

ロールバック段階で、Oracle はロールバック・セグメントをデータ・ファイルに適用します。ロールバック情報はトランザクション表に格納されます。Oracle は、コミットされていないトランザクションをこの表で検索し、見つかったものはすべて取り消します。たとえば、行を追加する SQL 文をユーザーがコミットしなかった場合、Oracle はトランザクション表でこれを検出し、変更を取り消します。

回復時にデータベースがマウントされた場合は、データベースをオープンするときに限りロールバックが行われます。回復時にデータベースがオープンし、かつ表領域がオフラインであった場合は、オンラインになったときにこの表領域でロールバックが行われます。

関連項目：Oracle 回復のメカニズムの詳細は、『Oracle8i 概要』を参照してください。

回復のタイプ

回復には、**インスタンス回復 (instance recovery)**、**クラッシュ回復 (crash recovery)** および **メディア回復 (media recovery)** という、3 つの基本タイプがあります。Oracle は、インスタンスの起動時に最初の 2 つのタイプの回復を自動的に実行します。コマンドの発行が必要なのはメディア回復のみです。

インスタンス回復

インスタンス回復は OPS 構成でのみ可能です。オープン・データベースで、あるインスタンスが別のインスタンスのクラッシュを検出したとき実行されます。クラッシュしていないインスタンスが、自動的に REDO ログを使用し、インスタンスに障害が発生したことにより失われた、データベース・バッファ内のコミット済みデータを回復します。インスタンスがクラッシュすると、Oracle は、そのインスタンスで処理中だったあらゆるトランザクションを取り消し、そのインスタンスによってかけられていたすべてのロックを、回復の完了後に消去します。

クラッシュ回復

クラッシュ回復は、単一インスタンス・データベースがクラッシュしたか、複数インスタンス・データベースの全インスタンスがクラッシュした場合に実行されます。クラッシュ回復では、インスタンスがデータベースをオープンし、回復処理を実行する必要があります。通常、クラッシュ後または SHUTDOWN ABORT 後にデータベースをオープンした最初のインスタンスが、クラッシュ回復を自動的に実行します。

メディア回復

クラッシュ回復およびインスタンス回復とは異なり、メディア回復はコマンドを入力して実行します。メディア回復では、オンラインおよびアーカイブ REDO ログの他、(RMAN を使用している場合) 増分バックアップも使用し、復元されたバックアップを現行の状態にするか、特定の時点の状態に更新します。この回復方法をメディア回復と呼ぶのは、通常メディア障害の発生時に実行するためです。

メディア回復

メディア回復とは、REDO レコードまたは (RMAN を使用している場合) 増分バックアップを使用し、復元されたデータ・ファイルを現在の状態、あるいは指定された時間の状態にまで回復することです。メディア回復で復元できるのは、データベース全体、表領域またはデータ・ファイルです。どの場合も、復元されたバックアップを使用します。メディア回復は、**完全回復 (complete recovery)** と **不完全回復 (incomplete recovery)** の 2 つに大きく分けることができます。

完全回復

完全回復とは、REDO データまたは増分バックアップを、データベース、表領域またはデータ・ファイルのいずれかのバックアップと組み合わせて使用し、できるだけ現行に近い状態に更新することです。これを完全回復と呼ぶのは、REDO 変更のすべてを Oracle が適用するためです。通常、メディア回復は、メディア障害によってデータ・ファイルまたは制御ファイルが破損した場合に実行します。

完全回復の要件 完全回復は、データベース、表領域またはデータ・ファイルに対して実行します。データベース全体の完全回復を実行する場合は、次の手順に従います。

- データベースをマウントする。
- 回復するデータ・ファイルがすべてオンラインになっているか確認する。
- データベース全体のバックアップ、または回復するファイルのバックアップを復元する。
- オンライン REDO ログとアーカイブ REDO ログの両方、またはその一方を適用する。

表領域またはデータ・ファイルの完全回復を実行する場合は、次の手順に従います。

- データベースがオープンしている場合は、回復する表領域またはデータ・ファイルをオフラインにする。
- 回復するデータ・ファイルのバックアップを復元する。
- オンライン REDO ログとアーカイブ REDO ログの両方、またはその一方を適用する。

関連項目 : RMAN を使用した完全回復の方法は、9-17 ページの「**完全回復の実行**」を参照してください。O/S コマンドを使用した完全メディア回復の方法は、14-19 ページの「**完全メディア回復の実行**」を参照してください。

不完全回復

不完全回復とは、バックアップを使用して、データベースの非現行バージョンを作成することです。したがって、最新のバックアップ以降に生成された REDO データをすべて適用するわけではありません。通常、不完全回復を実行するのは次の場合です。

- メディア障害によって、オンライン REDO ログの一部あるいはそのすべてが破損した。

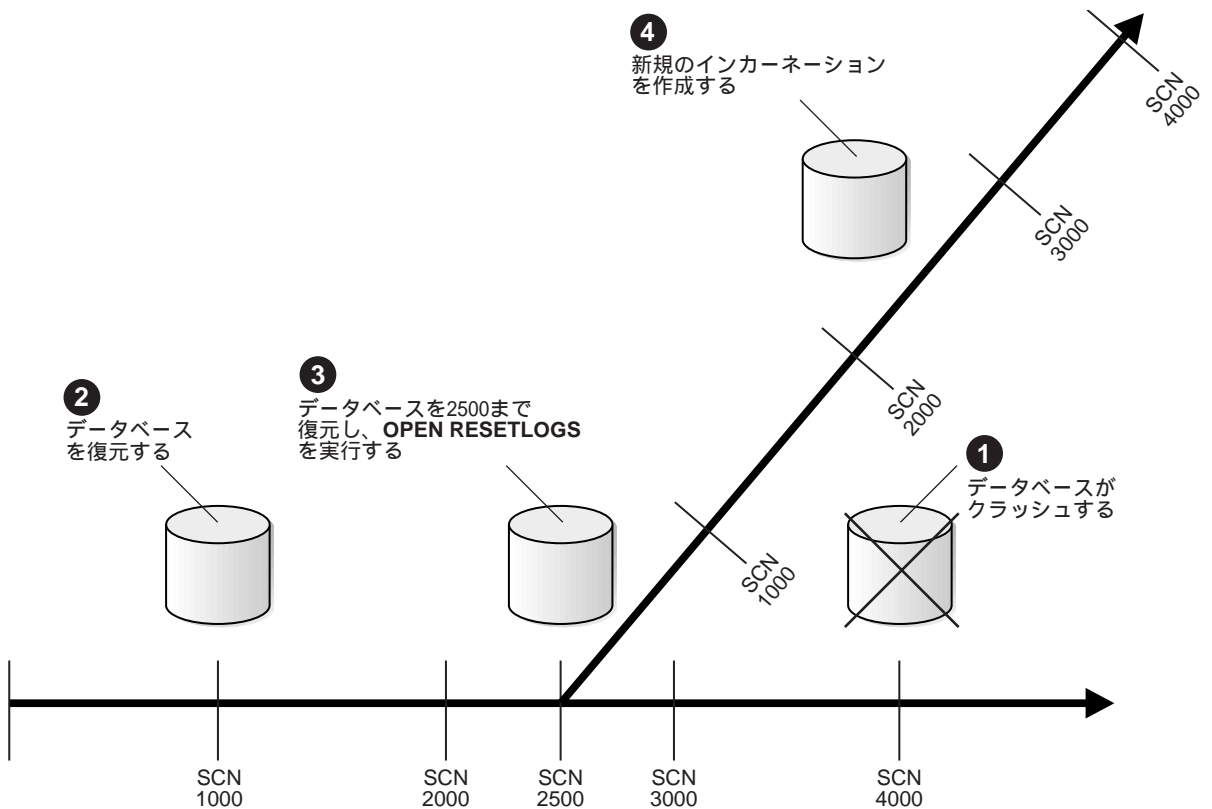
- ユーザーが誤って表を削除したなどの、ユーザー・エラーによってデータが損失した。
- アーカイブ REDO ログが見つからず完全回復が実行できない。
- 現行の制御ファイルが失われたため、バックアップ制御ファイルを使用して、データベースを開く必要がある。

不完全メディア回復を実行するには、回復する時点よりも前に作成したバックアップのデータ・ファイルをすべて復元する必要があります。さらに、回復の完了後 RESETLOGS オプションでデータベースを開く必要があります。この RESETLOGS 処理により、データベースのインカーネーションが新規作成されます。RESETLOGS の実行後に古いインカーネーションで生成されたアーカイブ REDO ログは、新しいインカーネーションではすべて無効になります。

図 1-2 に、アーカイブ REDO ログが失われたため、SCN2500 の時点の状態にまでしか回復できなかった、データベースの事例を示します。ここでは、SCN4000 でデータベースのクラッシュが発生しています。SCN1000 のバックアップを復元し、完全回復に備えました。しかし、アーカイブ REDO ログのうち 1 つが破損していました。失われたログの前のログには、SCN2500 が含まれています。そのため、この時点の状態まで回復した後、RESETLOGS オプションを指定してオープンしました。

図に示したとおり、データベースの新規インカーネーションで生成された新しい変更は、最終的には SCN4000 となります。データベースの新規インカーネーションにおける、SCN2500 から SCN4000 までの変更は、古いインカーネーションにおける、SCN2500 から SCN4000 までの変更とはまったく異なります。Oracle では、古いインカーネーションのログは、新しいインカーネーションに適用できません。したがって、古いインカーネーションの SCN2500 より前から、新規インカーネーションにバックアップを復元できません。

図 1-2 データベースの新規インカーネーションの作成



表領域の Point-in-Time 回復 表領域の Point-in-Time 回復 ([tablespace point-in-time recovery: TSPITR](#)) 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回復できます。TSPITR は、次のような状況に最も有効です。

- 誤って実行した、表の削除またはトランケート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のみに影響を与えたその他の DML 文から回復する場合。
- 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 (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る場合)。

- 大規模なデータベース環境で、データベース全体をバックアップから復元するのではなく、表領域単位で復元する（B-4 ページの「[表領域の Point-in-Time 回復の計画](#)」を参照して方針を決定してください）。

関連項目 : RMAN を使用した TSPITR の実行方法は、[付録 A の「Recovery Manager による表領域の Point-in-Time 回復の実行」](#)を参照してください。O/S の TSPITR の実行方法は、[付録 B の「オペレーティング・システム環境での表領域の Point-in-Time 回復の実行」](#)を参照してください。

回復オプション データベースを最新の状態に完全回復しない場合、回復を終了させる時点 Oracle に指示する必要があります。選択できる回復のタイプは次のとおりです。

回復のタイプ	内容
時間ベースの回復 (time-based recovery)	指定された時点の状態にデータを回復する。
取消しベースの回復 (cancel-based recovery)	CANCEL コマンドが発行される時点まで回復する。
変更ベースの回復 (change-based recovery)	指定された SCN の状態に回復する。
ログ順序回復 (log sequence recovery)	指定されたログ順序番号の状態に回復する。

関連項目 : RMAN を使用した不完全回復の方法は、9-23 ページの「[不完全回復の実行](#)」を参照してください。O/S コマンドを使用した不完全メディア回復の方法は、14-25 ページの「[不完全メディア回復の実行](#)」を参照してください。

RESETLOGS オプションによるデータベースのオープン 不完全回復を実行する場合は、必ずオンライン REDO ログをリセットしてデータベースをオープンする必要があります。リセットされたデータベースの新しいバージョンを、新規インカーネーションと呼びます。

RESETLOGS オプションを指定してデータベースをオープンすると、一部の REDO を破棄するよう Oracle に指示したことになります。破棄された REDO が次回以降の回復で適用されることはありません。たとえば、データベースの最新のログ順序番号が 100 の場合に、ログ順序番号 50 の状態に回復したとしても、RESETLOGS をオープンしないと、データベースのログ順序番号は再び 100 となります。その際、100 という古いアーカイブ・ログを使ってデータベースを回復しようとする、データ・ファイルが破損したり、内部エラーが生成される可能性があります。

RESETLOGS オプションを指定してデータベースを開いた場合は、必ず全データ・ファイルに新しい SCN とタイムスタンプが付与されます。これらの値はアーカイブ REDO ログのヘッダーにも記述されます。SCN およびタイムスタンプが合致しない限り、Oracle がアーカイブ REDO ログをデータ・ファイルに適用することはありません。したがって、

RESETLOGS 処理を実行すれば、古いアーカイブ・ログによってデータ・ファイルが破損することがなくなります。

オンライン REDO ログをリセットした後、データベース全体のバックアップを作成します。RESETLOGS 処理前に作成されたバックアップは、通常、新規インカーネーションでは有効とされません。ただし、この原則には次のように例外もあります。Oracle が、RESETLOGS 前のアーカイブ REDO ログにアクセスせずに回復が実行できる場合に限り、RESETLOGS 前のバックアップでも復元が可能です。

次に示すような RESETLOGS 前のバックアップが、新規インカーネーションで復元できます。

- 読取り専用にされた後の表領域のバックアップ（RESETLOGS 前に再度書き込み可能にされなかった場合のみ）
- NORMAL モードでオフラインにされた後の表領域のバックアップ（RESETLOGS 前に再度オンラインにされなかった場合のみ）
- 回復が終了し、かつ RESETLOGS をオープンする前に作成された読み書き可能表領域のバックアップ。つまり、バックアップから RESETLOGS までの間に、それ以上の回復もデータ・ファイルの変更もしなかった場合です。

RESETLOGS の直前に作成されたのではない読み書き可能表領域のバックアップは、復元できないことに注意してください。この制限は、読み書き可能表領域のデータ・ファイルに、バックアップから RESETLOGS までの間に変更がなかった場合にも適用されます。バックアップのデータ・ファイル・ヘッダーに設定されたチェックポイントは、制御ファイルのチェックポイントよりも古くなるため、Oracle はアーカイブ・ログを検索して、変更を適用する必要があるかどうかを判断する必要があります。また、RESETLOGS 前のアーカイブ・ログは新規インカーネーションでは無効となります。

関連項目 : RESETLOGS モードでデータベースを再起動する方法は、14-32 ページの「[メディア回復後のデータベースのオープン](#)」を参照してください。

回復方法の選択

物理ファイル回復の基本的な方法は 2 つあります。次の中から選択できます。

- RMAN ユーティリティを使用して回復を自動化する。
- SQL/SQL*Plus コマンドを実行し、データベースを手動で回復する。

RMAN による回復

RMAN の基本的な回復コマンドは、**restore** および **recover** です。ディスク上のバックアップ・セットまたはイメージ・コピーのデータ・ファイルを、現行の場所か新しい場所に復元する場合は、RMAN を使用します。アーカイブ REDO ログが含まれているバックアップ・セットも復元できます。

リカバリ・カタログを使用する場合には、RMAN は以前の全バックアップに関する重要なメタデータをすべて保持します。リカバリ・カタログを使わない場合には、RMAN は制御ファイルから必要なメタデータを取得します。

RMAN の **recover** コマンドを使用してメディア回復を実行し、増分バックアップを適用します。不完全メディア回復を実行するには、**set until** コマンドを使用します。バックアップ、そしてコピーの回復および復元の操作は、RMAN を使用すると完全に自動化されます。

関連項目 : データベースの復元方法は、9-2 ページの「[データ・ファイル、制御ファイルおよびアーカイブ REDO ログの復元](#)」を参照してください。RMAN を使用した完全回復の方法は、9-15 ページの「[データ・ファイルの回復](#)」を参照してください。RMAN の構文は、[第 11 章の「Recovery Manager のコマンド構文」](#)を参照してください。

SQL*Plus による回復

RMAN を使用しない場合は、SQL*Plus ユーティリティを使ってファイルの復元および回復を実行します。実行可能なものを次に示します。

- SQL*Plus の RECOVER コマンド (推奨)
- SQL の ALTER DATABASE RECOVER 文

使用しているオペレーティング・システムが Oracle Enterprise Manager をサポートしている場合は、GUI 環境で復元操作および回復操作が実行できます。詳細は、『Oracle Enterprise Manager 管理者ガイド』を参照してください。

いずれの場合も、データベース、表領域またはデータ・ファイルの回復が可能です。回復を実行する前に、次の手順を実行する必要があります。

1. 回復するファイルを選択する。多くの場合ビュー `VSRECOVER_FILE` を使用します。
2. メディア障害が原因で永久的な破損が生じたファイルのバックアップを復元する。バックアップを作成していなくても、データ・ファイルに最初に作成された時点からの REDO ログあり、かつ制御ファイルに破損したファイルの名前が含まれていれば、回復が実行できる場合があります。

データ・ファイルを元の場所に復元できない場合は、別の場所を選択し、制御ファイルにその場所を示します。

3. 必要なアーカイブ REDO ログ・ファイルを復元する。

上記の操作が完了したら、RMAN の **recover** コマンドまたは SQL*Plus の RECOVER 文を発行します。

関連項目 : RMAN を使用した完全回復の方法は、[第 9 章の「Recovery Manager による復元と回復」](#)を参照してください。オペレーティング・システム回復の方法は、[第 14 章の「オペレーティング・システム環境での回復の実行」](#)を参照してください。

SQL*Plus ユーティリティの使用方法は、SQL*Plus ユーザーズ・ガイドおよびリファレンスを参照してください。SQL コマンドの構文の詳細は、『Oracle8i SQL リファレンス』を参照してください。

データ構造の管理

この章では、バックアップおよび回復の成功の鍵となる、データベース構造の管理方法を説明します。この項では、次の項目について説明します。

- [バックアップおよび回復のデータ構造の概要](#)
- [制御ファイルの管理](#)
- [オンライン REDO ログの管理](#)
- [アーカイブ REDO ログの管理](#)

関連項目 : データ構造の概説は、『Oracle8i 概要』を参照してください。管理の詳細は、『Oracle8i 管理者ガイド』を参照してください。Parallel Server を使用している場合は、『Oracle8i Parallel Server 概要および管理』を参照してください。

バックアップおよび回復のデータ構造の概要

バックアップおよび回復の最も有効な方針は、前もって計画を立てることです。データの消失を避けるには、データ消失の可能性をさまざまに予測し、これに基づいて防護策を講じる必要があります。

あらかじめ計画を立てる上で重要となるポイントは、データベース構造の効果的な管理です。たとえば、制御ファイルが破損したとき、データベースのクラッシュを防ぐには何をしたらよいのでしょうか。ディスク障害が発生したときに、アーカイブ REDO ログの消失を防ぐには何をすべきでしょうか。データ・ファイルを除き、バックアップおよび回復計画の立案に、最も重要なデータ構造を次に示します。

- 制御ファイル
- オンライン REDO ログ
- アーカイブ REDO ログ (ARCHIVELOG モードで実行する場合)

これらのデータ構造が破損したり使用できなくなると、消失したデータの回復が不可能になります。

十分なリソースを確保していれば、次に示したデータ管理の基本計画に従うことにより、データ消失の可能性を低くできます。

- データ・ファイルを O/S レベルまたはハードウェア・レベルでミラー化する。
- 現行の制御ファイルを多重化またはミラー化し、少なくとも 2 つを別個のディスクでメンテナンスする。O/S またはハードウェアのミラー化を選択すると、システムを完全に作動させたまま、あらゆるメディア障害からの回復が可能になります。
- 2 つのメンバーで構成されるオンライン REDO ログ・グループを少なくとも 3 つメンテナンスする。グループのメンバーは、それぞれ別のディスク、別のコントローラに配置します。
- REDO ログを複数の宛先にアーカイブし、異なるメディアに頻繁にバックアップする。複数のバックアップをとり、それぞれを複数の異なるメディア装置に配置することをお勧めします。

注意： この章は、制御ファイル、オンライン REDO ログおよびアーカイブ REDO ログそれぞれの機能と、これらの基本的な管理方法を理解していることを前提としています。そうでない場合は、『Oracle8i 概要』および『Oracle8i 管理者ガイド』の関連する章を参照してください。

制御ファイルの管理

制御ファイルは、データベース・スキーマの情報が含まれている、小さなバイナリ・ファイルです。これは、データベースを正しく起動し操作するために必要な、データベース内の最も重要なファイルの一つです。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き込み可能にしてください。なんらかの理由で制御ファイルにアクセスできないと、データベースがマウントできなくなり、回復も困難になります。

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するために必要です。制御ファイルの情報を変更できるのは Oracle Server のみであり、ユーザーはデータベースの制御ファイルを編集できません。

制御ファイルがバックアップおよび回復時に重要な理由は、このファイルがさまざまな特性を持っているためです。たとえば、制御ファイルは次に示すような機能を持っています。

- データベース名（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）を識別する。
- 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・ファイルの名前と場所を記録する。
- アーカイブ REDO ログの名前と場所を記録する。
- データベースを同期化する際に必要となる、チェックポイントおよびログ順序番号を格納する。
- Recovery Manager バックアップの情報を格納する（Recovery Manager を使用している場合）。RMAN がオプションで使用するリカバリ・カタログによって、必要な情報が制御ファイルから取り出されます。
- データベースをマウント、オープンおよびメンテナンスするために、アクセス可能にしておく必要がある。

この項では、制御ファイルの管理に関連する次の項目について説明します。

- [制御ファイル情報の表示](#)
- [構造変更後の制御ファイルのバックアップ](#)
- [複数の制御ファイルのメンテナンス](#)
- [制御ファイルの消失からの回復](#)

制御ファイル情報の表示

制御ファイル管理の第 1 ステップは、情報の取得方法を習得することです。次に示すデータ・ディクショナリ・ビューには、役立つ情報が表示されます。

ビュー	説明
V\$CONTROLFILE	制御ファイルのファイル名を表示する。
V\$DATABASE	制御ファイルが現行のものかバックアップか、また、いつ作成されたかを示す。バックアップである場合は、最新のタイムスタンプも示します。

たとえば、次の問合せではデータベースの制御ファイルが表示されます。

```
SELECT name FROM v$controlfile;

NAME
-----
/vobs/oracle/dbs/cf1.f
/vobs/oracle/dbs/cf2.f
2 rows selected.
```

制御ファイルのタイプを表示するには、V\$DATABASE ビューに問合せを行います。

```
SELECT controlfile_type FROM v$database;

CONTROL
-----
BACKUP
```

次のコマンドを使用すると、データベースの制御ファイル、データ・ファイルおよび REDO ログ・ファイルがすべて表示されるので便利です。

```
SELECT member FROM v$logfile
UNION ALL
SELECT name FROM v$datafile
UNION ALL
SELECT name FROM v$controlfile;

MEMBER
-----
/vobs/oracle/dbs/rdo_log1.f
/vobs/oracle/dbs/rdo_log2.f
/vobs/oracle/dbs/tbs_01.f
/vobs/oracle/dbs/tbs_02.f
/vobs/oracle/dbs/tbs_11.f
/vobs/oracle/dbs/tbs_12.f
/vobs/oracle/dbs/tbs_21.f
/vobs/oracle/dbs/tbs_22.f
```

```
/vobs/oracle/dbs/tbs_13.f  
/vobs/oracle/dbs/cf1.f  
/vobs/oracle/dbs/cf2.f  
11 rows selected.
```

関連項目：動的パフォーマンス・ビューの詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。

構造変更後の制御ファイルのバックアップ

ユーザーが、データベースのデータ・ファイルまたはオンライン REDO ログを、追加、改名あるいは削除するたびに、Oracle は制御ファイルを更新し、この物理構造の変更を反映します。Oracle はこれらの変更を記録して次に示すファイルを識別します。

- 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。
- データベースを回復する必要があるときに必要となるファイル、または使用可能となるファイル。

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バックアップを作成してください。作成していない場合に、制御ファイルの破損が発生すると、バックアップ制御ファイルが障害の発生時点のデータベースの状態を正しく反映しなくなります。

制御ファイルはバイナリのコピーを生成するか、またはテキストのトレース・ファイルにバックアップします（宛先は USER_DUMP_DEST の初期化パラメータで指定します）。テキストのトレース・ファイルのスクリプトを実行すれば、制御ファイルの再作成ができます。制御ファイルのバックアップは、次に示したコマンドのうち、いずれかを実行した後で行います。

- ALTER DATABASE [ADD | DROP] LOGFILE
- ALTER DATABASE [ADD | DROP] LOGFILE MEMBER
- ALTER DATABASE [ADD | DROP] LOGFILE GROUP
- ALTER DATABASE [ARCHIVELOG | NOARCHIVELOG]
- ALTER DATABASE RENAME FILE
- CREATE TABLESPACE
- ALTER TABLESPACE [ADD | RENAME] DATAFILE
- ALTER TABLESPACE [READ WRITE | READ ONLY]
- DROP TABLESPACE

SQL コマンドを使用して制御ファイルのバイナリ・バックアップを作成する方法

1. データベースがマウントもオープンもされていなければ、SQL*Plus を使用してマウントまたはオープンする。

```
SQL> ALTER DATABASE MOUNT;
```

2. 次の ALTER DATABASE 文を発行し、バックアップ制御ファイルの宛先を指定する。

```
SQL> ALTER DATABASE BACKUP CONTROLFILE TO '/oracle/backup/cf.f';
```

制御ファイルをテキストのトレース・ファイルにバックアップする方法

制御ファイルをトレース・ファイルにバックアップしておく、このファイルのスクリプトを使用して、制御ファイルを再作成できます。

1. データベースがマウントもオープンもされていなければ、SQL*Plus を使用してマウントまたはオープンする。

```
SQL> ALTER DATABASE MOUNT;
```

2. 次のコマンドを発行する。

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

3. USER_DUMP_DEST ディレクトリにあるトレース・ファイルのスクリプトを実行する。オフラインの表領域と読取り専用の表領域のどちらがあるかによって、スクリプトの編集を先に行う必要がある場合があります。

たとえば、UNIX でトレース・ファイル rman_ora_839.trc を編集するには、次のように入力します。

```
% vi rman_ora_839.trc
```

```
*** SESSION ID:(8.1) 1998.12.09.13.26.36.000
*** 1998.12.09.13.26.36.000
# The following commands will create a new control file and use it
# to open the database.
# Data used by the recovery manager 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 "RMAN" NORESETLOGS ARCHIVELOG
    MAXLOGFILES 32
    MAXLOGMEMBERS 2
    MAXDATAFILES 32
    MAXINSTANCES 1
    MAXLOGHISTORY 1012
LOGFILE
    GROUP 1 '/oracle/dbs/tl_log1.f' SIZE 200K,
```



```

GROUP 2 '/oracle/dbs/tl_log2.f' SIZE 200K
DATAFILE
  '/oracle/dbs/tbs_01.f',
  '/oracle/dbs/tbs_02.f',
  '/oracle/dbs/tbs_11.f',
  '/oracle/dbs/tbs_12.f',
  '/oracle/dbs/tbs_21.f',
  '/oracle/dbs/tbs_22.f',
CHARACTER SET WE8DEC
;
# Configure snapshot controlfile filename
EXECUTE SYS.DBMS_BACKUP_RESTORE.CFILESETSNAPSHOTNAME('/oracle/dbs/snapcf_
rman.f');
# Recovery is required if any of the datafiles 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;
# No tempfile entries found to add.

```

関連項目 : 制御ファイルの管理の詳細は、『Oracle8i 管理者ガイド』を参照してください。トレース・ファイル編集の実例は、13-12 ページの「[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](#)」を参照してください。

複数の制御ファイルのメンテナンス

オンライン REDO ログ・ファイルの場合と同様に、Oracle では、制御ファイルを多重化できます。つまり、同じ内容の複数のコピーをオープンし、書込みを行うことができます。Oracle は、制御ファイルの各コピーに同じデータを書き込みます。またミラー化することも可能です。つまり、O/S で制御ファイルのコピーを複数の物理ディスクに書き込むことができます。

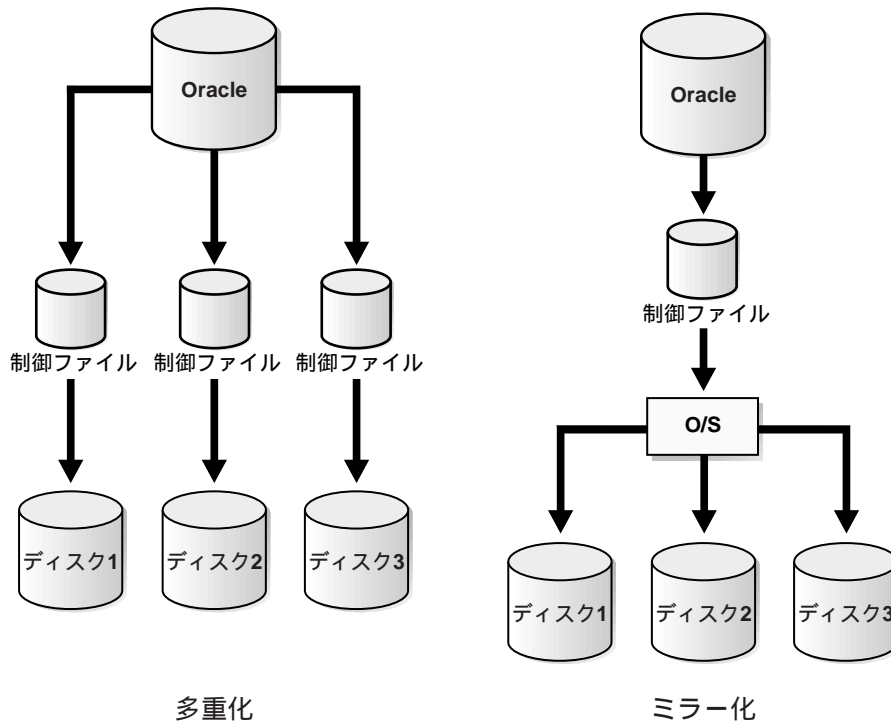
O/S レベルのミラー化の方が、Oracle レベルの多重化よりも好ましいことがあります。O/S ミラー化は、通常いずれかのミラーの障害を許容しますが、Oracle はそのようになっていないためです。Oracle による多重化時に、ミラーのいずれかで障害が発生すると、インスタンスが停止します。その場合、ユーザーは次のいずれかの処置が必要です。

- 障害の発生したディスクを修復した後、正常な制御ファイルを以前の場所にコピーし、再起動する。
- 正常な制御ファイルを新しい場所にコピーし `init.ora` ファイルの `CONTROL_FILES` パラメータを適宜編集する。

O/S またはハードウェアのミラー化で、確保される冗長性は多重化の場合と同じです。ただし、障害が発生しても、可用性が損なわれることはほとんどありません。

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きます。制御ファイルがデータベースの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。制御ファイルの多重化またはミラー化を行っていない場合、回復がさらに困難になります。したがって、各データベースでは、多重化またはミラー化した制御ファイルを使用してください。

図 2-1 制御ファイルの多重化およびミラー化



制御ファイルの多重化

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と、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、別のディスクにある現行の制御ファイルのコピーが使用できます。障害の発生したディスクを修復し、正常な制御ファイルを前の場所にコピーすれば、メディア回復を実行することなくインスタンスを容易に再起動できます。

ディスクが修復できない場合は、CONTROL_FILES の初期化パラメータを編集して新しい場所を指定し、制御ファイルの正常なコピーをここにコピーします。

制御ファイルの多重化の唯一の欠点は、制御ファイルの更新処理（データ・ファイルの追加やデータベースのチェックポイントなど）にかかる時間が少し長くなることです。ただし、このパフォーマンスのオーバーヘッド増加は、通常は重要視する必要はありません。このことは、複数の同時書込みが実行可能なオペレーティング・システムに特に当てはまります。パフォーマンスが多少犠牲になっても、複数の制御ファイルを併用することの方が重要です。

注意： オラクル社は、最低 2 つの制御ファイルを別個のディスクにメンテナンズすることをお勧めします。

多重化された制御ファイルの、注意すべき特徴を次に示します。

- データベースのパラメータ・ファイルの初期化パラメータ CONTROL_FILES に、最低 2 つのファイル名が記述される。
- CONTROL_FILES のパラメータに記述された最初のファイルは、データベース運用中に Oracle Server が読み取る唯一のファイルである。
- 制御ファイルがデータベース運用中に使用不能になると、そのインスタンスは運用不能になり異常終了する。障害が発生した制御ファイルの場所に、正常な制御ファイルをコピーします。メディア障害によってディスクにアクセスできない場合は、新しい場所に正常な制御ファイルをコピーし init.ora ファイルを編集します。

制御ファイルのミラー化

ディスクのミラー化をサポートしているオペレーティング・システムは、ミラー化ディスクが利用可能です。ミラー化ディスクを使用すると、Oracle は複数の物理ディスクを 1 つのディスクとして認識します。Oracle はデータを 1 回書き込み、O/S はこのデータをそれぞれの物理ディスクに書き込みます。各ファイルは互いのミラーとなります。すなわち、複製となります。

ディスク・ミラー化の利点は、ディスクの 1 つが使用不能になっても、その他のディスクで処理を継続できることです。したがって、制御ファイルは単一ポイントの障害から保護されます。制御ファイルをミラー化ディスク・システムに格納すると、制御ファイルのアクティブなコピーを書き込むよう Oracle を設定するのみであることに注目してください。

制御ファイルの消失からの回復

制御ファイルの回復または再作成が必要と考えられる実例を次に示します。

- メディア障害が原因で制御ファイルが破損または消失した。制御ファイルを多重化またはミラー化し、それぞれを別個のディスクに保管しておけば、これを回避できる可能性が高くなります。

- データベースの名前を変更した。制御ファイルは該当するデータベースの名前を記録しているため、そのデータベースの再作成が必要となります。

制御ファイルの現行コピーを使用した制御ファイルの破損の回復

この手順は、CONTROL_FILES パラメータに指定された制御ファイルのうちの 1 つが破損しているが、その制御ファイルのディレクトリにはアクセスが可能であり、さらに多重化またはミラー化された現行コピーを保持しているという状況を前提としています。

1. インスタンスを停止した後、O/S コマンドを使用して、不良制御ファイルを正常なコピーで上書きする。

```
% cp '/disk2/copy/cf.f' '/disk1/oracle/dbs/cf.f';
```

2. SQL*Plus を起動し、データベースをマウントまたはオープンする。

```
SQL> STARTUP MOUNT;
```

制御ファイルの現行コピーを使用した制御ファイルの永続的な破損の回復

この手順は、CONTROL_FILES パラメータに指定された制御ファイルのうちの 1 つが、永続的なメディア障害によりアクセス不能になったが、多重化またはミラー化された現行コピーは保持しているという状況を前提としています。

1. インスタンスを停止した後、O/S コマンドを使用して、制御ファイルの現行コピーを新規かつアクセス可能な場所にコピーする。

```
% cp '/disk2/copy/cf.f' '/disk3/copy/cf.f';
```

2. init.ora の CONTROL_FILES パラメータを編集し、障害の発生した場所を新しい場所に置き換える。

```
CONTROL_FILES = '/oracle/dbs/cf1.f','/disk3/copy/cf.f'
```

3. SQL*Plus を起動し、データベースをマウントまたはオープンする。

```
SQL> STARTUP MOUNT;
```

オンライン REDO ログの管理

回復操作で最も重要な構造と考えられるものはオンライン REDO ログ (redo log) です。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。Oracle データベースの各インスタンスには、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。

警告： 現行のオンライン・ログはバックアップしないでください。現行のオンライン・ログのバックアップを復元すると、このバックアップが REDO スレッドの終わりとみなされます。REDO スレッドに追加の REDO が生成されている可能性があるため、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 スレッドの終わりと誤って検出されて処理が途中で打ち切れ、データベースが破損するおそれがあります。

各データベース・インスタンスには、専用の REDO ログ・グループ (redo log groups) があります。このようなオンライン REDO ログ・グループは、多重化されているかどうかにかかわらず、インスタンスのオンライン REDO のスレッド (thread) と呼びます。通常の構成では、1 つのデータベース・インスタンスのみが 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。そのため、各インスタンスにはそれぞれ専用のスレッドがあります。

注意： このマニュアルは、Oracle Parallel Server を使用し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ています。したがって、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、『Oracle8i Parallel Server 概要および管理』を参照してください。

関連項目： オンライン REDO ログの概説は、『Oracle8i 概要』を参照してください。オンライン REDO ログの管理方法の詳細は、『Oracle8i 管理者ガイド』を参照してください。

オンライン REDO ログ情報の表示

次に示すデータ・ディクショナリ・ビューには、アーカイブ REDO ログに役立つ情報が示されます。

ビュー	説明
V\$LOG	オンライン REDO ログ・グループ、グループのメンバー数およびアーカイブされたログを示す。
V\$LOGFILE	REDO ログ・グループのメンバーのファイル名および状態情報を示す。

たとえば、次の問合せではアーカイブの必要なオンライン REDO ログ・グループが表示されます。

```
SELECT group#, sequence#, status, archived FROM v$log;
```

GROUP#	SEQUENCE#	STATUS	ARC
1	43	CURRENT	NO
2	42	INACTIVE	YES

2 rows selected.

各ログ・グループのメンバーを表示するには、次のように V\$LOGFILE ビューに問い合わせます。

```
SELECT group#, member FROM v$logfile;
```

GROUP#	MEMBER
1	/oracle/dbs/t1_log1.f
2	/oracle/dbs/t1_log2.f

2 rows selected.

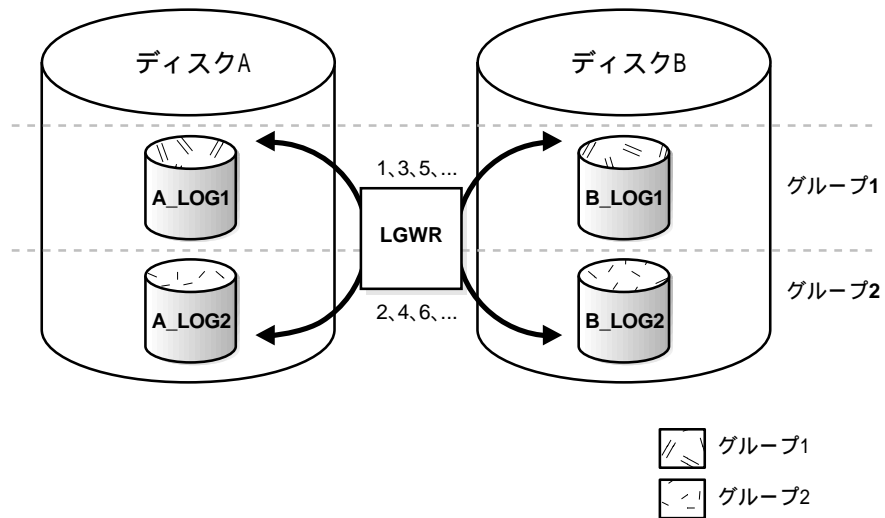
関連項目: データ・ディクショナリ・ビューの詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。

オンライン REDO ログ・ファイルの多重化

Oracle には、損傷が発生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。オンライン REDO ログ・ファイルを**多重化 (multiplexing)** すると、LGWR は、同じ情報を複数の同一視できるオンライン REDO ログ・ファイルに同時に書き込むので、単一ポイントの障害を排除できます。REDO ログを O/S レベルでミラー化するという方法をとることもできますが、この場合は O/S またはハードウェアに起因する破損が発生する危険性があります。多くの場合、オンライン・ログを多重化するという方法が最も適切です。

警告： REDO ログ・ファイルは多重化しておくか、または O/S レベルでミラー化しておくことをお勧めします。回復が必要な場合に REDO データが失われていると、重大な結果を招くことがあります。

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。図 2-2 では、ファイル A_LOG1 および B_LOG1 は、グループ 1 のメンバーで、A_LOG2 および B_LOG2 は、グループ 2 のメンバーになると構成されています。グループの各メンバーは、それぞれまったく同じサイズになっている必要があります。

LGWR が同一のログ順序番号を割り当てているため、同じグループ内の各メンバーは同時にアクティブになる（LGWR によって同時に書込みが行われる）ことに注意してください。図 2-2 では、LGWR はまずファイル A_LOG1 および B_LOG1 に書込みを実行し、続いて A_LOG2 および B_LOG2 に書込みを実行するというように処理を行います。異なるグループのメンバー、たとえば A_LOG1 および B_LOG2 に、LGWR が同時に書込みを行うことはありません。

オンライン REDO ログの障害に対する処置

LGWR からグループ内のメンバーへの書込みができない場合、Oracle はそのメンバーに STALE のマークを付し、LGWR トレース・ファイルとデータベースの ALERT ログに、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・メッセージを書き込みます。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。

状況	処理
LGWR がグループの少なくとも 1 つのメンバーへの書込みに成功した。	通常どおりに書込みが行われる。LGWR はグループ内の使用可能なメンバーに書込みを行い、使用不能なメンバーは無視します。
アーカイブが必要なため、ログ・スイッチの時点で次のグループに LGWR がアクセスできない。	グループが使用可能になるまで、すなわちアーカイブされるまで、データベース処理が一時的に停止する。
ディスク障害が原因で、LGWR がログ・スイッチの時点で次のグループのどのメンバーにもアクセスできない。	Oracle はエラーを戻し、データベース・インスタンスが停止する。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。
次のグループのどのメンバーにもアクセスできず、失われた REDO ログを過ぎた時点までデータベースのチェックポイントが進んでいる。	Oracle は、REDO ログに記録されたデータをデータ・ファイルに保存しているため、メディア回復の必要はない。この場合は、アクセスできない REDO ログ・グループを削除するだけで済みます。
ARCHIVELOG モードで運用している場合に、アーカイブされていない REDO ログを削除する。	そのログが削除される前に ALTER DATABASE CLEAR UNARCHIVED LOG を発行してアーカイブを使用禁止にする。
LGWR が書込み中、グループのどのメンバーにもアクセスできない。	Oracle はエラーを戻し、データベース・インスタンスが即時停止する。 この場合、メディア回復を実行する必要がある場合があります。ログのあるメディアが実際には失われていない場合（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場合があります。この場合は、ドライブをオンにして Oracle にインスタンス回復を行わせるだけで済みます。

関連項目：多重化オンライン REDO ログの構成方法の詳細は、『Oracle8i 管理者ガイド』を参照してください。

アーカイブ REDO ログの管理

データベースを ARCHIVELOG モードで運用している場合、Oracle ではオンライン REDO ログの満杯になったグループ（[アーカイブ REDO ログ \(archived redo log\)](#)）を、1 つ以上のオフラインの宛先に保存することができます。アーカイブとは、オンライン REDO ログをアーカイブ REDO ログに変換する処理のことです。

アーカイブ・ログは次のような場合に使用します。

- データベースを回復する。
- スタンバイ・データベースを更新する。
- データベースの履歴に関する情報を LogMiner ユーティリティで取得する。

アーカイブ REDO ログ・ファイルとは、オンライン REDO ログ・グループのメンバーのうち、同じ内容でかつ満杯になったメンバーの 1 つをコピーしたものです。このファイルには、グループ内の同じ内容のメンバーに存在する REDO エントリが含まれ、そのグループに対応する一意のログ順序番号も保存されています。たとえば、グループ 1 にメンバー・ファイル A_LOG1 および B_LOG1 が含まれているときに、オンライン REDO ログを多重化すると、同じ内容のメンバーのうちの 1 つが ARCn プロセスでアーカイブされます。万一 A_LOG1 が破損しても、ARCn は、同じ内容の B_LOG1 をアーカイブできます。

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、LGWR はそのグループを再使用できず、その結果上書きもできません。つまり、アーカイブ REDO ログには、アーカイブを使用可能にした後に作成された各オンライン REDO グループのコピーが含まれることになります。現行のオンライン・ログの内容をバックアップする場合は、必ず同ログをアーカイブし、さらにアーカイブしたログをバックアップするという方法が最もよい方法です。

オンライン REDO ログをアーカイブすることで、アーカイブを使用可能にした後にデータベースに加えたすべての変更のコピーが保存されます。メディア障害が発生しても、このアーカイブ REDO ログを使用すれば失われたデータを回復できます。

関連項目：アーカイブ REDO ログの管理方法、および LogMiner の使用方法の詳細は、『Oracle8i 管理者ガイド』を参照してください。スタンバイ・データベースの管理方法は、[第 16 章の「スタンバイ・データベースの管理」](#)を参照してください。

NOARCHIVELOG モードおよび ARCHIVELOG モードの選択

この項では、データベースを NOARCHIVELOG モードまたは ARCHIVELOG モードで運用する場合に考慮が必要な点について説明します。項目は次のとおりです。

- [NOARCHIVELOG モードでのデータベースの運用](#)
- [ARCHIVELOG モードでのデータベースの運用](#)

NOARCHIVELOG モードでのデータベースの運用

データベースを NOARCHIVELOG モードで運用する場合、オンライン REDO ログのアーカイブは使用禁止にします。データベースの制御ファイルは、満杯となったグループにもアーカイブが必要ないことを示しています。したがって、ログ・スイッチが発生し、満杯のグループがアクティブでなくなると、LGWR はそのグループを再使用できます。

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、またはオペレータ・エラーが起きた場合にデータベース内のデータを 1 つも失わないためには、ARCHIVELOG モードを使う必要があります。満杯のオンライン REDO ログ・ファイルをアーカイブすることにより、管理作業が増える可能性があることに注意してください。

ログをアーカイブしていないと、Oracle がログを切り替えて書き込みを開始したとき、そのログに含まれている REDO をすべて失うことになります。データベースを NOARCHIVELOG モードで実行すると、次のような結果が得られます。

- 最新の全データベース・バックアップの状態にまでしか、データベースを（回復ではなく）復元できない。適用するアーカイブ・ログが存在しないため、バックアップにアーカイブ・ログを適用できません。
- データベースのオペレーティング・システム・バックアップは、データベースを正しく停止した場合でなければ実行できない。
- 全体データベース・バックアップの復元、およびこのデータベースのオープン、データベースを正しくクローズしてバックアップをとっていた場合のみ可能である。一貫性のないバックアップを復元できるのは、REDO ログが破損しておらず、かつバックアップ実行後に生成された REDO がすべてこの REDO ログに含まれている場合に限られます。
- オンライン表領域バックアップは実行できない。また、データベースが ARCHIVELOG モードで運用されていたときにとったオンライン表領域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 NOARCHIVELOG モードで運用する場合は、定期的かつ頻繁に全データベース・バックアップを実行してください。そのようにしないと、古いバックアップを復元しなければならなくなり、日単位、週単位、場合によっては月単位分の変更を失う場合があります。

ARCHIVELOG モードでのデータベースの運用

データベースを ARCHIVELOG モードで実行する場合、Oracle ではオンライン REDO ログをアーカイブする必要があります。アーカイブは手動で実行することも、自動アーカイブを使用可能にしておくこともできます。

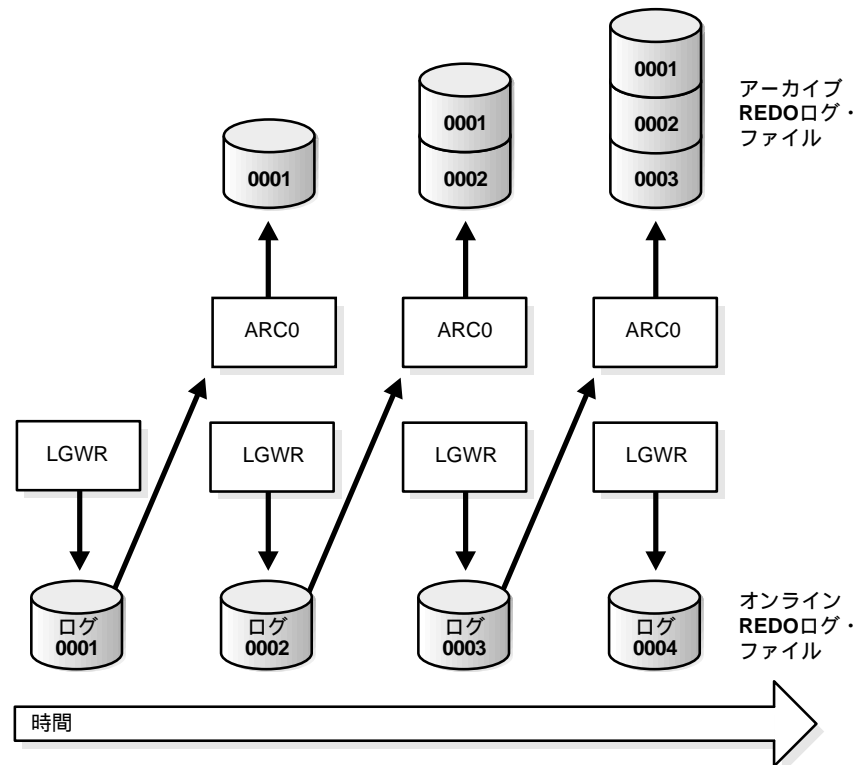
ARCHIVELOG モードでは、データベース制御ファイルは、満杯でかつアーカイブ前のオンライン REDO ログ・ファイルのグループを LGWR が使用できないことを示しています。満杯になったグループが ARC n で利用できるようになるのは、ログ・スイッチが発生した直後です。グループが正しくアーカイブされると、Oracle はこのグループを再使用できます。

満杯になったグループをアーカイブする利点は次のとおりです。

- データベース・バックアップとオンライン・ログおよびアーカイブ・ログのバックアップを組み合わせることにより、コミットされたトランザクションをすべて回復することが可能になる。
- データベースを、指定した時間、SCN またはログ順序番号の状態に回復できる。

- データベースをオープンした状態で表領域のバックアップをとることができる。
- データベースをオープンした状態で表領域を回復できる。
- 元のデータベースのアーカイブ REDO ログを継続してスタンバイ・データベースに適用することにより、元のデータベースと同じ状態にスタンバイ・データベースを保つことができる。

図 2-3 ARCHIVELOG モードでのオンライン REDO ログ・ファイルの使用方法



関連項目：ARCHIVELOG モードおよび自動アーカイブを有効にする方法は、『Oracle8i 管理者ガイド』を参照してください。スタンバイ・データベースの管理回復オプションの詳細は、16-15 ページの「スタンバイ・データベースの回復モードのメンテナンス」を参照してください。

アーカイブ REDO ログ情報の表示

次に示すデータ・ディクショナリ・ビューには、アーカイブ REDO ログに関する有益な情報が表示されます。

ビュー	説明
V\$DATABASE	データベースが ARCHIVELOG モードと NOARCHIVELOG のどちらで運用されているかを識別する。
V\$ARCHIVED_LOG	制御ファイルのアーカイブ・ログ情報を表示する。
V\$ARCHIVE_DEST	現行インスタンス、全アーカイブ先の宛先、現行の値、モード、およびそれらの状態が記述される。
VSLOG	データベースのオンライン REDO ログ・グループがすべて表示され、アーカイブの必要なグループが示される。
V\$LOG_HISTORY	アーカイブされたログおよびそれぞれの SCN 範囲など、ログの履歴情報が記述される。

たとえば、次の問合せではアーカイブの必要なオンライン REDO ログ・グループが表示されます。

```
SELECT group#, archived FROM sys.v$log;
```

```
GROUP#      ARC
-----  ---
1             YES
2             NO
```

現行のアーカイブ・モードを確認するには、V\$DATABASE ビューに問合せを行います。

```
SELECT log_mode FROM sys.v$database;
```

```
LOG_MODE
-----
NOARCHIVELOG
```

SQL*Plus 文の ARCHIVE LOG LIST を使用しても、接続されたインスタンスのアーカイブ情報を表示できます。

```
ARCHIVE LOG LIST;

Database log mode                ARCHIVELOG
Automatic archival              ENABLED
Archive destination              /oracle/log
Oldest online log sequence      30
Next log sequence to archive    31
Current log sequence number     33
```

現行のインスタンスのアーカイブ REDO ログ設定に関する必要な情報は、この表示からすべて取得できます。

- データベースは ARCHIVELOG モードで運用されている。
- 自動アーカイブが使用可能になっている。
- アーカイブ REDO ログの宛先（オペレーティング・システム固有）
- 満杯になったオンライン REDO ログ・グループのうち、最も古いもののログ順序番号は 30 である。
- 満杯になったオンライン REDO ログ・グループのうち、次にアーカイブされるもののログ順序番号は 31 である。
- 現行のオンライン REDO ログ・グループのログ順序番号は 33 である。

アーカイブの必要な次のログ順序番号以上現行ログ順序番号未満の順序番号が付されている REDO ログ・グループは、すべてアーカイブする必要があります。たとえば、前述の表示は、順序番号 31 および 32 のオンライン REDO ログ・グループをアーカイブしなければならないことを示しています。

関連項目：データ・ディクショナリ・ビューの詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。

複数の場所への REDO ログのアーカイブ

アーカイブ REDO ログの宛先は、1 つのみの指定も複数指定も可能です。ファイル破損およびメディア障害の危険を避けるため、ログは複数のディスクにアーカイブすることをお薦めします。

アーカイブ・ログの場所の数を指定するには、初期化パラメータ内の相互排他となっている 2 つのセットのうち、いずれかを設定します。

- LOG_ARCHIVE_DEST_ *n* (*n* は、1 から 5 の整数)
- LOG_ARCHIVE_DEST とオプションの初期化パラメータ LOG_ARCHIVE_DUPLEX_DEST

最初の方法では、LOG_ARCHIVE_DEST_ *n* パラメータを使用して、1 ~ 5 つの間で、アーカイブの宛先を指定します。LOG_ARCHIVE_DEST_1、LOG_ARCHIVE_DEST_2 などというように、末尾の数字のパラメータで、一意的に宛先を識別します。パス名を指定するには LOCATION キーワードを、ネット・サービス名を指定するには SERVICE キーワードをそれぞれ使用します（スタンバイ・データベースとあわせて使用する場合）。

2 つ目の方法では、最大 2 つの場所が指定できます。プライマリ・アーカイブの宛先を指定するには LOG_ARCHIVE_DEST パラメータを、オプションのセカンダリの場所を指定するには LOG_ARCHIVE_DUPLEX_DEST をそれぞれ指定します。Oracle が REDO ログをアーカイブする場合は、パラメータのいずれかのセットで指定されたすべての宛先にアーカイブします。

注意： アーカイブ・ログを O/S レベルでミラー化することもできます。

LOG_ARCHIVE_DEST_*n*を使用したアーカイブの宛先の設定方法

1. LOG_ARCHIVE_DEST_*n* を編集し、1 ~ 5 つの宛先を指定する。たとえば、次のように入力します。

```
LOG_ARCHIVE_DEST_1 = 'LOCATION=/disk1/arc/'
LOG_ARCHIVE_DEST_2 = 'LOCATION=/disk2/arc/'
LOG_ARCHIVE_DEST_3 = 'LOCATION=/disk3/arc/'
```

2. LOG_ARCHIVE_FORMAT パラメータを編集する。ログ順序番号をファイル名に含める場合は %s を、スレッド番号を含める場合は %t をそれぞれ使用します。大文字 (%S および %T) を使用し、ゼロを使ってファイル名を左側に埋め込む。たとえば、次のように入力します。

```
LOG_ARCHIVE_FORMAT = arch%t_%s.arc
```

たとえば、この設定では、スレッド 1 のログ順序番号 100 ~ 102 に次のファイルが生成されます。

```
/disk1/arc/arch1_100.arc, /disk1/arc/arch1_101.arc, /disk1/arc/arch1_102.arc,
/disk2/arc/arch1_100.arc, /disk2/arc/arch1_101.arc, /disk2/arc/arch1_102.arc,
/disk3/arc/arch1_100.arc, /disk3/arc/arch1_101.arc, /disk3/arc/arch1_102.arc
```

3. データベースがオープンしていれば、SQL*Plus セッションを起動してデータベースを停止する。たとえば、次のように入力します。

```
SHUTDOWN IMMEDIATE;
```

4. データベースをマウントするかオープンして設定を有効にする。たとえば、次のように入力します。

```
STARTUP;
```

LOG_ARCHIVE_DEST および LOG_ARCHIVE_DUPLEX_DEST でのアーカイブの宛先の設定方法

1. init.ora ファイルを編集し、LOG_ARCHIVE_DEST および LOG_ARCHIVE_DUPLEX_DEST パラメータに対して宛先の指定をする。データベースがオープンしていれば、ALTER SYSTEM コマンドを使用して動的にこのパラメータを編集することもできます。

たとえば、パラメータを次のように変更します。

```
LOG_ARCHIVE_DEST = '/disk1/arc'
LOG_ARCHIVE_DUPLEX_DEST_2 = '/disk2/arc'
```

2. LOG_ARCHIVE_FORMAT パラメータを編集する。ログ順序番号をファイル名に含める場合は %s を、スレッド番号を含める場合は %t をそれぞれ使用します。大文字 (%S および %T) を使用し、ゼロを使ってファイル名を左側に埋め込む。データベースがオープンしていれば、ALTER SYSTEM コマンドを使用してこのパラメータを修正できます。

たとえば、次のように入力します。

```
LOG_ARCHIVE_FORMAT = arch_%t_%s.arc
```

たとえば、この設定では、スレッド 1 のログ順序番号 300 ~ 302 に次のファイルが生成されます。

```
/disk1/arc/arch_1_300.arc, /disk1/arc/arch_1_301.arc, /disk1/arc/arch_1_302.arc,  
/disk2/arc/arch_1_300.arc, /disk2/arc/arch_1_301.arc, /disk2/arc/arch_1_302.arc
```

3. データベースがオープンしていれば、SQL*Plus セッションを起動して停止する。たとえば、次のように入力します。

```
SHUTDOWN IMMEDIATE;
```

4. データベースをマウントするかオープンして init.ora ファイルの設定を有効にする。たとえば、次のように入力します。

```
STARTUP;
```

Oracle には、便利なアーカイブ・オプションが多数用意されています。次の方法の詳細は『Oracle8i 管理者ガイド』を参照してください。

- アーカイブの宛先ごとの状況情報を取得する。たとえば、特定の宛先にアーカイブできないといった問題があるのかどうかを確認できます。
- アーカイブのモードを標準アーカイブ送信 (normal archiving transmission) またはスタンバイ転送 (standby transmission) に指定する。標準転送では、ファイルがローカル・ディスクに送信されます。スタンバイ転送では、ファイルがネットワークを介してローカル・スタンバイ・データベース、またはリモート・スタンバイ・データベースに転送されます。
- Oracle が正常にアーカイブする必要がある宛先の最小数を指定する。
- 失敗した宛先に対する ARCn の再アーカイブの試行時点および回数を指定する。
- データベース・インスタンス 1 つにつき最大 10 の ARCn プロセスを指定し、アーカイブ操作をパラレル化する。
- LogMiner ユーティリティを使用し、オンラインおよびアーカイブ REDO ログの内容を分析する。

バックアップおよび回復計画の立案

この章では、効果的なバックアップおよび回復計画を立案するにあたっての、指針および考慮点について説明します。この項では、次の項目について説明します。

- バックアップ・タイプの選択
- バックアップ方法の選択
- バックアップ形式の選択
- バックアップ計画の立案
- 回復計画の立案

バックアップ・タイプの選択

バックアップ計画を立案する際には、どのタイプのバックアップを実行できるか把握する必要があります。物理バックアップは、どのタイプの場合も、ファイルのバックアップもファイルのグループのバックアップも可能です。この項では、次の項目の定義および説明を行います。

- [全体データベース・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](#)

論理バックアップはエクスポートともいいます。詳細は、『Oracle8i ユーティリティ・ガイド』を参照してください。

全体データベース・バックアップ

[全体データベース・バックアップ \(whole database backup \)](#) には、制御ファイルのバックアップと、データベースに属するすべてのデータベース・ファイルを含める必要があります。全体データベース・バックアップは、最も一般的なバックアップの形態です。

全体データベース・バックアップでは、データベースを特定のアーカイブ・モードで運用する必要はありません。ただし、全体データベース・バックアップを実行する場合は、ARCHIVELOG モードでのバックアップと NOARCHIVELOG モードでのバックアップから生じる結果の違いに注意してください (2-15 ページの「[NOARCHIVELOG モードおよび ARCHIVELOG モードの選択](#)」を参照してください)。

全体データベース・バックアップは、一貫性のある場合とない場合があります。この項では、次の項目について説明します。

- [一貫性バックアップ](#)
- [非一貫性バックアップ](#)

関連項目 : RMAN コマンドを使用してデータベースをバックアップする方法の詳細は、[第 8 章の「Recovery Manager でのバックアップおよびコピーの作成」](#)を参照してください。

O/S コマンドを使用してデータベースをバックアップする方法の詳細は、[第 13 章の「オペレーティング・システム環境でのバックアップの実行」](#)を参照してください。

一貫性バックアップ

データベース全体の[一貫性バックアップ \(consistent backup \)](#)とは、すべての読み書き可能なデータ・ファイルおよび制御ファイルが、同じ SCN でチェックポイントされているバックアップの形態です。また、オンラインかつ読み書き可能なデータ・ファイルはいずれも " ファジー " ではありません。つまり、SCN 以降の変更はヘッダーに含まれていません。

Oracle は、データ・ファイルのヘッダーすべてを、制御ファイルのデータ・ファイル・ヘッダー情報と照合して、復元されたバックアップに一貫性があるかを確認します。

制御ファイルおよびデータ・ファイルは、データベースのチェックポイント時に一貫性を持つようになります。一貫性バックアップで、古い SCN を持つことが許されている表領域は読取り専用または NORMAL モードでオフラインにされた表領域のみです。このような表領域には、変更が加えられておらず、回復が必要ないため、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。オフライン・データ・ファイルのチェックポイント SCN と、制御ファイルのオフライン SCN が合致している場合、Oracle は同データ・ファイルに REDO が必要ないことを検知します。

重要なのは、REDO ログを適用しなくても、一貫性のある全体データベース・バックアップを復元すれば、データベースをオープンできる点です。そのデータは、すでに一貫性を持っているためです。したがって、復元されたデータ・ファイルを正しくするためのアクションは必要ありません。これは、1 年を経過した一貫性のあるデータベースのバックアップでも、メディア回復および Oracle によるインスタンス回復を実行せずに復元できることを意味します。

警告： 回復処理を行わずに全体データベース・バックアップを復元する時は、一貫性のある全体データベース・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のみを使用してください。回復処理を実行するとき、現行の制御ファイルがある場合、古い制御ファイルは復元しないでください。ただし、データベース構造が現在の構造と異なっていたときの状態にする Point-in-Time 回復を実行する場合を除きます。

一貫性のある全体データベース・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。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、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。ただし、データベースを読取り専用モードでオープンしていた場合を除きます。インスタンス回復はオープン時に実行する必要があります。

一貫性のある全体データベース・バックアップは、NOARCHIVELOG モードで運用されているデータベースにおける唯一の有効なバックアップ方法です。一貫性のないバックアップでは、REDO を適用して一貫性を確保する必要があるためです。さらに、NOARCHIVELOG モードでは、Oracle は REDO レコードをアーカイブせずに上書きするためです。

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としたり、これを現行以外の時点の状態にバックアップするためには、メディア回復を実行します。現行の制御ファイルを使用して回復を実行した場合、Oracle はメディア回復をデータ・ファイル・ヘッダー内の最も低いチェックポイント SCN から開始します。これに対し、バックアップ制御ファイルを使用した場合は、制御ファイルの SCN またはデータ・ファイル・ヘッダーの最も低い SCN の低い方を使ってメディア回復を実行します。

メディア回復を実行するにはアーカイブ REDO ログを適用します。Recovery Manager を使用している場合は、増分バックアップとアーカイブ・ログの両方が、いずれか一方を適用してメディア回復を実行します。REDO データは、すべてアーカイブ REDO ログおよびオンライン REDO ログにあります。

非一貫性バックアップ

データベース全体の**非一貫性バックアップ (inconsistent backup)** は、読み書き可能データ・ファイルおよび制御ファイル全体が、同じ SCN に関してチェックポイントされているとは限らないバックアップの形態です。たとえば、あるデータ・ファイルに SCN100 が含まれている一方、別のデータ・ファイルには SCN95 が含まれている状態になります。Oracle は、これらの SCN が一貫性を持つまで、つまり、オンライン REDO ログに記録されたすべての変更がデータ・ファイルに適用されるまで、データベースをオープンしません。

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必要がある場合は、データベース全体の非一貫性バックアップを実行する以外の選択肢はありません。たとえば、オープンしているデータベースのオフライン表領域のバックアップは、他の表領域と一貫性が保たれていません。これは、表領域のバックアップが行われている間にもデータベースの一部が変更されてディスクに書き込まれるためです。オンラインおよびオフライン・データ・ファイルのデータ・ファイル・ヘッダーには、一貫性のない SCN が含まれている場合があります。オープン・バックアップを作成するには、データベースを ARCHIVELOG モードで実行する必要があります。

データベースを ARCHIVELOG モードで実行している場合は、異なる時点で作成したデータ・ファイルのバックアップを使用して、全体データベース・バックアップを作成できます。たとえば、データベースに 7 つの表領域がある場合に、制御ファイルとそれぞれ 1 つの異なる表領域を毎晩バックアップしていくと、1 週間でデータベースの全表領域と制御ファイルのバックアップが作成されます。このようなバックアップを、全体データベース・バックアップと見なすことができます。

クローズ状態の非一貫性バックアップ システム・クラッシュの後、または SHUTDOWN ABORT の後でデータベースをバックアップする場合は、クローズ状態の非一貫性バックアップを作成することもできます。このタイプのバックアップは、データベースが ARCHIVELOG モードで実行されている場合に限り有効です。同モードで実行されている場合は、バックアップに一貫性を持たせる際、オンライン REDO ログとアーカイブ REDO ログの両方が使用できるためです。

注意： クローズ状態の非一貫性バックアップを NOARCHIVELOG モードで作成することはお薦めしません。

NOARCHIVELOG モードで実行しているデータベースのバックアップを作成する場合は、必ず IMMEDIATE または NORMAL オプションを使用して、正しくデータベースをクローズしてください。NOARCHIVELOG モードで運用中のデータベースに、一貫性のない全体データベース・バックアップを使用できるのは、バックアップ前の変更を含む REDO ログ

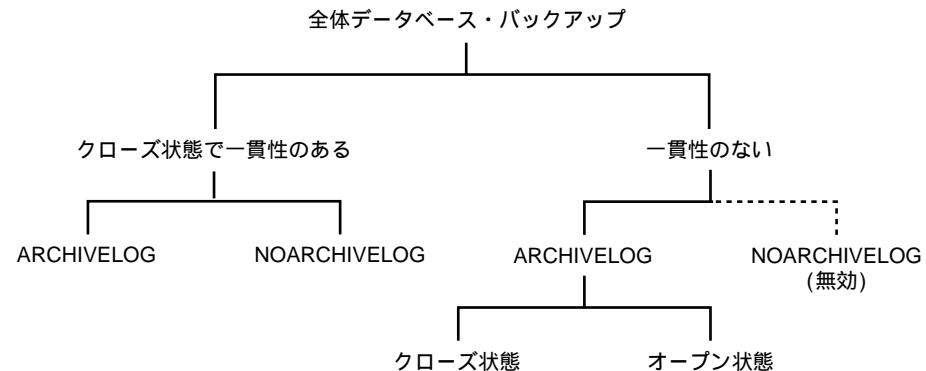
が、バックアップを復元して使用できる場合に限られます。ただしこのような状況はまず発生しません。

非一貫性バックアップ・ファイルのデータ・ファイル・ヘッダーには、それぞれ異なった SCN が含まれています（通常にシャットダウンすれば、これらの SCN の一貫性は保たれます）。また、データベースを NOARCHIVELOG モードで運用した場合は、失われた変更に適用するアーカイブ REDO ログが存在しません。これらが NOARCHIVELOG での非一貫性バックアップをお勧めしない理由です。そのため、NOARCHIVELOG モードで運用されていたデータベースが異常終了により停止した場合、そのバックアップは、回復に使用できないため、RMAN ではこのデータベースのバックアップをとることができません。

事実として念頭に置かなければならないのは、データベースを NOARCHIVELOG モードで実行している場合は、回復処理を一切行わなくても使用が可能なバックアップを作成すべきだということです。ただし、ログの REDO を適用してバックアップを回復する必要がある場合は、そのようにすることはできません。REDO を使用してデータベースの一貫性を確保する必要がある場合は、ARCHIVELOG モードで運用してください。

図 3-1 に、全体データベース・バックアップを行う場合の、さまざまな選択肢を示します。

図 3-1 全体データベース・バックアップのオプション



アーカイブされていない REDO ログ・ファイルのアーカイブ オープン状態のバックアップ、またはクローズ状態の非一貫性バックアップを実行した後は、アーカイブされていない REDO ログを必ずアーカイブして、これらのバックアップの回復に必要な REDO を確保してください。データベースがオープンしているときに、Oracle を現行のログから解放し、このログをアーカイブされていないその他すべてのログとともにアーカイブする場合は、次に示す SQL 文を発行します。

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

データベースがマウント、オープンまたはクローズのいずれかの状態にあるときに、現行以外のすべての REDO ログを Oracle にアーカイブさせる場合は、次の SQL 文を発行します。

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

データベースがマウント、オープンまたはクローズのいずれかの状態にあるときに、特定のログ・グループをアーカイブする場合は、次の SQL 文を発行します。*integer* には、グループの数を入力します。

```
SQL> ALTER SYSTEM ARCHIVE LOG GROUP integer;
```

上記の処理が済んだら、バックアップ開始後に生成されたアーカイブ REDO ログをすべてバックアップします。これにより、該当のバックアップが使用でき、さらに元のアーカイブ・ログのディスクからの削除も可能になります。バックアップ時に作成されたアーカイブ REDO ログがすべて確保されないと、バックアップの一貫性保持に必要な REDO レコードが確保されないため、バックアップの回復が実行できなくなります。

制御ファイルのバックアップ 一貫性のある全体データベース・バックアップを行う場合を除き、BACKUP CONTROLFILE オプションを指定して ALTER DATABASE コマンドを実行し、制御ファイルのバイナリ・バックアップを行ってください。バックアップの宛先は引用符で指定します。たとえば、次のように入力します。

```
SQL> ALTER DATABASE BACKUP CONTROLFILE TO '/oracle/copy/cf.f';
```

制御ファイルはトレース・ファイルにバックアップすることもできます。制御ファイルの再作成が必要な場合は、USER_DUMP_DEST パラメータで指定された位置にある、トレース・ファイルのスクリプトを使用します。次の構文を使用します。

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

トレース・バックアップには、1 つ大きな問題があることに注意してください。つまり、古い制御ファイルで作成された以前のバックアップの記録がここには残らないという問題です。

関連項目 : SQL 文を使用した制御ファイルのバックアップの詳細は、13-11 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。

表領域のバックアップ

表領域のバックアップとは、表領域を構成するデータ・ファイルのバックアップのことです。その理由は、データ・ファイルが物理ファイルであるのに対し（物理的なバックアップが可能なのは物理ファイルのみ）、表領域は論理的なグルーピングであるためです。たとえば、表領域 TBS_2 にデータ・ファイル 2、3、4 が含まれていたとすると、表領域 TBS_2 のバックアップにより、これら 3 つのデータ・ファイルがバックアップされます。

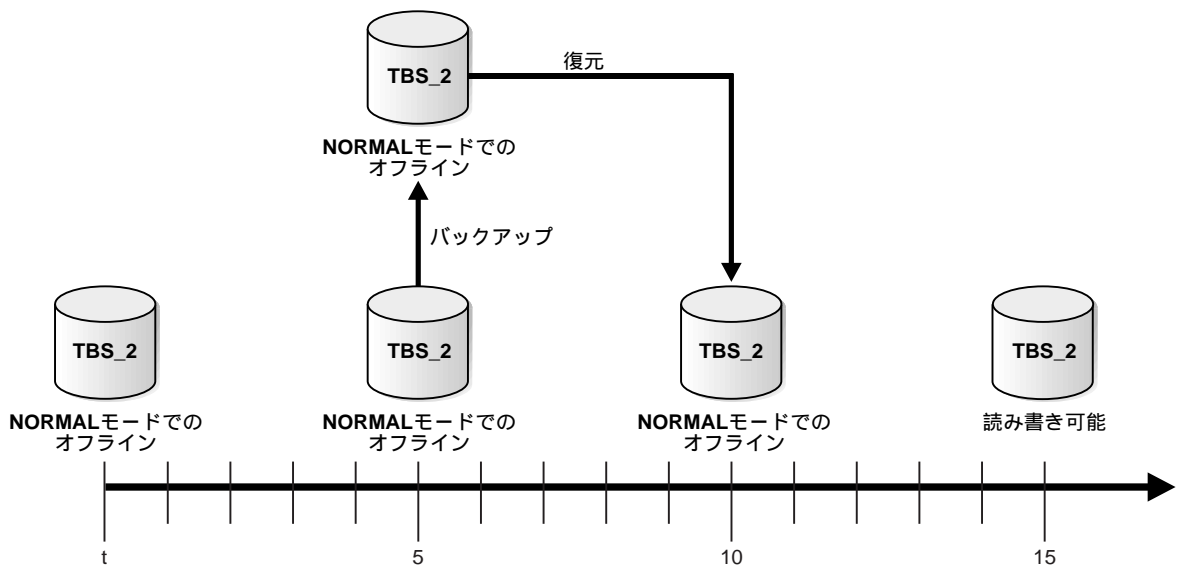
オンラインであるかオフラインであるかを問わず、表領域のバックアップが有効なのは、データベースが ARCHIVELOG モードで運用されている場合のみです。これは、復元された

表領域とデータベース内のその他の表領域とを、一貫性のある状態にするために REDO が必要となるからです。

NOARCHIVELOG モードで運用中のデータベースにおいて、表領域のバックアップが有効となるのは、その表領域が現在読取り専用または NORMAL モードでオフラインされた場合のみです。これらを回復するには REDO を必要としないため、これらは例外的なケースです。たとえば、[図 3-2](#) の実例を見てみましょう。

1. t 日のある時点で表領域 TBS_2 を NORMAL モードでオフラインにする。
2. $t + 5$ 日に TBS_2 のバックアップを作成する。
3. $t + 5$ 日に作成したバックアップを使用して、 $t + 10$ 日に表領域 TBS_2 を復元する。
4. $t + 15$ 日に表領域 TBS_2 を読み書き可能とする。

図 3-2 NOARCHIVELOG モードでの表領域のバックアップ



$t + 5$ から $t + 10$ の間は、このオフライン表領域に変更が加えられていないため、Oracle ではメディア回復を必要としません。ただし、 $t + 15$ で、この表領域を読取り書き込み両用とし、続いて $t + 5$ のバックアップの復元を試行した場合、Oracle では $t + 15$ 以降の変更について、メディア回復が必要となります。その結果、必要な REDO がすべてオンライン REDO ログにある場合にしかデータベースをオープンできなくなります。

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。データ・ファイルのバックアップは、表領域のバックアップほど一般的ではありませんが、ARCHIVELOG データベースでは有効です。

NOARCHIVELOG モードのデータベースにおいて、データ・ファイルのバックアップが有効となるのは、表領域内のデータ・ファイルがすべてバックアップされている場合のみです。データ・ファイルがすべてバックアップされていなければ、データベースを復元できません。さらに、データ・ファイルは、読取り専用か、または NORMAL モードでオフラインになっている必要があります。

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コピーのことです。データベースがマウントされていれば、次の SQL 文を発行できます。controlfile_location には、バックアップの名前を入力します。

```
ALTER DATABASE BACKUP CONTROLFILE TO 'controlfile_location';
```

トレース・ファイルにバックアップすることもできます。トレース・ファイルには、制御ファイルを再作成するためのスクリプトが含まれています。その文は次のようになります。

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

RMAN の backup current controlfile コマンドを使用することもできます。

関連項目 : SQL*Plus による制御ファイルのバックアップは、2-5 ページの「構造変更後の制御ファイルのバックアップ」を参照してください。RMAN による制御ファイルのバックアップは、8-6 ページの「制御ファイルのバックアップ」を参照してください。

バックアップ方法の選択

物理バックアップを作成するには、Recovery Manager または O/S コマンドを使用し、論理バックアップを作成するには、Export などのユーティリティを使用します。

表 3-1 バックアップ方法の種類と要件

バックアップ方法	タイプ	使用できるバージョン	要件
Recovery Manager (RMAN)	物理的	Oracle バージョン 8.0 以上	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	物理的	全バージョン	O/S のバックアップ・ユーティリティ (例 : UNIX dd)
Export	論理的	全バージョン	該当せず

表 3-1 バックアップ方法の種類と要件（続き）

バックアップ方法	タイプ	使用できるバージョン	要件
Enterprise Backup Utility (EBU)	物理的	Oracle7 のみ	メディア・マネージャ

Recovery Manager

Recovery Manager (RMAN) ユーティリティは、Oracle におけるバックアップおよび回復処理を管理するものです。RMAN は、制御ファイルに格納されたデータベースの情報を使用して、データベース・ファイルの位置の特定、バックアップおよび回復操作を自動的に行います。ここで言うデータベース・ファイルには、データ・ファイル、制御ファイル、アーカイブ REDO ログなどが含まれます。RMAN には次のような機能があります。

- データベース、表領域、データ・ファイルおよびアーカイブ REDO ログのバックアップおよび復元を実行する。
- 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関するすべてのアクションについて印刷可能なログを生成する。
- リカバリ・カタログまたはターゲット・データベースの制御ファイルを使用して、メディア復元操作と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。
- メディア・マネージャのデータをチェックし、バックアップされたファイルが引き続き使用可能であるか確認する。
- ユーザーが指定した適用に必要な REDO の量の限界に基づき、バックアップが必要なデータ・ファイルを検索する。
- Oracle Enterprise Manager の GUI インタフェースを使用する。
- Oracle Enterprise Manager を介して、バックアップのスケジュールを自動的に組む。

関連項目 : Recovery Manager の概説は、4-2 ページの「[Recovery Manager の概要](#)」を参照してください。

Recovery Manager のメタデータ

Recovery Manager は、バックアップおよび回復操作に必要な情報を、制御ファイルまたは [リカバリ・カタログ \(recovery catalog\)](#) と呼ばれるオプションのリポジトリから取得します。リカバリ・カタログは、RMAN がメンテナンスする情報の格納場所です。その情報はターゲット・データベースの制御ファイルから取得したものです。復元および回復操作の際、RMAN はリカバリ・カタログのデータを参照して、サーバー・プロセスに指示を出します。

関連項目 : Recovery Manager のメタデータの概説は、4-12 ページの「[Recovery Manager のメタデータ](#)」を参照してください。RMAN のメタデータの管理方法は、第 6 章の「[Recovery Manager のメタデータの管理](#)」を参照してください。

メディア管理

磁気テープ記憶装置を使用して Oracle データベースのバックアップを行う場合は、RMAN に加えて [メディア・マネージャ \(media manager\)](#) を使う必要があります。メディア・マネージャとは、サード・パーティのソフトウェア・ユーティリティで、テープ・ドライブなどの順次メディアのロード、ラベル付け、およびアンロードを行う機能があり、データのバックアップおよび回復に使用します。

Oracle の Backup Solutions Program (BSP) を使用するサード・パーティ・ベンダーは、自社の製品と Recovery Manager を容易に統合することができます。BSP のオンライン情報は、次の URL からアクセスできます。

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

メディア管理ソフトウェアを使用すると、ディスクとテープなどの 3 次記憶装置にバックアップするという柔軟性が加わります。ディスクにのみバックアップを行う場合には、メディア・マネージャは必要ありません。

関連項目 : メディア管理の概説は、4-15 ページの「[メディア管理](#)」を参照してください。RMAN で使用できるようにメディア・マネージャを設定する方法の詳細は 5-19 ページの「[メディア・マネージャの構成](#)」を参照してください。

Enterprise Manager

通常、RMAN はコマンドラインのユーティリティとして使用されますが、Oracle Enterprise Manager を使って RMAN バックアップを実行することもできます。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 によるバックアップおよび回復の詳細は、『Oracle Enterprise Manager 管理者ガイド』を参照してください。

オペレーティング・システム (O/S)

O/S のシステム固有のコマンドを使用して、O/S バックアップを実行できます。たとえば、UNIX のファイルをバックアップするには、`cp` コマンドを使用します。この場合、O/S バックアップを制御するには、UNIX のスクリプトを記述し、メンテナンスする必要があります。

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・システム固有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 固有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・エクスポートは、全体データベース・バックアップの代替となるものではありません。

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の詳細は、『Oracle8i ユーティリティ・ガイド』を参照してください。

Enterprise Backup Utility

Oracle の Enterprise Backup Utility (EBU) は、Oracle7 データベースのバックアップを自動化するユーティリティです。Oracle8i データベースとの互換性はありません。

各バックアップ方法の機能の比較

表 3-2 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 3-2 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 コマンドは使用しない。	BEGIN/END BACKUP コマンドを使用する。	一貫性バックアップを生成するには、大量の RBS が必要となる場合がある。
増分バックアップ	サポートされている。	サポートされていない。	サポートされていない。
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。

表 3-2 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
バックアップの自動化	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）の名前と位置を自動的に識別します。	サポートされていない。バックアップ対象のファイルは、手動で指定する必要があります。	サポートされている。全体、ユーザー単位または表単位でバックアップを実行します。
バックアップ・カタログ	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のみ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。
順次メディアへのバックアップ	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。RMAN は、プロキシ・コピーもサポートしています。これは、メディア・マネージャによるデータ転送を可能にする機能です。	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うか、またはメディア・マネージャが管理します。	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スクリプト	サポートされている。	サポートされていない。	サポートされている。

バックアップ形式の選択

バックアップの形式は、バックアップの作成方法によって変わってきます。ファイルは次に示す形式でバックアップできます。

- [バックアップ・セット](#)
- [イメージ・コピー](#)
- [オペレーティング・システムによるバックアップ](#)
- [論理バックアップ](#)

バックアップ・セット

RMAN の **backup** コマンドを発行する場合（かつ **proxy** オプションを指定しない場合）は、バックアップ・セットを作成します。バックアップ・セットとは、1 つ以上の物理的な**バックアップ・ピース (backup piece)** で構成される論理構造のことです。バックアップ・セットは、1 つのバックアップ・ピースで構成されるのが普通です。バックアップ・セットでは次のようなことができます。

- アーカイブ REDO ログ、またはデータ・ファイルを含めることができます。ただし両方同時に含めることはできません。
- ディスクまたはテープなどの 3 次記憶装置に書き込むことができます。
- 全体バックアップを構成することも、増分バックアップを構成することも可能である。
- 複数の O/S ファイルにまたがることができます。

バックアップ・セットは、Oracle 独自の形式で書かれます。したがって、Oracle インスタンスがバックアップ・セットのファイルを使用できるようになるには、RMAN がこれらをインスタンスが使用可能な形式に復元する必要があります。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。バックアップ・セット内のデータ・ファイルを復元するには、RMAN のコマンドを使用する必要があります。

バックアップするデータ・ファイルを指定すると、Oracle サーバー・セッションはそのファイルを読み取り、バックアップ・セットを作成します。RMAN バックアップの前に、ALTER TABLESPACE BEGIN BACKUP 文を実行する必要はありません（RMAN バックアップにおけるブロック読取りの詳細は、4-54 ページの「**論理ブロックの破損の検出**」を参照してください）。

RMAN は、データ・ファイル、制御ファイルまたはアーカイブ REDO ログをバックアップ・セットに含めることができます。全体データベース・バックアップを実行すると、RMAN はそのデータベースの全ファイルをバックアップし、さらに制御ファイルもバックアップします。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、RMAN に指示することもできます。

注意： アーカイブ REDO 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。

関連項目： Recovery Manager バックアップの概要は、4-28 ページの「**バックアップ・セット**」を参照してください。RMAN バックアップの作成方法の詳細は、第 8 章の「**Recovery Manager でのバックアップおよびコピーの作成**」を参照してください。RMAN の **backup** コマンドの詳細は、11-21 ページの「**backup**」を参照してください。

イメージ・コピー

次のオブジェクトの**イメージ・コピー**（**image copy**）を作成するには、RMAN の **copy** コマンドを使用します。

- データ・ファイル
- 制御ファイル
- アーカイブ REDO ログ

このコマンドを発行すると、（O/S ルーチンではなく）Oracle サーバー・セッションがデータ・ファイルを読み取り、そのコピーをディスクに書き込みます。RMAN バックアップの前に、ALTER TABLESPACE BEGIN BACKUP 文を実行する必要はありません。

イメージ・コピーは Oracle インスタンスでただちに使用できます。つまりこれはすでにインスタンスで使用可能な形式になっています。イメージ・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。

注意： プロキシ・コピーは、テープにとることができます。プロキシ・コピーはイメージ・コピーではありませんが、**backup** コマンドで **proxy** オプションを指定した場合に生成され、かつメディアによって管理されるバックアップの一種です。

関連項目：RMAN のイメージ・コピーの概要は、4-45 ページの「**イメージ・コピー**」を参照してください。RMAN のイメージ・コピーの作成方法の詳細は、8-10 ページの「**イメージ・コピーの作成**」を参照してください。RMAN の **copy** コマンドと **backup** コマンドの詳細は、11-48 ページの「**copy**」および 11-21 ページの「**backup**」を参照してください。

オペレーティング・システムによるバックアップ

オペレーティング・システム（O/S）のバックアップを作成するには、UNIX の **dd** など、オペレーティング・システムのコマンドを使用します。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あれば、どの形式を使用しても、ディスクまたはテープへの書込みが可能です。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。

関連項目：オペレーティング・システムのバックアップの作成方法の詳細は、第 13 章の「**オペレーティング・システム環境でのバックアップの実行**」を参照してください。

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 独自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実行できます。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ありません。

関連項目 : Export ユーティリティの詳細は、『Oracle8i ユーティリティ・ガイド』を参照してください。

バックアップ計画の立案

Oracle データベースを作成する前に、メディア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定しておいてください。バックアップ計画を立案せずにデータベースを作成すると、ディスク障害により、データ・ファイル、オンライン REDO ログ・ファイルまたは制御ファイルが破損した場合、回復を実行できなくなることがあります。

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定するための、一般的なガイドラインを示します。計画の詳細は操作環境の制約によって異なります。ただし、どのようなバックアップ計画を実行に移した場合でも、できるだけ次のガイドラインに従うようにしてください。

- [ARCHIVELOG または NOARCHIVELOG の運用モードの決定](#)
- [多重化された制御ファイル、オンライン REDO ログおよびアーカイブ REDO ログ](#)
- [頻繁で定期的なバックアップの実行](#)
- [構造変更時のバックアップの実行](#)
- [使用回数の多い表領域の頻繁なバックアップ](#)
- [回復不能な操作の後のバックアップ](#)
- [古いバックアップの保管](#)
- [分散データベースにおけるバックアップの制約](#)
- [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](#)
- [オンライン REDO ログのバックアップ](#)

ARCHIVELOG または NOARCHIVELOG の運用モードの決定

Oracle データベースを作成するにあたっては、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定しておいてください。次の質問について考慮してください。

- **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場合は、データベースは ARCHIVELOG モードで運用し、多重オンライン REDO ログ、多重制御ファイルおよび多重アーカイブ REDO ログを使用するのが理想的です。ある程度の量であればデー

タが失なわれてもよい場合は、NOARCHIVELOG モードで運用すれば余分なメンテナンス作業を削減することができます。

- **現行ではなく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？**データベースへの誤った変更を修正するために、不完全回復を実行する場合は、ARCHIVELOG モードで運用し、構造上の変更を行うときには制御ファイルのバックアップを実行します。任意の時点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、不完全回復を補完します。
- **データベースを常時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、全体データベース・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。

質問への回答が済み、使用するモードが決定したら、それぞれのガイドラインに従ってください。

- **NOARCHIVELOG モードでのバックアップ**
- **ARCHIVELOG モードでのバックアップ**

NOARCHIVELOG モードでのバックアップ

データベースを NOARCHIVELOG モードで運用している場合、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のみです。次のガイドラインに従ってください。

- 失ってもよい作業量に応じて、全体データベース・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、一貫性のある全体データベース・バックアップをとります。1 日分の作業の消失しか許されない場合は、一貫性のある全体データベース・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。このような場合は、ARCHIVELOG モードでデータベースを運用します。
- 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全体データベース・バックアップをとってください。これにより、全体データベース・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

ARCHIVELOG モードでデータベースを使用している場合は、ARCn によって、オンライン REDO ログ・ファイルが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、ディスク障害から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)までを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。

- データベースを作成した後、全体データベース・バックアップを実行する。この最初の全体データベース・バックアップには、関連する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： 全体データベース・バックアップを初めて実行するときは、まずデータベースが ARCHIVELOG モードになっているかどうかを確認してください。ARCHIVELOG モードでないと、バックアップの制御ファイルに、NOARCHIVELOG モードの設定が含まれてしまいます。

- 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し、データベースのバックアップを最新の状態に保つ。その後は、全体データベース・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ておく必要がある場合は、データベースがクローズした状態を行うことはできません。
- オープン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実行し、データベースの情報のバックアップを更新する。これにより、最初にとった全体データベース・バックアップを補完できます。特に、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り、データベースの回復にかかる時間を短縮す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 データ (または増分バックアップ) が少なくて済みます。

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要件に依存します。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あることが必要です。復元されたデータ・ファイルには、適切な REDO ログ・ファイルを適用して、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る。データベースを ARCHIVELOG モードで運用しており、かつそのデータベースがオープン状態になっている場合は、RMAN を使用するか、または BACKUP CONTROLFILE オプションを指定した上で、ALTER DATABASE 文を使用してください。

多重化された制御ファイル、オンライン REDO ログおよびアーカイブ REDO ログ

制御ファイル、オンライン REDO ログおよびアーカイブ REDO ログは、バックアップおよび回復を実行する際に重要となるファイルです。これらのうちのどのファイルが消失しても、回復不能なデータ損失の原因となることがあります。次に示すものをメンテナンスしておいてください。

- 異なるディスク・コントローラにマウントされた、異なるディスクに、制御ファイルのコピーを最低 2 つ保管しておく。Oracle でこれらのコピーを多重化することも、使用中の O/S でこれらをミラー化することもできます。
- オンライン REDO ログの複数のコピーを、異なるディスクに保管する。オンライン REDO データは、インスタンス、クラッシュおよびメディア回復を実行する場合に重要です。
- アーカイブ REDO ログの複数のコピーを、異なるディスクに保管する。可能であれば、異なるメディアに保管する。

関連項目：データベース構造の管理も考慮したバックアップおよび回復計画の詳細は、[第 2 章の「データ構造の管理」](#)を参照してください。Oracle の全データ構造の概要は、『Oracle8i 概要』を参照してください。

頻繁で定期的なバックアップの実行

回復計画においては、頻繁なバックアップが重要です。バックアップの頻度は、データベースの次のような変更の割合または頻度に基づいて決定します。

- 表の追加および削除
- 既存の表における行の追加および削除
- 表におけるデータの更新

ユーザーが多量の DML を生成した場合は、それに応じてデータベースのバックアップの回数を増やします。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。

バックアップ・スクリプトを作成するには、Recovery Manager か O/S コマンドを使用します。RMAN スクリプトはリカバリ・カタログに格納されます。日常的かつ定期的にバックアップ操作を実行する場合に特に有効な方法です。

関連項目：RMAN に格納するスクリプトの概要は、4-10 ページの「[ストアド・スクリプト](#)」を参照してください。格納スクリプトの作成、削除、置換および表示の詳細は、6-21 ページの「[リカバリ・カタログへのスクリプトの格納](#)」を参照してください。

構造変更時のバックアップの実行

データベースの変更は、管理者もユーザーも行います。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
- 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、改名または削除

データベース内でバックアップの必要な部分は、選択したアーカイブ・モードによって異なります。

モード	アクション
ARCHIVELOG	構造変更の前後に、制御ファイルのバックアップを作成する (BACKUP CONTROLFILE オプションを指定した上で ALTER DATABASE コマンドを使用する)。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。
NOARCHIVELOG	修正の直前および直後に、一貫性のある全体データベース・バックアップを作成する。

使用回数の多い表領域の頻繁なバックアップ

バックアップ計画を強固なものにするためには、全体データベース・バックアップを定期的に行うだけでは十分ではないと、多くの管理者は考えています。ARCHIVELOG モードで運用している場合は、ある表領域の全データ・ファイルをバックアップすることも、1つのデータ・ファイルのみをバックアップすることもできます。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。

たとえば、全体データベース・バックアップを週に1回、日曜日に行うとします。データベースの更新量が多い週の金曜日にメディア障害が発生したとすると、回復時に多大な量の REDO データを適用する必要があります。アクセス頻度の高い表領域は、週に3回バックアップをとることになると、復元したファイルを障害発生時の状態にロールフォワードする際、データに適用する変更の数は少なくて済みます。

回復不能な操作の後のバックアップ

UNRECOVERABLE オプションを使用し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 として作成した場合、REDO データのログがとられないため、既存のバックアップからこれらのオブジェクトを回復することができなくなります。

注意： RMAN 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 オプションの詳細は、『Oracle8i SQL リファレンス』の CREATE TABLE ... AS SELECT コマンドおよび CREATE INDEX コマンドを参照してください。

RESETLOGS オプション使用後のデータベース全体のバックアップ

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、障害が発生した場合は、データベースをオープンした後に行った作業はすべて失われます。

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

- データベースの新規インカーネーションを作成し、新しい RESETLOGS SCN をデータベースの全ファイルのヘッダーに記入する。
- ログ順序番号を 1 にリセットする。
- オンライン REDO ログ・ファイルがあれば再フォーマットし、なければ新規作成する。

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

オンライン REDO ログをリセットすると、バックアップおよび回復計画に、重大な結果をもたらします。RESETLOGS 前のバックアップは復元できません。したがって、新規インカーネーションでは古いバックアップは無価値です。ただし次のような例外もあります。

- 読取り専用表領域のバックアップで、RESETLOGS 前に読み書き可能に再変更されなかったもの。
- NORMAL モードでオフラインされた表領域のバックアップで、RESETLOGS 前にオンラインに再変更されなかったもの。
- 読み書き可能な表領域のバックアップで、不完全回復の実行後、RESETLOGS 直前までに作成されたもの。

関連項目： 詳細は、14-34 ページの「[RESETLOGS 以前のバックアップからの回復](#)」を参照してください。

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。これを行わないと、作成したバックアップが一貫性のないものとなり、回復を実行しないとデータベースをオープンできなくなります。回復を実行するための REDO データが使用できないため、Oracle はデータベースをオープンしません。

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）全体データベース・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るまでの時間
- 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、データベースを稼働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を行う必要があります。オープン状態のデータベースのバックアップを実行する場合は危険も伴います。つまり、このバックアップが完了しないうちに、別のメディア障害が発生すると、RESETLOGS オプションを指定したデータベースのオープン以降に行われた変更がすべて失われます。RESETLOGS オプションを指定してデータベースをオープンするより前に作成されたバックアップは、データベースのインカーネーションの回復に使用できません。

注意： この規則には、1 つだけ例外があります。14-34 ページの「RESETLOGS 以前のバックアップからの回復」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。可能な限り、一貫性のある全体データベース・バックアップをとるようにしてください。

古いバックアップの保管

古いバックアップを保管する基本的な理由は次の 2 つです。

- 最新のバックアップより以前のある時点の状態にする不完全回復を実行する場合には、古いバックアップが必要となる。
- 最新のバックアップが破損している。

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。たとえば、2 月の 1 日と 14 日にバックアップを作成し、同月の終わりの時点でデータベースを 2 月 7 日の状態に回復することにした場合は、2 月 1 日のバックアップを使用する必要があります。

NOARCHIVELOG モードで運用されているデータベースの場合、使用するバックアップは一貫性のある全体データベース・バックアップであることが必要です。このバックアップを使用してメディア回復を実行することはできません。ARCHIVELOG モードで運用されているデータベースの場合は、全体データベース・バックアップは次のようになります。

- 回復に REDO が使用できるため、一貫性を持たせる必要がない。
- 指定した回復時点よりも前に作成する必要がある（制御ファイルに、その回復で指定された時点のデータベースの構造が反映されている必要があります）。
- データ・ファイルを指定した時点の状態に回復するのに必要な、アーカイブ・ログをすべて含める必要がある。

保護を強化するため、現行バックアップの直前のバックアップ（および関連するアーカイブ REDO ログ）を複数保存してください。これにより、最新のバックアップが使用できない場合でも、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 オプションを指定してデータベースをオープンしていた場合は、ログがリセットされた時点よりも後の状態にする回復に、既存のバックアップを使用することはできません。したがって、データベースを停止し、一貫性のある全体データベース・バックアップをとることが必要となります。このようにすれば、RESETLOGS オプションを使用した後のデータベース変更についても、回復できます。

分散データベースにおけるバックアップの制約

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、その分散データベース・システムを構成するデータベースは、すべて同じアーカイブ・モードで運用する必要があります。次に示す結果と制約に注意してください。

モード	制約	結果
ARCHIVELOG	正しくクローズ	各ノードのバックアップは自律的に、すなわち同期をとらず個別に実行できる。
NOARCHIVELOG	正しくクローズ	グローバルな分散データベース回復を計画している場合は、一貫性のある全体データベース・バックアップをグローバルな同一時刻に実行する必要があります。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。

関連項目：分散システムでのメディア回復の詳細は、14-32 ページの「[メディア回復後のデータベースのオープン](#)」を参照してください。

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用すると、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してください。この方法は、RMAN を使用して、リカバリ・カタログのバックアップを行う場合に便利です。

データベースのエクスポートは、全体データベース・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ないことに注意してください。

関連項目 : Export ユーティリティの詳細は、『Oracle8i ユーティリティ・ガイド』を参照してください。

オンライン REDO ログのバックアップ

データ・ファイルおよび制御ファイルをバックアップするのには必要ですが、オンライン REDO ログをバックアップするのは危険です。次の理由から、オンライン・ログのバックアップはとらないでください。

- メディア障害からオンライン・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。オンライン・ログの多重化とは、グループに複数のログ・メンバーを割り当て、それぞれを異なるディスクおよび異なるディスク・コントローラで持つことです。
- ARCHIVELOG モードでデータベースを使用している場合は、ARCn がすでに REDO ログをアーカイブしています。
- 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、一貫性のある全体データベース・バックアップしかありません。このタイプのバックアップのファイルは一貫性があり、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。

オンライン REDO ログのバックアップにおける危険とは、意図せずに、誤って復元してしまう可能性があることです。オンライン・ログの復元が原因となって、回復プロセスで重大な問題が発生する場合があります。バックアップされたオンライン・ログの復元による、回復への重大な悪い影響を次の 2 つの例で示します。

使用例 1: オンライン REDO ログの意図しない復元

障害が発生すると、単純なミスを犯しがちです。データベースの復元中、DBA およびシステム管理者は、さまざまな危険にさらされます。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。これにより、DBA は、完全回復を実行しようとしているにもかかわらず、不完全回復を実行することになり、上書きされた REDO ログに含まれていた大切なトランザクションを回復できなくなります。

使用例 2: 複数のパラレル REDO ログの時間の流れの作成

単一インスタンス・データベースに対して、REDO ログの時間の流れを誤って複数作成してしまうことがあります。ただし、オンライン・ログが復元できないように作成すれば、このミスは防ぐことができます。データベースは RESETLOGS オプションを指定してオープンする必要があります。これにより、新しい REDO ログの他、データベースの新規インカーネーションも効果的に作成されます。

一貫性バックアップからデータベースを復元し、いかなる回復も実行しないことが、最善の方法であるという状況が発生した場合、オンライン・ログを復元し、RESETLOGS オプションを指定したデータベースのオープンを避けるのが安全であると考えられるでしょう。しかし、そのようにした場合、前回の時間線で生成していたログ順序番号を、データベースが再び生成してしまいます。

そのような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は、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、ログをリセットしていた場合、データベースの新規インカーネーションを作成します。新規インカーネーションが作成したアーカイブ REDO ログは、新規インカーネーションにしか適用されません。

注意： Recovery Manager および EBU は、オンライン REDO ログのバックアップを行いません。

回復計画の立案

Oracle は、回復処理に役立つプロシージャおよびツールを多数用意しています。効果的な回復計画を立案するため、次の事柄を行うようにしてください。

- [バックアップおよび回復計画のテスト](#)
- [非メディア障害への対処](#)
- [メディア障害からの回復](#)

バックアップおよび回復計画のテスト

本番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実行してください。これにより、計画が完全なものであるかどうかを判断することができ、実際の環境における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。

Recovery Manager を使用している場合は、本番データベースのバックアップを使用して **duplicate** コマンドでテスト・データベースを作成できます。データベースの複製方法の詳細

細は、[第 10 章の「Recovery Manager による複製データベースの作成」](#)を参照してください。

非メディア障害への対処

回復計画を立案する場合は、メディア回復が主な関心事となりますが、非メディア障害の基本的なタイプと、それぞれの原因および解決策も理解しておく必要があります。

- 文障害
- ユーザー・プロセス障害
- ユーザー・エラー
- インスタンス障害
- メディア障害

文障害

文障害とは、Oracle プログラムで文を処理する際に発生する論理障害のことです。通常、文障害が発生すると、Oracle または O/S はエラー・コードとメッセージを戻します。[表 3-3](#)に、文障害の代表的な原因と解決策を示します。

表 3-3 文障害の典型的な原因と解決策

問題	解決策
アプリケーションで論理エラーが発生した。	論理フローが正常となるよう、エラーを生成したプログラムを修正する。このタイプの問題を解決する場合は、プログラム開発者の助力が必要になることがあります。
ユーザーが不正なデータを表に入力しようとした。	不正な SQL 文を修正し、再発行する。
不十分な権限でユーザーが操作を行おうとした。たとえば、SELECT 権限のみで表にデータを挿入しようとした。	該当する文の実行に必要なデータベース権限をユーザーに与える。
ユーザーが割当て制限を超えて表を作成しようとした。	ALTER USER コマンドを発行して割当て制限を変更する。
ユーザーが表に対して INSERT または UPDATE を試行した結果、表領域の空き領域が十分でないにもかかわらず領域が割り当てられた。	表領域に領域を追加する。データ・ファイルに対しては RESIZE および AUTOEXTEND を使用することもできます。

ユーザー・プロセス障害

ユーザー・プロセス障害とは、Oracle データベースにアクセスするユーザー・プログラムで発生したあらゆる障害のことです。ユーザー・プロセス障害の原因は多様です。代表的な使用例は次のとおりです。

- セッション時、ユーザーが異常切断を行った。
- ユーザー・セッションが異常終了した。たとえば、クライアント / サーバー構成において、データベースに接続中であるにもかかわらずユーザーがクライアントを再起動したなどの場合です。
- ユーザーのプログラムでアドレス例外が呼び出され、セッションが終了した。

多くの場合、ユーザー・プロセス障害の解決に、なんらかの処置をとる必要はありません。ユーザー・プロセスが単に機能しなくなったのみであれば、Oracle や他のユーザー・プロセスが影響を受けることはありません。ユーザー・プロセスが異常終了しても、通常は PMON バックグラウンド・プロセスによるクリーン・アップで対処できます。

ユーザー・エラー

ユーザー・エラーとは、ユーザーがデータをデータベースに追加または削除するときに発生するあらゆるミスのことです。ユーザー・エラーの代表的な原因は次のとおりです。

- ユーザーが誤って表の全体または一部を削除した。
- ユーザーが表内の行をすべて削除した。
- ユーザーがデータをコミットした後からエラーに気付いた。

データが失われた表の論理バックアップでも、場合によっては単にそのデータを再び表にインポートできることがあります。ただし、場合によっては、ある種の不完全メディア回復を行わなければ、そのようなエラーを修復できない場合もあります。

データベースの Point-in-Time 回復 (DBPITR) と表領域の Point-in-Time 回復 (TSPITR) のいずれかを使用することができます。これらの不完全回復の違いを次の表に示します。

タイプ	説明	手順
DBPITR	<div>1. バックアップされたデータベースを復元する。</div> <div>2. エラー発生直前の状態にロールフォワードする。</div> <div>3. RESETLOGS を指定してオープンする。</div>	<div>RMAN 回復は、9-23 ページの「不完全回復の実行」を参照。</div> <div>O/S 回復は、14-25 ページの「不完全メディア回復の実行」を参照。</div>

タイプ	説明	手順
TSPITR	<ol style="list-style-type: none"> 1. 補助インスタンスを作成する。 2. 補助インスタンスの表領域をエラー発生直前の状態に回復する。 3. データをプライマリ・データベースに再びインポートする 	<p>RMAN TSPITR は付録 A の「Recovery Manager による表領域の Point-in-Time 回復の実行」を参照。</p> <p>O/S TSPITR は、付録 B の「オペレーティング・システム環境での表領域の Point-in-Time 回復の実行」を参照。</p>

インスタンス障害

インスタンス障害は、インスタンスが異常終了すると発生します。次のような場合、インスタンス障害が発生することがあります。

- 停電によりサーバーがクラッシュした。
- ハードウェアの問題によりサーバーが使用不能になった。
- O/S がクラッシュした。
- Oracle バックグラウンド・プロセスの 1 つに障害が発生した。
- SHUTDOWN ABORT 文を発行した。

インスタンス障害が発生した場合、Oracle は自動的にインスタンス回復を実行します。必要なことはデータベースの起動のみです。Oracle は、データベースが正しく停止されていないことを自動的に検知し、REDO ログ内の REDO レコードを、コミット済みかどうかにかかわらずデータ・ファイルに適用します。そして、コミットされていないデータをロールバックします。Oracle は、さらにデータ・ファイルと制御ファイルの同期をとり、データベースをオープンします。

メディア障害からの回復

メディア障害は、データに対する最も重大な問題です。メディア障害とは、データベースの運用に必要なファイルへの書込み、または同ファイルの読取りを、コンピュータが試行したときに発生する物理的な問題です。メディアの問題のうち、代表的なものを挙げると次のようになります。

- いずれかのデータベース・ファイルを保持しているディスク・ドライブで、ヘッド・クラッシュが発生した。
- データ・ファイル、オンライン REDO ログ、アーカイブ REDO ログまたは制御ファイルのいずれかが破損したか、誤って削除または上書きされた。

メディア障害からの回復を実行する際に使用する方法は、発生したメディア障害のタイプによって大きく異なります。たとえば、破損したデータ・ファイルの回復を行うときの方法と、制御ファイルの消失の回復を行うときの方法は異なります。

メディア回復の基本的な手順は次のとおりです。

- 回復するファイルを選択する。
- 必要なメディア回復のタイプを判断する。すなわち、完全か不完全か、オープン・データベースかクローズ・データベースかを決定します。
- 次のような必要なファイルのバックアップまたはコピーを復元する。データ・ファイルの回復には、データ・ファイル、制御ファイルおよびアーカイブ REDO ログが必要です。

注意： バックアップを作成していなくても、必要となる REDO ログがあり、かつ制御ファイルに破損したファイルの名前が含まれていれば、回復を実行できます。ファイルを元の場所に復元できない場合は、新しい場所を選択し、制御ファイルにその場所を示す必要があります。

- REDO レコードを適用し（Recovery Manager を使用している場合は、これに加えて増分バックアップも適用するか、または増分バックアップのみを適用する）、データ・ファイルを回復する。
- データベースを再度オープンする。不完全回復を実行している場合は、RESETLOGS モードでデータベースをオープンする必要があります。

関連項目： RMAN によるメディア回復の方法の詳細は、[第 9 章の「Recovery Manager による復元と回復」](#)を参照してください。O/S コマンドによるメディア回復の方法の詳細は、[第 14 章の「オペレーティング・システム環境での回復の実行」](#)を参照してください。

回復するファイルの判別

まず最初に、何を回復するのかを判断する必要があります。いくつかのタイプの障害では明白です。たとえば、ハードウェアのクラッシュであれば、データベース全体を回復する必要があります。ある 1 つのデータ・ファイルが破損した場合があります。そのような場合は、表 VSRECOVER_FILE を使用すると、回復が必要なものを判断できることがあります。

回復のタイプの選択

メディア回復を実行する場合は、[完全回復（complete recovery）](#)と[不完全回復（incomplete recovery）](#)のいずれかを選択します。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・ファイルに対しても実行できます。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は、Point-in-Time (PITR) 回復とも呼ばれます。この方式では、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を行った場合は、以後これを継続して実行し、完全メディア回復となるようにするか、あるいは、RESETLOGS 操作でデータベースの新規のインカーネーションを作成し、不完全メディア回復を終了する必要があります。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。次の中から選択できます。

- **時間ベースの回復 (time-based recovery)** 指定された時点の状態にデータを回復する。
- **取消しベースの回復 (cancel-based recovery)** CANCEL コマンドが発行される時点まで回復する。
- **変更ベースの回復 (change-based recovery)** 指定された SCN の状態に回復する。
- **ログ順序回復 (log sequence recovery)** 指定されたログ順序番号の状態まで回復する。

重要かつ特殊なタイプのメディア回復の一つに、表領域の Point-in-Time 回復 (TSPITR) があります。TSPITR を使用すると、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回復させることができます。

選択すべき回復のタイプは状況によって異なります。表 3-4 に、代表的な使用例と方針を示します。

表 3-4 メディア障害と回復方針の代表例

消失したファイル	アーカイブ・モード	状態	方法
1 つ以上のデータ・ファイル	NOARCHIVELOG	クローズ状態	一貫性のあるデータベースのバックアップから、データベース全体を復元する。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変更内容はすべて失われます。 注意: NOARCHIVELOG モードで運用中の場合、データベースを完全に回復できるのは、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ていない時のみです。
1 つのオンライン REDO ログおよび 1 つ以上のデータ・ファイル	NOARCHIVELOG	クローズ状態	一貫性のあるバックアップから、データベース全体を復元する。最後のバックアップ以降の変更はすべて失われます。
1 つ以上のデータ・ファイルおよびすべての制御ファイル	NOARCHIVELOG	クローズ状態	一貫性のあるバックアップから、データベース全体と制御ファイルを復元する。最後のバックアップ以降の変更はすべて失われます。
1 つ以上のデータ・ファイル	ARCHIVELOG	オープン状態	データベースをオープンしたまま、表領域回復またはデータ・ファイル回復を実行する。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 1 つ以上のデータ・ファイルおよび 1 つのオンライン REDO ログ	ARCHIVELOG	クローズ状態	データベースの不完全回復を実行し、オンライン REDO ログが失われた時点の状態にする。
回復に必要な 1 つのアーカイブ REDO ログおよび 1 つ以上のデータ・ファイル	ARCHIVELOG	オープン状態	失われたデータ・ファイルが含まれた表領域に対し、TSPITR を実行し、利用できる最新の REDO ログの時点の状態にする。
1 つ以上のデータ・ファイルとすべての制御ファイルの両方、またはすべての制御ファイル	ARCHIVELOG	オープンされていない状態	失われたファイルをバックアップから復元し、データ・ファイルを回復する。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 1 つ以上のデータ・ファイルとすべての制御ファイルの両方、またはすべての制御ファイル。同じく回復に必要なアーカイブ REDO ログまたはオンライン REDO ログ	ARCHIVELOG	オープンされていない状態	データベースの不完全回復を実行する。失われたログおよびそれ以降のすべてのログに含まれていた変更は、すべて失われます。

データ・ファイルおよび必要なアーカイブ REDO ログのバックアップの復元

バックアップ復元の方法は、データのバックアップに RMAN を使用するか、または O/S コマンドを使用するかによって異なります。RMAN を使用する場合は、**restore** コマンドを発行し、RMAN にデータの転送を実行させます。O/S コマンドを使用する場合は、復元するのはどのファイルであるかを確認した上で、バックアップを手動で必要な場所にコピーする必要があります。

メディア障害が発生したために、またはなんらかの理由で、停止中のデータベースを現行以外の時点の状態に復元する必要があるとします。バックアップを使用してデータ・ファイルの 1 つを復元するとき、Oracle は、データ・ファイル・ヘッダー内のチェックポイント SCN と、制御ファイルに記録されたデータ・ファイル・ヘッダー・チェックポイント SCN の間の非一貫性を、データベースのオープン時に検知します。その際、Oracle は、制御ファイルおよびデータ・ファイル・ヘッダーに記録されたチェックポイント SCN のうち、最も小さいものを選択した上で、指定したログ順序番号からメディア回復を開始するかどうかを確認するメッセージを表示します。メディア回復を実行しなければならないオンライン・データ・ファイルがある場合は、データベースをオープンすることはできません。

メディア回復が実際には必要ないにもかかわらず、その必要があるというメッセージを Oracle が表示するケースは一つしかありません。ALTER TABLESPACE ...BEGIN BACKUP を発行したことにより、**ホット・バックアップ・モード (hot backup mode)** になっている表領域がある場合に、システムがクラッシュすると、Oracle はメディア回復が必要というメッセージを次の起動時に発行します。ただし、バックアップを復元していないため、ここでは実際にはメディア回復は必要ありません。この場合は、ALTER DATAFILE ... END BACKUP コマンドを発行して、メディア回復を行わないでください。RMAN バックアップではこのような問題は発生しないことに注目してください。

関連項目 : RMAN によるデータベースの復元方法の詳細は、9-2 ページの「**データ・ファイル、制御ファイルおよびアーカイブ REDO ログの復元**」を参照してください。O/S コマンドによるデータ・ファイル復元の方法の詳細は、14-5 ページの「**ファイルの復元**」を参照してください。

メディア回復の開始

物理ファイル回復の基本的な方法は 2 つあります。次のことが可能です。

- RMAN を使用して回復を自動化する。
- SQL 文または SQL*Plus 文を実行する。

回復方法は、どのバックアップ方式を選択するかにより変わります。たとえば、どのコマンドを使用してバックアップを行ったかによります。

- RMAN の **backup** コマンドを使用した場合、バックアップは RMAN 特有の形式で書き込まれるため、RMAN の **restore** コマンドおよび **recover** コマンドで回復を実行する必要があります。

- RMAN の **copy** コマンドを使用した場合は、RMAN および O/S コマンドのいずれでも回復を実行できる。
- O/S コマンドを使用した場合は、O/S ユーティリティでファイルを復元し、RECOVER 文 (SQL*Plus) または ALTER DATABASE RECOVER 文 (SQL) でファイルを回復する必要がある。

注意： 唯一の例外は、O/S バックアップがディスクに保管されたイメージ・コピーである場合です。RMAN を使用して、これらのイメージ・コピーをデータ・ファイル・コピーとして登録していた場合は、RMAN による復元が可能です。

RMAN による回復 RMAN は、バックアップおよび回復操作を支援する、強力なツールです。RMAN を使用して回復を実行すると、次のことが可能になります。

- ファイル・コピーと RMAN 特有の形式のバックアップ・セットの両方を復元および回復できる。
- リカバリ・カタログを使用することによって管理エラーを最小限に抑えることができる。
- RMAN の増分バックアップ機能を使用して、回復にかかる時間を最小限に抑えることができる。
- バックアップ、データ・ファイル・コピー、回復不能なファイルなどに関する、詳細なレポートを生成できる。
- ファイル・システムまたはリカバリ・カタログに格納されたスクリプトを使用して、ジョブを自動化できる。
- メディア・マネージャとともに使用することによって、テープからバックアップを復元できる。

関連項目： Recovery Manager による回復の概要は、4-47 ページの「[ファイルの復元](#)」および 4-49 ページの「[メディア回復](#)」を参照してください。RMAN 回復の実行方法の詳細は、第 9 章の「[Recovery Manager による復元と回復](#)」を参照してください。RMAN の **recover** コマンドの詳細は、11-89 ページの「[recover](#)」を参照してください。

SQL*Plus の RECOVER コマンドによる回復 RMAN を使用しない場合でも、O/S コマンドを使用してバックアップを復元し、SQL*Plus コマンドを使用してメディア回復を実行できます。SQL*Plus コマンドを使用すると、次のことができます。

- RMAN の **copy** コマンドを使用して生成した O/S コピーとファイルの両方を復元できる。
- メディア回復を手動で制御できる。

回復に使用できる基本的な SQL*Plus コマンドは次の 3 つです。

- RECOVER DATABASE
- RECOVER TABLESPACE
- RECOVER DATAFILE

これらのコマンドは、それぞれ ALTER DATABASE 文の副文句にもなっていることに注意してください。RECOVER 句を指定して ALTER DATABASE 文を使用するのではなく、SQL*Plus の RECOVER コマンドを使用することをお勧めします。SQL*Plus の RECOVER コマンドの詳細は、『SQL*Plus ユーザーズ・ガイドおよびリファレンス』を参照してください。

各コマンドは、同じ基準を使用して、どのファイルが回復可能であるかを判断します。Oracle は、回復対象のファイルをロックできない場合、エラーを表示します。2 つの回復セッションが同じファイルを回復したり、メディア回復が使用中のファイルに対して行われたりすることが、このエラー表示により防止されます。いずれの場合も、データベース、表領域またはデータ・ファイルの回復が可能です。

関連項目 : SQL の ALTER DATABASE RECOVER 文と SQL*Plus の RECOVER の違いの詳細は、14-8 ページの「[メディア回復文の使用方法](#)」を参照してください。

第 II 部

Recovery Manager の使用方法

Recovery Manager の概要

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

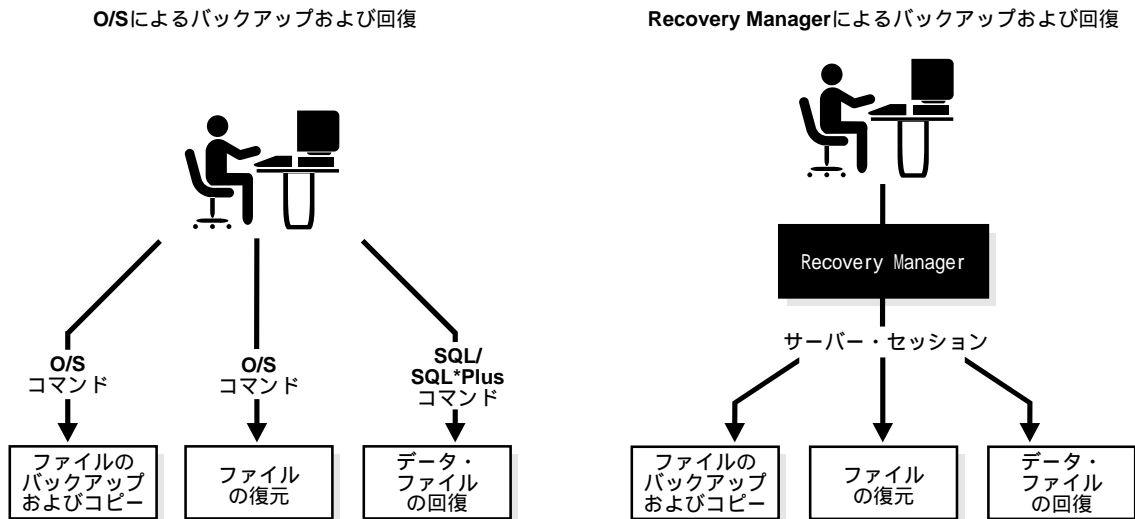
- Recovery Manager の概要
- Recovery Manager のコマンド
- Recovery Manager のメタデータ
- メディア管理
- リストとレポート
- チャンネルの割当て
- バックアップ・セット
- バックアップ・タイプ
- イメージ・コピー
- ファイルの復元
- メディア回復
- データベースの複製
- 整合性チェック

Recovery Manager の概要

Recovery Manager (RMAN) は、Oracle ツールの一つで、データ・ファイル、制御ファイルおよびアーカイブ REDO ログの、バックアップ、復元および回復に使用します。O/S プロンプトからコマンド行ユーティリティとして RMAN を起動することも、GUI ベースの Enterprise Manager の一部である Backup Manager を使用することもできます。

RMAN を使用すると、以前は手動で実行していたさまざまなバックアップおよび回復タスクを自動化できます。たとえば、各データ・ファイルに対応するバックアップを検索する、これらを O/S コマンドで正しい場所にコピーする、そして適用するアーカイブ・ログを選択するというタスクが、RMAN によって自動的に行われます。

図 4-1 RMAN と O/S におけるバックアップおよび回復手順の比較



RMAN を起動すると、次のような処理が行われます。

- クライアントで RMAN ユーザー・セッションが開始される。
- RMAN によって 2 つのデフォルト・サーバー・セッションが作成され、ターゲット・データベースに接続される。**ターゲット・データベース (target database)** とは、バックアップまたは復元するデータベースのことです。
- ディスク、またはテープ上で I/O を実行する場合、RMAN では、ディスクまたはデバイスごとに 1 つのチャンネルを割り当てる必要がある。チャンネルはサーバー・セッションに対応します。

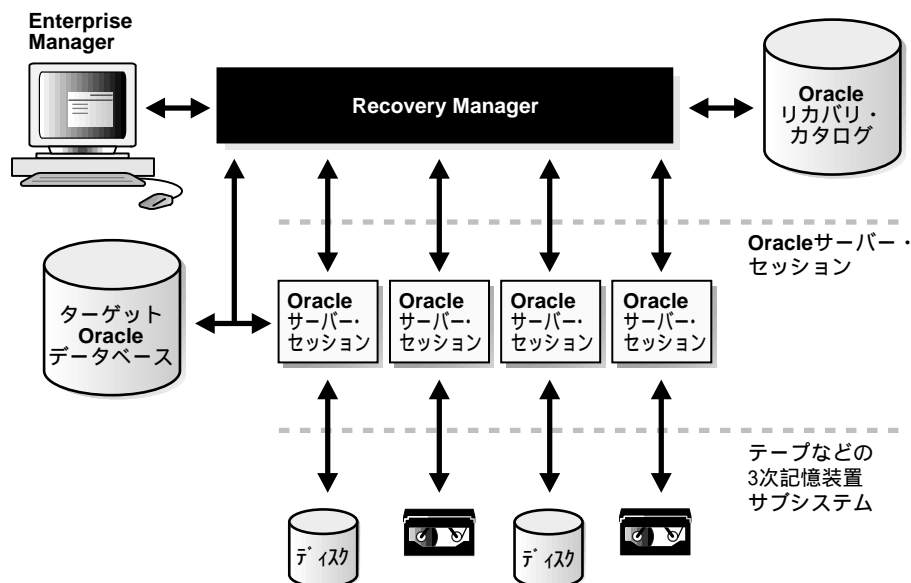
- リカバリ・カタログに接続した場合、RMAN はそのリカバリ・カタログ・データベースのサーバー・セッションを作成する。

RMAN を使用して、ターゲット・データベースに接続する場合、RMAN はサーバー・セッションを使い、PL/SQL インタフェースを通じてバックアップおよび回復操作を実行します。RMAN はそのバックアップおよびコピーを物理的にディスクに格納します。メディア管理ソフトウェアを使用している場合はテープに格納します。

RMAN は、そのバックアップおよび回復操作に関するメタデータを、情報のリポジトリであるリカバリ・カタログに格納するか、または制御ファイルにのみ格納します。通常、リカバリ・カタログは、別のデータベースに格納されます。リカバリ・カタログを使わない場合には、RMAN は制御ファイルをメタデータのリポジトリとして使用します。

注意： Recovery Manager は、リリース 8.0 以上の Oracle データベースとのみ互換性を持っています。Oracle7 Enterprise Backup Manager (EBU) とは互換性がありません。

図 4-2 Recovery Manager とオプションのリカバリ・カタログを併用した場合



関連項目： RMAN の互換性およびアップグレードの詳細は、『Oracle8i 移行ガイド』を参照してください。

Recovery Manager の機能

RMAN を使用すると、バックアップおよび回復の重要な手順が自動化されます。たとえば、Recovery Manager には次のような機能があります。

- サーバー・セッションを使用して、データベース、表領域、データ・ファイルおよびアーカイブ REDO ログの、バックアップおよび復元を実行する。
- 一度でも使用されたデータ・ブロックのみが含まれるよう、データ・ファイルのバックアップを圧縮する。
- 頻繁に実行されるバックアップおよび回復操作を、スクリプトに格納する。
- 増分バックアップを実行する。増分バックアップは、前回のバックアップ以降に変更されたデータ・ブロックのみをバックアップします。
- 本番データベースをテスト用に複製する。
- サード・パーティのメディア管理ソフトウェアを使用する。
- バックアップおよび回復操作すべての印刷可能なメッセージ・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的にパラレル化する。
- バックアップ制御ファイルを使用してバックアップを復元した場合、復元されたデータ・ファイルの構造を反映するよう、バックアップ制御ファイルを自動的に調整する。
- 回復に必要な REDO の量についてユーザーが指定した制限に基づき、バックアップが必要なデータ・ファイルを検索する。
- クロスチェックを実行し、メディア管理カタログにアーカイブされた情報が引き続き使用可能であるか確認する。
- 指定したバックアップが復元可能かどうかテストする。

Recovery Manager の機能の確認

RMAN は、次に示すようなものではありません。

- バックアップおよび回復操作を実行するための唯一の選択肢である。O/S コマンドを使用してファイルをバックアップおよび復元し、SQL/SQL*Plus 文で回復するという方法も選択できます。
- リカバリ・カタログである。リカバリ・カタログは、RMAN の操作に関する情報が含まれたオプションのスキーマです。RMAN は、バックアップおよびコピーをリカバリ・カタログに格納するのではなく、バックアップおよびコピーに関する情報をリカバリ・カタログに格納するということにも注意してください。
- リリース 8.0 以前の Oracle データベースと互換性がある。

- Oracle7 ユーティリティの Enterprise Backup Utility (EBU) である。RMAN は、EBU とは互換性がありません。
- ある O/S 固有のバックアップ・ユーティリティである。RMAN は、さまざまな異なるオペレーティング・システムで実行が可能な汎用ユーティリティです。

Recovery Manager のコマンド

RMAN コマンドを使用すると、バックアップおよび回復にかかわるすべての操作を管理することができます。Oracle リリース 8.0 で使用した RMAN コマンドは、すべてリリース 8.1 でも使用できることに注目してください。

この項では、次の項目について説明します。

- [Recovery Manager PL/SQL パッケージ](#)
- [Recovery Manager におけるコマンドのコンパイルおよび実行](#)
- [Recovery Manager コマンドのタイプ](#)
- [Recovery Manager コマンドのユーザー例外](#)
- [Recovery Manager のコマンド・エラー](#)

関連項目 : RMAN コマンドとその構文の詳細は、[第 11 章の「Recovery Manager のコマンド構文」](#)を参照してください。

Recovery Manager PL/SQL パッケージ

RMAN の実行可能ファイルは、PL/SQL プロシージャを使用してコマンドを解釈します。PL/SQL パッケージの主な機能は 2 つあります。

- RMAN のメタデータを制御ファイルまたはリカバリ・カタログにメンテナンスする。
- Oracle および O/S と通信し、バックアップ・セットおよびイメージ・コピーの作成、復元および回復を行う。

DBMS_RCVCAT パッケージおよび DBMS_RCVMAN パッケージは、**create catalog** コマンドで作成します。RMAN は、リカバリ・カタログに情報をメンテナンスする場合は DBMS_RCVCAT を、リカバリ・カタログまたは制御ファイルに問合せを行う場合は DBMS_RCVMAN を使用します。

DBMS_BACKUP_RESTORE パッケージは、dbmsbkrs.sql スクリプトおよび prvtbkrs.plb スクリプトで作成します。このパッケージは、catproc.sql スクリプトが実行されると、各 Oracle データベースに自動的にインストールされます。このパッケージは、Oracle サーバーおよびオペレーティング・システムとインタフェースをとり、Recovery Manager の指示に従って、バックアップおよび復元操作に I/O サービスを提供します。

関連項目 : DBMS_RCVCAT パッケージ、DBMS_RCVMAN パッケージおよび DBMS_BACKUP_RESTORE パッケージの詳細は、『Oracle8i パッケージ・プロシージャ リファレンス』を参照してください。

Recovery Manager におけるコマンドのコンパイルおよび実行

Recovery Manager におけるコマンドの処理は、次の 2 つの段階に分かれます。

- **コンパイル**
- **実行**

Recovery Manager で **backup** などのコマンドを発行すると、コマンド処理にはさまざまなフェーズがあることを示すメッセージが RMAN により生成されます。**backup tablespace** コマンドの場合のサンプル出力を次に示します。RMAN-03xxx で始まる RMAN のメッセージに注意してください。

```
RMAN-03022: compiling command: backup
RMAN-03023: executing command: backup
RMAN-08008: channel chl: starting full datafile backupset
RMAN-08502: set_count=48 set_stamp=346765191 creation_time=15-OCT-98
RMAN-08010: channel chl: specifying datafile(s) in backupset
RMAN-08522: input datafile fno=00017 name=/oracle/dbs/tbs_14.f
RMAN-08522: input datafile fno=00003 name=/oracle/dbs/tbs_11.f
RMAN-08522: input datafile fno=00004 name=/oracle/dbs/tbs_12.f
RMAN-08522: input datafile fno=00007 name=/oracle/dbs/tbs_13.f
RMAN-08013: channel chl: piece 1 created
RMAN-08503: piece handle=/oracle/dbs/lgaamds7_1_1 comment=NONE
RMAN-08525: backup set complete, elapsed time: 00:00:04
RMAN-03023: executing command: partial resync
RMAN-08003: starting partial resync of recovery catalog
RMAN-08005: partial resync complete
RMAN-08031: released channel: chl
```

コンパイル

RMAN は、コンパイル・フェーズで次の 2 つの処理を実行します。

- 名前の変換
- PL/SQL ジョブ・ステップの作成

名前の変換 RMAN の多くのコマンドは、どのオブジェクトに対して操作するかを指定する必要があります。コマンドに指定されたオブジェクトを、コマンドが実際に操作の対象とする 1 つ以上のエンティティのリストに RMAN が変換しますが、この操作を名前の変換と呼びます。ここで言うエンティティには、データ・ファイル、アーカイブ REDO ログ、データ・ファイルのコピーおよび制御ファイルのコピーなどが含まれます。たとえば、コマンド **backup database** を入力すると、RMAN はキーワード **database** を、データベースの全データ・ファイルのリストに変換します。同様に、次のコマンドを発行した場合、RMAN は、

archivelogRecordSpecifier 句（コマンドの **from ... until ...** 部分）を、アーカイブ REDO ログのリストに変換します。

```
backup archivelog from time='xxx' until time='yyy'
```

PL/SQL ジョブ・ステップの作成 RMAN は、自動的に 1 つ以上の PL/SQL プログラムを生成し、これを実行して作業を行います。基本的に、RMAN は、発行したコマンドを PL/SQL プログラムにコンパイルします。RMAN には、PL/SQL エンジンのコピーが組み込まれており、実行フェーズでは、PL/SQL プログラムを内部で実行します。これらのプログラムは、ターゲット・データベースに対してリモート・プロシージャ・コールを行い、バックアップ、復元およびその他必要な操作を実行します。

1 つの RMAN コマンドで、複数の PL/SQL プログラムを生成する場合があります。たとえば、**backup database** コマンドを実行すると、作成することになるバックアップ・セット 1 つにつき 1 つの PL/SQL プログラムが生成されます。同様に、**restore database** コマンドを実行すると、復元する必要があるバックアップ・セット 1 つにつき 1 つの PL/SQL プログラムが生成されます。

実行

RMAN は、コンパイル・フェーズでコンパイルした PL/SQL プログラムを、実行フェーズでスケジュールおよび実行します。RMAN は、割り当てられた各チャネル（各サーバー・セッション）に、1 つの PL/SQL プログラムを割り当てます。各チャネルは、割り当てられた PL/SQL プログラムを同時に実行します。たとえば、3 つのチャネルを割り当てた場合、RMAN は 3 つの PL/SQL プログラムを同時に実行します。

RMAN が複数の PL/SQL プログラムを同時に実行できるのは、これらのプログラムがターゲット・データベースに対して実行するリモート・プロシージャ・コール（RPC）で、非ブロック化ユーザー・プログラム・インタフェース（UPI）が使用されているためです。これにより、あるチャネルが非ブロック化 RPC コールを実行したとき、RMAN は別のチャネルに切り替えができます。RMAN は内部のポーリング・メカニズムを使用して、非ブロック化 RPC コールの完了を検出します。非ブロック化コールの完了後、RMAN は PL/SQL プログラムを再開します。

Recovery Manager コマンドのタイプ

RMAN で使用されるコマンドには、スタンドアロン・コマンドと、ジョブ・コマンドの 2 つの基本的なタイプがあります。スタンドアロン・コマンドは、**change**、**crosscheck** および **delete** コマンドを除き、自己完結しています。これに対して、ジョブ・コマンドは、**run** コマンドの大カッコ内で実行する必要があります。

ターゲットおよびオプションのリカバリ・カタログに接続した後は、**run** コマンドの内部でほとんどの RMAN コマンドを実行します。**run** 文の代表的な例を次に示します。

```
run {
    allocate channel c1 type 'sbt_tape';
    restore database;
```

```
        recover database;  
    }
```

関連項目 : RMAN の構文の詳細は、[第 11 章の「Recovery Manager のコマンド構文」](#)を参照してください。

スタンドアロン・コマンド

スタンドアロン・コマンドは、ジョブ・コマンドとは異なり、**run** コマンドの内部のサブ・コマンドとして実行されません。それ自体で実行可能なコマンドの一部を次に示します。

- **catalog**
- **change**
- **create catalog**、**drop catalog**、**upgrade catalog**
- **create script**、**delete script**、**replace script**
- **crosscheck**
- **delete expired backupset**
- **list**
- **report**

一部のコマンドは、実行前に **allocate channel for maintenance** コマンドを実行する必要がありますので、厳密にはスタンドアロンではありません。

関連項目 : スタンドアロン・コマンドの構文の詳細は、11-122 ページの「[rmanCmd](#)」を参照してください。

ジョブ・コマンド

ジョブ・コマンドは、スタンドアロン・コマンドとは異なり、**run** コマンドの大カッコ内で実行する必要があります。ジョブ・コマンドの例を次に示します。

- **allocate channel**
- **backup**
- **copy**
- **duplicate**
- **recover**
- **restore**
- **switch**

RMAN は、**run** コマンド・ブロックの内部で、ジョブ・コマンドを順次実行します。ブロック内部のコマンドに障害が発生すると、RMAN は処理を停止します。それ以降のコマンド

は実行されません。コマンド実行の単位は、実際には **run** コマンドが定義しています。**run** ブロック内部の最後のコマンドが完了すると、Oracle はブロック内に割り当てられていた I/O バッファや I/O スレーブ・プロセスなどのあらゆるサーバー側リソースを解放します。

関連項目 : ジョブ・コマンドの構文の詳細は、11-125 ページの「**run**」を参照してください。

コマンド例外

ほとんどのコマンドは、スタンドアロン・コマンドか、ジョブ・コマンドのどちらかに属しています。**run** ブロック外でジョブ・コマンドを発行したり、**run** ブロック内でスタンドアロン・コマンドを発行すると、RMAN は構文エラー・メッセージを発行します。次に示す例外は、スタンドアロン・コマンドとジョブ・コマンドの両方で機能します。

- @
- @@
- host
- send
- shutdown
- startup
- sql

コマンド行引数

RMAN ではさまざまなコマンド行引数がサポートされており、RMAN への接続時に指定できます。ほとんどの引数はコマンド行で指定します。例外は **target** および **catalog** で、この 2 つは RMAN の起動後コマンド行でも **connect** コマンドでも指定できます。**connect** コマンドを使用すると、セキュリティの問題が発生することがあるコマンド行へのパスワードの入力を回避できます。

関連項目 : コマンド行のオプションの詳細は、11-39 ページの「**cmdLine**」を参照してください。

Recovery Manager コマンドのユーザー例外

RMAN は、コマンドライン・インタプリタ (CLI) を使用しているため、対話モードでもバッチ・モードでもコマンドを実行できます。コマンド行で **log** オプションを指定し、RMAN 出力をログ・ファイルに書き込むこともできます。

対話モード

RMAN コマンドを対話的に実行するには、RMAN を起動した後、コマンド行インタフェースにコマンドを入力します。たとえば、次のように、RMAN を UNIX コマンド・シェルから起動し、対話コマンドを実行することができます。

```
% rman target sys/sys_pwd@prod1 catalog rman/rman@rcat
RMAN> run {
2> allocate channel d1 type disk;
3> backup database;
4> }
```

バッチ・モード

RMAN コマンドをファイルに入力しておくと、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 (command file)**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様に実行されます。

バッチ・モードでは、RMAN はコマンド・ファイル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。RMAN は、コマンド・ファイル全体を解析した後、各コマンドをコンパイルおよび実行します。オペレーティング・システムのジョブ制御機能を使用して、定期的スケジュールされたバックアップをとる場合は、バッチ・モードが最適です。

次の例では、4-9 ページの「**対話モード**」の RMAN のサンプル・スクリプトが、b_whole_10.rcv というコマンド・ファイルに入力されています。このファイルを O/S のコマンド行から実行し、次の様にログ・ファイル rman_log.f に出力を書き込むことができます。

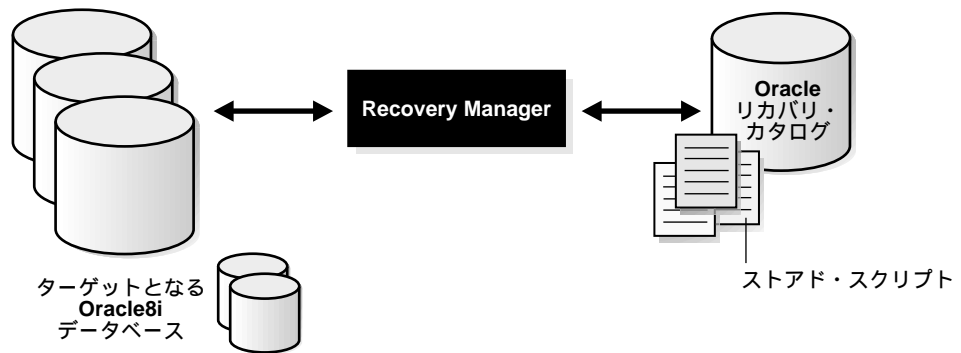
```
% rman target / catalog rman/rman@rcat @b_whole_10.rcv log rman_log.f
```

関連項目 : RMAN のコマンド行オプションの詳細は、11-39 ページの「**cmdLine**」を参照してください。

ストアド・スクリプト

ストアド・スクリプト (stored script) とは、リカバリ・カタログに格納された一連の RMAN コマンドのことで、中カッコで囲まれています。ストアド・スクリプトをメンテナンスしておくと、データベースのバックアップ、復元および回復のための一連のコマンドを、計画、開発またはテストできます。ストアド・スクリプトを使うと、オペレータ・エラーの可能性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 4-3 Recovery Manager のストアド・スクリプト



ストアド・スクリプトを作成するには、スクリプトを対話的に RMAN のコマンド行インタフェースに入力するか、または RMAN 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。

ストアド・スクリプトの例を次に示します。

```
replace script b_whole_10 {
    # back up 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/prod1/df/df_t%t_s%s_p%p'
        (database);
    sql 'ALTER SYSTEM ARCHIVE LOG CURRENT';
    backup
        filesperset 20
        format '/dev/backup/prod1/al/al_t%t_s%s_p%p'
        (archivelog all
        delete input);
}
```

ストアド・スクリプトを確認するには、リカバリ・カタログ・ビュー RC_STORED_SCRIPT に問い合わせます。

```
SQL> SELECT * FROM rc_stored_script;
```

DB_KEY	DB_NAME	SCRIPT_NAME
1	RMAN	full_backup
1	RMAN	incr_backup_0
1	RMAN	incr_backup_1
1	RMAN	incr_backup_2
1	RMAN	log_backup

関連項目：スクリプトの詳細は、6-21 ページの「リカバリ・カタログへのスクリプトの格納」を参照してください。 /demo ディレクトリに格納されたサンプル・スクリプトも参照してください。

Recovery Manager のコマンド・エラー

RMAN コマンドが正常に実行されたかどうかを判断することは重要なことです。たとえば、RMAN を使用した無人バックアップのスクリプトを書こうとしたとき、バックアップが成功したか、あるいは失敗したかを知りたい場合があります。

RMAN にエラーが発生したかどうかを確認する最も簡単な方法は、リターン・コードを調べることです。RMAN は、エラーが発生していなければ 0、発生していれば 1 を返します。たとえば、C シェルを使用して UNIX を実行している場合、RMAN は \$status というシェル変数にリターン・コードを出力します。

次に簡単な方法は、Recovery Manager の出力で、文字列 RMAN-00569 を検索することです。これは、エラー・スタックのメッセージ番号です。RMAN のエラーの前には、必ずこのエラー・メッセージがあります。出力に RMAN-00569 メッセージがなければ、エラーはありません。構文エラーのサンプル出力を次に示します。

```
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-00558: error encountered while parsing input commands
RMAN-01005: syntax error: found "}": expecting one of: "archivelog, backup, backupset,
channel, comma, controlfilecopy, current, database, datafile, datafilecopy, delete,
diskratio, filesperset, format, include, (, parms, pool, ;, skip, setsize, tablespace,
tag"
RMAN-01007: at line 1 column 58 file: standard input
```

Recovery Manager のメタデータ

RMAN メタデータとは、RMAN がバックアップ、回復およびメンテナンス操作を実行する際に使用する、ターゲット・データベースに関する情報のことです。この情報を格納するリカバリ・カタログ (recovery catalog) を作成するか、またはこの情報をターゲット・データベースの制御ファイルにのみ格納できます。RMAN は、制御ファイルのみで、主要なバックアップおよび回復操作をすべて実行できますが、リカバリ・カタログを使用している場合にしか機能しない RMAN コマンドも一部あります。(5-6 ページの「制御ファイルを RMAN メタデータに使用した場合」に、カタログのみで機能するコマンドのリストがあります)。

リカバリ・カタログをメンテナンスするのは RMAN のみです。ターゲット・データベースがリカバリ・カタログに直接アクセスすることはありません。RMAN は、データベース構造、アーカイブ REDO ログ、バックアップ・セットおよびデータ・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・カタログの中に移します。

関連項目 : RMAN のメタデータの管理方法は、第 6 章の「[Recovery Manager のメタデータの管理](#)」を参照してください。

メタデータのリカバリ・カタログへの格納

リカバリ・カタログとは、RMAN によって使用およびメンテナンスされる、ターゲット・データベースに関する情報のリポジトリです。リカバリ・カタログは既存のデータベースに保管できるため、リカバリ・カタログ格納のための追加データベースを用意する必要はありません。RMAN は、制御ファイルから取得したリカバリ・カタログの情報を使用して、要求されたバックアップおよび復元操作の実行方法を決定します。

リカバリ・カタログには、次の情報が含まれています。

- データ・ファイルおよびアーカイブ REDO 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。ユーザーが作成した、名前付きの一連の RMAN コマンドおよび SQL コマンド

関連項目 : リカバリ・カタログの異なるリリースのデータベースとの互換性の詳細は、『Oracle8i 移行ガイド』を参照してください。

リカバリ・カタログの再同期化

リカバリ・カタログは、ターゲット・データベースの制御ファイルから、重要な RMAN のメタデータを取得します。リカバリ・カタログを再同期化することにより、RMAN が制御ファイルから取得した情報が、現行の状態に保たれます。

再同期化には、完全再同期化か、または部分再同期化があります。部分再同期化では、RMAN は、現行の制御ファイルを読み取って変更済みデータを更新しますが、データベースの物理スキーマ (physical schema) に関するメタデータは再同期化しません。ここで物理スキーマとは、データ・ファイル、表領域、REDO スレッド、ロールバック・セグメント (データベースがオープンしている場合のみ) およびオンライン REDO ログのことを指します。完全再同期化では、RMAN は、スキーマ・レコードも含めてすべての変更済みレコードを更新します。

あるコマンドを RMAN で発行すると、RMAN は完全再同期化または部分再同期化が必要となるタイミングを自動的に検出し、必要に応じていずれかの方法で同期化を実行します。完全再同期化は、**resync catalog** コマンドを発行して実行することもできます。RMAN を一日に 1 回程度の割合で実行し、**resync catalog** コマンドを発行してカタログを現行の状態を保証することをお勧めします。

RMAN は、リカバリ・カタログを再同期化するたびに、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 (snapshot control file)** 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に、RMAN に制御ファイルの一貫性のあるビューがあることを保証します。スナップショット制御ファイルは、RMAN で短期間に使用するものであるため、リカバリ・カタログには登録されません。RMAN は、スナップショット制御ファイルのチェックポイントをリカバリ・カタログに記録し、リカバリ・カタログの現行の状態を示します。

Oracle8i 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、どの時点でも、1 つの RMAN 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 RMAN 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よ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は指定することができます。その方法については、5-3 ページの「**スナップショットの制御ファイル位置の決定**」を参照してください。

関連項目：リカバリ・カタログの再同期化の方法は、6-23 ページの「**リカバリ・カタログの再同期化**」を参照してください。**resync catalog** の構文は、11-119 ページの「**resync**」を参照してください。

リカバリ・カタログのバックアップ

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。そのため、リカバリ・カタログの消失は、重大な結果を招くことがあります。リカバリ・カタログは頻繁にバックアップしてください。

リカバリ・カタログは Oracle データベースにあるので、RMAN 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。他にも、バックアップ・オプションがいくつかあります。

リカバリ・カタログが壊れた場合に、使用可能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、カタログの一部を回復できます。ただし、常にリカバリ・カタログの有効な最新のバックアップを持つようにしてください。

関連項目 : リカバリ・カタログのバックアップ方法は、6-31 ページの「[リカバリ・カタログのバックアップ](#)」を参照してください。

メタデータの制御ファイルへの格納

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、RMAN 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のみを使う操作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、このモードが特に適しています。リカバリ・カタログのみ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となるためです。

Oracle が、この操作モードでサポートしていない機能は次のとおりです。

- 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および回復

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場合には、次のことを実行してください。

- 最低 2 つの多重化制御ファイル、またはミラー化制御ファイルを、それぞれ別のディスクで使用する。
- 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存する（[第 7 章の「Recovery Manager によるリストおよびレポートの生成」](#)を参照）。すべての Recovery Manager のバックアップ・ログを保存してください。

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、カスタマ・サポートに連絡することです。カスタマ・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。
 - バックアップされているファイルの名前。
 - それらのファイルのバックアップ時期。
 - それらのファイルを含むバックアップ・ピースの名前。
-

関連項目 : リカバリ・カタログを使用しない場合に使用不可となるコマンドのリストは、5-6 ページの「[制御ファイルを RMAN メタデータに使用した場合](#)」を参照してください。

メディア管理

磁気テープ記憶装置を使用してデータベースのバックアップを行う場合は、RMAN は **メディア・マネージャ (media manager)** を使用する必要があります。メディア・マネージャとは、テープ・ドライブなどの順次メディアのロード、ラベル付け、およびアンロードを行う機能を持つユーティリティで、データのバックアップおよび回復に使用します。

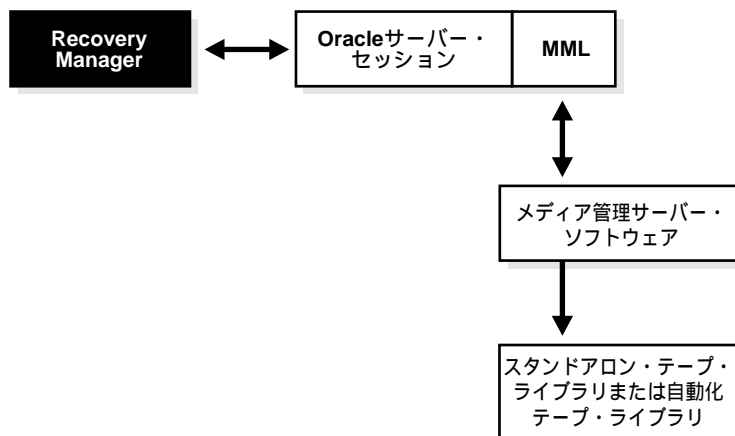
メディア管理製品の中には、Oracle データ・ファイルとバックアップ装置との間のデータの移動を、すべて管理できるものもあります。そのような製品には、格納サブシステムとメディア・サブシステムとの間の高速接続などのテクノロジーが使用されている場合があります。プライマリ・データベース・サーバーのバックアップの負荷は、このテクノロジーでほとんど削除できます。

RMAN を使用して、テープなどの順次メディアにバックアップをとる場合は、メディア管理ソフトウェアを Oracle ソフトウェアに統合する必要があります。バックアップ・データの書き込み先がディスクの場合は、Oracle をメディア管理ライブラリ (MML) ソフトウェアに接続する必要がないことに注意してください。

RMAN は、Oracle7 Enterprise Backup Utility (EBU) と同じメディア管理 API を使用しているため、EBU と同じメディア・マネージャとは互換性があります。ただし、所有しているメディア・ソフトウェアが、RMAN との互換性を有しているかについては、当該メディア・ベンダーに問い合わせる必要があります (4-19 ページの「[Backup Solutions Program](#)」を参照)。

図 4-4 に、Oracle に統合されたメディア・マネージャのアーキテクチャを示します。

図 4-4 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使用されるものと同じです。前述のダイアグラムの MML 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およびソースとして使い、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。

メディア・マネージャを使用したバックアップおよび復元処理

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 {
    # Allocating a channel of type 'sbt_tape' specifies a media management device
    allocate channel chl type 'sbt_tape';
    backup datafile 10;
}
```

Recovery Manager はこ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・セッションに対して、バックアップ要求を送信します。Oracle サーバー・セッションは出力チャンネルをメディア管理装置として識別し、メディア・マネージャに対して、テープをロードし、出力を書き込むように要求します。

メディア・マネージャは、テープにラベルを付け、テープおよび各テープ上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。それがない場合、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。

メディア・マネージャは、バックアップ操作および復元操作を行います。ファイルの復元では、次のようなステップが発生します。

1. 特定のファイルの復元が Oracle から要求される。
2. メディア・マネージャは、そのファイルが含まれているテープを判別し、これを読み取る。
3. メディア・マネージャは、読み取った情報を Oracle サーバー・セッションに戻す。
4. Oracle セッションは、そのファイルをディスクに書き込む。

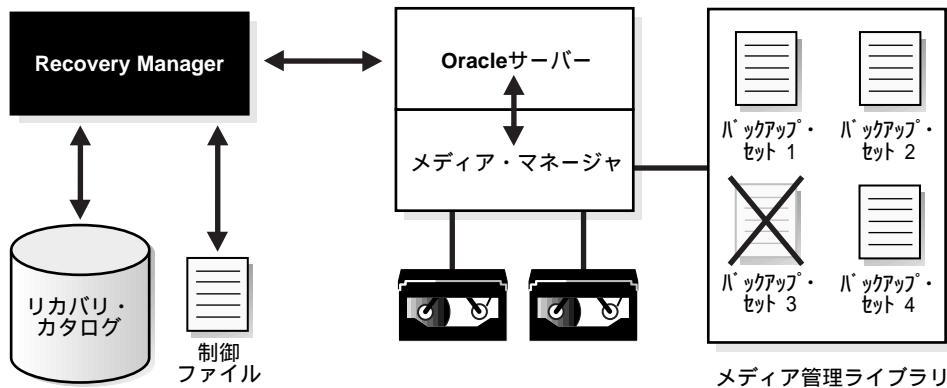
メディア・マネージャのクロスチェック

場合によってはメディア管理ライブラリのテープが、使用不可能な場合があります。RMAN は、クロスチェックを実行し、使用できるバックアップ・ピースがあるかを判断します。これにより、リカバリ・カタログと、メディア管理カタログとの同期が保たれます。

たとえば、**crosscheck backup** を発行すると、テープ上のバックアップをすべてチェックできます。特定のバックアップを RMAN が検索できなかった場合は、RMAN メタデータ内で**期限切れ**になった (expired) ものとして状態を変更します。バックアップおよびコピーの

状態を確認するには、**list** コマンドを発行するか、または該当するリカバリ・カタログ・ビューにアクセスします。

図 4-5 クロスチェック



関連項目：クロスチェックの実行方法は、6-9 ページの「[RMAN メタデータのクロスチェック](#)」を参照してください。**crosscheck** コマンドおよび **change ... crosscheck** コマンドの構文は、11-57 ページの「[crosscheck](#)」および 11-35 ページの「[change](#)」を参照してください。

プロキシ・コピー

Oracle は、そのメディア管理 API に、**プロキシ・コピー (proxy copy)** の機能を統合しています。各ベンダーは、この API を使用することにより、バックアップおよび復元操作の制御機能を持つメディア管理ソフトウェアを開発することができます。RMAN は、バックアップまたは回復が必要なファイルのリストをメディア・マネージャに提供します。そして、メディア・マネージャは、データ移動の方法と時期に関するすべての決定を行います。

backup proxy コマンドを使用してファイルのバックアップを試行するたびに、RMAN はメディア・マネージャに問合せを行い、プロキシ・コピーを実行できるかどうかを確認します。メディア・マネージャがそのファイルのプロキシ・コピーを実行できない場合、RMAN は従来のバックアップ・セットを使用してバックアップを行います。例外として、**proxy only** オプションを使用した場合が挙げられます。このオプションを指定すると、プロキシ・コピーできない場合は、Oracle からエラー・メッセージが発行されます。

Oracle は、プロキシ・コピーされた各ファイルのレコードを、制御ファイルに記録します。RMAN は、この情報を使用してリカバリ・カタログの再同期化を行います。プロキシ・コピーの情報を取得するには、動的パフォーマンス・ビューである、**VSPROXY_DATAFILE** および **VSPROXY_ARCHIVEDLOG** にアクセスします。プロキシ・バックアップの状態を削除したり、変更するには、**change ... proxy** コマンドを使用します。

注意： プロキシ・コピーをサポートする RMAN をプロキシ・コピーをサポートしないターゲット・データベースで使用する場合、RMAN はバックアップ・セットの作成にプロキシ・コピーをしません。プロキシ・コピーを使用してバックアップを作成し、Oracle をプロキシ・コピーをサポートしないバージョンにダウングレードした場合、RMAN は復元にプロキシ・コピー・バックアップをしません。また、利用できる最良のファイルがプロキシ・コピーである場合、RMAN は警告メッセージを発行します。

関連項目： V\$PROXY_DATAFILE および V\$PROXY_ARCHIVEDLOG の詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。backup コマンドの構文は、11-21 ページの「**backup**」を参照してください。

メディア・マネージャのテスト

新しいクライアント・プログラムの一つに、`sbttest` があります。これは、メディア管理ソフトウェアのテストを行うためのスタンドアロンのプログラムで、Oracle とリンクしてテープへのバックアップを実行します。これを使用するのは、バンドルされた Legato Storage Manager を使用しても、別のベンダーのメディア管理製品を使用しても、Oracle でバックアップの作成または復元ができない場合です。`sbttest` プログラムは、必ずカスタマ・サポートの指示に従って使用してください。

Backup Solutions Program

オラクルの Backup Solutions Program (BSP) は、オラクルのメディア管理ライブラリ (MML) 仕様に適合したメディア管理製品群です。MML インタフェースに適合したソフトウェアを使用すると、Oracle サーバー・セッションは、ファイルのバックアップおよび復元のコマンドを、メディア・マネージャに発行できるようになります。メディア・マネージャは、このコマンドが要求するテープのロード、ラベル付けおよびアンロードを実行します。

MML 準拠製品の一つに、Legato Storage Manager (LSM) があります。これは、複数の一般的なプラットフォームで使用可能です。各プラットフォーム用のオラクルのソフトウェアには、LSM が付属しています。使用しているオラクルのソフトウェアに LSM が付属している場合は、『Legato Storage Manager 管理者ガイド』を参照し、その機能を確認してください。その他の BSP メディア製品が付属している場合は、使用しているプラットフォーム固有のマニュアルを参照してください。

他にも、メディア管理ベンダーから入手可能な、各プラットフォーム用のメディア管理製品があります。BSP のウェブ・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、詳細な情報が必要な場合は、オラクル社の営業担当に問い合せてください。

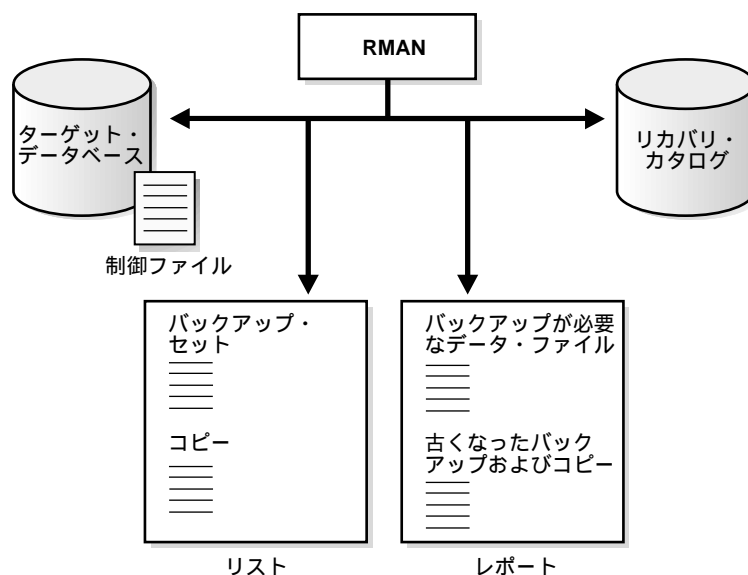
特定のメディア管理製品を使用する場合は、その製品が Oracle の BSP に属する製品かどうかをそのベンダーに直接問い合わせて確認してください。オラクル社は、RMAN との互換性に関して、メディア・ベンダーに保証を与えるものではありません。したがって、可用性、バージョンの互換性および機能については、オラクル社ではなくすべてそのメディア・ベンダーに直接問い合わせてください。

リストとレポート

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するには、RMAN の **report** コマンドおよび **list** コマンドを使用します。RMAN は、メタデータ・リポジトリからこれらの情報を取得します。ここで言うメタデータ・リポジトリとは、制御ファイルまたはリカバリ・カタログのことです。

list コマンドでは、RMAN のメタデータ・リポジトリの内容がリスト表示されますが、**report** コマンドでは、さらに詳しい分析が行われます。RMAN は、これらのコマンドの出力を、スクリーンまたはログ・ファイルに書き込みます。

図 4-6 RMAN のリストとレポート



バックアップおよびコピーのリスト

リカバリ・カタログに対する問合せを行い、その内容のリストを作るには、**list** コマンドを使用します。このコマンドは、次のリスト表示に使用します。

- 指定したデータ・ファイル・リストのバックアップ
- 指定したデータ・ファイル・リストのイメージ・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
- 指定した表領域リストのメンバーであるデータ・ファイルのイメージ・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）アーカイブ REDO ログのバックアップ
- 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）アーカイブ REDO ログのイメージ・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：バックアップおよびイメージ・コピーの生成方法は、7-2 ページの「[リストの生成](#)」を参照してください。**list** コマンドに関する参照情報は、11-76 ページの「[list](#)」を参照してください。

バックアップ、コピーおよびデータベース・スキーマに関するレポート

RMAN のレポートは、バックアップおよび回復状況の分析を行うために使用します。

RMAN レポートは、次のような質問に答えます。

- どのデータ・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データ・ファイルか？
- 利用できるバックアップまたはコピーの数が n より小さいので、バックアップが必要となるデータ・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？
- 回復不能操作が原因で回復不能になっているデータ・ファイルはどれか？
- データベースの現行の物理スキーマ、または過去のある時点の物理スキーマは何か？
- 孤立したバックアップはどれか？すなわち、データベースのインカーネーションには属しているが、このインカーネーションが現行のインカーネーションの直接の先祖でないため、復元操作に使用できないバックアップはどれか？

report need backup コマンドおよび **report unrecoverable** コマンドを定期的に発行し、メディア回復に必要なバックアップが使用可能で、かつ妥当な時間内にメディア回復が完了するような状態にしておいてください。

report コマンドを使うと、冗長であるか、または回復不能であるために、削除が可能なバックアップ・セットおよびデータ・ファイル・コピーのリストが示されます。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。

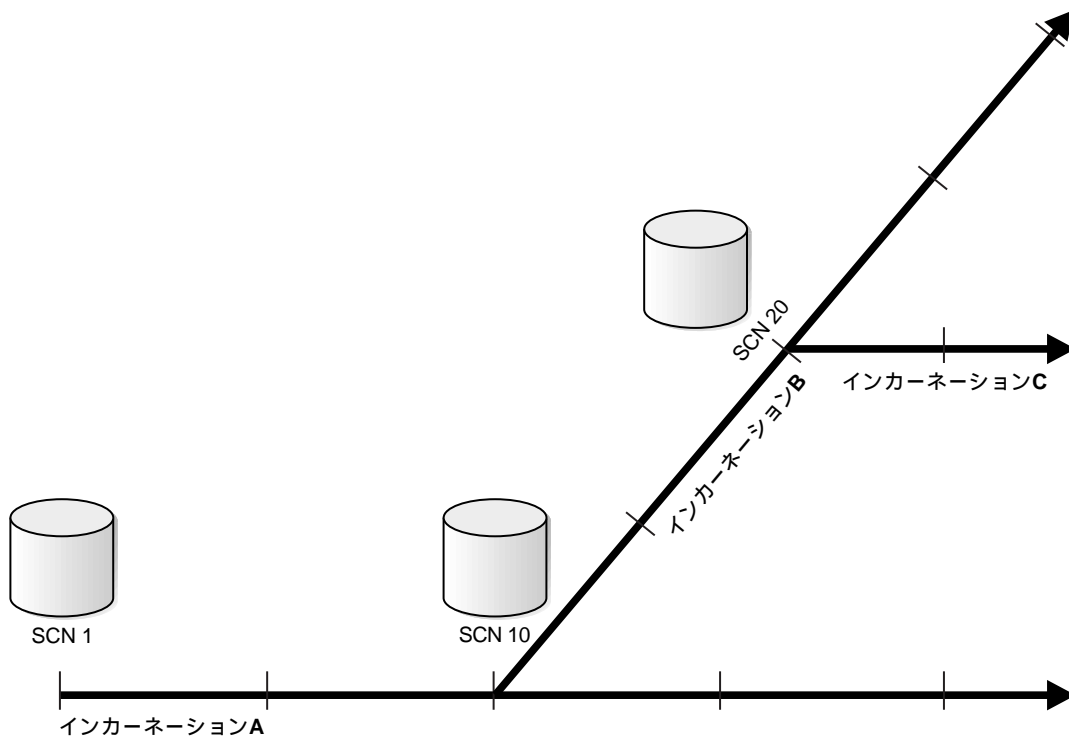
注意： バックアップを持たないデータ・ファイルは、回復不能とはみなされません。そのようなデータ・ファイルは、作成した時点からの REDO ログが残っていれば、CREATE DATAFILE コマンドを使用して回復することができます。

孤立したバックアップのレポート作成

report コマンドを使用すると、**孤立したバックアップ (orphaned backups)** のリストを表示できます。孤立したバックアップとは、データベースのインカーネーションには属しているものの、このインカーネーションが現行のインカーネーションの直接の先祖でないため、使用できないバックアップのことを指します。

図 4-7 はインカーネーションの例です。

図 4-7 孤立したバックアップ



データベースのインカーネーション A は、SCN1 から開始されています。SCN10 で、RESETLOGS 処理を実行してインカーネーション B を作成し、さらに SCN20 で、インカーネーション B に再度 RESETLOGS 処理を実行して、新しいインカーネーション C を作成したとします。

現行のインカーネーションによって、孤立するバックアップも変わりますが、その関係を次の表に示します。

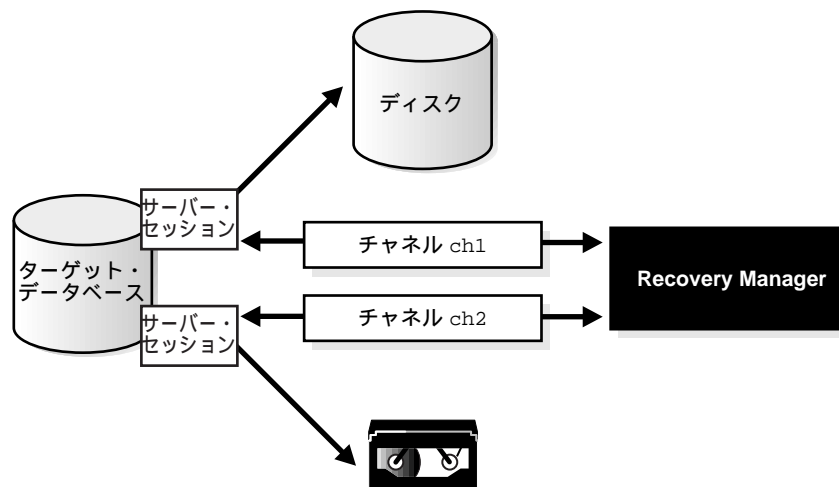
現行のインカーネーション	孤立したバックアップ	使用可能なバックアップ (孤立していないバックアップ)
インカーネーション A	インカーネーション B および C から作成した全バックアップ	インカーネーション A から作成した全バックアップ
インカーネーション B	SCN10 より後にインカーネーション A から作成したバックアップ インカーネーション C から作成した全バックアップ	SCN10 より前にインカーネーション A から作成したバックアップ インカーネーション B から作成した全バックアップ
インカーネーション C	SCN10 より後にインカーネーション A から作成した全バックアップ SCN10 より後にインカーネーション B から作成した全バックアップ	SCN10 より前にインカーネーション A から作成した全バックアップ SCN20 より前にインカーネーション B から作成した全バックアップ インカーネーション C から作成した全バックアップ

関連項目：レポートの生成方法は、7-5 ページの「[レポートの生成](#)」を参照してください。
report コマンドに関する参照情報は、11-103 ページの「[report](#)」を参照してください。

チャンネルの割当て

バックアップおよび回復コマンドを実行するには、あらかじめ**チャンネル (channel)** を割り当てておく必要があります。割り当てられた各チャンネルは、ターゲット・データベースまたは補助データベース (**duplicate** コマンドで作成されたデータベースまたは TSPITR で使用する一時データベース) のインスタンスでサーバー・セッションを起動し、これらのデータベースと RMAN との接続を確立します。バックアップおよび回復操作は、このサーバー・セッションで実行されます。割り当てられたサーバー・セッションと通信する RMAN セッションは 1 つしかありません。

図 4-8 チャンネルの割当て



allocate channel コマンド (run コマンド内で実行) および **allocate channel for maintenance** コマンド (RMAN プロンプトで実行) では、サーバー・セッションが、バックアップ、復元およびメンテナンス操作を実行する際に使用する、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、使用している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。

警告： Oracle バックアップのハードウェアによる多重化は推奨しません。

関連項目： **allocate channel** コマンドに関する参照資料は、11-9 ページの「[allocate](#)」を参照してください。 **allocate channel for maintenance** コマンドに関する参照資料は、11-13 ページの「[allocateForMaint](#)」を参照してください。

チャンネル制御オプション

チャンネル制御コマンドは、次のような場合に使用します。

- **backup**、**copy**、**restore**、**recover** の各コマンドを実行するときに RMAN が使う O/S リソースを制御する。
- バックアップの並列度を調整する (**backup** コマンドの **filesperset** パラメータと組み合わせて使用する)。
- I/O 帯域幅の消費制限を指定する (**set limit channel ... readrate**)。
- 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 channel ... 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 channel ... maxopenfiles`)
- ベンダー固有のコマンドをメディア・マネージャに送信する (`send`)

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。他のプラットフォームでは、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。読み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、リソースが割り当てられないオペレーティング・システムもあります。

注意： `type disk` を指定した場合は、サーバー・セッションの作成以外では 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メンテナンス・チャンネルを割り当てておく必要があります。`change ... crosscheck` の発行前も同様です。メンテナンス・チャンネルはメンテナンス・タスクにのみ使用します。このチャンネルをバックアップまたは復元ジョブの入力チャンネルまたは出力チャンネルとして使用することはできません。メンテナンス・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。

関連項目： `set` コマンドに関する参照資料は、11-130 ページの「[set](#)」を参照してください。
`allocate channel for maintenance` コマンドに関する参照資料は、11-13 ページの「[allocateForMaint](#)」を参照してください。

チャンネルの平行化

チャンネルは複数割り当てることができるため、1 つの `RMAN` コマンドで、複数のバックアップおよびイメージ・コピーを、平行に読み書きすることが可能です。したがって、割り当てるチャンネル数によって、コマンドの並列度が変わってきます。テープにバックアップを行う場合は、物理装置 1 つにつき 1 つのチャンネルを割り当てますが、ディスクにバックアップを行う場合は、必要な数のみチャンネルを割り当てることができるため、最大のスループットが得られます。

各 `allocate channel` コマンド、または各 `allocate auxiliary channel` コマンドは、別の接続を使ってターゲット・データベースまたは補助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。

パラレル化の度合いを左右する要素

RMAN は、**backup**、**copy** および **restore** の各コマンドの並列度を内部で調整します。指定する必要があるのは、次のもののみです。

- 複数の **allocate channel** コマンド
- バックアップ、コピーまたは復元の対象となるオブジェクト

RMAN 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 つのコマンド内のみに適用されます。したがって、データ・ファイルの 5 個のコピーが必要な場合は、**copy** コマンドを個別に 5 回発行するのではなく、5 個のコピーすべてを指定した **copy** コマンドを 1 回発行します。

次の RMAN スクリプトは、シリアルにファイル・コピーを作成します。ここでは、これらのファイルをバックアップするために、**copy** コマンドが 5 回使用されています。一度にアクティブになるチャンネルは 1 つのみです。

```
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;
  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';
}
```

次の文は、同じ例を**パラレル化 (parallelization)** します。1 つの RMAN の **copy** コマンドで、5 個のチャンネルが使用可能になり、5 個のファイルがコピーされます。5 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;
  copy datafile 5 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';
}
```

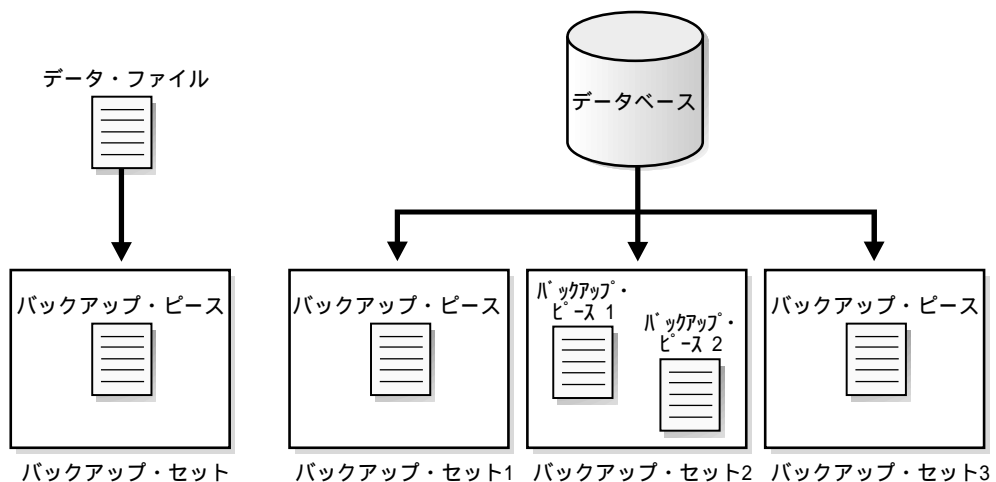
関連項目 : OPS 構成におけるパラレル化については、『Oracle8i Parallel Server 概要および管理』を参照してください。

バックアップ・セット

backup コマンドを実行すると、1 つ以上のバックアップ・セットが作成されます。論理構造であるバックアップ・セット (**backup set**) は、1 つ以上の物理的なバックアップ・ピース (**backup piece**) で構成されています。バックアップ・ピースとは、バックアップ・データ・ファイル、制御ファイルまたはアーカイブ REDO ログで構成された O/S ファイルのことです。ある 1 つのファイルを複数のバックアップ・セットに分割したり、アーカイブ REDO 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・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。バックアップ・セットは、RMAN 固有の形式で書かれています。これに対し、イメージ・コピーは、追加処理を行わずに使用できます。

図 4-9 1 つ以上のバックアップ・ピースで構成されるバックアップ・セット



ファイルのバックアップをとるときには、ターゲット・データベースをマウントするか、オープンする必要があります。データベースがマウント状態でかつマウント前に異常終了していなければ、RMAN は一貫性のあるバックアップを作成します。制御ファイルは現行のものを使用する必要があります。

データベースを ARCHIVELOG モードで運用している場合は、ターゲット・データベースのオープンおよびクローズはいずれも可能です。正しくクローズする必要はありません（ただし、バックアップを一貫性のあるものにするために、正しくクローズすることをお勧めします）。データベースを NOARCHIVELOG モードで運用している場合は、正しくクローズしてからバックアップをとる必要があります。

注意： 読み書き可能として指定されていないトランスポータブル表領域については、バックアップを作成できません。

この項では、次の項目について説明します。

- バックアップ・セットの格納
- バックアップ・セットの圧縮
- バックアップ・ピースのファイル名
- バックアップ・ピースのサイズ
- バックアップ・セットの数
- 多重バックアップ・セット
- 二重バックアップ・セット
- バックアップの平行化
- バックアップ・エラー

関連項目：バックアップの作成方法は、第 8 章の「[Recovery Manager でのバックアップおよびコピーの作成](#)」を参照してください。backup コマンドの詳細は、11-21 ページの「[backup](#)」を参照してください。

バックアップ・セットの格納

RMAN で作成したバックアップ・セットは、ディスクまたはテープなどの 3 次記憶装置に書き込むことができます。type disk を指定した場合は、ディスクにバックアップを行う必要があります。Oracle データ・ファイルの格納が可能な装置であれば、どの装置にもバックアップを行うことができます。つまり、CREATE TABLESPACE *tablespace_name* DATAFILE *filename* という文が有効であれば、*filename* は有効なバックアップ・パス名となります。

Oracle がサポートしており、かつ Oracle で使用可能な順次出力装置またはメディア管理システムを、任意のオペレーティング・システムで使用すると、磁気テープなどの順次出力メディアにバックアップ・セットを書き込むことができます。sbt_tape を指定すると、メディア管理ソフトウェアがサポートしているメディアであれば、どのメディアに対してもバックアップを行うことができます。

RMAN では、テープに対して直接アーカイブを行うことはできませんが、ディスクのアーカイブ REDO ログをテープにバックアップすることは可能です。delete input オプションを指定すると、RMAN はファイルのバックアップを行った後、そのファイルを削除します。また、回復時に必要なアーカイブ・ログは、テープからディスクに自動的に移動されます。

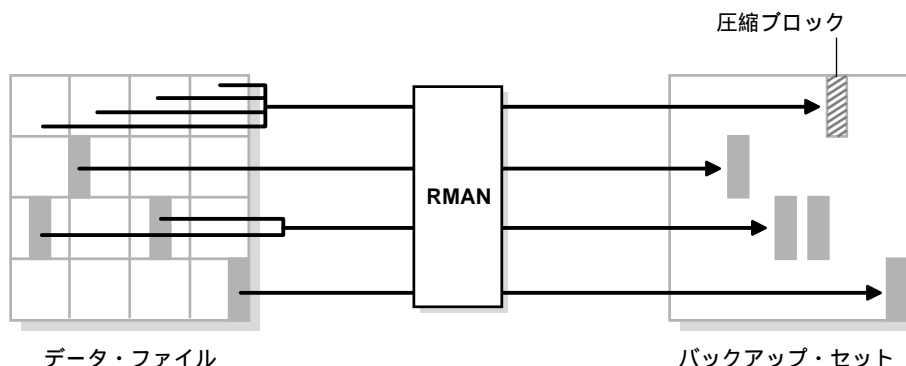
バックアップ・セットの圧縮

RMAN は、バックアップに対して**圧縮 (compression)**を行います。したがって、サーバー・セッションは、使用されたことのないデータ・ブロックの書き込みは行いません。ただし、データ・ファイルのイメージ・コピーについては、常にすべてのデータ・ブロックが含まれます。

データ・ファイル内のデータ・ブロックは、バッファに分割されます。RMAN は、バッファ内に使用済みデータ・ブロックを見つけると、その使用済みブロックのみをバックアップ・セットに書き込みます。未使用の入力ブロックだけで 4 つ連続するバッファを見つけた場合、RMAN は、1 つの (サイズ DB_BLOCK_SIZE の) 圧縮ブロックをバックアップ・セットに書き込みます。

バッファのサイズを設定する場合は、DB_FILE_DIRECT_IO_COUNT の初期化パラメータを使用します。たとえば、パラメータの値を 64K に設定するとします。この場合、RMAN は、入力ファイル内の連続した未使用の入力ブロック 256K につき、1 つの圧縮ブロックを書き込みます。

図 4-10 バックアップ・セットの圧縮



バックアップ・ピースのファイル名

バックアップ・ピースの名前を指定する場合は、RMAN に一意の名前を自動的に指定させるか、または **format** パラメータを使用します。ファイル名を指定しない場合、RMAN は %U 置換変数を使用して、一意の名前を保証します。置換変数は **backup** コマンドで取得します。これを使用することで一意のファイル名が生成できます。

関連項目 : **format** パラメータおよび **backup** コマンドの代替変数に関する参照資料は、11-21 ページの「**backup**」を参照してください。

バックアップ・ピースのサイズ

バックアップ・セットは、いずれも最低 1 つのバックアップ・ピースで構成されています。ディスクへの書込みを行うとき、バックアップ・ピースのサイズをメディア・マネージャまたは O/S がサポートする最大ファイルサイズに制限するには、**backup** コマンドの **setsize** オペランドを使用します。バックアップ・ピースのサイズを制限しない場合、Oracle は 1 つのファイルのみで構成されるバックアップ・セットを生成します。

RMAN は、バックアップ・セットの各ピースを順次書き込みます。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。RMAN は、バックアップのパーティション化を自動で行います。

関連項目 : **setsize** パラメータの詳細は、11-21 ページの「**backup**」を参照してください。

バックアップ・セットの数

backupSpec 句を使用すると、バックアップ対象をリスト表示したり、その他の便利なオプションを指定できます。RMAN が作成するバックアップ・セットの数は次に示した事柄によって変わります。

- 指定した **backupSpec** 句の数。
- 各 **backupSpec** 句内に指定または暗黙指定された入力ファイルの数。
- **filesperset** パラメータ。バックアップ・セットのファイル数が制限されます。

backupSpec 句を 1 つ指定することにより、1 つ以上のバックアップ・セットが生成されます。**backupSpec** 句に指定または自動的に選択された、入力ファイルの数が **filesperset** の制限を超えた場合、RMAN は複数のバックアップ・セットを作成します。**filesperset** の制限を指定しない場合は、1 つの **backupSpec** あたり 1 つのバックアップ・セットが生成されます。

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、出力テープ・デバイスのストリーミングを維持するのに必要な量を確保するため、またはバックアップによって特定のデータ・ファイルによる帯域幅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。

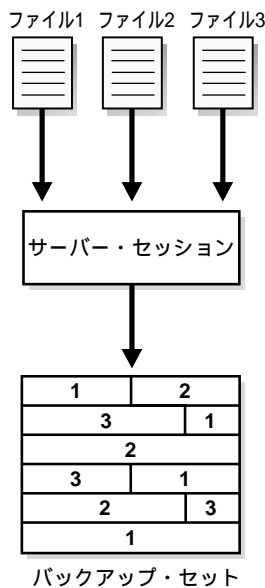
バックアップ・セット内のファイル数が少ないほど、個々のファイルの復元が速くなります。これは、他のデータ・ファイルに属していてスキップする必要があるデータの量が少なくなるからです。アーカイブ・ログを含むバックアップ・セットの場合は、同じ時間間隔のログをグループ化して、1 つのバックアップ・セットに入れることをお勧めします。同じ時間間隔のログは同時に復元する必要がある場合が多いからです。

関連項目 : **backupSpec** 句については、11-21 ページの「**backup**」を参照してください。

多重バックアップ・セット

Oracle では、同一のバックアップ・セットのデータ・ファイル・ブロックを多重化します。これはすなわち、バックアップ・セットの全データ・ファイルのデータ・ブロックが、他のデータ・ブロックとともに混在していることを意味します。図 4-11 に示したとおり、RMAN は 3 つのデータ・ファイルを 1 つのバックアップ・セットにバックアップできます。このバックアップ・セットには、1 つのバックアップ・ピースしか含まれていません。このバックアップ・ピースは、3 つの入力ファイルの各ブロック・コンポーネントが混在して構成されています。

図 4-11 データ・ファイルの多重化



Oracle によって同時にバックアップ・セットにバックアップされるデータ・ファイルの数を制御するには、**filesperset** パラメータを使用します。同時実行性の制御機能は、1 つのデータ・ファイルを多数の読取り要求で満杯にすることなく、テープ・デバイスのストリーミングを継続する場合に役立ちます。読取り要求が多数になると、オンライン・パフォーマンスが低下することがあります。読取りレートを制限するには、**set limit channel** コマンドの **readrate option** を使用します。

ファイルを多重化すると、次のようなことが可能となります。

- 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。自動的にパーティション化を選択するよう RMAN を設定する。

- パフォーマンスの高い順次出力デバイスに対して、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、ストリームを連続して提供する。デバイスに対してストリームを連続して提供することは、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・セットに制御ファイルを含める。この場合、最初に制御ファイルに書き込みが行われるため、そのブロックはデータ・ファイル・ブロックと多重化されません。
- バックアップ・セットをデータ・ファイルで構成するか、またはアーカイブ REDO ログで構成するかを選択する。ただし両方同時に含めることはできません。データ・ファイルとアーカイブ REDO ログを同じバックアップに書き込むことはできません。Oracle においては、多重化バックアップの各オブジェクトの論理ブロック・サイズは、それぞれ同じにする必要があるためです。

関連項目：バックアップ・パフォーマンスの調整方法は、『Oracle8i チューニング』を参照してください。バックアップの多重化の方法は、8-18 ページの「[バックアップにおけるデータ・ファイルの多重化](#)」を参照してください。

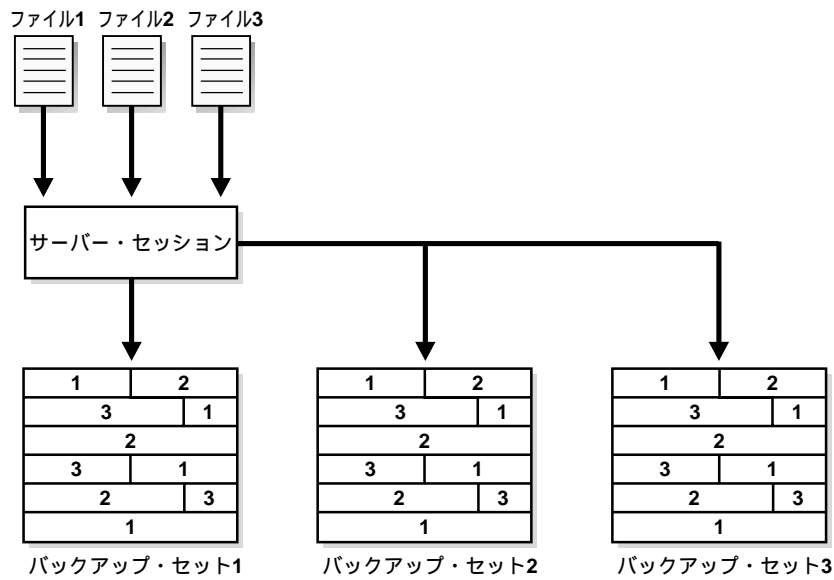
backup コマンドの **filesperset** パラメータに関する参照資料は、11-21 ページの「[backup](#)」を参照してください。**run** コマンドの **set** オプションに関する参照資料は、11-134 ページの「[set_run_option](#)」を参照してください。

二重バックアップ・セット

RMAN は、アーカイブ REDO ログまたはデータ・ファイル・バックアップ・セットのコピーを複数作成するための、効率的な機能を提供しています。**set duplex** コマンドの発行により、バックアップ・セットの同内容のコピーを最高 4 つまで作成できます。

多重化と二重化を混同しないでください。多重化とは、複数の入力ファイルを 1 つのバックアップ・セットにまとめることで、二重化とは、同内容のバックアップ・セットを複数出力することです。RMAN では、入力ファイルの多重化および出力の二重化が可能です。

図 4-12 多重バックアップ・セットの二重化



関連項目：バックアップの二重化の方法は、8-22 ページの「バックアップ・セットの二重化」を参照してください。**set duplex** コマンドに関する参照情報は、11-134 ページの「set_run_option」を参照してください。

バックアップの平行化

複数のバックアップ・セットを作成し、複数のチャネルを割り当てる処理は、RMAN が自動的に平行化します。複数のバックアップ・セットの書き込みも平行に行われます。割り当てが行われたサーバー・セッションは、指定されたファイルのバックアップ処理を分割します。

注意： 1つのバックアップ・セットを複数のチャネルに分散することはできません。

RMAN は、1つのデバイスに1つのバックアップ・セットを自動的に割り当てます。**channel** パラメータを使用すれば、backupSpec の全バックアップ・セットが特定のチャネルに書き込まれるよう Oracle を設定できます。その例を次に示します。

```
run {
  allocate channel ch1 type disk;
  allocate channel ch2 type disk;
```

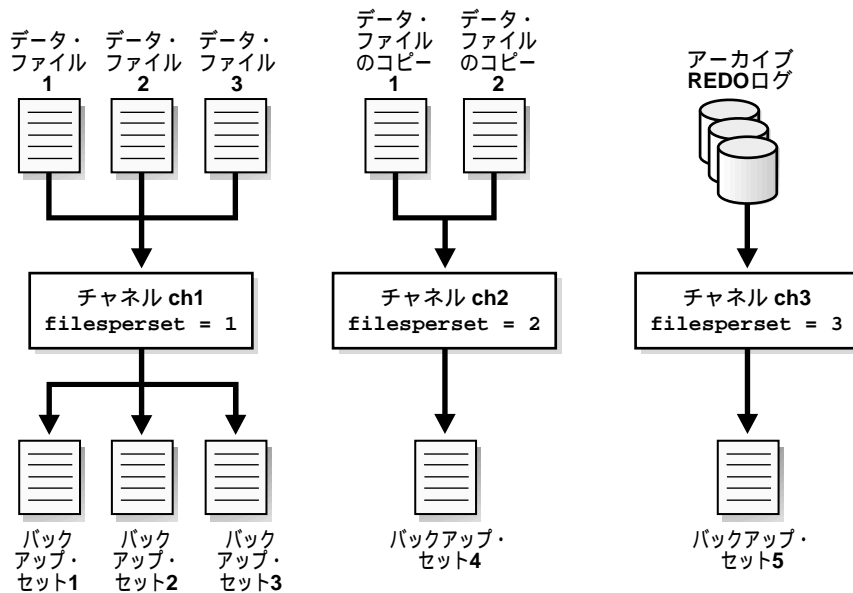
```

allocate channel ch3 type disk;
backup
  (datafile 1,2,3 filesperset = 1
   channel ch1)
  (controlfilecopy '/oracle/copy/cf.f' filesperset = 1
   channel ch2)
  (archivelog all filesperset = 3
   channel ch3);
}

```

表 4-13 にパラレル化の例を示します。この例では、チャンネル 1 がデータ・ファイルを、チャンネル 2 がデータ・ファイルのコピーを、そしてチャンネル 3 がアーカイブ REDO ログをそれぞれバックアップします。

図 4-13 バックアップのパラレル化



関連項目：チャンネル割当てとパラレル化の関係の概要は、4-26 ページの「[チャンネルのパラレル化](#)」を参照してください。バックアップのパラレル化の方法は、8-23 ページの「[バックアップ・セットのパラレル化](#)」を参照してください。**backup** コマンドの **channel** パラメータに関する参照資料は、11-21 ページの「[backup](#)」を参照してください。

バックアップ・エラー

RMAN は、主に 2 つのタイプのバックアップ・エラーを処理する能力を有しています。I/O エラーと破損ブロックです。ファイルの読取り時またはバックアップ・ピースへの書込み時、RMAN が I/O エラーを見つけると、システムはこれらのジョブを異常終了します。RMAN は、エラーの発生時に書込み中だったバックアップ・セットを、書き込み直す必要があります。ただし、異常終了前に書込みが済んでいたバックアップ・セットはすべて保持されます。

RMAN は、破損していると既に判断されているデータ・ファイル・ブロックをバックアップにコピーします。破損しているとまだ判断されていないデータ・ファイル・ブロックを見つけた場合、RMAN は、メディア障害が発生したという情報をデータブロック・ヘッダーに付けてこのブロックをバックアップに書き込みます（このデータ・ファイルの **set maxcorrupt** が 0 以外の値で、かつ破損の数がこの制限を超えていない場合）。いずれの場合も、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。制御ファイルのこれらのレコードにアクセスするには、V\$BACKUP_CORRUPTION ビューを使用します。

指定したデータ・ファイル、または指定したデータ・ファイルの一覧における、未検出のブロック破損数に制限を設定するには、**set maxcorrupt** コマンドを使用します。**backup** コマンドまたは **copy** 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 RMAN が 1 つの破損ブロックも許容しないことを意味します。

関連項目 : 分裂ブロックおよび破損ブロックの詳細は、4-53 ページの「[整合性チェック](#)」を参照してください。V\$BACKUP_CORRUPTION の詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。**set maxcorrupt** に関する参照資料は、11-134 ページの「[set_run_option](#)」を参照してください。

バックアップ・タイプ

Recovery Manager では、作成するバックアップのタイプを制御することができます。RMAN のバックアップは次のように分類できます。

- 全体または増分
- オープンまたはクローズ
- 一貫性または非一貫性

バックアップ・タイプ	定義
全体	<p>非増分バックアップを指す。つまり、この方式ではデータ・ファイル内の使用済みの全データ・ブロックがバックアップされます。</p> <p>注意: 全体バックアップは、全体データベース・バックアップとは異なります。全体データベース・バックアップとは、全データ・ファイルのバックアップおよび現行の制御ファイルで構成されるバックアップです。</p>
増分	<p>前回の増分バックアップ以降に変更されたブロックのみで構成されるデータ・ファイルをバックアップしたもの。増分バックアップには、基礎として機能する増分レベル 0 のバックアップが必要です。全体バックアップは、増分レベル 0 として使用することはできません。ただし、level パラメータを使用して作成した RMAN コピーは増分レベル 0 として使用することが可能です。</p>
オープン状態	<p>オープン状態になっているターゲット・データベースの任意の部分のバックアップ。</p> <p>注意: ALTER TABLESPACE BEGIN BACKUP 文を使用して、表領域をホット・バックアップ・モードにしないでください。RMAN は、別の方法を使用してホット・バックアップの一貫性を保ちます。</p>
クローズ状態	<p>マウントはされているが、オープン状態にはなっていないターゲット・データベースの任意の部分のバックアップ。クローズ状態のバックアップは、一貫性の有無を問わず実行できます。</p> <p>注意: リカバリ・カタログを使用する場合は、カタログのデータベースをオープンする必要があります。</p>
一貫性	<p>マウント前にクラッシュしていない、または ABORT オプションで停止されていないデータベースを、(オープンしたときではなく) マウントしたときにとったバックアップ。データ・ファイル・ヘッダーのチェックポイント SCN は、制御ファイルのヘッダー情報に合致しており、そのチェックポイント以降の変更が含まれているデータ・ファイルはありません。一貫性バックアップは、回復を実行することなく復元できます。</p>
非一貫性	<p>次のような状態になっているターゲット・データベースの任意の部分のバックアップ。</p> <ul style="list-style-type: none"> ■ オープン状態になっている。 ■ マウント前にクラッシュした。 ■ マウント前に ABORT オプションを使用して停止した。 <p>非一貫性バックアップを一貫性のあるものにするには、回復操作を行う必要があります。</p>

全体バックアップ

全体バックアップでは、ファイル全体が読み込まれ、未使用のデータ・ファイル・ブロックを除いた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。サーバー・セッションは、アーカイブ REDO ログまたは制御ファイルのバックアップ時、ブロックのスキップは行いません。

注意： 全体バックアップは全体データベース・バックアップとは異なります。全体バックアップの全体とは、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。

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。つまり、全体バックアップを行っても、後続の増分バックアップに含まれるブロックは変わりません。

Oracle では、次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（全データ・ファイルおよび現行の制御ファイル）

アーカイブ REDO ログが含まれたバックアップ・セットは、常に全体バックアップとなることに注意してください。

増分バックアップ

増分バックアップでは、ファイル全体が読み込まれますが、バックアップされるデータ・ブロックは、前回のバックアップ以降に変更されたブロックのみです。Oracle では、データ・ファイル、表領域またはデータベース全体の増分バックアップを、作成および復元できます。RMAN は、増分バックアップ・セットに制御ファイルを含めることができますが、その場合は制御ファイル全体を含めることになります。制御ファイルでスキップされるブロックはありません。

増分バックアップを作成する主な理由は次のとおりです。

- メディア・マネージャが使用するテープを節約できる。
- ネットワークを介してバックアップする場合、ネットワークの帯域幅を節約できる。
- テープ書込み I/O に使用できるテープ帯域幅の合計は、ディスク読取り I/O に使用できるディスク帯域幅の合計よりもはるかに少ない。

これらの基準がいずれも適用されない場合は、全体バックアップの方が望ましい選択です。増分バックアップを適用すると、回復に要する時間が長くなる上、コストの節約もわずかであるためです。

この項では、次の項目について説明します。

- マルチレベル増分バックアップ
- 増分バックアップの仕組み
- 差分増分バックアップ
- 累積増分バックアップ
- 増分バックアップ計画

マルチレベル増分バックアップ

RMAN では、**マルチレベル増分バックアップ (multi-level incremental backups)** を作成することが可能です。増分レベルは、それぞれ 0、1、2 というように整数で表されます。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。このバックアップでは、データが含まれているブロックがすべてコピーされます。レベル n の増分バックアップを生成するときに、 n が 0 より大である場合は、次のいずれかのバックアップが作成されます。

- レベル n 以下の最後のバックアップ以降に変更されたすべてのブロック (増分バックアップのデフォルトのタイプで、これを**差分バックアップ (differential backup)**と呼びます)。
- レベル $n-1$ 以下の最新のバックアップ以後に使用された全ブロック (これを**累積バックアップ (cumulative backup)**と呼びます)。

マルチレベル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る必要がないことです。RMAN は、データ・ファイルのすべてのブロックを読み取る必要があるため、全体バックアップにかかる時間と増分バックアップにかかる時間はほぼ同じです (バックアップの出力がボトルネックにならない場合)。

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のみがコピーされます。バックアップ・ファイルのサイズを決定する要素は、変更されたブロックの数と増分バックアップのレベルのみです。

注意： ほとんどの場合、差分バックアップよりも累積バックアップの方がよりよい選択です。

増分バックアップの仕組み

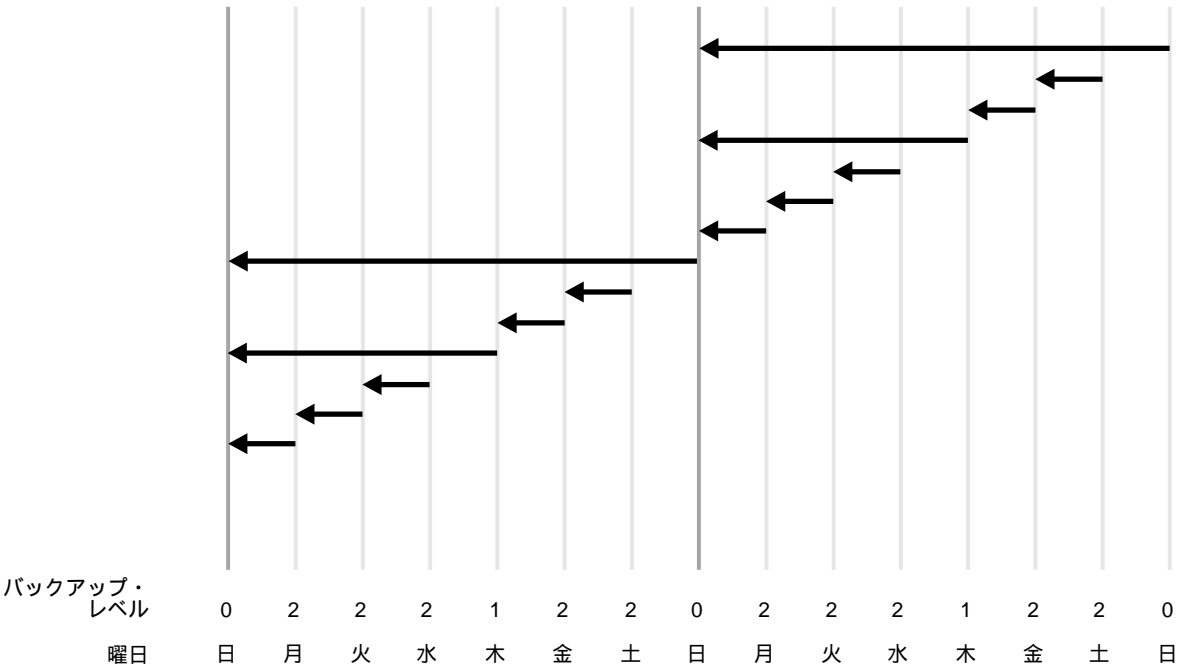
データ・ファイルの各データ・ブロックには SCN が含まれています。この SCN は、そのブロックに対して行われた最後の変更の時点の SCN です。増分バックアップ時、RMAN は入力ファイルの各データ・ブロックの SCN を読み取り、これを親増分バックアップのチェッ

クポイント SCN と比較します。この親バックアップは、RMAN が両 SCN の比較に使用するバックアップです。したがって、これはレベル 0 のバックアップとするか、または増分レベルに応じて、レベル 1 またはレベル 2 のバックアップとします。入力データ・ブロックの SCN が親のチェックポイント SCN よりも大きい場合、RMAN はこのブロックをコピーします。

差分増分バックアップ

差分レベル *n* の増分バックアップの場合、RMAN は、レベル *n* 以下の最後のバックアップ以降に変更されたブロックをすべてバックアップします。たとえば、差分レベル 2 のバックアップでは、レベル 1 とレベル 2 のバックアップのどちらが新しいかを RMAN が判断し、新しい方のバックアップ以降に変更されたブロックをすべてバックアップします。利用できるレベル 1 がない場合、RMAN は基礎となるレベル 0 のバックアップ以降に変更されたブロックをすべてコピーします。増分バックアップは、デフォルトでは差分に設定されています。

図 4-14 差分増分バックアップ (デフォルト)



上記の例では次のようになります。

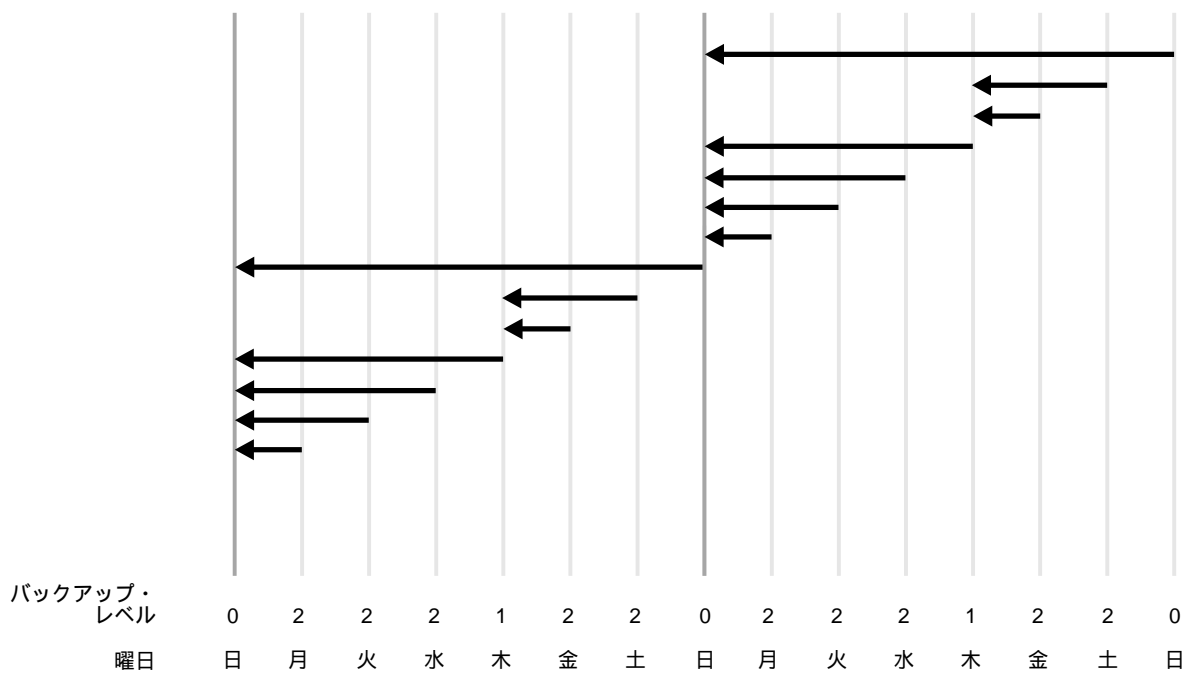
- 日曜日
レベル 0 の増分バックアップでは、このデータベース内でこれまでに使われたことのあるすべてのブロックのバックアップがとられます。
- 月曜日
レベル 2 の差分増分バックアップでは、レベル n 以下の最後の増分バックアップ以降に変更されたすべてのブロックのバックアップがとられます。この例では、レベル 2 以下の最後の増分バックアップは日曜日のレベル 0 のバックアップなので、日曜日以降に変更されたブロックのみのバックアップがとられます。
- 火曜日
レベル 2 の差分増分バックアップでは、レベル n 以下の最後の増分バックアップ以降に変更されたすべてのブロックのバックアップがとられます。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 のバックアップなので、月曜日以降に変更されたブロックのみのバックアップがとられます。
- 水曜日
レベル 2 の増分バックアップでは、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 のバックアップなので、火曜日以降に変更されたブロックのみのバックアップがとられます。
- 木曜日
レベル 1 の増分バックアップでは、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 の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みのバックアップがとられます。
- 金曜日
レベル 2 の増分バックアップでは、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 のバックアップなので、木曜日のレベル 1 のバックアップ以降に変更されたブロックのみのバックアップがとられます。
- 土曜日
レベル 2 の増分バックアップでは、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 のバックアップなので、金曜日のレベル 2 のバックアップ以降に変更されたブロックのみ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。

累積増分バックアップ

Oracle には、レベル 1 以上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。累積レベル n の増分バックアップの場合、RMAN は、レベル $n-1$ 以下の最後のバックアップ以降に使用されたブロックをすべてバックアップします。たとえば、レベル 2 の累積バックアップでは、レベル 1 のどのバックアップが最後に行われたかを RMAN が判断し、最後のレベル 1 のバックアップ以後に変更されたブロックをすべてコピーします。利用できるレベル 1 のバックアップがない場合、RMAN は基礎となるレベル 0 のバックアップ以降に変更されたブロックをすべてコピーします。

累積増分バックアップを使うと、特定レベルからの増分バックアップが 1 つのみで済むので、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差分バックアップよりも増加します。

図 4-15 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

レベル 0 の増分バックアップでは、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

レベル 2 の累積増分バックアップでは、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 の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み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 の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)。

- 水曜日

レベル 2 の累積増分バックアップでは、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 のバックアップなので、日曜日以降に変更された全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

レベル 1 の累積増分バックアップでは、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 のバックアップなので、日曜日以降に変更された全ブロックのバックアップがとられます。

- 金曜日

レベル 2 の累積増分バックアップでは、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 のバックアップなので、木曜日以降に変更された全ブロックのバックアップがとられます。

- 土曜日

レベル 2 の累積増分バックアップでは、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 のバックアップなので、木曜日以降に変更された全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。

増分バックアップ計画

バックアップ計画は、許容しうる平均回復時間（MTTR）に基づいて選択します。たとえば、レベル 0 のバックアップを月に 1 度、レベル 1 の累積バックアップを週に 1 度、そしてレベル 2 の累積バックアップを毎日実行するという、3 つのレベルのバックアップ計画を適用できます。この計画では、完全回復を実行する際に、1 日分の量を超える REDO を適用する必要はありません。

レベル 0 のバックアップをとる回数を決定する場合は、データの 50% 以上が変更されたら必ずレベル 0 の新しいバックアップをとるようにするというのがおおよその目安です。データの変更新率が予測できる場合は、増分バックアップのサイズを確認すれば、いつレベル 0 の新しいバックアップを行うのが適切かがわかります。次のような問合せを行うと、バックアップされたブロックが少なくとも 50% ある各データ・ファイルのバックアップ・セットに、いくつかのブロックが書き込まれたかがわかります。

```
SELECT file#, incremental_level, completion_time, blocks, datafile_blocks
FROM v$backup_datafile
WHERE incremental_level > 0 AND blocks / datafile_blocks > .5
ORDER BY completion_time;
```

差分バックアップまたは累積バックアップのブロック数を、基礎となるレベル 0 のバックアップと比較してください。たとえば、レベル 1 の累積バックアップを作成したのみの場合に、レベル 1 の最新のバックアップが、基礎となるレベル 0 のバックアップのほぼ半分のサイズであるときは、レベル 0 の新しいバックアップを作成します。

バックアップの制約

RMAN がバックアップ操作を実行するのは、データベースがインスタンスによってマウントされているか、またはオープンされている場合に限られます。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他のどのインスタンスもオープンしてはいけません。

RMAN は、データ・ファイル、アーカイブ REDO ログおよび制御ファイルのバックアップをサポートしています。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。

- パラメータ・ファイル
- 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
- トランスポータブル表領域により組み込まれた表領域（読み書き可能に指定される前）

関連項目：パラレル・サーバー環境におけるバックアップの制約の詳細は、『Oracle8i Parallel Server 概要および管理』を参照してください。

イメージ・コピー

イメージ・コピー (image copy) には、回復のために現状のままで使用できる 1 つのデータ・ファイル、アーカイブ REDO ログ・ファイルまたは制御ファイルが含まれています。イメージ・コピーを作成するには、RMAN の **copy** コマンドを使用するか、または UNIX の **cp** コマンドなどの O/S コマンドを使用します。

RMAN の **copy** コマンドで作成したイメージ・コピーは、Oracle サーバー・セッションで作成されたという点を除き、ある 1 つのファイルの O/S コピーと同等です。サーバー・セッションは、ファイル内のブロックの妥当性検査や、コピーの制御ファイルへの登録などの追加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で、バックアップ・セットとは異なります。RMAN は、ディスクにのみイメージ・コピーの書込みを行います。

RMAN イメージ・コピー

イメージ・コピーを作成するには、RMAN の **copy** コマンドを使用します。オリジナルのファイルを置き換える必要があり、かつイメージ・コピーがデータ・ファイルのものである場合は、そのコピーは復元する必要はありません。その代わりに Oracle には **switch** コマンドが用意されています。このコマンドは、制御ファイルがそのコピーをポイントするよう設定し、さらにコピーがスイッチされたことを示すようリカバリ・カタログを更新するものです。この場合、**switch** コマンドを発行することと、SQL 文である ALTER DATABASE RENAME DATAFILE を発行することは同等です。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。

RMAN 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。この操作は、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・ログのみです。

O/S イメージ・コピー

Oracle は、RMAN 以外のメカニズムで作成したイメージ・コピーもサポートしています。このイメージ・コピーは、O/S コピーとも呼ばれています。たとえば、UNIX の **cp** コマンドで作成したデータ・ファイルのコピーは O/S コピーです。このような O/S コピーは、**restore** または **switch** コマンドで使用する前に、RMAN でカタログ化しておく必要があります。

O/S コピーは、データベースがオープンされているときでもクローズされているときでも作成できます。データベースがオープンされていて、データ・ファイルが NORMAL モードでオフラインにされていない場合は、表領域をホット・バックアップ・モードにする必要があ

ります。つまり、SQL 文である ALTER TABLESPACE BEGIN BACKUP を発行してコピーを作成します。

警告： 表領域をホット・バックアップ・モードにしないでオンライン・バックアップを作成すると、Oracle が破損ブロックを生成することがあります。4-54 ページの「[論理ブロックの破損の検出](#)」の項を参照してください。

ミラー化ディスク・ボリュームにデータ・ファイルを格納しているサイトもあります。この場合は、ミラーを解除することでイメージ・コピーの作成が可能になります。ミラーを解除した後、RMAN に新しい O/S コピーの存在を通知すると、復元操作にこのコピーが使用できるようになります。そのコピーでは復元ができなくなったときは、**change ... uncatalog** コマンドを使用して、そのことを RMAN に通知する必要があります。この例では、ミラーを（ミラー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** コマンドを使ってリカバリ・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。

関連項目： コピーのカタログ化の方法は、6-30 ページの「[O/S バックアップのカタログ化の方法](#)」を参照してください。**change** コマンドに関する参照情報は、11-35 ページの「[change](#)」を参照してください。

バックアップおよびイメージ・コピーのタグ

バックアップ・セットおよびイメージ・コピー（RMAN で作成したコピーまたは O/S で作成したコピー）には、**タグ (tag)** 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。タグとは、weekly_backup など、バックアップ・セットやファイル・コピーの記号名のことを指します。**restore** または **change** コマンドを実行する場合は、ファイル名を指定するのではなくこのタグを指定します。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・ファイルについては、そのタグは、最新の適正なファイルを指します。デフォルトでは、Recovery Manager は、タグまたは **set until** 句で修飾されている場合を除き、最新のバックアップを選択して復元します。Point-in-Time 回復の場合、指定されたファイルが含まれた最も適正な現行バックアップが、最新のバックアップではないことがあります。

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、最新のコピーを指すとみなされます。したがって、次のようなコマンドで、datafile 3 を、月曜日の夜の最新のコピーに置き換えます。

```
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** のみ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： 入力ファイルを指定する際に、タグを **restore** コマンドまたは **switch** コマンドに指定すると、RMAN は、使用するバックアップ・セットまたはイメージ・コピーを選択するとき、合致しているタグが付されたバックアップ・セットのみをその候補と見なします。

関連項目： **switch** コマンドおよび **restore** コマンドに関する参照資料は、[第 11 章の「Recovery Manager のコマンド構文」](#)を参照してください。

ファイルの復元

ディスク上のバックアップ・セットまたはイメージ・コピーのデータ・ファイル、制御ファイルおよびアーカイブ REDO ログを復元するには、RMAN の **restore** コマンドを使用します。バックアップ・セットは Oracle 特有の形式で書かれるため、そのままインポートすることはできません。RMAN の **restore** コマンドを使用してバックアップ・セットを抽出する必要があります。これに対し、RMAN で作成されたイメージ・コピーは、追加の処理を行うことなく Oracle で使用することが可能です。

復元可能なものは次のとおりです。

- データ・ファイル
- 制御ファイル（現行またはコピー）
- アーカイブ REDO ログ

注意： 通常、アーカイブ REDO ログに対して復元を行うことはありません。この処理は、回復時に必要に応じて RMAN が自動的に実行するためです。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。

RMAN では、ファイル復元のプロシージャは完全に自動化されています。O/S で使用するバックアップまたはコピーを検索し、ファイルを手動で該当するディレクトリにコピーするといった手間がありません。RMAN は、サーバー・セッションに指示して、該当するバックアップまたはコピーを復元します。復元先は次のいずれかとなります。

- デフォルトの位置。ここに現在存在する同じ名前のファイルが上書きされます。
- 新しい位置。**set newname** コマンドで指定します。データ・ファイルを新しい位置に復元すると、Oracle はこれをデータ・ファイルのコピーと見なし、制御ファイルやリカバリ・カタログに記録します。

関連項目：バックアップ・セットおよびコピーの復元方法は、9-2 ページの「[データ・ファイル、制御ファイルおよびアーカイブ REDO ログの復元](#)」を参照してください。**restore** コマンドに関する参照情報は、11-113 ページの「[restore](#)」を参照してください。

復元処理時のファイル選択

RMAN 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・データベースの制御ファイル）を使って、使用可能な最善のバックアップ・セットまたはイメージ・コピーを、復元操作に使うために選択します。いずれも使用可能な場合は、バックアップ・セットではなくイメージ・コピーを選択します。複数の候補がある場合、RMAN は最新のバックアップ・セットまたはコピーを使用します。この場合、*untilClause* が指定されているかどうかも選択の基準に含まれます。

RMAN がバックアップ・セットまたはファイル・コピーを復元できるようになるには、**restore** コマンドのすべての設定が満たされている必要があります。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 **restore** コマンド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、RMAN は、復元ジョブのコンパイル段階でエラーを戻します。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は、新しいジョブを作成し、既存のバックアップ・セットまたはデータ・ファイル・コピーを含むデバイス用のチャンネルを指定してください。

復元の制約

restore コマンドには次のような制約があることに注意してください。

- 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らない。ただし、データベースをマウントする必要はありません。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。
- 復元する表領域またはデータ・ファイルをオフラインにするか、データベースをクローズする必要がある。
- 復元操作では、既存のデータ・ファイルを上書きするか、または **set newname** コマンドを使用し、その出力を新しいファイルに送信するかのいずれかである。
- 読み書き可能に指定されていないトランSPORTABLE 表領域を復元することはできない。
- RMAN を使用して、あるホストで作成されたイメージ・コピーを別のホストに復元することはできない。このようなファイルは、手動で移送するか、または **catalog** コマンドを使用してカタログ化しないと、別のホストに復元できません。

メディア回復

メディア回復とは、オンライン REDO ログ、アーカイブ REDO ログまたは増分バックアップを、復元されたデータ・ファイルに適用し、このファイルが現行の状態または指定した時点の状態になるよう更新することです。メディア回復を実行し、増分バックアップを自動的に適用するには、RMAN の **recover** コマンドを使用します。回復は、現行のデータ・ファイルに対してのみ実行できます。

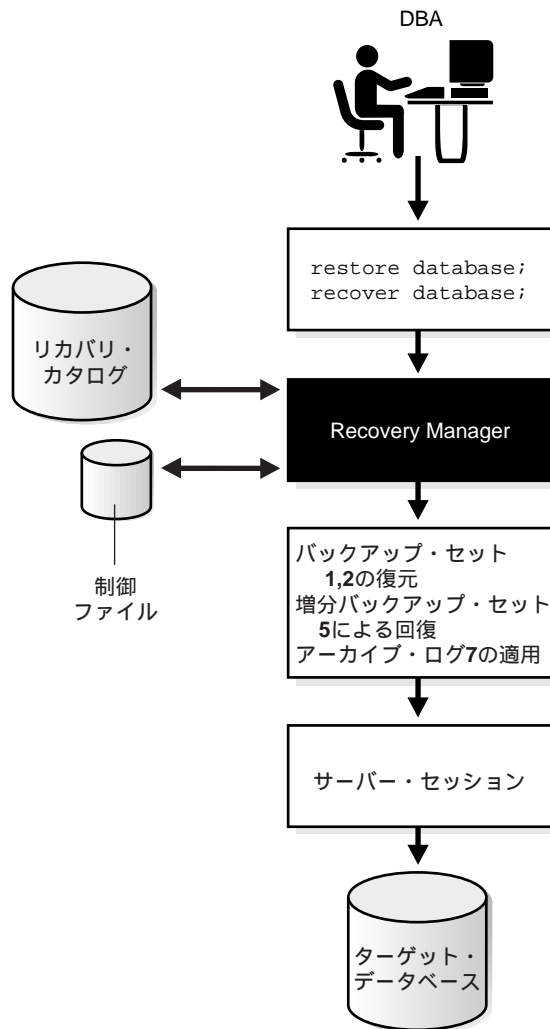
リカバリ・カタログは、できるだけメディア回復に使用できるようにしておいてください。使用できない場合、RMAN は、ターゲット・データベースの制御ファイルの情報を使用して回復を実行します。制御ファイルの回復が必要な場合は、リカバリ・カタログを使用可能にする必要があることに注意してください。リカバリ・カタログもターゲット・データベースの制御ファイルも使用できない場合、RMAN は操作を実行できません。

RMAN を使用した場合のメディア回復の基本的なステップは次のとおりです。

- 不完全回復を実行する場合は、**set until** コマンドを使用して、回復を終了する時刻、SCN またはログ順序番号を指定する。
- **restore** コマンドを使用して、必要なファイルを復元する。
- **recover** コマンドを使用して、データ・ファイルを回復する。

復元および回復の手順の詳細は、[第 9 章の「Recovery Manager による復元と回復」](#)を参照してください。

図 4-16 完全メディア回復の実行



関連項目：データ・ファイルの回復の方法は、9-15 ページの「[データ・ファイルの回復](#)」を参照してください。**recover** コマンドに関する参照情報は、11-89 ページの「[recover](#)」を参照してください。

増分バックアップおよび REDO レコードの適用

復元されたデータ・ファイルに対し、増分バックアップを適用するか、REDO を適用するか
の選択が必要になったときは、RMAN 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、RMAN は、最も長い
時間にわたっているものを自動的に選択します。

増分バックアップが見つからない場合、RMAN はアーカイブ REDO ログを探します。
ARCn が REDO ログをアーカイブすると、Oracle は即時にこれを制御ファイルに記録しま
す。Recovery Manager は、再同期化の際、この情報をリカバリ・カタログ内に反映し、
アーカイブ REDO ログをイメージ・コピーとして分類します。これらを表示するには **list** コ
マンドを使用します。

回復時、RMAN は、パラメータ・ファイルに指定されたデフォルトのディレクトリで、該
当するアーカイブ REDO ログを探します。これらがディスクで検索できない場合はバック
アップ・セットを検索し、さらに必要に応じてアーカイブ REDO ログを復元してメディア
回復を実行します。

デフォルトでは、RMAN は、`init.ora` ファイルに指定された現行のログ・アーカイブの
宛先にアーカイブ REDO ログを復元します。別の場所に復元する場合は **set archivelog
destination** コマンドを使用します。

関連項目 : **set archivelog destination** の構文は、11-134 ページの「**set_run_option**」を参照
してください。

不完全回復

RMAN は、完全回復も不完全回復も実行できます。**set until** コマンドを使用すると、時間、
SCN またはログ順序番号を制限値として指定した上で、不完全回復を実行できます。通常、
このコマンドは、**restore** コマンドおよび **recover** コマンドの発行前に使用します。不完全回
復を実行した後は、必ず **RESETLOGS** オプションを指定してデータベースを開き、その後す
ぐにデータベースをバックアップしてください。

表領域の Point-in-Time 回復

Recovery Manager には、自動化されている表領域の Point-in-Time 回復 (TSPITR) 機能が
あります。この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
は異なる時点で回復させることができます。RMAN の TSPITR は、次のような状況に最も有
効です。

- 誤って実行した、表の削除またはトランケート操作から回復する場合。
- 論理的に破損し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のみに影響を与えた他の
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物理データベースとは異なる時点で回復させる場合。
- 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 回復のみを実行すればよい場合。

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。

関連項目: RMAN を使用した TSPITR の実行方法は、[付録 A の「Recovery Manager による表領域の Point-in-Time 回復の実行」](#)を参照してください。

データベースの複製

バックアップおよび回復プロシージャの練習を行うテスト・データベースを作成するには、RMAN の **duplicate** コマンドを使用します。これは、ディスク上のプライマリ・データベースのファイルのバックアップ・セットを作成し、これを使用して新しいデータベースを作成するコマンドです。テスト・データベースは、本番データベースを常時稼働させておく必要がある場合に特に役立ちます。

複製操作の一環として、RMAN は次の処理を行います。

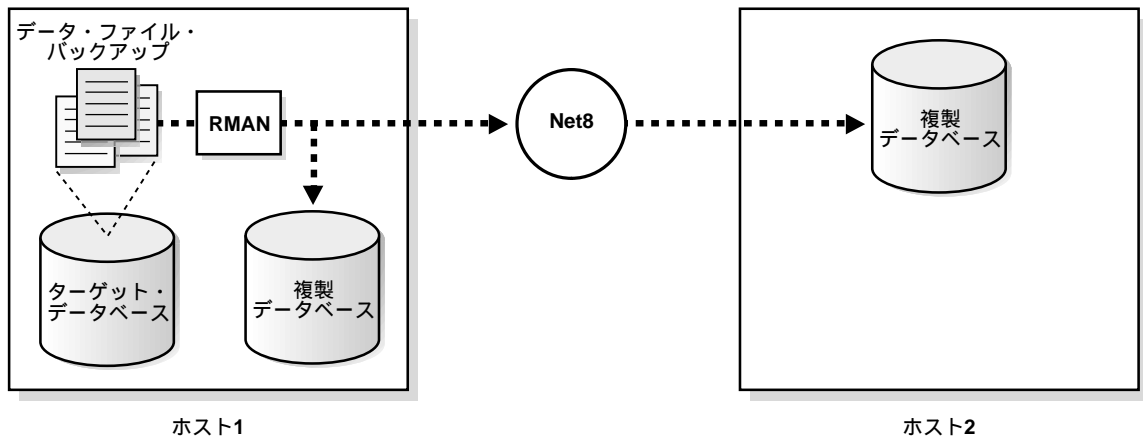
- ターゲット・データ・ファイルを復元して複製データベースを作成し、使用可能なアーカイブ REDO ログおよび増分バックアップをすべて使って不完全回復を実行する。
- RESETLOGS オプションを指定して、不完全回復を行った複製データベースを開き、オンライン REDO ログを作成する。
- その複製データベース用の、一意かつ新規のデータベース識別子を作成する。

RMAN による複製には次のような特徴もあります。次のことができます。

- **skip readonly** 句を使用して、読取り専用の表領域をスキップする。読取り専用の表領域は、デフォルトで含まれています。これらは削除しても、後で追加することが可能です。
- 複製データベースを新しいホストに作成する。同じディレクトリ構造であれば、**nofilenamecheck** オプションを使用すると、ターゲット・データ・ファイル名が複製データ・ファイルでも再使用できます。
- **set until** オプションを使用して複製データベースを作成すると、現行以外の状態に回復することができます。デフォルトでは、**duplicate** コマンドは、ターゲット・データベースの最新のバックアップを使用してデータベースを作成し、増分バックアップおよびアーカイブ REDO ログのバックアップに含まれた、一貫性のある最新の状態にまで回復するよう設定されています。
- リカバリ・カタログの有無に関係なく複製データベースを使用する。

- 複製データベースを、ターゲット・データベースと同じリカバリ・カタログに登録する。このオプションが可能なのは、複製時、複製データベースが新しいデータベース識別子を取得するためです。O/S ユーティリティを使用してターゲット・データベースをコピーした場合は、コピー・データベースの識別子が変わらないため、同じリカバリ・カタログに登録することはできません。

図 4-17 バックアップからの複製データベースの作成



複製データベースの作成の方法は、これを同じホストに作成するか、異なったホストに作成するか、また、複製ディレクトリの構造が、ターゲット・データベースのファイル・システムと同じかどうかで決まります。たとえば、同じディレクトリ構造およびファイル名を、複製データベースでも使用できる場合もあれば、**set newname** コマンドと **DB_FILE_NAME_CONVERT** 初期化パラメータの両方、あるいはその一方を使用して、ファイル名を再設定する必要がある場合もあります。

関連項目：複製データベースの作成の方法は、[第 10 章の「Recovery Manager による複製データベースの作成」](#)を参照してください。**duplicate** コマンド構文は、11-69 ページの「**duplicate**」を参照してください。

整合性チェック

Oracle では、バックアップが使用不能になったり、復元済みのデータ・ファイルが破損するような処理は実行できないようになっています。Oracle は、整合性チェックで次のことを行います。

- データベースの古いインカーネーションからバックアップを適用し、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情報を制御して、破損したバックアップ・ファイルを検出する。

物理ブロックの破損の検出

バックアップおよびコピーを実行しているのは Oracle サーバー・セッションであり、サーバー・セッション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 `alert.log` の中に記録されます。

RMAN は、バックアップ操作の完了時に破損情報について問い合わせ、リカバリ・カタログおよび制御ファイルにその情報を格納します。このデータにアクセスするには、`V$BACKUP_CORRUPTION` and `V$COPY_CORRUPTION` ビューを使用します。

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・ブロックを RMAN がバックアップ中に見つけると、サーバー・セッションはその破損ブロックをバックアップにコピーします。Oracle は、論理破損またはメディア破損のどちらかとしてこの破損を制御ファイルにレポートします。

破損ヘッダーが付いているが、まだ破損のマークはデータベースによって付けられていないデータ・ファイル・ブロックを RMAN が見つけた場合、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、そのブロックをバックアップに書き込みます。

注意： RMAN は、すべてのタイプの破損を検出できるわけではありません。

論理ブロックの破損の検出

RMAN は、物理破損チェックに合格したデータ・ブロックと索引ブロックの論理破損テスト機能を持っています（たとえば行ピースまたは索引エントリの破損）。論理破損を検出した場合、RMAN は、そのブロックを `alert.log` およびサーバー・セッションのトレース・ファイルに記録します。

あるファイルで検出された物理破損と論理破損の合計が、`maxcorrupt` に設定された値以下である場合、RMAN のコマンドは完了し、Oracle は `V$BACKUP_CORRUPTION` および `V$COPY_CORRUPTION` に破損ブロックの範囲を記録します。`maxcorrupt` を超過すると、コマンドはビューに記録を行わずに終了します。

注意： `copy` コマンドおよび `backup` コマンドの場合、`maxcorrupt` の設定は、ファイルが許容する物理破損と論理破損の合計数を表します。

オープン・バックアップ時の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、表領域を **ホット・バックアップ・モード (hot backup mode)** にしておく必要があります。それは、DBW_n により書き込み中であるブロックを、オペレーティング・システムがバックアップ用に読み取ってしまう可能性があるからです。このようなブロックを **分裂ブロック (fractured block)** と呼びます。

RMAN 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・セッションがデータ・ファイルを読み取ります。Oracle サーバー・セッションはすべての 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。このセッションで分裂ブロックが検出された場合、そのブロックは再読み込みされます。このため、Recovery Manager を使用してデータ・ファイルのバックアップまたはコピーを行う場合は、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： ホット・バックアップ・モードの詳細は、13-6 ページの「**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**」を参照してください。

Recovery Manager のスタート・ガイド

この章では、RMAN を使用した作業開始の方法を説明します。項目は次のとおりです。

- [Recovery Manager のセットアップ](#)
- [制御ファイルを RMAN メタデータに使用した場合](#)
- [RMAN への接続](#)
- [基本的な RMAN コマンドの使用方法](#)
- [メディア・マネージャの構成](#)
- [サンプル・スクリプトの使用法および使用例](#)

Recovery Manager のセットアップ

RMAN を使用する前に、次のことを決める必要があります。

- 認証にパスワード・ファイルを使用するか。
- NLS 環境変数をどのように設定するか。
- スナップショット制御ファイルをどこに保管するか。

パスワード・ファイルの使用

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、特に次のいずれかを行う場合には、パスワード・ファイルを使用する必要があります。

- RMAN をリモートで、つまりターゲット・データベースとは異なるマシンで実行する。
- データベース接続文字列のサービス名を指定して RMAN を使用する。
- OPS 環境でデータベースを実行しているときに、RMAN セッションを 1 つしか使用せずに、クラスタ内の複数のノードから同時にこのデータベースをバックアップする。

注意： Recovery Manager は、初期化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。バックアップおよび回復計画を立案する場合は、メディア障害からこれらのファイルを保護する方法も検討する必要があります。

関連項目： 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は、『Oracle8i Parallel Server 概要および管理』を参照してください。

NLS 環境変数の設定

RMAN 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。これらの変数によって、**restore**、**recover**、**report** などの、RMAN コマンドの時間を指定するパラメータで使用される形式が決まります。

次の例は、言語および日付書式の代表的な設定を示しています。

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

RMAN コマンドの日付の指定

RMAN コマンドで日付を指定する場合、次のいずれかの日付文字列を選択できます。

- NLS_DATE_FORMAT の設定に一致するフォーマットのリテラル文字列

- 'SYSDATE-10' または "TO_DATE('01/30/1997', 'MM/DD/YYYY')" など、データ・タイプが DATE 型の SQL 式。2 番めの例では、専用の日付書式マスクが含まれているため、現行の NLS_DATE_FORMAT 設定とは関係がないことに注意してください。

RMAN における日付設定の代表例を次に示します。

```
backup archivelog from time 'SYSDATE-31' until time 'SYSDATE-14';
restore database until time "TO_DATE('12/20/98', 'MM/DD/YY')";
```

データベース・キャラクタ・セットの指定

RMAN を使用してマウント状態でないデータベースに接続し、RMAN を接続した状態のまま後からデータベースをマウントする場合は、データベースが使用するキャラクタ・セットを指定するよう NLS_LANG 変数を設定します。

マウントされていないデータベースでは、デフォルトのキャラクタ・セット、すなわち US7ASCII が使用されます。使用しているキャラクタ・セットがこのデフォルト設定と異なる場合、RMAN はデータベースのマウント後、エラーを戻します。この問題を回避するには、ターゲット・データベースのキャラクタ・セットを指定するよう NLS_LANG を設定します。たとえば、キャラクタ・セットが WE8DEC である場合は、NLS_LANG パラメータを次のように設定します。

```
NLS_LANG=american_america.we8dec.
```

注意： NLS_LANG と NLS_DATE_FORMAT を、使用する NLS_DATE_FORMAT 用に設定してください。

関連項目： NLS_LANG パラメータおよび NLS_DATE_FORMAT パラメータの詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。

スナップショットの制御ファイル位置の決定

RMAN 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、[スナップショット制御ファイル \(snapshot control file\)](#) が作成されます。スナップショット制御ファイルのデフォルトの名前は、ポートによって異なります。スナップショット制御ファイルの名前を変更するには、**set snapshot controlfile name** コマンドを使用します。このコマンドに指定した名前が、RMAN でそれ以降作成されるスナップショット制御ファイルで使用されます。

たとえば、RMAN を起動し次のように入力します。

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

スナップショット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。この処理は、クラスタ内の複数のインスタンスが RMAN 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・プロセスが、名前と位置が同じスナップショット制御ファイルを作成できなければならないからです。たとえば、次のように入力します。

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

ある RMAN のジョブで制御ファイルのバックアップをすでに開始していても、別のジョブで新しいスナップショット制御ファイルを作成する必要がある場合、次のようなメッセージが表示されることがあります。

```
RMAN-08512: waiting for snapshot controlfile enqueue
```

通常、ジョブは、制御ファイルのエンキューを待機する必要がある場合、そのエンキューを取得するまでに多少の時間を要します。Recovery Manager は、エンキューの取得を最高 5 回試行した後、ジョブを停止します。通常、このような競合は、制御ファイルのバックアップを実行するジョブが 2 つあるときに発生します。この場合、先に制御ファイルのバックアップを開始した方が、メディア・マネージャのサービスを待機します。

関連項目：スナップショット制御ファイルを使用した、RMAN の同期化の概要は、4-13 ページの「[リカバリ・カタログの再同期化](#)」を参照してください。set コマンド構文は、11-130 ページの「[set](#)」を参照してください。

リカバリ・カタログの使用・不使用の決定

RMAN の使用を開始する際、最も重要な決定であると考えられるのは、RMAN のメタデータ格納にリカバリ・カタログを使用するかどうかということです。この項では、リカバリ・カタログを使用する場合と使用しない場合の、それぞれのコストと利点を説明します。カタログを作成する場合は、6-2 ページの「[リカバリ・カタログの作成](#)」を参照してください。

関連項目：RMAN メタデータの概要は、4-12 ページの「[Recovery Manager のメタデータ](#)」を参照してください。

リカバリ・カタログを RMAN メタデータに使用した場合

リカバリ・カタログを使用した場合、RMAN では、自動化されるバックアップおよび回復機能が増えます。このため、できるだけ RMAN でリカバリ・カタログを使用することをお勧めします。

リカバリ・カタログを使用する場合、RMAN では、リカバリ・カタログのスキーマと、そのスキーマで使用される関連の領域をメンテナンスする必要があります。リカバリ・カタログのスキーマのサイズは次のようになります。

- カatalogで管理するデータベースの数に依存する。
- カatalogに格納された Recovery Manager スクリプトの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数の増加に従って、増加する。

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および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。RMAN を使用して複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 RMAN 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。実際には ARCHIVELOG モードで運用することをお勧めします。

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。

- システム表領域
- 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・ファイル

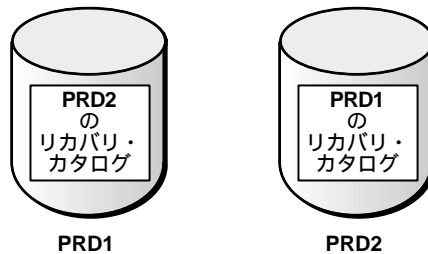
リカバリ・カタログを個別にメンテナンスした場合、DBA の裁量でそのデータベースを停止できるという別の利点があります。このデータベースで使用する領域のほとんどは、表領域 (SYSTEM、一時、ロールバックなど) のサポートに使用されます。

表 5-1 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 REDO 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 つの本番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。

図 5-1 リカバリ・カタログ・データベースとしての本番データベースの使用



本番データベースを他方のリカバリ・カタログとして機能させることにより、リカバリ・カタログ用の個別のデータベースを維持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： リカバリ・カタログのスキーマは、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てください。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：リカバリ・カタログの管理方法は、[第 6 章の「Recovery Manager のメタデータの管理」](#)を参照してください。

制御ファイルを RMAN メタデータに使用した場合

リカバリ・カタログを使用しない場合でも、RMAN を使用して処理を非常に効果的に実行することができます。RMAN は、必要な情報をターゲット・データベースの制御ファイルから取得します。リカバリ・カタログを使用しない場合、次のコマンドは使用できません。

- `change ... available`、`change ... unavailable`、`change ... uncatalog`、`change backupset ... crosscheck`、`change backuppiece ... crosscheck`
- `create catalog`、`upgrade catalog`、`drop catalog`
- `create script`、`delete script`、`replace script`、`print script`
- `crosscheck backup`

- delete expired backup
- list incarnation
- register database
- report schema at time
- reset database
- restore (使用できる制御ファイルがない場合)
- resync catalog
- set auxname

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。

- 最低 3 つの多重化制御ファイル、またはミラー化制御ファイルを、それぞれ別々のディスクで使用する。
- 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・ピースの名前の記録を確実に保存しておく。バックアップおよびコピーに関する情報を取得するには、**list** コマンドを使用します。RMAN のバックアップ・ログはすべて保存しておいてください。

警告： リカバリ・カタログを使用していないときに、すべての制御ファイルが消失した場合、復元および回復を可能にする唯一の手段は、オラクル社カスタマ・サポートに連絡することです。カスタマ・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たファイルの名前
 - 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
-
-

関連項目：制御ファイルを RMAN メタデータの専用リポジトリとして管理する方法の詳細は、6-36 ページの「[リカバリ・カタログを使用しない RMAN の管理方法](#)」を参照してください。

RMAN への接続

RMAN を使用するには、最初にこれに接続する必要があります。RMAN への接続は以下の操作のために必要です。

- 有効なユーザーであることを認証する。
- ターゲット・データベース、バックアップまたは復元の対象となるデータベースを指定する。
- リカバリ・カタログが含まれたデータベースを指定する（リカバリ・カタログを使用する場合）。
- 補助データベースを指定する（補助データベースを使用する場合）。

RMAN を起動する場合は、ターゲット・データベースに接続する必要があります。その場合、RMAN への接続方法は数種類あります。たとえば、RMAN は次のような方法で起動できます。

- リカバリ・カタログを使用して、または使用しないで起動する。
- O/S コマンド行または RMAN コマンド行でデータベースに接続して起動する。
- バッチ・モード（一連の RMAN コマンドで構成されるコマンド・ファイルを使用する）または対話モードで起動する。
- バッチ・モードで実行する場合、RMAN 出力が記録されるログ・ファイルを使用して起動する。
- ログ・ファイルへの追加、または同ファイルの上書きを行うことで起動する。

この項では、Recovery Manager の接続方法のサンプルを示します。

- [リカバリ・カタログを使用せずに RMAN に接続する場合](#)
- [リカバリ・カタログを使用して RMAN に接続する場合](#)
- [補助データベースへの接続](#)
- [RMAN からの切断](#)

関連項目：コマンド行オプションの一覧は、11-39 ページの「[cmdLine](#)」を参照してください。

リカバリ・カタログを使用せずに RMAN に接続する場合

これらの例では、次のことを前提としています。

<code>sys</code>	SYSDBA 権限を持ったユーザー
<code>target_pwd</code>	ターゲット・データベースの <code>orapwd</code> ファイルに指定されている、SYSDBA として接続するためのパスワード
<code>target_str</code>	ターゲット・データベースのネット・サービス名

O/S 認証を使用した RMAN への接続

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、ユーザー名が有効かどうかを、O/S 認証を使用して検証する必要があります。O/S 認証が使用できるのは、ローカルで接続する場合のみです。つまり、RMAN とターゲット・データベースが同じマシンに常駐していることが前提となります。

1. UNIX のコマンド行で、次のように入力する。

```
% ORACLE_SID=PROD1; export ORACLE_SID
```

2. 次の文を発行します。

```
% rman nocatalog
```

3. RMAN プロンプトで、次のように入力する。

```
RMAN> connect target /
```

注意： SYSDBA オプションは、RMAN が自動的に使用するので、指定する必要はありません。

パスワード・ファイルを使用した RMAN への接続

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、パスワードを使用して接続できます。パスワード・ファイルは、ローカル・アクセスにもリモート・アクセスにも使用します。ネット・サービス名を使用してリモートで接続する場合は、パスワード・ファイルを使用する必要があります。

O/S コマンド行からの接続 O/S コマンド行から接続する場合は、次のように入力します。ここでは、`sys_pwd` は SYS のパスワード、`target_str` はターゲット・データベースのネット・サービス名をそれぞれ示します。

```
% rman target sys/target_pwd@target_str nocatalog
```

RMAN プロンプトからの接続 RMAN を起動し、RMAN プロンプトからターゲット・データベースに接続するという方法もあります。

```
% rman nocatalog
RMAN> connect target sys/target_pwd@target_str
```

関連項目 : コマンド行のオプションの詳細は、11-39 ページの「[cmdLine](#)」を参照してください。**connect** コマンドの詳細は、11-44 ページの「[connect](#)」を参照してください。パスワード・ファイルの詳細は、『Oracle8i 管理者ガイド』を参照してください。

リカバリ・カタログを使用して RMAN に接続する場合

これらの例では、リカバリ・カタログをメンテナンスしていることの他、次のことを前提としています。

<i>sys</i>	SYSDBA 権限を持ったユーザー
<i>rman</i>	RECOVERY_CATALOG_OWNER 権限を持っているリカバリ・カタログの所有者
<i>target_pwd</i>	ターゲット・データベースの <i>orapwd</i> ファイルに指定されている、SYSDBA として接続するためのパスワード
<i>target_str</i>	ターゲット・データベースのネット・サービス名
<i>cat_pwd</i>	ターゲット・データベースの <i>orapwd</i> ファイルに指定されている、ユーザー RMAN 用パスワード
<i>cat_str</i>	リカバリ・カタログ・データベースのネット・サービス名

O/S 認証を使用した RMAN への接続

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、ユーザー名が有効かどうかを、O/S 認証を使用して検証する必要があります。O/S 認証が使用できるのは、ローカルで接続する場合のみです。つまり、RMAN とターゲット・データベースが同じマシンに常駐していることが前提となります。

1. RMAN をターゲット・データベースと同一マシンで実行する場合は、環境変数 `ORACLE_SID` をターゲット・データベースに設定する。たとえば、UNIX プロンプトで、次のように入力します。

```
% ORACLE_SID=PROD1; export ORACLE_SID
```

2. 次の文を発行し、ユーザー RMAN としてリカバリ・カタログに接続する。

```
% rman catalog rman/cat_pwd@cat_str
```

3. RMAN が起動したら、**connect target** コマンドを発行する (SYSDBA 権限を持っている場合)。

```
RMAN> connect target
```

パスワード・ファイルを使用した RMAN への接続

ターゲット・データベースおよびリカバリ・カタログ・データベースにパスワード・ファイルがある場合、パスワードを使用して接続できます。パスワード・ファイルは、ローカル・アクセスにもリモート・アクセスにも使用します。ネット・サービス名を使ってリモートで接続する場合は、パスワード・ファイルを使用する必要があります。

O/S コマンド行からの接続 O/S コマンド行から RMAN に接続するには、次のように入力します。

```
% rman target sys/target_pwd@target_str catalog rman/cat_pwd@cat_str
```

RMAN プロンプトからの接続 RMAN を起動し、RMAN プロンプトからターゲット・データベースに接続するという方法もあります。

```
% rman
RMAN> connect target sys/target_pwd@target_str
RMAN> connect catalog rman/cat_pwd@cat_str
```

補助データベースへの接続

duplicate コマンドを使用する場合、または RMAN TSPITR を実行する場合は、補助データベースに接続する必要があります。これらの例では、次のことを前提としています。

<i>sys</i>	SYSDBA 権限を持ったユーザー
<i>rman</i>	RECOVERY_CATALOG_OWNER 権限を持っているリカバリ・カタログの所有者
<i>target_pwd</i>	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、SYSDBA として接続するためのパスワード
<i>target_str</i>	ターゲット・データベースのネット・サービス名
<i>cat_pwd</i>	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、ユーザー RMAN 用パスワード
<i>cat_str</i>	リカバリ・カタログ・データベースのネット・サービス名
<i>aux_pwd</i>	補助データベースの orapwd ファイルに指定されている、SYSDBA として接続するためのパスワード
<i>aux_str</i>	補助データベースのネット・サービス名

補助データベースにパスワード・ファイルがある場合、パスワードを使用して接続できます。パスワード・ファイルは、ローカル・アクセスにもリモート・アクセスにも使用します。ネット・サービス名を使ってリモートで接続する場合は、パスワード・ファイルを使用する必要があります。

O/S コマンド行からの接続 O/S コマンド行から補助インスタンスに接続するには、次のように入力します。

```
% rman auxiliary sys/aux_pwd@aux_str
```

ターゲット、補助およびリカバリ・カタログの各データベースに接続するには、次に示すように（すべて 1 行で）発行します。

```
% rman target sys/target_pwd@target_str catalog rman/cat_pwd@cat_str \  
> auxiliary sys/aux_pwd@aux_str
```

RMAN プロンプトからの接続 RMAN を起動し、RMAN プロンプトから補助データベースに接続するという方法もあります。

```
% rman  
RMAN> connect auxiliary sys/aux_pwd@aux_str
```

ターゲット、補助およびリカバリ・カタログの各データベースに接続するには、次に示すように発行します。

```
% rman  
RMAN> connect target sys/target_pwd@target_str  
RMAN> connect catalog rman/cat_pwd@cat_str  
RMAN> connect auxiliary sys/aux_pwd@aux_str
```

関連項目 : **duplicate** コマンド構文は、11-69 ページの「[duplicate](#)」を参照してください。RMAN TSPITR 回復の実行方法は、[付録 A の「Recovery Manager による表領域の Point-in-Time 回復の実行」](#)を参照してください。

RMAN からの切断

RMAN から切断するには、RMAN プロンプトで **exit** と入力します。

```
RMAN> exit
```

基本的な RMAN コマンドの使用法

RMAN への接続方法がわかれば、バックアップ、回復およびメンテナンスの各操作はすぐに開始できます。この項の例を使って基本的なコマンドを試してみてください。

これらの例では、次の事柄を前提としています。

- ARCHIVELOG モードで運用する。

- ターゲット・データベースと同じマシンで RMAN を実行する。
- O/S 認証を使用してコマンド行から接続する。
- OPS 構成は実行しない。

次のタスクの実行方法がわかります。

- [RMAN への接続](#)
- [データベースのマウント](#)
- [現行スキーマのレポート](#)
- [データ・ファイルのコピー](#)
- [表領域のバックアップ](#)
- [バックアップおよびコピーのリスト表示](#)
- [復元の妥当性チェック](#)

RMAN への接続

最初のタスクはターゲット・データベースへの接続です。リカバリ・カタログを作成していた場合も、同じように接続できます。ただし、ここに示す例では、リカバリ・カタログは使用せずに接続します。

コマンド行で、次のように入力します。

```
% rman target / nocatalog
```

データベースがすでにマウントまたはオープンされている場合は、次のような出力が表示されます。

```
Recovery Manager: Release 8.1.5.0.0
```

```
RMAN-06005: connected to target database: RMAN (DBID=1237603294)  
RMAN-06009: using target database controlfile instead of recovery catalog
```

ここに表示されている DBID とは、データベースのデータベース識別子です。

データベースが起動していない場合、RMAN は接続時に次のメッセージを表示します。

```
RMAN-06193: connected to target database (not started)
```

関連項目：コマンド行の接続オプションは、11-39 ページの「[cmdLine](#)」を参照してください。

データベースのマウント

リカバリ・カタログを使用しない場合、RMAN はターゲット・データベースの制御ファイルから必要な情報を取得する必要があります。そのため、データベースはマウントまたはオープンする必要があります。ここに示す例では、データベースはマウントしますがオープンしません。

データベースが起動しない場合は、**startup** コマンドを発行し、パラメータ・ファイルを指定します（デフォルトの位置にない場合）。この例では、initPROD1.ora というパラメータ・ファイルを使用します。

```
RMAN> startup mount pfile=/oracle/dbs/temp/initPROD1.ora
```

```
RMAN-06196: Oracle instance started
RMAN-06199: database mounted
```

```
Total System Global Area      19799144 bytes

Fixed Size                      64616 bytes
Variable Size                  11001856 bytes
Database Buffers               8192000 bytes
Redo Buffers                   540672 bytes
```

```
RMAN>
```

RMAN は、データベースおよび REDO バッファのサイズなど、SGA のサイズを表示した後、RMAN プロンプトに戻ります。

データベースがオープンしている場合は、次のように発行して正しく停止した後マウントします。

```
RMAN> shutdown immediate
```

```
RMAN-06405: database closed
RMAN-06404: database dismounted
RMAN-06402: Oracle instance shut down
```

```
RMAN> startup mount pfile = initPROD1.ora # specify a parameter file if necessary
```

関連項目：startup の構文は、11-143 ページの「[startup](#)」を参照してください。

現行スキーマのレポート

この例では、ターゲット・データベースに含まれているデータ・ファイルをレポートするよう RMAN に指示します。次のように、**report schema** コマンドを使用します。

```
RMAN> report schema;
```

RMAN は、現行データベースに存在するデータ・ファイルを表示します。データベースの内容によって異なりますが、次のような出力が表示されます。

```

RMAN-03022: compiling command: report
Report of database schema
File K-bytes    Tablespace          RB segs Name
-----
1          47104 SYSTEM              ***    /oracle/dbs/tbs_01.f
2           978 SYSTEM              ***    /oracle/dbs/tbs_02.f
3           978 TBS_1                ***    /oracle/dbs/tbs_11.f
4           978 TBS_1                ***    /oracle/dbs/tbs_12.f
5           978 TBS_2                ***    /oracle/dbs/tbs_21.f
6           978 TBS_2                ***    /oracle/dbs/tbs_22.f
7           500 TBS_1                ***    /oracle/dbs/tbs_13.f
8           500 TBS_2                ***    /oracle/dbs/tbs_23.f
9           500 TBS_2                ***    /oracle/dbs/tbs_24.f
10          5120 SYSTEM              ***    /oracle/dbs/tbs_03.f
11          2048 TBS_1                ***    /oracle/dbs/tbs_14.f
12          2048 TBS_2                ***    /oracle/dbs/tbs_25.f

```

関連項目 : `report` の構文は、11-103 ページの「[report](#)」を参照してください。リストおよびレポートの作成の詳細は、第 7 章の「[Recovery Manager によるリストおよびレポートの生成](#)」を参照してください。

データ・ファイルのコピー

この例では、datafile 1 を新しいディレクトリにコピーします。コピーするデータ・ファイルはこのデータ・ファイルでなくても構いません。この例では、ディスク・チャネル c1 を割り当て、df1.bak というデータ・ファイル・コピーを作成します。

次のように、`copy` コマンドを使用します。

```

RMAN> run {
2> allocate channel c1 type disk;
3> copy datafile 1 to 'df1.bak';
4> }

```

次のような出力が表示されます。

```

RMAN-03022: compiling command: allocate
RMAN-03023: executing command: allocate
RMAN-08030: allocated channel: c1
RMAN-08500: channel c1: sid=12 devtype=DISK

RMAN-03022: compiling command: copy
RMAN-03023: executing command: copy
RMAN-08000: channel c1: copied datafile 1
RMAN-08501: output filename=/oracle/dbs/df1.bak recid=3 stamp=352381826
RMAN-08031: released channel: c1

```

RMAN-08000 は、コピーが成功したことを示すメッセージです。RMAN は、出力コピーの絶対パスによるファイル名をメッセージ RMAN-08501 に表示することに注意してください。

関連項目 : `copy` の構文は、11-48 ページの「[copy](#)」を参照してください。`allocate` の構文は、11-9 ページの「[allocate](#)」を参照してください。イメージ・コピーの作成の詳細は、8-10 ページの「[イメージ・コピーの作成](#)」を参照してください。

表領域のバックアップ

この例では、SYSTEM 表領域をディスクにバックアップします。別のオブジェクトをバックアップした場合も同様です。

ここでは、この表領域をバックアップのデフォルト・ディレクトリにバックアップします。そのディレクトリはポートによって異なります。UNIX システムでは、そのディレクトリは `$ORACLE_HOME/dbs` となります。`format` パラメータは指定しないので、RMAN によって自動的にそのバックアップに一意のファイル名が割り当てられます。

次のように、`backup` コマンドを使用します。

```
RMAN> run {  
2> allocate channel c1 type disk;  
3> backup tablespace system;  
4> }
```

次のような出力が表示されます。

```
RMAN-03022: compiling command: allocate  
RMAN-03023: executing command: allocate  
RMAN-08030: allocated channel: c1  
RMAN-08500: channel c1: sid=12 devtype=DISK  
  
RMAN-03022: compiling command: backup  
RMAN-03023: executing command: backup  
RMAN-08008: channel c1: starting full datafile backupset  
RMAN-08502: set_count=1 set_stamp=352382211 creation_time=18-DEC-98  
RMAN-08010: channel c1: specifying datafile(s) in backupset  
RMAN-08522: input datafile fno=00001 name=/vobs/oracle/dbs/tbs_01.f  
RMAN-08011: including current controlfile in backupset  
RMAN-08522: input datafile fno=00016 name=/oracle/dbs/tbs_03.f  
RMAN-08522: input datafile fno=00002 name=/oracle/dbs/tbs_02.f  
RMAN-08013: channel c1: piece 1 created  
RMAN-08503: piece handle=/oracle/dbs/lhaglr83_1_1 comment=NONE  
RMAN-08525: backup set complete, elapsed time: 00:00:27  
RMAN-08031: released channel: c1
```

RMAN-08525 は、バックアップ・セットの作成に RMAN が成功したことを示すメッセージです。RMAN は、バックアップ・ピースの絶対パスによるファイル名をメッセージ RMAN-08503 に表示することに注意してください。

関連項目 : `backup` の構文は、11-21 ページの「[backup](#)」を参照してください。イメージ・コピーの作成の詳細は、8-2 ページの「[バックアップの作成](#)」を参照してください。

バックアップおよびコピーのリスト表示

この例では、バックアップ・セットおよびイメージ・コピーをリスト表示します。次のように、**list** コマンドを発行します。

```
RMAN> list backup;
```

次のような出力が表示されます。

```
List of Backup Sets
Key       Recid       Stamp       LV Set Stamp Set Count Completion Time
-----
3         3           352382231  0 352382211 49      18-DEC-98

List of Backup Pieces
Key       Pc# Cp# Status       Completion Time       Piece Name
-----
2         1  1  AVAILABLE  18-DEC-98             /oracle/dbs/1hag1r83_1_1

List of Datafiles Included
File Name                               LV Type Ckp SCN    Ckp Time
-----
1      /oracle/dbs/tbs_01.f                0 Full 114149 18-DEC-98
2      /oracle/dbs/tbs_02.f                0 Full 114149 18-DEC-98
16     /oracle/dbs/tbs_03.f                0 Full 114149 18-DEC-98
```

RMAN は、作成したバックアップ・セットおよびバックアップピースを表示する他、これらのセットに含めたデータ・ファイルを表示します。

次にイメージ・コピーをリスト表示します。

```
RMAN> list copy;
```

次のような出力が表示されます。

```
List of Datafile Copies
Key       File S Completion time Ckp SCN    Ckp time       Name
-----
2         1  A 18-DEC-98          114148        18-DEC-98      /oracle/dbs/df1.bak
```

関連項目 : **list** 出力における列見出しの詳細は、11-76 ページの「[list](#)」を参照してください。
リストおよびレポートの作成の詳細は、7-2 ページの「[リストの生成](#)」を参照してください。

復元の妥当性チェック

最後に、メディア障害に備え、バックアップの復元が可能かどうかをチェックします。**list backup** コマンドの出力を使用して、バックアップ・セットの主キーを確認します。

List of Backup Sets						
Key	Recid	Stamp	LV	Set Stamp	Set Count	Completion Time

3	3	352382231	0	352382211	49	18-DEC-98

この例では、主キーは3となります。バックアップ・セットの主キーを **validate backupset** コマンドで使用します。

```
RMAN> run {
2> allocate channel c1 type disk;
3> validate backupset 3;
4> }
```

次のような出力が表示されます。

```
RMAN-03022: compiling command: allocate
RMAN-03023: executing command: allocate
RMAN-08030: allocated channel: c1
RMAN-08500: channel c1: sid=12 devtype=DISK

RMAN-03022: compiling command: validate
RMAN-03023: executing command: validate
RMAN-08096: channel c1: starting validation of datafile backupset
RMAN-08502: set_count=49 set_stamp=352382211 creation_time=18-DEC-98
RMAN-08023: channel c1: restored backup piece 1
RMAN-08511: piece handle=/oracle/dbs/lhagl83_1_1 params=NULL
RMAN-08098: channel c1: validation complete
RMAN-08031: released channel: c1
```

エラー・メッセージがない場合は、バックアップ・セットの復元が可能なことを RMAN が確認したことを意味します。エラーがある場合、RMAN は必ずエラー・バナーを表示し、エラーの性質を示したメッセージを表示します。

たとえば、接続されていないにもかかわらず、ターゲット・データベースでチャンネルの割当てを行おうとした場合は、次のように表示されます。

```
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure during compilation of command
RMAN-03013: command type: allocate
RMAN-06171: not connected to target database
```

関連項目 : `validate` の構文は、11-151 ページの「[validate](#)」を参照してください。バックアップおよびコピーの復元方法は、9-2 ページの「[データ・ファイル、制御ファイルおよびアーカイブ REDO ログの復元](#)」を参照してください。

メディア・マネージャの構成

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、次の項目について説明します。

- [メディア・マネージャとのリンク](#)
- [一意的なファイル名の生成](#)
- [ファイル・サイズの制限設定](#)
- [デバイス固有文字列のメディア・マネージャへの送信](#)

関連項目 : メディア管理の概要は、4-15 ページの「[メディア管理](#)」を参照してください。

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、次のことを実行してください。

- メディア・マネージャのソフトウェアおよびハードウェアのインストールと構成を行う。
- ベンダーからメディア管理ライブラリ (MML) インタフェース・ソフトウェアの提供を受け、Oracle Server にリンクする。この統合により、Oracle サーバー・セッションがメディア・マネージャを呼び出すことができますようになります。

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・システム固有の 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。

一意的なファイル名の生成

バックアップをメディア・マネージャに書き込む場合、RMAN が提供する置換変数を使用する必要があります (変数の一覧は 11-21 ページの「[backup](#)」を参照)。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フォーマット文字列によって判別されます。

`format` パラメータを指定しない場合、RMAN は %U 置換変数を使用して、一意的なファイル名を自動的に生成します。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内で一意の名前にしてください。

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。

ファイル・サイズの制限設定

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。RMAN は複数のデータ・ファイルを 1 つのファイルに多重化しますが、このバックアップ・ピースのサイズが、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ような状況では、ファイル・サイズが問題となります。

問題を回避するには、メディア管理ソフトウェアのマニュアルを参照し、運用時のファイル・サイズ制限を確認してください。RMAN からファイルを書き出す場合は、その制限を超えないようにしてください。バックアップ・ピースのファイル・サイズを制限するには、**set limit channel** コマンドで **kbytes** パラメータを使用します。\$ORACLE_HOME/rdbms/demo ディレクトリに、スクリプトの例があります。

関連項目：run コマンドに影響を及ぼすさまざまなパラメータについては、11-134 ページの「[set_run_option](#)」を参照してください。

デバイス固有文字列のメディア・マネージャへの送信

ベンダー固有の引用符付き文字列を、メディア管理ソフトウェアに送信するには、**send** コマンドを使用します。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、サポートされているコマンドを確認してください。使用できるものは次のとおりです。

- **send** コマンド。他のオペランドを使用せずに、割当て済みの全チャンネルに文字列を送信できます。
- **send device type** コマンド。指定したデバイス・タイプの全チャンネルに文字列を送信できます。
- **send channel** コマンド。コマンドに指定したチャンネルに文字列を送信できます。

関連項目：send コマンドの構文は、11-128 ページの「[send](#)」を参照してください。

メディア・マネージャのトラブルシューティング

メディア管理におけるトラブルシューティングを円滑化するため、Oracle には sbttest というクライアント・プログラムが付属しています。このプログラムは、RMAN にリンクしてテープへのバックアップを実行しますが、スタンドアロンでメディア管理ソフトウェアのテストを行うこともできます。バンドルされた Legato Storage Manager を使用しても、他のベンダーのメディア管理製品を使用しても、Oracle でバックアップの作成、復元ができなくなった場合、このプログラムを使用します。sbttest プログラムを使用する場合は、必ずオラクル社カスタマ・サポートの指示に従ってください。

サンプル・スクリプトの使用方法および使用例

\$ORACLE_HOME/rdbms/demo サブ・ディレクトリには（位置はオペレーティング・システムによって異なります）RMAN のサンプル・スクリプトが多数含まれています。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、十分な説明もなされています。これらのファイルは編集でき、サイトにあわせてカスタマイズできます。

最初のファイルには、データベースのバックアップ、復元および回復を実行するスクリプトが多数含まれています。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います。スクリプトのバックアップ、復元および回復を実行するにあたっての基礎として使用してください。その他のファイルには、バックアップ、回復または複製の実例が示されています。

コマンド・ファイルは、O/S コマンド行または RMAN プロンプトから実行します。リカバリ・カタログを使用している場合は、**create script** コマンドを使用してスクリプトを作成し、これを **run** コマンド内で実行する方法も選択できます。

関連項目 : RMAN のプロンプトからコマンド・ファイルを実行する方法は、11-122 ページの「[rmanCmd](#)」を参照してください。コマンド行からコマンド・ファイルを実行する方法は、11-39 ページの「[cmdLine](#)」を参照してください。ストアド・スクリプトの作成および実行の方法は、6-21 ページの「[リカバリ・カタログへのスクリプトの格納](#)」を参照してください。

Recovery Manager のメタデータの管理

この章では、RMAN のメタデータの管理方法を説明します。RMAN の使用方法によっては、このデータをリカバリ・カタログに格納することも、制御ファイルのみに格納することも可能です。この章では、次の項目について説明します。

- リカバリ・カタログの作成
- RMAN メタデータのメンテナンス
- リカバリ・カタログへのスクリプトの格納
- リカバリ・カタログのバックアップと回復
- リカバリ・カタログのアップグレード
- リカバリ・カタログの削除
- リカバリ・カタログを使用しない RMAN の管理方法

リカバリ・カタログの作成

リカバリ・カタログを使用するには、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・カタログ専用の表領域に設定することをお勧めしますが、必要に応じて、SYSTEM システム表領域にインストールすることもできます。

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ます。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、他方も失う可能性があります。

これ以降に示す例では、次の事柄を前提とします。

- パスワード CHANGE_ON_INSTALL のユーザー SYS が、リカバリ・カタログ・データベース RCAT の SYSDBA 権限を有する。
- リカバリ・カタログ・データベース RCAT に、RCVCAT という表領域があり、リカバリ・カタログはこの表領域に格納される。
- リカバリ・カタログ・データベースに、TEMP という表領域が存在する。
- 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。つまり、catalog.sql と catproc.sql が正常に実行されています。

関連項目： RMAN への接続方法は、5-8 ページの「[RMAN への接続](#)」を参照してください。リカバリ・カタログを保持することの長所と短所については、5-4 ページの「[リカバリ・カタログの使用・不使用の決定](#)」を参照してください。

リカバリ・カタログのスキーマの設定

1. SQL*Plus を起動し、リカバリ・カタログが含まれているデータベースに、管理者権限で接続する。たとえば、次のように入力します。

```
SQL> CONNECT sys/change_on_install@rcat
```

2. エラー・チェックに使用するログ・ファイルを作成する。たとえば、次のように入力します。

```
SQL> SPOOL create_rman.log
```

3. リカバリ・カタログのユーザーおよびスキーマを作成する。たとえば、次のように入力します。

```
SQL> CREATE USER rman IDENTIFIED BY rman  
2> TEMPORARY TABLESPACE temp
```

```
3> DEFAULT TABLESPACE rcvcat
4> QUOTA UNLIMITED ON rcvcat;
```

4. RECOVERY_CATALOG_OWNER ロール権限をスキーマの所有者に付与する。このロールは、リカバリ・カタログのメンテナンスおよび同カタログへの問合せの権限を、ユーザーに与えるものです。

```
SQL> GRANT recovery_catalog_owner TO rman;
```

5. その他の必要な権限を RMAN ユーザーに付与する。

```
SQL> GRANT connect, resource TO rman;
```

6. オペレーティング・システムに出力し、create_rman.log ファイルでエラーをチェックした後、作業を継続する。たとえば、UNIX ユーザーは次のように発行します。

```
SQL> host
% vi create_rman.log
```

リカバリ・カタログの作成

1. O/S コマンド行からリカバリ・カタログに接続する。たとえば、次のように入力します。

```
% rman catalog rman/rman@rcat log = create_rman.log
```

RMAN プロンプトから接続することもできます。

```
% rman log = create_rman.log
RMAN> connect catalog rman/rman@rcat
```

2. **create catalog** コマンドを発行してカタログを作成し、表領域を RCVCAT と指定する。

```
RMAN> create catalog tablespace rcvcat;
```

3. オペレーティング・システムに出力し、エラーがないかどうか create_rman.log ファイルをチェックした後、作業を継続する。たとえば、UNIX ユーザーは次のように発行します。

```
RMAN> host;
% vi create_rman.log
```

関連項目 : create catalog コマンドの構文は、11-52 ページの「[createCatalog](#)」を参照してください。GRANT 文および CREATE USER 文の SQL 構文は、『Oracle8i SQL リファレンス』を参照してください。

RMAN メタデータのメンテナンス

この項では、RMAN の情報リポジトリの管理方法を説明します。ここでは、リカバリ・カタログを使用していることを前提として説明を行います。制御ファイルを RMAN メタデータの専用リポジトリとして使用している場合でも、RMAN のメンテナンス・コマンドのほとんどが使用できます。カタログ専用コマンドの一覧は、5-6 ページの「[制御ファイルを RMAN メタデータに使用した場合](#)」を参照してください。

この項では、次の項目について説明します。

- [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・カタログからのデータベースの登録解除](#)
- [リカバリ・カタログのリセット](#)
- [バックアップまたはファイル・コピーの可用性の変更](#)
- [RMAN メタデータのクロスチェック](#)
- [バックアップおよびコピーの削除と、RMAN メタデータにおけるそれらの状態の更新](#)
- [バックアップおよびコピーの復元の妥当性チェック](#)
- [リカバリ・カタログの再同期化](#)
- [制御ファイル内のレコードの管理](#)
- [O/S バックアップのカタログ化](#)

リカバリ・カタログへのターゲット・データベースの登録

ターゲット・データベースに対して RMAN を使用するときは、ターゲット・データベースをリカバリ・カタログに登録します。そのようにしないと、RMAN は、ターゲット・データベースに関する情報をリカバリ・カタログに格納できなくなります。RMAN は、ターゲット・データベースの登録に必要な情報を、すべてターゲット・データベース自体から取得するためです。リカバリ・カタログには、複数のターゲット・データベースを登録できます。

注意： 1 つのデータベースは、1 回しか登録できません。

ターゲット・データベースの登録

1. ターゲット・データベースおよびリカバリ・カタログに接続する。たとえば、RMAN としてカタログ・データベース RCAT に接続するには、次のように発行します。

```
% rman target / catalog rman/rman@rcat
```

2. データベースがマウントされていない場合は、マウントしてオープンする。たとえば、次のように発行します。

```
startup mount;
```

3. 新規にデータベースを作成した場合や、既存の Oracle7 データベースを移行した場合、またはバージョン 8.0 以上のデータベースで使用するために RMAN をインストールした場合は、次のコマンドを発行します。

```
register database;
```

4. バージョン 8.0 以上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';
```

Oracle7 以降の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み取り専用とされていた表領域の一部であった場合のみです。6-28 ページの「[O/S バックアップのカタログ化](#)」の項を参照してください。

注意： Enterprise Backup Manager (EBU) で作成したバックアップは、RMAN では使用もカタログ化もできません。

5. RMAN は、オリジナルのアーカイブ REDO ログに関する情報を、ターゲット・データベースの制御ファイルから自動的に取得します。追加で作成したアーカイブ・ログの O/S バックアップ、または古くなったターゲット制御ファイルのログ・レコードがある場合はカタログ化します。たとえば、次のようにログをカタログ化します (log1... は、アーカイブ・ログのフル・ファイル名を指します)。

```
catalog archivelog 'log1', 'log2', 'log3', ... 'logN';
```

注意： 制御ファイルのどのログ・レコードが古いかを判断するには、ディスク上のログ数と、V\$ARCHIVED_LOG 内のログ・レコード数を比較します。

Oracle では、データベースを区別するため、[DB 識別子 \(DB identifier \)](#) という一意的に生成された内部的な番号を使用します。データベースを作成すると、Oracle が自動的にこの番号を作成します。

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 文を使用せずに) 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、RMAN で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。このような問題の発生を回避するには、**duplicate** コマンドを使用して、バックアップからデータベースをコピーし、新しいデータベース識別子を生成するようにします。

データベース識別子の重複により障害が発生した場合は、別の Oracle データベース・ユーザーを使用して **create catalog** を再実行すると、別のユーザーのスキーマ内に 2 番目のリカバリ・カタログが作成されます。その後、データベース識別子が重複したデータベースを、この新スキーマ内の新たに作成したリカバリ・カタログに登録できます。

注意： 同一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 RMAN 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。

関連項目： **register** コマンドの構文は、11-94 ページの「[register](#)」を参照してください。**catalog** コマンドの構文は、11-32 ページの「[catalog](#)」を参照してください。**duplicate** コマンドの構文は、11-69 ページの「[duplicate](#)」を参照してください。データベースの移行に関する問題は、『Oracle8i 移行ガイド』を参照してください。

リカバリ・カタログからのデータベースの登録解除

RMAN では、データベースの登録も登録解除も実行できます。この処理は、実際に必要かどうか確認して行ってください。この処理の実行後、RMAN は、データベースのどのバックアップも回復できなくなるためです。

データベースの登録解除

1. RMAN を起動し、ターゲット・データベースに接続する。RMAN を使用してターゲット・データベースに接続する際に表示される、DBID の値をメモしておいてください。たとえば、次のように入力します。


```
% rman target sys/change_on_install@prod1 nocatalog
```

RMAN-06005: connected to target database: RMAN (DBID=1231209694)
2. 制御ファイルに記録されたコピーおよびバックアップ・セットをリスト表示し、**change ... delete** 文を発行してこれらを O/S から削除する。7-2 ページの「[リストの生成](#)」および 6-13 ページの「[バックアップおよびコピーの削除と、RMAN メタデータにおけるそれらの状態の更新](#)」を参照してください。
3. SQL*Plus を使用してリカバリ・カタログ・データベースに接続した後、リカバリ・カタログで次の問合せを実行し、DB 表から該当する行を見つける。DB_ID の値はステップ 1 で取得した値と同じに設定します。たとえば、次のように入力します。

```
SQL> SELECT db_key, db_id FROM db WHERE db_id = 1231209694;
```


この問合せにより戻される行は 1 行です。

```
DB_KEY      DB_ID
-----
1 1237603294
1 row selected.
```

4. リカバリ・カタログに接続した状態で、次のように入力する。DB_KEY および DB_ID は、ステップ 2 の問合せで得られた行の対応する列を示します。

```
SQL> EXECUTE dbms_rcvcat.unregisterdatabase(db_key, db_id)
```

たとえば、次のように入力します。

```
SQL> EXECUTE dbms_rcvcat.unregisterdatabase(1 , 1237603294)
```

注意： DBMS_RCVCAT.UNREGISTERDATABASE パッケージは、Oracle8 のどのデータベースでも有効です。

リカバリ・カタログのリセット

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 RMAN を使う場合は、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あらかじめ RMAN に通知しておく必要があります。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** コマンドを使用して RMAN に指示します。この新しいインカーネーション・レコードは現行のインカーネーションを示します。以降ターゲット・データベースが行う、すべてのバックアップおよびログ・アーカイブは、RMAN によって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のみで、データベースをリセットしなかった場合は、RMAN 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、RMAN はデータベースが RESETLOGS オプションを使用してオープンされたことを認識します。

RESETLOGS オプションを使用し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** コマンドを使用し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します。

リカバリ・カタログの古いインカーネーションへのリセット

1. 必要なデータベース・インカーネーション・主キーを指定する。インカーネーション・キーの値を取得するには、**list** コマンドを発行します。

```
list incarnation;
```

```
List of Database Incarnations
```

DB Key	Inc Key	DB Name	DB ID	CUR	Reset SCN	Reset Time
1	2	PROD1	1224038686	NO	1	02-JUL-98
1	582	PROD1	1224038686	YES	59727	10-JUL-98

- データベースを古いインカーネーションにリセットする。たとえば、次のように入力します。

```
reset database to incarnation 2;
```

- データベースをリセットしたら、**restore** コマンドおよび **recover** コマンドを発行し、前のインカーネーションからデータベース・ファイルを復元および回復します。そして、RESETLOGS オプションでデータベースをオープンします。たとえば、次のように入力します。

```
run {
  allocate channel ch1 type disk;
  restore database;
  recover database;
  sql "ALTER DATABASE OPEN RESETLOGS";
}
```

関連項目 : `reset database` コマンドの構文は、11-111 ページの「[reset](#)」を参照してください。
`list` コマンドの構文は、11-76 ページの「[list](#)」を参照してください。

バックアップまたはファイル・コピーの可用性の変更

unavailable オプションは、バックアップまたはコピー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。この操作では、**type maintenance** のチャンネルを割り当てる必要はありません。

注意 : `change ... unavailable` および `change ... available` を実行する場合は、リカバリ・カタログを使用する必要があります。

バックアップ・ピースまたはコピーへの available、unavailable のマークの付与

- バックアップまたはコピーに **unavailable** とマークするには、`change ... unavailable` コマンドを発行します。たとえば、次のように入力します。

```
change datafilecopy '/oracle/backup/cf_c.f' unavailable;
change backupset 12 unavailable;
```

2. 以前に使用不能とされたファイルを使用可能にするには、**change ... available** を発行すると、その状態が **available** に戻ります。

```
change datafilecopy '/oracle/backup/cf_c.f' available;
change backupset 12 available;
```

関連項目 : **change** コマンドの構文は、11-35 ページの「**change**」を参照してください。

RMAN メタデータのクロスチェック

バックアップおよびコピーは、ディスクまたはテープから消失したり、破損したりする可能性があるため、RMAN のメタデータ・リポジトリには、古い情報が含まれることがあります。リカバリ・カタログまたは制御ファイルのバックアップ・セットおよびイメージ・コピーの情報と、ディスクまたはメディア管理カタログの対応する情報との同期を保つためには、**クロスチェック (crosscheck)** を実行します。

特定のファイルをチェックするには、**change ... crosscheck** コマンドまたは **crosscheck backup** コマンドを使用します。これらのコマンドは、O/S ファイルやメタデータ・レコードを削除するものではないことに注意してください。これらの操作には別のコマンドを使用する必要があります。

クロスチェック・コマンドの違いを次の表に示します。

コマンド	カタログが必要か？	目的
change ...crosscheck	backupset オプションおよび backuppiece オプションの場合のみ	<p>バックアップまたはコピーが存在するかどうかを確認する。バックアップ・ピースを見つけられない場合、RMAN はこれらに expired のマークを付与します。それ以外のタイプのファイル、つまり、イメージ・コピーおよびアーカイブ REDO ログについては、不在である場合には deleted とマークされます。</p> <p>ファイルがディスク上にある場合、RMAN はそのヘッダーに対して問合せを行います。その他のデバイス・タイプの場合、RMAN はメディア・マネージャに対して問合せを行い、ファイルがメディア・カタログに存在するかどうかを確かめます。</p>

コマンド	カタログが必要か？	目的
crosscheck backup	必要	<p>ディスクまたはテープに格納されたバックアップが存在するかどうかを確認する。バックアップとは、バックアップ・セットまたはメディア管理のプロキシ・コピーのことを指します。</p> <p>このコマンドでは、available または expired とマークされたバックアップ・セットのみチェックされます。type disk の場合はバックアップ・ピースが検査され、type 'sbt_tape' の場合はメディア・マネージャに対して問合せが行われます。このコマンドは、指定したチャンネルで作成されたバックアップのみ処理します。</p> <p>RMAN は、見つけれられないバックアップ・ピースについては削除せず、expired のマークを付与します。</p>

関連項目 : ファイルを削除する方法、およびメタデータ・レコードを更新する方法は、6-13 ページの「[バックアップおよびコピーの削除と、RMAN メタデータにおけるそれらの状態の更新](#)」を参照してください。**crosscheck** コマンドの構文は、11-57 ページの「[crosscheck](#)」を参照してください。

バックアップのクロスチェック

ディスクまたはテープ上のバックアップの状態をチェックするには、クロスチェック機能を使用します。バックアップがディスク上にある場合は、**change backupset ... crosscheck** コマンドおよび **crosscheck backup** コマンドを使用すると、バックアップ・ピースのヘッダーが有効であるかどうかを確認できます。テープ上のバックアップについては、そのバックアップが存在するかのみこれらのコマンドでチェックされます。

RMAN は、読み取れないまたは存在しないバックアップ・ピースには、**list** コマンドの出力およびリカバリ・カタログ・ビューで **expired** のマークを付与します。**expired** とマークされても、現在は使用可能になっているバックアップ・ピースについては、RMAN は、**list** コマンドの出力およびリカバリ・カタログ・ビューで **available** のマークを付与します。

チェックするバックアップ・セットまたはバックアップ・ピースのリストが必要な場合は、**change backupset ... crosscheck** を使用します。クロスチェックを特定のデバイス・タイプ、オブジェクト・タイプあるいはデータ範囲に限定し、RMAN でバックアップ・セットまたはバックアップ・ピースのリストを生成する場合は **crosscheck backup** を使用します。

注意 : **change** コマンドは、リカバリ・カタログまたは制御ファイルに記録されたファイルでのみ有効です。これは、**catalog** および **resync from controlfilecopy** を除き、その他のコマンドについても同様です。

クロスチェックするバックアップの一覧の表示

1. **type maintenance** のチャンネルを割り当てる。

```
allocate channel for maintenance type 'sbt_tape';
```

2. **list** コマンドを発行して、チェックするバックアップ・ピース、バックアップ・セットまたはプロキシ・コピーを指定する。

```
list backup;
```

3. 指定したバックアップ・セットが存在するかどうかチェックする。この例では、主キーが 1338、1339 および 1340 のバックアップ・セットが残っているかどうかをチェックします。

```
RMAN> change backupset 1338, 1339, 1340 crosscheck;
```

```
RMAN-03022: compiling command: change
RMAN-08074: crosschecked backup piece: found to be 'EXPIRED'
RMAN-08517: backup piece handle=/oracle/dbs/2eafnuj3_1_1 recid=77
stamp=352057957
RMAN-08074: crosschecked backup piece: found to be 'AVAILABLE'
RMAN-08517: backup piece handle=/oracle/dbs/2dafnuj2_1_1 recid=78
stamp=352057957
RMAN-08074: crosschecked backup piece: found to be 'AVAILABLE'
RMAN-08517: backup piece handle=/oracle/dbs/2fafnuj3_1_1 recid=79
stamp=352057960
```

RMAN は、使用不能になっているバックアップ・セットには、**expired** のマークを付与します。**expired** とマークされても現在は使用可能になっているバックアップ・セットに対しては、RMAN は **available** とマークします。

4. 割り当てられたメンテナンス・チャンネルを解放する。

```
release channel;
```

クロスチェックするバックアップの一覧の RMAN による自動的な生成

1. **type maintenance** のチャンネルを割り当てる

```
allocate channel for maintenance type 'sbt_tape';
```

2. 指定したデータベース、表領域、データ・ファイル、制御ファイルまたはアーカイブ REDO ログのバックアップがあるかどうかをチェックする。時間順序に基づいて、クロスチェックの範囲を限定する。

ここでは、データ・ファイル `tbs_8.f` の過去 6 か月間のすべてのバックアップをチェックします。

```
crosscheck backup of datafile "/oracle/dbs/tbs_8.f" completed after
'SYSDATE-180';
```

RMAN は、使用不能になっているバックアップ・セットには、**expired** のマークを付与します。**expired** とマークされても現在は使用可能になっているバックアップ・セットに対しては、RMAN は **available** とマークします。

- 3. 割り当てられたメンテナンス・チャンネルを解放する。

```
release channel;
```

イメージ・コピーのクロスチェック

ディスクまたはテープ上のデータ・ファイルのイメージ・コピー、制御ファイルまたはアーカイブ REDO ログが有効であるかどうかを確認するには、**change ... crosscheck** コマンドを使用します。指定したイメージ・コピーまたはアーカイブ REDO ログが見つけれなかった場合、RMAN はこれらの状態を更新して **deleted** に変更します。**deleted** 状態のコピーおよびログを確認するには、該当するリカバリ・カタログ・ビューに問合せを行います。

RMAN では、アーカイブ REDO ログはイメージ・コピーとして認識されることに注意してください。なんらかの理由で 1 つ以上のアーカイブ REDO ログが使用不能になった場合は、**change archivelog all crosscheck** コマンドを発行して、そのログに **deleted** と RMAN にマークさせます。このコマンドを実行する場合、リカバリ・カタログを使用する必要はありません。

イメージ・コピーをクロスチェックするには

- 1. RMAN に接続する。リカバリ・カタログの使用は任意です。たとえば、次のいずれかを入力します。

```
% rman target / nocatalog
% rman target / catalog rman/rman@rcat
```

- 2. **list** コマンドを発行して、チェックするイメージ・コピーを指定する。この例では、記録されていたすべてのイメージ・コピーおよびアーカイブ REDO ログをリスト表示します。

```
list copy of database archivelog all;
```



```
List of Datafile Copies
Key      File S Completion time Ckp SCN      Ckp time      Name
-----
1262     1    A 18-AUG-98          219859      14-AUG-98      /oracle/dbs/copy/tbs_01.f
```



```
List of Archived Log Copies
Key      Thrd Seq      S Completion time Name
-----
789      1      1      A 14-JUL-98      /oracle/work/arc_dest/arcr_1_1.arc
790      1      2      A 11-AUG-98      /oracle/work/arc_dest/arcr_1_2.arc
```

```
791      1      3      A 12-AUG-98      /oracle/work/arc_dest/arcr_1_3.arc
```

3. **change ... crosscheck** を使用して、指定したコピーが存在するかどうかをチェックする。存在していなければ、RMAN はその状態を更新して **deleted** に変更します。この例では、主キーが 1262 のデータ・ファイル・コピーが存在しているかどうかをチェックします。

```
change datafilecopy 1262 crosscheck;
```

```
RMAN-03022: compiling command: change
```

```
RMAN-06154: validation succeeded for datafile copy
```

```
RMAN-08513: datafile copy filename=/oracle/dbs/copy/tbs_01.f recid=1
stamp=351194732
```

RMAN がそのコピーを検証できない場合は次のように表示されます。

```
RMAN-06153: validation failed for datafile copy
```

アーカイブ REDO ログ削除後の、RMAN メタデータの更新

1. RMAN に接続する。リカバリ・カタログの使用は任意です。この例では、次のいずれかを入力します。

```
% rman target / nocatalog
```

```
% rman target / catalog rman/rman@rcat
```

2. **change ... crosscheck** コマンドを発行して、不在のアーカイブ REDO ログのメタデータ・レコードを更新する。この例では、すべてのアーカイブ REDO ログをチェックします。

```
change archivelog all crosscheck;
```

関連項目 : **crosscheck** コマンドの構文は、11-57 ページの「[crosscheck](#)」を参照してください。**change** コマンドの構文は、11-35 ページの「[change](#)」を参照してください。**list** コマンドの構文は、11-76 ページの「[list](#)」を参照してください。

バックアップおよびコピーの削除と、RMAN メタデータにおけるそれらの状態の更新

RMAN では、バックアップおよびコピーを削除した後、制御ファイルまたはリカバリ・カタログにおけるそれらの状態を **deleted** に更新できます。**deleted** 状態のバックアップおよびコピーは、**list** コマンドの出力には表示されないことに注意してください。これらはリカバリ・カタログ・ビューに検索を行う必要があります。

代表的な使用例を次に示します。

コマンドまたはスクリプト	カタログが必要か？	目的
<code>change ... uncatalog</code>	必要	<p>リカバリ・カタログから指定したバックアップまたはコピーのレコードを削除する。物理バックアップまたはコピーは、このコマンドでは削除されません。これらのレコードのみ削除されます。</p> <p>注意: バックアップ制御ファイルから再同期化を行う場合、削除したレコードがカタログに再び表示されることがあります。</p>
<code>prgrmanc.sql</code>	必要	<p>状態が deleted になっているバックアップまたはコピーのレコードを、カタログからすべて同時に削除する。物理バックアップまたはコピーは、このスクリプトでは削除されません。これらのレコードのみ削除されます。</p> <p>注意: バックアップ制御ファイルから再同期化を行う場合、削除したレコードがカタログに再び表示されることがあります。</p>
<code>delete expired backup</code>	必要	<p>バックアップ・セットのレコードの状態を、expired から deleted に更新する。期限切れのバックアップが残っている場合、RMAN はこれらについても物理的に削除します。</p> <p>通常、このコマンドは、クロスチェックの実行後に発行します。クロスチェックを実行すると、アクセス不能なバックアップには expired のマークが付与されます。</p>
<code>change ... delete</code>	必要なし	<p>物理バックアップ・セット、イメージ・コピーまたはアーカイブ REDO ログを削除し、これらのメタデータ・レコードの状態を deleted に更新する。</p> <p>delete expired コマンドとは異なり、change ... delete コマンドは、expired とマークされたバックアップまたはコピーのみではなく、あらゆるバックアップまたはコピーで有効です。</p>

関連項目 : `change` コマンドの構文は、11-35 ページの「[change](#)」を参照してください。
`delete` コマンドの構文は、11-63 ページの「[deleteExpired](#)」を参照してください。リカバリ・カタログ・ビューの詳細は、[第 12 章の「リカバリ・カタログ・ビュー」](#)を参照してください。

O/S ユーティリティを使用して削除したバックアップまたはコピーのリカバリ・カタログ・レコードの更新

1. **type maintenance** のチャンネルを割り当てる。

```
allocate channel for maintenance type disk;
```

2. オペレーティング・システムから O/S コマンドを使用して削除したバックアップまたはコピーに対し、**change ... uncatalog** コマンドを発行する。この例では、制御ファイルおよび datafile 1 のコピーへの参照を削除します。

```
change controlfilecopy '/oracle/backup/cf_c.f' uncatalog;
change datafilecopy '/oracle/backup/df_1_c.f' uncatalog;
```

3. 割り当てられたチャンネルを解放する。

```
release channel;
```

バックアップおよびコピーの物理的な削除およびそれらのメタデータ・レコードの更新

この処理では、リカバリ・カタログを使用する必要はありません。

1. 期限切れのバックアップ・セットおよびコピーが存在するかどうかチェックする。これらの主キーを取得するには、**list** コマンドを使用します。

```
# lists backups of database files and logs
list backup of database archivelog all;
list copy;
```

2. **type delete** のチャンネルを割り当てる。

```
allocate channel for delete type 'sbt_tape';
```

3. **change ... delete** コマンドを発行して両方の物理ファイルを削除し、リカバリ・カタログ・レコードのステータスを **deleted** に更新する。この例では、主キーが 101 のバックアップ・ピースと、主キーが 63 の制御ファイル・コピーを、テープから削除します。

```
change backuppiece 101 delete;
change controlfilecopy 63 delete;
```

4. 割り当てられたメンテナンス・チャンネルを解放する。

```
release channel;
```

カタログ内の期限切れのバックアップ・レコードの状態を deleted に更新する方法（および期限切れであるにもかかわらずリカバリ・カタログに存在しているバックアップ・ピースを削除する方法）

この処理では、リカバリ・カタログを使用する必要があります。

1. オプションで、クロスチェック操作（6-9 ページの「[RMAN メタデータのクロスチェック](#)」を参照）を実行して、アクセス不可能なバックアップまたは不在のバックアップすべてに **expired** とマークする。**delete expired backup** コマンドは、期限切れのバックアップでのみ有効です。

2. **type delete** のチャンネルを割り当てる。

```
allocate channel for delete type disk;
```

3. **delete expired backup** を発行して、**expired** のマークが付与されたバックアップを探し、それらの状態を **deleted** に更新する。この例では、リカバリ・カタログに登録されたバックアップのうち、期限切れになっているものをすべて更新します。

```
delete expired backup;
```

4. 割り当てられたチャンネルを解放する。

```
release channel;
```

リカバリ・カタログから、特定のコピー・レコードまたはバックアップ・レコードのみを削除する方法（バックアップおよびコピーを物理的に削除しない）

1. 期限切れのイメージ・コピーまたはバックアップを探す。これらの主キーを取得するには、**list** コマンドを使用します。たとえば、次のように入力します。

```
list copy;
```

2. **change ... uncatalog** コマンドを発行し、指定したバックアップおよびコピーへの参照を、リカバリ・カタログから削除する。この例では、制御ファイルのコピーのレコードと、主キーが 4833 のデータ・ファイル・コピーのレコードを削除します。

```
change controlfilecopy '/oracle/backup/cf_c.f' uncatalog;
change datafilecopy 4833 uncatalog;
```

change ... uncatalog では、O/S からファイルは削除されないことに注意してください。削除されるのは、リカバリ・カタログのレコードのみです。

3. 該当するリカバリ・カタログ・ビュー、つまり、RC_DATAFILE_COPY や RC_CONTROLFILE_COPY などの内容を確認し、指定したレコードが削除されたかどうかを確認する。たとえば、この問合せでは、コピー 4833 のレコードが削除されたことが確認できます。

```
SQL> SELECT cdf_key, status FROM rc_datafile_copy WHERE cdf_key = 4833;
CDF_KEY      S
-----
0 rows selected.
```

リカバリ・カタログから、すべてのコピー・レコードまたはバックアップ・レコードを削除する方法（バックアップおよびコピーを物理的に削除しない）

\$ORACLE_HOME/rdbms/prgrmanc.sql スクリプトを使用すると、削除されたバックアップおよびコピーのリカバリ・カタログ・レコードを、すべて同時に削除することができます。この操作を実行するのは、バックアップの履歴が不要な場合のみです。

関連項目：change コマンドの構文は、11-35 ページの「[change](#)」を参照してください。

1. **type maintenance** のチャンネルを割り当てる。

```
allocate channel for maintenance type 'sbt_tape';
```

2. **change ... delete** コマンドを発行して、必要なレコードの状態を **deleted** に更新し、ファイルを O/S またはメディア・マネージャから削除する。アーカイブ REDO ログ、バックアップ・セット、制御ファイル・コピーまたはデータ・ファイル・コピーの主キーを取得するには、**list** コマンドを発行するか、または該当するリカバリ・カタログ・ビューに対して問合せを行います。

```
change backupset 100, 101, 102, 103 delete;
```

3. 割り当てられたメンテナンス・チャンネルを解放する。

```
release channel;
```

4. SQL*Plus セッションを開始し、リカバリ・カタログに接続する。この例では、ユーザー RMAN としてデータベース RCAT に接続します。

```
% sqlplus rman/rman@rcat
```

5. prgrmanc.sql スクリプトを実行する。このスクリプトは、\$ORACLE_HOME/rdbms/admin ディレクトリに格納されています。

```
SQL> @prgrmanc.sql
```

RMAN は、状態が **deleted** になっているレコードを、すべてリカバリ・カタログから削除します。

リカバリ・カタログからのインカーネーション・レコードの削除

1. **type maintenance** のチャンネルを割り当てる。

```
allocate channel for maintenance type 'sbt_tape';
```

2. **change ... delete** コマンドを発行し、必要のないバックアップ・ピース、アーカイブ REDO ログおよびイメージ・コピーの状態を、**deleted** に更新する。アーカイブ REDO ログ、バックアップ・セット、制御ファイル・コピーまたはデータ・ファイル・コピーの主キーを取得するには、**list** コマンドを発行するか、または該当するリカバリ・カタログ・ビューに対して問合せを行います。

```
change backupset 100, 101, 102, 103 delete;
```

3. 割り当てられたメンテナンス・チャンネルを解放する。

```
release channel;
```

4. SQL*Plus セッションを開始し、リカバリ・カタログに接続する。この例では、ユーザー RMAN としてデータベース RCAT に接続します。

```
% sqlplus rman/rman@rcat
```

5. RC_DATABASE_INCARNATION リカバリ・カタログ・ビューに問合せを行い、削除するレコードを有するインカーネーションの DBINC_KEY の値を取得する。

```
SQL> SELECT * FROM rc_database_incarnation;
```

6. 次の DML 文を実行する。key_value は、DBINC_KEY の値を示します。

```
SQL> DELETE FROM dbinc WHERE dbinc_key=key_value;
```

RMAN は、指定したインカーネーションのレコードを、リカバリ・カタログから削除します。

関連項目 : RC_DATABASE_INCARNATION リカバリ・カタログ・ビューの詳細は、12-15 ページの「[RC_DATABASE_INCARNATION](#)」を参照してください。

バックアップおよびコピーの復元の妥当性チェック

復元の**妥当性チェック (validation)**とは、ファイルの復元を実際に行わず、そのテストを行うことです。復元のテストは、データベース全体に対して行うことも、指定した表領域、データ・ファイルまたは制御ファイルに対して行うこともできます。バックアップまたはコピーが復元できるかどうかをテストするには、**restore ... validate** コマンドおよび **validate backupset** コマンドを使用します。これらは、それぞれ次の場合に使用します。

- **restore ... validate** は、テストするバックアップ・セットまたはコピーを、RMAN に自動的に選択させる場合に使用する。
- **validate backupset** は、テストするバックアップ・セットを自分で指定する場合に使用する。

関連項目 : **restore** コマンドの構文は、11-113 ページの「[restore](#)」を参照してください。
validate コマンドの構文は、11-151 ページの「[validate](#)」を参照してください。

妥当性チェックを行うバックアップ・セットまたはコピーを RMAN に自動的に選択させる方法

1. データベース全体の妥当性チェックを行う場合は、そのデータベースをクローズする必要があります。個別のファイルの妥当性チェックを行う場合は、データベースをクローズするか、またはチェック対象のデータ・ファイルをオフラインにします。この例で

は、データベースの状態のチェック、インスタンスの停止、そしてマウントを行います。

```
SQL> SELECT status FROM v$instance;
```

```
STATUS
-----
```

```
OPEN
```

```
1 row selected.
```

```
SQL> STARTUP FORCE MOUNT;
```

2. RMAN を起動し、ターゲット・データベースおよびオプションのリカバリ・カタログ・データベースに接続する。

```
% rman target / catalog rman/rman@rcat
```

3. 妥当性チェックをデータベース全体に対して行うのであれば、**list** コマンドを発行し、主キーを明示して、妥当性チェックを行うバックアップ・セットおよびコピーを指定する。

```
list backupset;
```

```
list copy;
```

4. バックアップ・セットおよびコピーの復元の妥当性チェックを行う。この例では、バックアップ制御ファイル、SYSTEM 表領域および全アーカイブ REDO ログのディスクからの復元について、妥当性チェックを行います。

```
run {
  allocate channel ch1 type disk;
  allocate channel ch2 type tape;
  restore controlfile validate;
  restore tablespace 'system' validate;
  restore archivelog all validate;
}
```

5. 出力をチェックする。エラー・メッセージ・スタックがあり、次のようになっている場合は、妥当性チェックを行うファイルのバックアップまたはコピーがないことを示しています。

```
RMAN-06026: some targets not found - aborting restore
```

エラー・メッセージ・スタックがあり、出力が次のようになる場合は、指定したファイルの復元に問題があることを示しています。

```
RMAN-03002: failure during compilation of command
```

```
RMAN-03013: command type: restore
```

```
RMAN-03007: retryable error occurred during execution of command: IRESTORE
```

```
RMAN-07004: unhandled exception during command execution on channel c1
```

```

RMAN-10035: exception raised in RPC: ORA-19505: failed to identify file
           "oracle/dbs/lfafv9gl_1_1"
ORA-27037: unable to obtain file status
SVR4 Error: 2: No such file or directory
Additional information: 3
RMAN-10031: ORA-19624 occurred during call to DBMS_BACKUP_
RESTORE.RESTOREBACKUPPIECE

```

エラー・スタックがない場合は、ファイルの妥当性チェックに RMAN が成功したことを示しています。

妥当性チェックを行うバックアップ・セットの指定

1. RMAN を起動し、ターゲット・データベースおよびオプションのリカバリ・カタログ・データベースに接続する。

```
% rman target / catalog rman/rman@rcat
```

2. **list** コマンドを発行して、チェックするバックアップ・セットを指定する。

```
list backup of database archivelog all;
```

3. バックアップ・セットの復元の妥当性チェックを行う。この例では、バックアップ制御ファイル、SYSTEM 表領域および全アーカイブ REDO ログのディスクからの復元について、妥当性チェックを行います。

```

run {
    allocate channel ch1 type disk;
    validate backupset 1121;
}

```

4. 出力をチェックする。RMAN-08024 というメッセージがあれば、指定したバックアップ・セットの復元に RMAN が成功したことを示しています。

```

RMAN-03022: compiling command: allocate
RMAN-03023: executing command: allocate
RMAN-08030: allocated channel: ch1
RMAN-08500: channel ch1: sid=10 devtype=DISK

RMAN-03022: compiling command: validate
RMAN-03023: executing command: validate
RMAN-08016: channel ch1: starting datafile backupset restore
RMAN-08502: set_count=47 set_stamp=346169465 creation_time=08-OCT-98
RMAN-08023: channel ch1: restored backup piece 1
RMAN-08511: piece handle=/vobs/oracle/dbs/lfaa483p_1_1 params=NULL
RMAN-08024: channel ch1: restore complete
RMAN-08031: released channel: ch1

```

リカバリ・カタログへのスクリプトの格納

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一連の RMAN コマンドのことです。これにより、頻繁に実行される RMAN コマンドのコレクションに、共通リポジトリがもたらされます。

たとえば、夜間バックアップを実行するために必要な RMAN 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、O/S 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、RMAN がどのマシン上で実行されているかに関係なく、どの DBA でも RMAN を使用してスクリプトにアクセスできるので便利です。

RMAN では、次の事柄が実行できます。

- スクリプトを作成し、これをリカバリ・カタログに格納する。
- ストアド・スクリプトを実行する。
- ストアド・スクリプトを置き換える。
- スクリプトをリカバリ・カタログから削除する。
- ストアド・スクリプトを、メッセージ・ログ・ファイルまたは画面へ出力する。
- 全ストアド・スクリプトの一覧を取得する。

ストアド・スクリプトの作成

1. RMAN を起動し、リカバリ・カタログ・データベースに接続する。たとえば、次のように入力します。

```
% rman catalog rman/rman@rcat
```

2. 必要なスクリプトを書く。たとえば、このスクリプトでは、データベースおよびアーカイブ REDO ログがバックアップされます。

```
create script b_whole{
    allocate channel ch1 type disk;
    allocate channel ch2 type disk;
    backup database;
    sql 'ALTER SYSTEM ARCHIVE LOG ALL';
    backup archivelog all;
}
```

3. 出力をチェックする。RMAN-08085 というメッセージがあれば、スクリプトの作成およびリカバリ・カタログへの格納が成功したことを示しています。

```
RMAN-03022: compiling command: create script
RMAN-03023: executing command: create script
RMAN-08085: created script b_whole
```

ストアド・スクリプトの実行

1. RMAN を起動し、リカバリ・カタログ・データベースおよびターゲット・データベースに接続する。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

2. **run** コマンドを発行し、スクリプトを実行する。RMAN は、スクリプトの内容を **run** の大カッコに挿入します。スクリプト内でチャンネルの割当てをすでに行っていた場合は、ここでそれを繰り返す必要はないことに注意してください。

```
run { execute script b_whole; }
```

関連項目 : **execute script** コマンドの構文は、11-125 ページの「[run](#)」を参照してください。

ストアド・スクリプトの置換え

1. RMAN を起動し、リカバリ・カタログ・データベースおよびターゲット・データベースに接続する。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

2. **replace script** コマンドを発行し、ストアド・スクリプトを置き換える。たとえば、このコマンドでは、**b_whole** が次のように置き換えられます。

```
replace script b_whole {  
    allocate channel ch1 type 'sbt_tape';  
    backup database;  
}
```

関連項目 : **replace script** コマンドの構文は、11-98 ページの「[replaceScript](#)」を参照してください。

ストアド・スクリプトの削除

1. RMAN を起動し、リカバリ・カタログ・データベースに接続する。たとえば、次のように入力します。

```
% rman catalog rman/rman@rcat
```

2. **delete script** コマンドを発行し、指定したストアド・スクリプトを削除する。

```
delete script 'b_whole';
```

関連項目 : **delete script** コマンドの構文は、11-65 ページの「[deleteScript](#)」を参照してください。

ストアド・スクリプトのメッセージ・ログへの出力

1. RMAN を起動し、リカバリ・カタログ・データベースおよびターゲット・データベースに接続する。ここでは、メッセージ・ログに出力するので **log** 引数を指定します。たとえば、`rman_log` を指定するとして、次のように入力します。

```
% rman target / catalog rman/rman@rcat log rman_log
```

2. **print script** コマンドを発行し、スクリプトをログに書き込む。

```
print script b_whole;
```

3. O/S に出力し、O/S ユーティリティを使用してスクリプトを表示する。たとえば、次のように入力します。

```
RMAN> host;  
% vi rman_log
```

関連項目 : `replace script` コマンドの構文は、11-87 ページの「[printScript](#)」を参照してください。

全ストアド・スクリプトの一覧の取得

1. SQL*Plus を起動し、リカバリ・カタログ・データベースに接続する。たとえば、データベース RCAT に接続するのであれば、次のように入力します。

```
% sqlplus rman/rman@rcat
```

2. `RC_STORED_SCRIPT` ビューで `SELECT` 文を発行する。

```
SQL> SELECT script_name FROM rc_stored_script;  
SCRIPT_NAME  
-----  
backupdb  
binc  
bincl  
3 rows selected.
```

関連項目 : `RC_STORED_SCRIPT` ビューについては、12-26 ページの「[RC_STORED_SCRIPT](#)」を参照してください。

リカバリ・カタログの再同期化

リカバリ・カタログの再同期化 ([resynchronization](#)) を実行する場合、RMAN 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、失われている情報または変更された情報を含めて、リカバリ・カタログを更新します。

再同期化を実行する場合は、完全再同期化か、部分再同期化を選択できます。部分再同期化の場合、RMAN は、現行の制御ファイルを読み取って変更済み情報を更新しますが、データベースの物理スキーマ (physical schema) に関するメタデータは再同期化しません。ここで言う物理スキーマとは、データ・ファイル、表領域、REDO スレッド、ロールバック・セグメント (データベースがオープンしている場合のみ) およびオンライン REDO ログのことです。完全再同期化の場合、RMAN は、データベース・スキーマも含めてすべての変更済みレコードを更新します。

注意： RMAN は、制御ファイルがバックアップである場合、部分再同期化を実行しますが、完全再同期化は実行しません。

再同期化の実行時、RMAN はスナップショット制御ファイルを作成し、リカバリ・カタログと比較します。そして、失われている情報または変更された情報を含めて、リカバリ・カタログを更新します。一定のコマンドが実行されるとき、RMAN は必要に応じて部分再同期化または完全再同期化を実行します (詳細は 11-119 ページの「resync」を参照)。完全再同期化を実行するには、resync catalog を発行します。

この項では、次の項目について説明します。

- 再同期化の対象
- 再同期化を実行する時期
- 再同期化の方法

再同期化の対象

RMAN の再同期化の対象となるレコードのタイプを表 6-1 で説明します。

表 6-1 再同期化時に更新されるレコード

レコード	説明
ログ履歴	オンライン REDO ログ・スイッチが発生すると作成される。
アーカイブ REDO 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、既存のアーカイブ・ログをコピーしたとき、またはアーカイブ REDO ログのバックアップ・セットを復元したときに作成される。RMAN 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。
バックアップ履歴	バックアップ・セット、バックアップ・ピース、バックアップ・セット・メンバーおよびファイル・コピーに関連したレコード。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。

表 6-1 再同期化時に更新されるレコード (続き)

レコード	説明
物理スキーマ	<p>データ・ファイルおよび表領域に関連したレコード。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。</p> <p>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、ターゲットに現行の制御ファイルがマウントされている場合のみです。</p> <p>ターゲット・データベースに、バックアップ制御ファイル、新しく作られた制御ファイルまたは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 controlfilecopy コマンドを使用したときには、物理スキーマ情報は更新されません。</p>

再同期化を実行する時期

RMAN は、ある一定の状況で、必要に応じて完全再同期化または部分再同期化を実行します。ターゲット・データベースがマウントされている場合、次のコマンドを実行すると、(スキーマ・メタデータの変更の有無に応じて) 完全再同期化または部分再同期化が自動的に行われます。リカバリ・カタログ・データベースは、これらのコマンドが実行されたときに使用可能となります。

- **backup**
- **copy**
- **crosscheck**
- **delete expired backupset**
- **duplicate**
- **list**
- **recover**
- **report**
- **restore**
- **switch**

次の例に基づいて、手動による再同期化を実行します。

リカバリ・カタログが使用できなかった場合の再同期化 リカバリ・カタログが使用不可の状態で **backup** コマンドまたは **copy** コマンドを発行した場合、カタログ・データベースのオープン後に、**resync catalog** コマンドを使用して手動でデータベースを再同期化してください。

ARCHIVELOG モードで運用している場合の再同期化 ARCHIVELOG モードで運用している場合、リカバリ・カタログは定期的に再同期化する必要があります。リカバリ・カタログは、REDO ログ・スイッチが発生したり、REDO ログがアーカイブされても自動的に更新されないためです。ただし、Oracle は、ログ・スイッチおよびアーカイブ REDO ログに関する情報は、制御ファイルに格納します。この情報は、定期的にリカバリ・カタログ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、Oracle に設定されている REDO ログのアーカイブの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。

物理データベースの変更後の再同期化 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・カタログの再同期化を行います。REDO 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。

物理スキーマは、次の操作を行った場合に変更されます。

- 表領域を追加または削除する。
- 既存の表領域に新しいデータ・ファイルを追加する。
- ロールバック・セグメントを追加または削除する。

現行の制御ファイルから再同期化する場合、RMAN 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、それらのレコードのみを再同期化します。ターゲット・データベースがオープン状態の場合、RMAN 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 TSPITR に使われます）も更新します。

バックアップ制御ファイルから再同期化する場合、RMAN 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 ... crosscheck** コマンドおよび **crosscheck** コマンドを使用して、すでに存在していないファイルのレコードを削除する必要がある場合があります。

再同期化の方法

resync catalog コマンドを発行し、リカバリ・カタログの完全再同期化を実行します。

リカバリ・カタログの完全再同期化の実行

1. リカバリ・カタログをオープンする（まだオープンしていない場合）。たとえば、次のように入力します。

```
SQL> STARTUP pfile=initRCAT.ora
```

2. RMAN を起動し、ターゲット・データベースおよびリカバリ・カタログ・データベースに接続する。たとえば、次のように入力します。

```
% rman target sys/change_on_install@prod1 catalog rman/rman@rcat
```

3. ターゲット・データベースをマウントする（まだマウントしていない場合）。

```
startup mount;
```

4. **resync catalog** コマンドを発行する。

```
resync catalog;
```

```
RMAN-03022: compiling command: resync
```

```
RMAN-03023: executing command: resync
```

```
RMAN-08002: starting full resync of recovery catalog
```

```
RMAN-08004: full resync complete
```

関連項目 : **resync catalog** コマンドの構文は、11-119 ページの「[resync](#)」を参照してください。

制御ファイル内のレコードの管理

リカバリ・カタログは、情報についてはターゲット・データベースの制御ファイルに依存しています。したがって、制御ファイル内の情報の最新性が、リカバリ・カタログの有効性を左右します。

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。

- 実行するバックアップの回数
- Oracle が生成するアーカイブ REDO ログの数
- 制御ファイル内にその情報を格納しておく日数

CONTROL_FILE_RECORD_KEEP_TIME パラメータを使用して、Oracle が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。ここで指定した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、制御ファイルも大きくなります。

リカバリ・カタログは、CONTROL_FILE_RECORD_KEEP_TIME の設定値よりも短い間隔で再同期化してください。このパラメータで指定した日数より後は、Oracle は最新の情報で制御ファイルの情報を上書きするためです。リカバリ・カタログの再同期化を行っていないにもかかわらず、Oracle が情報を上書きした場合、この情報をリカバリ・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 O/S 固有の 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、『Oracle8i リファレンス・マニュアル』を参照してください。バックアップおよび回復時の制御ファイルの管理の詳細は、2-3 ページの「[制御ファイルの管理](#)」を参照してください。制御ファイル管理の詳細は、『Oracle8i 管理者ガイド』を参照してください。

O/S バックアップのカタログ化

RMAN 以外の手段で作成したファイル・コピーの存在を RMAN に認識させておくことができます。この項では、次の項目について説明します。

- [カタログ化の目的](#)
- [O/S バックアップのカタログ化の時期](#)
- [O/S バックアップのカタログ化の方法](#)

カタログ化の目的

RMAN コマンドを使用してバックアップまたはイメージ・コピーを作成していない場合、リカバリ・カタログにはこれらのレコードは残りません。O/S ユーティリティを使用してバックアップを作成した場合は、手動でその旨を RMAN に通知する必要があります。次のような場合に RMAN の `catalog` コマンドを使用します。

- O/S のデータ・ファイル・コピー、アーカイブ REDO ログのコピーまたは制御ファイルのコピーをリカバリ・カタログおよび制御ファイルに追加する。
- 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。これにより、このコピーを増分バックアップ計画の基礎とし、後で増分バックアップを実行することができるようになります。
- RMAN をインストールする前に作成した、Oracle 8 のデータベースのバックアップが存在していることを記録する。
- Oracle8i に移行する前に作成された、読取り専用または NORMAL モードでオフラインにされたデータベース・ファイルの Oracle7 バックアップが存在することを記録する。

O/S バックアップのカタログ化の時期

次のような場合は必ずバックアップをカタログ化します。

- O/S ユーティリティを使用してコピーを作成する。
- 移行前に、NORMAL モードでオフラインにされたまたは読取り専用になっていた Oracle7 の O/S バックアップをカタログ化する。

注意： バックアップ・セットおよびバックアップ・ピースを再カタログ化することはできません。

ホット O/S バックアップおよびコールド O/S バックアップのカタログ化 コールド O/S バックアップを行う場合、すなわち、UNIX の `cp` コマンドを使用してデータ・ファイルをコピーする場合は必ずカタログ化を行ってください。Oracle8i は、`ALTER TABLESPACE BEGIN/END BACKUP` コマンドをサポートしており、データベースの O/S バックアップはこれを使用してオープンします。RMAN 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、RMAN で認識できます。

カタログ化するバックアップは、次のようにしてください。

- ディスク上でアクセス可能である。
- 単一ファイルの完全なイメージ・コピーである。
- 一貫性または非一貫性データベース・バックアップ、表領域、データ・ファイル、制御ファイルまたはアーカイブ REDO ログのバックアップのいずれかであること。RMAN 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、ミラーを解除すれば O/S コピーとして使用できます。この例では、ミラーを解除した後で、`catalog` コマンドを使用して RMAN 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。ミラーを復元する場合は、`change ... uncatalog` コマンドを発行して、ファイル・コピーを削除することを、RMAN にあらかじめ通知します。

Oracle7 の O/S バックアップのカタログ化 RMAN は、次にあげる特殊な状況を除き、Oracle7 ファイルをカタログ化することはできません。Oracle7 から Oracle8i への移行時には、移行ユーティリティを実行する前に、Oracle7 データベースを正しく停止しておく必要があります。このとき、Oracle7 データ・ファイルの O/S バックアップを作成し、これを RMAN でカタログ化することができます。Oracle がこれらを受け入れることができるのは、これらを回復する際に、古いデータベースの REDO を必要としないからです。RMAN は、他のバックアップが存在しなければ、これらのバックアップを Oracle8i で復元できます。

RMAN でのカタログ化が可能な、Oracle7 バックアップの生成の例を次に示します。

1. データ・ファイルを読取り専用か、または NORMAL モードでオフラインにする。
2. データベースをクローズする。データベースをクローズする前に、ファイルを読み書き可能に戻さないでください。
3. オフラインまたは読取り専用ファイルの O/S バックアップを作成する。
4. データベースを Oracle8i に移行する。

移行前のバックアップは、移行に先立って作成するバックアップと同じ内容です。したがって、これも RMAN によるカタログ化が可能です。

O/S バックアップのカタログ化の方法

O/S バックアップの情報をリカバリ・カタログに波及させるには、**catalog** コマンドを使用します。

O/S バックアップをカタログ化するには

1. O/S ユーティリティを使用してバックアップを作成する。この例では、データ・ファイルをバックアップします。

```
% cp $ORACLE_HOME/dbs/sales.f $ORACLE_HOME/backup/sales.bak';
```

2. RMAN を起動し、ターゲット・データベースおよびリカバリ・カタログ・データベースに接続する。たとえば、次のように入力します。

```
% rman target change_on_install@prod1 catalog rman/rman@rcat
```

3. **catalog** コマンドを発行する。たとえば、次のように入力します。

```
catalog datafilecopy '$ORACLE_HOME/backup/sales.bak';
```

```
RMAN-03022: compiling command: catalog
```

```
RMAN-03023: executing command: catalog
```

```
RMAN-08050: cataloged datafile copy
```

```
RMAN-08513: datafile copy filename=/oracle/backup/sales.bak recid=121  
stamp=342972501
```

```
RMAN-03023: executing command: partial resync
```

```
RMAN-08003: starting partial resync of recovery catalog
```

```
RMAN-08005: partial resync complete
```

関連項目 : **catalog** コマンドの構文は、11-32 ページの「[catalog](#)」を参照してください。

リカバリ・カタログのバックアップと回復

バックアップおよび回復計画には、リカバリ・カタログを含めます。リカバリ・カタログをバックアップしていない場合に、ディスク・クラッシュが発生すると、データの一部またはすべてを失う可能性があります。このような、望ましくない結果を避けるため、リカバリ・カタログのバックアップおよび回復方法を検討します。

この項では、次の項目について説明します。

- [リカバリ・カタログのバックアップ](#)
- [リカバリ・カタログの回復](#)
- [リカバリ・カタログの再作成](#)

リカバリ・カタログのバックアップ

リカバリ・カタログのバックアップ計画を立案する場合は、次のガイドラインに従ってください。

- 定期的なバックアップの励行
- 物理バックアップに適合したメソッドの選択
- 適切な場所へのリカバリ・カタログの格納
- 論理バックアップの作成

定期的なバックアップの励行

リカバリ・カタログは、ターゲット・データベースと同じ頻度でバックアップします。たとえば、ターゲット・データベースに対し、全体データベース・バックアップを週に 1 回行っている場合は、毎回その直後にリカバリ・カタログもバックアップします。バックアップされたカタログには、その前のターゲット・バックアップのレコードが記載されます。したがって、カタログを復元する必要がある場合は、これを使用すればターゲット・データベースのバックアップも復元できます。

物理バックアップに適合したメソッドの選択

リカバリ・カタログをバックアップする場合、O/S バックアップまたは RMAN バックアップの選択が可能です。O/S バックアップを作成する利点は、RMAN またはリカバリ・カタログに依存せず、O/S コマンドを使用して復元できることです。O/S バックアップの手順は、第 13 章の「オペレーティング・システム環境でのバックアップの実行」を参照してください。

RMAN を使用してリカバリ・カタログ・データベースをバックアップする場合は、障害の発生を考慮したりリカバリ・カタログの復元方法を検討する必要があります。たとえば、次のような方法が選択できます。

- もう一つのリカバリ・カタログを別のデータベース内に、すなわち、バックアップする本番データベース内に作成する。カタログ作成の方法は、6-2 ページの「リカバリ・カタログの作成」を参照してください。
- リカバリ・カタログの制御ファイルを、リカバリ・カタログの RMAN メタデータ・リポジトリとして使用し、ALTER DATABASE BACKUP CONTROLFILE TO 'filename' 文を使用して、同制御ファイルを頻繁にバックアップする。O/S コマンドを使用した、制御ファイルのバックアップ方法は、13-11 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。

適切な場所へのリカバリ・カタログの格納

データベースの RMAN メタデータを含んだリカバリ・カタログは、ターゲット・データベースと同じデータベースには絶対に格納しないでください。たとえば、データベース PROD1 のカタログは、PROD1 に格納してはいけません。PROD1 のリカバリ・カタログは、

保護対象のデータとは分離された状態でのみ有効です。PROD1 が完全なメディア障害状態に陥ったとき、PROD1 のリカバリ・カタログが PROD1 に格納されていた場合は、回復に使用するカタログがなくなることになります。

このことは、リカバリ・カタログ・データベースをバックアップする場合に特に重要です。たとえば、データベース RCAT に、ターゲット・データベース PROD1 のリカバリ・カタログ・メタデータが含まれていたとします。リカバリ・カタログを使用して、RCAT をバックアップすることにしたのですが、このカタログをどこに格納すればよいかわかりません。RCAT のメタデータが含まれたカタログを、データベース RCAT にそのまま格納すると、メディア障害が発生して RCAT が失われた場合、RCAT の復元が困難になるのみでなく、PROD1 も保護されないままになってしまいます。

論理バックアップの作成

リカバリ・カタログのデータのバックアップを作成する場合は、Export ユーティリティを使用すると便利です。カタログのエクスポートは、リカバリ・カタログの復元が必要な場合、最も柔軟性があります。Oracle8i データベースであれば、既存のどのデータベースにも復元が可能になるためです。

コマンド行からリカバリ・カタログの論理エクスポートを実行する方法

1. コマンド行でエクスポートを実行する。その際、次の事を必ず行うようにします。

- a. リカバリ・カタログの所有者として接続する。
- b. OWNER オプションを指定する。
- c. 出力ファイルを指定する。

たとえば、データベース PROD1 のカタログの所有者が RMAN であれば、UNIX のコマンド行で次のとおり発行すれば、カタログをファイル `cat.dmp` にエクスポートできます。

```
% exp rman/rman@prod1 file=cat.dmp owner=rman
```

2. 出力をチェックし、成功したかどうかを確認する。

```
Export terminated successfully without warnings.
```

関連項目 : Export ユーティリティの使用方法は、『Oracle8i ユーティリティ・ガイド』を参照してください。O/S バックアップの作成方法は、[第 13 章の「オペレーティング・システム環境でのバックアップの実行」](#)を参照してください。

リカバリ・カタログの回復

カタログの回復方法は、カタログのバックアップ方法によって異なります。次の方法が選択できます。

- [O/S コマンドによるカタログの回復](#)

- [RMAN によるカタログの回復](#)
- [カタログの論理バックアップのインポート](#)

O/S コマンドによるカタログの回復

オペレーティング・システムのコマンドを使用してリカバリ・カタログをバックアップしていた場合は、O/S コマンドを使用してカタログのバックアップを復元し、SQL*Plus コマンドを発行してこれを回復します。方法は、リカバリ・カタログ全体を回復するか、またはリカバリ・カタログが格納されている表領域のみを回復するかによって異なります。手順は、[第 14 章の「オペレーティング・システム環境での回復の実行」](#)を参照してください。

RMAN によるカタログの回復

RMAN を使用してリカバリ・カタログを回復する場合は、[第 9 章の「Recovery Manager による復元と回復」](#)を参照してください。

カタログの論理バックアップのインポート

Export を使用して作成したリカバリ・カタログの論理バックアップは、Import を使用して回復します。リカバリ・カタログがメディア障害によって破損した場合は、次の操作を行います。

1. 別のデータベースに新規ユーザーを作成する。リカバリ・カタログ・データベースに新規ユーザーを作成するための推奨 SQL 構文は、6-2 ページの「[リカバリ・カタログの作成](#)」を参照してください。
2. カatalog・データを Export ファイルからインポートする。コマンド行でインポートを実行する。その際、次の事柄を必ず行ってください。
 - a. リカバリ・カタログの新規所有者として接続する。
 - b. FROMUSER パラメータを使用して、旧所有者を指定する。
 - c. TOUSER パラメータを使用して、新規所有者を指定する。
 - d. インポート・ファイルを指定する。

ここでは、次のようにします。

- データベース PROD1 のカタログの旧所有者は RMAN とする。
- 新規リカバリ・カタログ・データベース NEW_CAT のユーザーは RCAT とする。
- カatalogのエクスポートが含まれたファイルは cat.dmp とする。

```
imp userid=rcat/rcat_pwd@new_cat file=cat.dmp fromuser=rman touser=rcat
```

3. インポートされたカタログ・データを使用して、ターゲット・データベースの復元および回復を行う。

関連項目：Import ユーティリティの使用方法は、『Oracle8i ユーティリティ・ガイド』を参照してください。

リカバリ・カタログの再作成

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、カタログの再作成を実行する必要があります。

古いカタログの内容を部分的に再作成する場合は、2 通りの方法が選択できます。

- **catalog** コマンドを発行して、アーカイブ・ログ、バックアップ制御ファイルおよびデータ・ファイル・コピーを再カタログ化する。
- **resync catalog from backup controlfile** コマンドを使用して、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。RMAN は、再カタログ化す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、すでに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。そのようなレコードは、**change ... crosscheck** コマンドまたは **crosscheck backup** コマンドを発行して削除する必要があります。

リカバリ・カタログのアップグレード

RMAN の実行可能ファイルが必要とするものよりも古いリカバリ・カタログを使用している場合は、これをアップグレードする必要があります。たとえば、RMAN のバージョン 8.1 で、8.0 のリカバリ・カタログを使用している場合は、リカバリ・カタログをアップグレードする必要があります。

RMAN の実行可能ファイルが必要とするバージョンよりも新しいリカバリ・カタログを使用しているときに、**upgrade catalog** を発行するとエラーが表示されます。リカバリ・カタログが現行のものであれば、**upgrade catalog** コマンドの実行が RMAN では可能です。それにより、パッケージの再作成が必要に応じて随時可能になります。しかしアップグレードが必要というわけではありません。アップグレード時にエラー・メッセージが生成されていないかどうかメッセージ・ログをチェックしてください。

リカバリ・カタログをアップグレードする方法

1. RMAN を起動し、ターゲット・データベースおよびリカバリ・カタログ・データベースに接続する。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

```
RMAN-06008: connected to recovery catalog database
```

```
RMAN-06186: PL/SQL package rcat.DBMS_RCVCAT version 08.00.04 in RCVCAT
database is too old
```

2. **upgrade catalog** コマンドを発行する。

```
RMAN> upgrade catalog
```

```
RMAN-06435: recovery catalog owner is rman
```

```
RMAN-06442: enter UPGRADE CATALOG command again to confirm catalog upgrade
```

3. 確認のため、**update catalog** コマンドを再度入力する。

```
RMAN> upgrade catalog
```

```
RMAN-06408: recovery catalog upgraded to version 08.01.05
```

```
RMAN-06452: DBMS_RCVMAN package upgraded to version 08.01.05
```

```
RMAN-06452: DBMS_RCVCAT package upgraded to version 08.01.03
```

関連項目 : **upgrade catalog** コマンドの構文は、11-149 ページの「[upgradeCatalog](#)」を参照してください。互換性および移行の詳細は、『Oracle8i 移行ガイド』を参照してください。

リカバリ・カタログの削除

リカバリ・カタログが不要となった場合は、表領域からリカバリ・カタログのスキーマを削除します。**drop catalog** コマンドを実行すると、リカバリ・カタログからすべての情報が削除されます。したがって、リカバリ・カタログのスキーマのバックアップがない場合は、このカタログが管理する全ターゲット・データベースの全バックアップは使用不能になります。

複数のターゲット・データベースが登録されているカタログから、あるデータベースを1つのみ "登録解除" する場合、**drop catalog** コマンドの使用は適切ではありません。カタログを削除して、あるターゲット・データベースの情報を削除しようとすると、全ターゲット・データベースの情報が削除されてしまいます。

リカバリ・カタログのスキーマを削除する方法

1. RMAN を起動し、ターゲット・データベースおよびリカバリ・カタログ・データベースに接続する。

```
% rman catalog rman/rman@rcvcat
```

2. 確認のため、**drop catalog** コマンドを2回入力する。

```
RMAN> drop catalog
```

```
RMAN-06435: recovery catalog owner is rman
```

```
RMAN-06436: enter DROP CATALOG command again to confirm catalog removal
```

```
RMAN> drop catalog
```

注意： リカバリ・カタログを削除しても、バックアップのレコードは制御ファイルに残ります。これらのレコードを除去するには、制御ファイルを再作成してください。

関連項目： `drop catalog` コマンドの構文は、11-68 ページの「[dropCatalog](#)」を参照してください。

リカバリ・カタログを使用しない RMAN の管理方法

RMAN は、リカバリ・カタログがなくても問題なく機能します。リカバリ・カタログは、実際には情報を制御ファイルから取得しています。ただし、リカバリ・カタログを使用しない場合は、次のタスクを実行する必要があります。

- [制御ファイルのメタデータのメンテナンス](#)
- [制御ファイルのバックアップ](#)

関連項目： 制御ファイルの概説は、『Oracle8i 概要』を参照してください。バックアップおよび回復時の制御ファイルの重要性は、[第 6 章の「Recovery Manager のメタデータの管理」](#)を参照してください。制御ファイルの管理の詳細は、『Oracle8i 管理者ガイド』を参照してください。

制御ファイルのメタデータのメンテナンス

RMAN には、バックアップおよびコピーを物理的に削除するコマンドの他、バックアップのレコードをチェックしたり削除するコマンドが複数用意されています。これらのメンテナンス手順の詳細は、6-4 ページの「[RMAN メタデータのメンテナンス](#)」を参照してください。これらのコマンドのほとんどは、リカバリ・カタログの使用不使用にかかわらず有効です。

カタログが必要なコマンドのリストは、5-6 ページの「[制御ファイルを RMAN メタデータに使用した場合](#)」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でも、次のようなことが可能なため、メンテナンスの機能が失われるようなことはほとんどありません。

- `restore ... validate` コマンドまたは `validate backupset` コマンドを使用して、バックアップおよびコピーが使用可能かどうか、そして復元可能かどうかテストできる。この手順の説明は、6-18 ページの「[バックアップおよびコピーの復元の妥当性チェック](#)」を参照してください。
- `change ... delete` を使用して、古くなったレコードを制御ファイルから削除したり、必要ない、あるいは破損したバックアップおよびコピーを O/S から物理的に削除できる。この手順の説明は、6-13 ページの「[バックアップおよびコピーの削除と、RMAN メタデータにおけるそれらの状態の更新](#)」を参照してください。

制御ファイルのバックアップ

制御ファイルを RMAN メタデータの専用のリポジトリとして使用する場合は、多重化または O/S のミラー化を行って代替の制御ファイルをメンテナンスします。この制御ファイルは頻繁にバックアップするようにしてください。バックアップがないときに、制御ファイルが失われると、その内部の RMAN バックアップおよび RMAN コピーの情報もすべて失われてしまいます。

制御ファイルの多重化およびミラー化の方法は、2-3 ページの「[制御ファイルの管理](#)」を参照してください。制御ファイルの O/S バックアップの作成方法は、13-12 ページの「[物理ファイルへの制御ファイルのバックアップの作成](#)」および 13-12 ページの「[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](#)」を参照してください。

Recovery Manager によるリストおよびレポートの生成

この章では、Recovery Manager を使用して、バックアップおよびイメージ・コピーのリストおよびレポートを作成する方法を説明します。取り上げる項目は次のとおりです。

- [バックアップおよび回復計画におけるリストとレポートの使用](#)
- [リストの生成](#)
- [レポートの生成](#)
- [リストとレポートの例](#)

バックアップおよび回復計画におけるリストとレポートの使用

何をバックアップまたはコピーしたか、そして何をバックアップまたはコピーする必要があるかを確認するには、**report** コマンドおよび **list** コマンドを使用します。この情報は、リカバリ・カタログを使用するかしないかにかかわらず利用でき、効果的なバックアップ計画を策定する上でとても役に立ちます。RMAN のメタデータを、現行に合ったものにする方法は、第 6 章の「[Recovery Manager のメタデータの管理](#)」を参照してください。

list コマンドを実行すると、RMAN のバックアップ（バックアップ・セットおよびプロキシ・コピー）およびイメージ・コピーがすべて表示されます。**report** コマンドでは、さらに複雑な分析が可能となります。たとえば、バックアップが必要なデータ・ファイルと廃止されたバックアップ・ピースについてのレポートを生成することが可能です。Oracle では、**report** コマンドおよび **list** コマンドの出力を、スクリーンに表示することもログ・ファイルに書き込むこともできます。

関連項目： **list** コマンドの構文は、11-76 ページの「[list](#)」を参照してください。**report** コマンドの構文は、11-103 ページの「[report](#)」を参照してください。

リストの生成

リカバリ・カタログまたは制御ファイルに問合せを行い、その内容のリストを作成するには、**list** コマンドを使用します。**list** コマンドの目的は、どのバックアップまたはコピーが使用できるかを判断することです。たとえば、次のものをリスト表示できます。

- RMAN メタデータに記録された、バックアップ（バックアップ・セットおよびプロキシ・コピー）またはイメージ・コピーのすべて。
- 指定したデータベース、表領域、データ・ファイル、アーカイブ REDO ログ、制御ファイルのバックアップまたはイメージ・コピー。
- 時間、パス名、デバイス名、タグまたは回復可能性に基づいて限定した、バックアップおよびイメージ・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。

バックアップが必要なものを確認するには RMAN メタデータを使用します。具体的には、次のようにします。

- 出力表の STATUS 列に、使用できるバックアップおよびイメージ・コピーをすべてリスト表示する。
- バックアップするデータ・ファイル、アーカイブ REDO ログおよび制御ファイルをすべて出力に含める。
- バックアップおよびコピーを最新の状態にする。

イメージ・コピーおよびバックアップのリストの生成

1. RMAN を起動し、ターゲット・データベースに接続する。必要に応じて、リカバリ・カタログ・データベースにも接続します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

出力をログ・ファイルに書き込む場合は、起動時にログ・ファイルを指定します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat log '/oracle/log/mlog.f'
```

2. **list copy** コマンドおよび **list backup** コマンドを実行する。of *listObjList* パラメータを指定しない場合の、**list** のデフォルト設定は **of database** です。

```
list copy of database archivelog all; # lists datafiles and archived redo logs
list backup; # lists backup sets, backup pieces, and proxy copies
```

3. 出力をチェックする。**list** 出力の列見出しにはさまざまな種類があります。それぞれの説明は、11-76 ページの「**list**」を参照してください。サンプルの出力を次に示します。

```
list copy of database archivelog all;
```

List of Datafile Copies

Key	File	S	Completion time	Ckp SCN	Ckp time	Name
1262	1	A	18-AUG-98	219859	14-AUG-98	/vobs/oracle/dbs/copy/tbs_01.f

List of Archived Log Copies

Key	Thrd	Seq	S	Completion time	Name
789	1	1	A	14-JUL-98	/vobs/oracle/work/arc_dest/arcr_1_1.arc
790	1	2	A	11-AUG-98	/vobs/oracle/work/arc_dest/arcr_1_2.arc
791	1	3	A	12-AUG-98	/vobs/oracle/work/arc_dest/arcr_1_3.arc

```
list backup;
```

List of Backup Sets

Key	Recid	Stamp	LV	Set Stamp	Set Count	Completion Time
1174	12	341344528	0	341344502	16	14-AUG-98

List of Backup Pieces

Key	Pc#	Cp#	Status	Completion Time	Piece Name
1176	1	1	AVAILABLE	14-AUG-98	/vobs/oracle/dbs/0ga5h07m_1_1

```
Controlfile Included
```

Ckp SCN	Ckp time				
219857	14-AUG-98				
List of Datafiles Included					
File Name	LV	Type	Ckp SCN	Ckp Time	
1 /vobs/oracle/dbs/tbs_01.f	0	Full	199843	14-AUG-98	
2 /vobs/oracle/dbs/tbs_02.f	0	Full	199843	14-AUG-98	

オブジェクトまたはその他の条件で限定して、コピーおよびバックアップのリストを生成する方法

- 1. RMAN を起動し、ターゲット・データベースに接続する。必要に応じて、リカバリ・カタログ・データベースにも接続します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcvcat
```

- 2. オブジェクトで限定するには、**of listObjList** 条件を指定した上で、**list copy** または **list backup** を使用する。たとえば、次のように入力します。

```
list backup of database; # lists backups of all files in database
list copy of datafile '/oracle/dbs/tbs_1.f'; # lists copy of specified datafile
list backup of tablespace SYSTEM; # lists all backups of SYSTEM tablespace
list copy of archivelog all; # lists all archived redo logs and copies of logs
list backup of controlfile; # lists all control file backups
```

タグ、デバイス・タイプ、ファイル名のパターンまたは時間オプションを組み合わせ、検索を限定することもできます。たとえば、次のように入力します。

```
list backup tag 'weekly_full_db_backup'; # by tag
list copy of datafile '/oracle/dbs/tbs_1.f' type 'sbt_tape'; # by type
list backup like '/oracle/backup/tbs_4%'; # by filename pattern
list backup of archivelog until time 'SYSDATE-30'; # by time
list copy of datafile 2 completed between '10-DEC-1998' and '17-DEC-1998'; # by time
```

- 3. 出力をチェックする。ここでは、datafile 1 のコピーのリストのサンプル出力を示します。

```
RMAN> list copy of datafile 1;

RMAN-03022: compiling command: list

List of Datafile Copies
Key      File S Completion time Ckp SCN      Ckp time      Name
-----
3        1      A 18-DEC-98      114148      18-DEC-98      /vobs/oracle/dbs/df1.bak
```

関連項目 : *listObjList* の構文は、11-85 ページの「[listObjList](#)」を参照してください。**list** 出力の列にはさまざまな種類があります。それぞれの説明は、11-78 ページの「[List Output](#)」を参照してください。

レポートの生成

RMAN メタデータからさらに詳しい情報を取得する場合は、レポートを生成します。次のような事柄を調べるには、**report** コマンドを使用します。

- 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？
- 回復不能とされたファイルはどれか？
- 不要なバックアップまたはコピーはどれか？ それは削除可能か？
- 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？
- どのバックアップおよびコピーがディスクにあるか？ また、テープにあるのはどれか？

注意： レポートの正確性を期すために、RMAN メタデータは現行のものを使用する必要があります。また、**change**、**uncatalog** および **crosscheck** の各コマンドを適切に使用し、すべてのバックアップおよびコピーが正しいものとなるよう更新する必要もあります。RMAN のメタデータのメンテナンス方法は、6-4 ページの「[RMAN メタデータのメンテナンス](#)」を参照してください。

レポートから得た情報は、バックアップおよび回復計画を策定する上で非常に重要です。具体的には、**report need backup** コマンドおよび **report unrecoverable** コマンドを定期的に使用して次のようにします。

- 回復に必要なバックアップが利用できるようにする。
- 回復が妥当な時間内で完了するようにする。つまり、平均回復時間（MTTR）を最小限に抑えるようにする。

バックアップが必要なオブジェクトのレポートの作成

1. RMAN を起動し、ターゲット・データベースに接続する。必要に応じて、リカバリ・カタログ・データベースにも接続します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

出力をログ・ファイルに書き込むには、起動時にファイルを指定します。

```
% rman target / catalog rman/rman@rcat log "/oracle/log/mlog.f"
```

- 必要に応じて、**crosscheck** コマンドを発行し、バックアップの状態を更新する。
change ... crosscheck コマンドを発行すると、イメージ・コピーの状態を更新できます
(主キーを使用してイメージ・コピーを指定する場合は、**list** コマンドを発行すると主
キーが取得できます)。

クロスチェック・セッションの例を次に示します。

```
# must allocate maintenance channel for crosscheck
allocate channel for maintenance type disk;
crosscheck backup; # crosschecks all backups
change datafile copy 100,101,102,103,104,105,106,107 crosscheck; # specified by
key
change archivelog copy 50,51,52,53,54 crosscheck; # specified by key
release channel;
```

- need backup** オプションを使用して、どのデータ・ファイルに新規バックアップが必要かを識別し、日数のしきい値または増分バックアップ数のしきい値でレポートの範囲を限定する。RMAN は、**days** パラメータの値よりも古いバックアップは、すべて新規バックアップを必要としているものとして認識します。バックアップを回復するためには、アーカイブ REDO ログ数と同じ **days** が必要となるためです。

たとえば、次のように入力します。

```
report need backup days = 7 database; # needs at least 7 days of logs to
recover
report need backup days = 30 tablespace system;
report need backup days = 14 datafile '/oracle/dbs/tbs_5.f';
```

incremental パラメータも指定できます。データ・ファイルの完全な回復のために、指定した数より多くの増分バックアップが必要な場合、RMAN はそのデータ・ファイルに新規のバックアップが必要であるものとみなします。たとえば、次のように入力します。

```
report need backup incremental = 1 database;
report need backup incremental = 3 tablespace system;
report need backup incremental = 5 datafile '/oracle/dbs/tbs_5.f';
```

- レポートをチェックし、新規バックアップが必要なデータ・ファイルをバックアップする。

関連項目 : **report** 出力の列見出しにはさまざまな種類があります。それぞれの説明は、11-107 ページの「[Report Output](#)」を参照してください。

不要なバックアップのレポートの作成

- RMAN を起動し、ターゲット・データベースに接続する。必要に応じて、リカバリ・カタログ・データベースにも接続します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

出力をメッセージ・ログ・ファイルに書き込む場合は、起動時にそのファイルを指定します。

```
% rman target / catalog rman/rman@rcat log "/oracle/log/mlog.f"
```

2. 必要に応じて、**crosscheck** コマンドを発行し、バックアップの状態を更新する。
change ... crosscheck コマンドを発行すると、イメージ・コピーの状態を更新できます（主キーを使用してコピーを指定する場合は、**list** コマンドを発行すると主キーが取得できます）。

```
allocate channel for maintenance type disk;
crosscheck backup;
change datafile copy 100,101,102,103,104,105,106,107 crosscheck;
change archivelog copy 50,51,52,53,54 crosscheck;
release channel;
```

3. 回復に不要とされたバックアップを、**obsolete** オプションを使用して識別する。
redundancy パラメータを使用すると、バックアップまたはコピーの冗長性の最低レベルを指定できます（このパラメータを指定しない場合の **redundancy** のデフォルト値は 1 です）。

データ・ファイルのコピーは、少なくとも *integer* で指定した数以上の、このファイルの最新のバックアップが存在する場合には不要と判断されます。データ・ファイル・バックアップ・セットは、そのバックアップ・セットに含まれた各ファイルの最新のバックアップまたはイメージ・コピーが、少なくとも *integer* で指定した数以上存在する場合には不要と判断されます。たとえば、次のように入力します。

```
# Lists backups or copies that have at least 2 more recent backups or copies
report obsolete redundancy = 2;
```

指定した時間、SCN またはログ順序番号以前の最新のバックアップ・セットまたはコピーに対して冗長性チェックを実行する場合は、**untilClause** を使用します。

```
# Obsolete if there are at least 2 copies/backups that are no more than 2 weeks
old.
report obsolete redundancy = 2 until time 'SYSDATE-14';
report obsolete until scn 1000;
report obsolete redundancy = 3 until logseq = 121 thread = 1;
```

4. 現行のインカーネーションの直接の親元でないインカーネーションに属しているため使用できなくなっているバックアップおよびコピーを、**orphan** オプションを使用してリスト表示する。

```
report obsolete orphan;
```

孤立したバックアップの説明は、4-22 ページの「[孤立したバックアップのレポート作成](#)」を参照してください。

5. レポートをチェックし、不要なバックアップを削除する。

```

RMAN> report obsolete;

RMAN-03022: compiling command: report
Report of obsolete backups and copies
Type                Recid  Stamp      Filename
-----
Backup Set          4      345390311
Backup Piece        4      345390310 /oracle/dbs/04a9cf76_1_1

RMAN> allocate channel for delete type disk;

RMAN-03022: compiling command: allocate
RMAN-03023: executing command: allocate
RMAN-08030: allocated channel: delete
RMAN-08500: channel delete: sid=11 devtype=DISK

RMAN> change backuppiece '/oracle/dbs/04a9cf76_1_1' delete;

RMAN-03022: compiling command: change
RMAN-03023: executing command: change
RMAN-08073: deleted backup piece
RMAN-08517: backup piece handle=/oracle/dbs/04a9cf76_1_1 recid=4 stamp=345390310
RMAN-03023: executing command: partial resync
RMAN-08003: starting partial resync of recovery catalog
RMAN-08005: partial resync complete

```

回復不能なバックアップのレポートの作成

1. RMAN を起動し、ターゲット・データベースに接続する。必要に応じて、リカバリ・カタログ・データベースにも接続します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

出力をメッセージ・ログ・ファイルに書き込む場合は、起動時にそのファイルを指定します。

```
% rman target / catalog rman/rman@rcat log "/oracle/log/mlog.f"
```

2. **report** コマンドの **unrecoverable** オプションを使用して、最後のバックアップ以後、回復不能操作が行われたオブジェクトを含むデータ・ファイルはどれであることを確認する。

```
report unrecoverable database; # Examines all datafiles.
```


指定した時点のデータベース・スキーマのレポートの作成

過去のある時点、SCN またはログ順序番号のデータベース・スキーマのレポートを作成するには、リカバリ・カタログを使用する必要があります。

1. RMAN を起動し、ターゲット・データベースおよびリカバリ・カタログ・データベースに接続する。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

出力をメッセージ・ログ・ファイルに書き込む場合は、起動時にそのファイルを指定します。

```
% rman target / catalog rman/rman@rcat log "/oracle/log/mlog.f"
```

2. **report schema** を発行し、現行の状態のターゲット・データベースの全データ・ファイルおよび表領域のリストを取得する。

```
report schema;
```

untilClause を使用して、過去のある時点、SCN またはログ順序番号を指定する。

```
report schema at time 'SYSDATE-14';
report schema at scn 1000;
report schema at logseq 100;
```

3. 出力をチェックする。サンプル出力を次に示します。

```
RMAN> report schema at scn 1000;
RMAN-03022: compiling command: report
```

```
Report of database schema
File K-bytes      Tablespace          RB segs Name
-----
1          35840 SYSTEM                YES    /vobs/oracle/dbs/tbs_01.f
2           978 SYSTEM                YES    /vobs/oracle/dbs/tbs_02.f
3           978 TBS_1                  NO     /vobs/oracle/dbs/tbs_11.f
4           978 TBS_1                  NO     /vobs/oracle/dbs/tbs_12.f
5           978 TBS_2                  NO     /vobs/oracle/dbs/tbs_21.f
6           978 TBS_2                  NO     /vobs/oracle/dbs/tbs_22.f
```

回復する時点のデータベースのスキーマがわかるため、この種の情報は不完全回復に役立ちます。

関連項目 : **report** コマンドの構文は、11-103 ページの「**report**」を参照してください。**list** コマンドの構文は、11-76 ページの「**list**」を参照してください。

リストとレポートの例

レポートおよびリスト生成の例を次にいくつか挙げます。

- バックアップおよびコピーのリストの作成
- リストを使用した不要なバックアップおよびコピーの判別
- バックアップの必要なデータ・ファイルのレポート
- 回復不能データ・ファイルのレポート
- 不要なバックアップおよびコピーのレポート
- 不要なバックアップの削除
- データベース・スキーマの履歴レポートの生成

バックアップおよびコピーのリストの作成

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、これによってターゲット・データベースの制御ファイルの内容を問合せできます。リストを修飾するパラメータは複数あります。

次の例では、1998 年 11 月 1 日以降に作成された、表領域 TBS_1 のデータ・ファイルの全バックアップをリスト表示します。

```
list backup of tablespace tbs_1 completed before 'Nov 1 1998 00:00:00';
```

次の例では、メディア管理デバイス上のバックアップ・セットまたはプロキシ・コピーをすべてリスト表示します。

```
list backup of database device type 'sbt_tape';
```

次の例では、copy サブ・ディレクトリ内のタグ weekly_df2__copy を使用して、datafile 2 の全コピーをリスト表示します。

```
list copy of datafile 2 tag weekly_df2_copy like '/copy/%';
```

次の例では、リカバリ・カタログに登録されたデータベース・インカーネーションをすべてリスト表示します。

```
list incarnation of database;
```

リストを使用した不要なバックアップおよびコピーの判別

削除可能なバックアップおよびコピーを判別するには、**list** コマンドを使用します。たとえば、11 月 2 日にデータベースのフル・バックアップを作成し、このデータベースをそれより前の日付に回復する必要がないことがわかっている場合に、次のレポートにリスト表示されたバックアップおよびイメージ・コピーは削除できます。

```
list backup of database completed before 'Nov 1 1998 00:00:00';  
list copy completed before 'Nov 1 1998 00:00:00';
```

バックアップの必要なデータ・ファイルのレポート

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;
```

次のコマンドは、5 日間以上バックアップ（全体または増分）を作成していない、表領域 SYSTEM の全データ・ファイルがレポートされます。

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

次のコマンドは、2 つ以上のバックアップまたはコピーがテープに格納されていないために、バックアップが必要となっているデータ・ファイルは 1 ~ 5 のうちどれかをレポートします。

```
report need backup redundancy 2 datafile 1,2,3,4,5 device type 'sbt_tape';
```

回復不能データ・ファイルのレポート

次の例では、テープ上のデータ・ファイルのうち、最新の完全または増分バックアップ後に行われた変更が記録されていないために、新規バックアップが必要となっているものをすべてレポートします。

```
report unrecoverable database device type 'sbt_tape';
```

不要なバックアップおよびコピーのレポート

次のコマンドは、ディスク上のバックアップおよびコピーのうち、最新バックアップまたはコピーがすでに 3 つ以上使用可能になっているために、不要と判断されたものをすべてレポートします。

```
report obsolete redundancy 3 device type disk;
```

次のコマンドは、テープ上のバックアップのうち、1 週間以内に作成されたバックアップが少なくともすでに 2 つ存在するため、不要と判断されたものをすべてレポートします。

```
report obsolete redundancy 2 until time 'SYSDATE-7' device type 'sbt_tape';
```

次のコマンドは、現行のインカーネーションの直接の親元でないインカーネーションに属しているため、不要と判断されたデータ・ファイルをレポートします。

```
report obsolete orphan;
```

たとえば、インカーネーション A とインカーネーション B という、2 つの子を有する親インカーネーションで、インカーネーション B が現行のインカーネーションであるとする、A の子は親のない孤立した状態となります。

不要なバックアップの削除

ここでは、次のものを削除すると仮定します。

- 最新のコピーが少なくとも 2 つあるデータ・ファイル・コピー。
- バックアップ・セットに含まれた、各データ・ファイルの最新のバックアップまたはイメージ・コピーが、少なくとも 2 つあるデータ・ファイル・バックアップ。

1. 冗長性を 2 に設定してレポートを生成する。

```
report obsolete redundancy 2;
```

```
RMAN-03022: compiling command: report
```

```
Report of obsolete backups and copies
```

Type	Recid	Stamp	Filename
Datafile Copy	23	345392880	/vobs/oracle/dbs/tbs_01.copy
Datafile Copy	22	345392456	/vobs/oracle/dbs/tbs_01_copy.f
Backup Set	31	345552065	
Backup Piece	31	345552061	/vobs/oracle/dbs/0va9hd5o_1_1
Backup Set	23	345399397	
Backup Piece	23	345399391	/vobs/oracle/dbs/0ma9co2p_1_1
Backup Set	20	345397468	
Backup Piece	20	345397464	/vobs/oracle/dbs/0ka9cm6l_1_1

2. **change ... delete** コマンドを、コピーおよびバックアップに対して発行する。主キーを取得するには、ファイル名を使用するか、または **list** コマンドを発行します。

```
allocate channel for delete type disk;
change backuppiece '/vobs/oracle/dbs/0va9hd5o_1_1', '/vobs/oracle/dbs/0ma9co2p_1_1',
'/vobs/oracle/dbs/0ka9cm6l_1_1' delete;
change datafilecopy '/vobs/oracle/dbs/tbs_01.copy', '/vobs/oracle/dbs/tbs_01_copy.f'
delete;
release channel;
```

3. 必要に応じて、SQL*Plus セッションを開始して、リカバリ・カタログ・ビューに対して問合せを行い、コピーおよびバックアップ・ピースの状態が **deleted** に変更されたかチェックする（**list** 出力には、**deleted** 状態のファイルは表示されません）。

```
SQL> SELECT handle FROM rc_backup_piece WHERE status = 'D';
SQL> SELECT name FROM rc_datafile_copy WHERE status = 'D';
```

データベース・スキーマの履歴レポートの生成

次のコマンドは、現在、1 週間前、2 週間前、1 か月前の各データベース・スキーマをレポートします。

```
report schema;  
report schema at time 'SYSDATE-7';  
report schema at time "TO_DATE('12/20/98','MM/DD/YY')";
```

次のコマンドは、SCN953 のデータベース・スキーマをレポートします。

```
report schema at scn 953;
```

次のコマンドは、スレッド 2 のログ順序番号 12 のデータベース・スキーマをレポートします。

```
report schema at logseq 12 thread 2;
```

Recovery Manager でのバックアップおよびコピーの作成

この章では、Recovery Manager を使って、バックアップおよびコピーの操作を管理する方法を説明します。項目は次のとおりです。

- [バックアップの作成](#)
- [イメージ・コピーの作成](#)
- [バックアップおよびコピー処理の監視](#)
- [バックアップおよびコピーの例](#)

バックアップの作成

次に示すオブジェクトをバックアップするには、RMAN の **backup** コマンドを使用します。

- データベース（全データ・ファイルおよび現行の制御ファイル）
- 表領域
- データ・ファイル（現行またはイメージ・コピー）
- アーカイブ REDO ログ
- 制御ファイル（現行またはイメージ・コピー）

RMAN は、これらのファイルをバックアップする際、1 つ以上のバックアップ・セットにまとめます。**backup** コマンドに対しては、バックアップ・ピースのファイル名、各セットに組み込むファイル数、そして各入力ファイルで操作するチャンネルを指定するためのパラメータを設定できます。

RMAN バックアップは、データベースがオープンされているときでもクローズされているときも作成できます。クローズ状態のバックアップが、一貫性バックアップになるか非一貫性バックアップになるかは、データベースの停止の方法によって異なります。オープン状態のバックアップは、必ず非一貫性のバックアップとなります。一貫性バックアップは、回復を実行することなく復元できます。非一貫性バックアップは、復元時になんらかのメディア回復が必要となります。ただしその他の点では、一貫性バックアップと同様に有効です。

RMAN バックアップは、さらに全体バックアップと増分バックアップに分けられます。全体バックアップとは非増分のバックアップです。すなわち、使用済みのブロックがすべてバックアップされます。

この項では、次の手順について説明します。

- [一貫性バックアップおよび非一貫性バックアップの作成](#)
- [全体データベース・バックアップの作成](#)
- [表領域およびデータ・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 REDO ログのバックアップ](#)
- [増分バックアップの作成](#)

関連項目 : RMAN バックアップの概要は、4-28 ページの「[バックアップ・セット](#)」を参照してください。**backup** コマンドの構文は、11-21 ページの「[backup](#)」を参照してください。RMAN バックアップの各タイプの説明は、4-36 ページの「[バックアップ・タイプ](#)」を参照してください。

一貫性バックアップおよび非一貫性バックアップの作成

この章では、データベースがオープン状態のとき、またはクローズ状態のときの、バックアップの作成手順について説明します。クローズ状態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通りがあります。オープン状態のバックアップは、非一貫性バックアップのみです。Oracle では、NOARCHIVELOG モードでの非一貫性バックアップは不可能であることに注意してください。

一貫性バックアップを作成する場合、データベースは次のようになっている必要があります。

- マウントされているがオープンされていない。
- 最後にオープンされたときに、クラッシュまたは異常終了していない。

これらの条件が満たされていない場合、そのバックアップは非一貫性バックアップとなります。

全体データベース・バックアップの作成

データベースをクローズしても差し支えない場合は、クローズ状態のデータベース全体の一貫性バックアップを作成することをお勧めします。データベースを停止できない場合はオープン・バックアップのみ作成可能です。

全体データベース・バックアップの作成

1. RMAN を起動し、ターゲット・データベースに接続する。必要に応じて、リカバリ・カタログ・データベースにも接続します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

出力をログ・ファイルに書き込むには、起動時にファイルを指定します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat log /oracle/log/mlog.f
```

2. 一貫性バックアップを作成する場合は、正しく停止されていたことを確認してからデータベースをマウントする（ただしオープンしない）。データベースがオープンされている場合、またはマウント状態でも前回オープンされたときに正しく停止されていない場合は、非一貫性バックアップとなります。
3. type **disk** のチャネル、または type '**sbt_tape**' チャネルを 1 つ以上割り当てる。この例では、全データ・ファイルおよび制御ファイルをバックアップします。ここでは **format** パラメータを指定していません。したがって、RMAN は、各バックアップ・ピースに一意の名前を自動的に割り当て、これらをポート固有のデフォルトの位置に格納します（UNIX の場合は \$ORACLE_HOME/dbs ）。

```
run {
    allocate channel ch1 type disk;
```

```
backup database;
sql 'ALTER SYSTEM ARCHIVE LOG CURRENT'; # archives current redo log
sql 'ALTER SYSTEM ARCHIVE LOG ALL'; # archives all un-archived redo logs
}
```

必要に応じて、**format** パラメータを使用して、バックアップ・ピースのファイル名を指定します。たとえば、次のように入力します。

```
run {
  allocate channel ch1 type 'sbt_tape';
  backup database
  format '/oracle/backup/%U'; # %U generates a unique filename
}
```

必要に応じて、**tag** パラメータを使用して、バックアップ・ピースのタグを指定します。たとえば、次のように入力します。

```
run {
  allocate channel ch1 type 'sbt_tape';
  backup database
  tag = 'weekly_backup'; # gives the backup a tag identifier
}
```

4. **list** コマンドを発行して、バックアップ・セットおよびバックアップ・ピースのリストを表示する。

表領域およびデータ・ファイルのバックアップ

表領域およびデータ・ファイルのバックアップは、データベースがオープン状態のときでもクローズ状態のときでも実行できます。オープン・データベース・バックアップは、必ず非一貫性バックアップとなることに注意してください。ALTER DATABASE BEGIN BACKUP は、オンライン表領域バックアップの作成前には発行しないでください。

表領域のバックアップ

1. RMAN を起動し、ターゲット・データベースに接続する。必要に応じて、リカバリ・カタログ・データベースにも接続します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

出力をメッセージ・ログ・ファイルに書き込む場合は、起動時にそのファイルを指定します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat log "/oracle/log/mlog.f"
```

2. 一貫性バックアップを作成する場合は、正しく停止されていたことを確認してからデータベースをマウントする（ただしオープンしない）。データベースがオープンまたはマ

ウントされていても、前回オープンされたときに正しく停止されていない場合は、非一貫性バックアップとなります。

3. 1 つ以上のチャンネルを割り当てた後、**backup** コマンドを発行する。この例では、3 つの表領域と制御ファイルをバックアップします。その際、**filesperset** パラメータを使用して、各バックアップ・セットに 4 つ以上のデータ・ファイルが組み込まれないよう指定しています。

```
run {
    allocate channel ch1 type disk;
    allocate channel ch2 type disk;
    allocate channel ch3 type disk;
    backup filesperset = 3
        tablespace inventory, sales
        include current controlfile;
}
```

4. **list** コマンドを発行して、バックアップ・セットおよびバックアップ・ピースのリストを表示する。

データ・ファイルのバックアップ

1. RMAN を起動し、ターゲット・データベースに接続する。必要に応じて、リカバリ・カタログ・データベースにも接続します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

出力をメッセージ・ログ・ファイルに書き込む場合は、起動時にそのファイルを指定します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat log "/oracle/log/mlog.f"
```

2. 一貫性バックアップを作成する場合は、正しく停止されていたことを確認してからデータベースをマウントする（ただしオープンしない）。データベースがオープンまたはマウントされていても、前回オープンされたときに正しく停止されていない場合は、非一貫性バックアップとなります。
3. 1 つ以上のチャンネルを割り当てた後、**backup** コマンドを発行する。この例では、データ・ファイル 1 ~ 6 と制御ファイルのイメージ・コピーをバックアップします。

```
run {
    allocate channel ch1 type disk;
    backup
        (datafile 1,2,3,4,5,6
        filesperset 3)
    datafilecopy '/oracle/copy/tbs_1_c.f';
}
```

4. **list** コマンドを発行して、バックアップ・セットおよびバックアップ・ピースのリストを表示する。

データ・ファイル・コピーのバックアップ

1. RMAN を起動し、ターゲット・データベースに接続する。必要に応じて、リカバリ・カタログ・データベースにも接続します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

出力をメッセージ・ログ・ファイルに書き込む場合は、起動時にそのファイルを指定します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat log '/oracle/log/mlog.f'
```

2. データベースをマウントまたはオープンする。
3. 1 つ以上のチャンネルを割り当てた後、**backup** コマンドを発行する。この例では、データ・ファイル・コピー `df1.copy` をテープにバックアップします。

```
run {  
    allocate channel ch1 type 'sbt_tape';  
    backup datafilecopy '/oracle/copy/df1.copy';  
}
```

4. **list** コマンドを発行して、バックアップ・セットおよびバックアップ・ピースのリストを表示する。

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は、データベースがオープン状態のときもクローズ状態のときも作成できます。RMAN は、スナップショット制御ファイルを使用して、読取り一貫性を確保します。

全体データベース・バックアップには、現行の制御ファイルが自動的に含まれます。ただし、現行の制御ファイルには、全体データベース・バックアップのレコードは含まれていません。全体データベース・バックアップのレコードを含む制御ファイル・バックアップを取得するには、全体データベース・バックアップを実行した後で、制御ファイルのバックアップを作成します。

include current controlfile オプションを指定すれば、どのようなバックアップにも制御ファイルのバックアップを含めることができます。

現行の制御ファイルのバックアップ

1. RMAN を起動し、ターゲット・データベースに接続する。必要に応じて、リカバリ・カタログ・データベースにも接続します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

出力をログ・ファイルに書き込む場合は、起動時にログ・ファイルを指定します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat log "/oracle/log/mlog.f"
```

2. データベースをマウントまたはオープンする。
3. 1 つ以上のチャンネルを割り当てた後、**backup** コマンドを発行する。この例では、現行の制御ファイルをテープにバックアップし、さらにタグを使用します。

```
run {
    allocate channel chl type 'sbt_tape';
    backup current controlfile
    tag = mondayPMbackup;
}
```

4. 必要に応じて、**list** コマンドを発行して、バックアップ・セットおよびバックアップ・ピースのリストを表示する。

制御ファイル・コピーのバックアップ

1. RMAN を起動し、ターゲット・データベースに接続する。必要に応じて、リカバリ・カタログ・データベースにも接続します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

出力をログ・ファイルに書き込む場合は、起動時にログ・ファイルを指定します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat log "/oracle/log/mlog.f"
```

2. データベースをマウントまたはオープンする。
3. 1 つ以上のチャンネルを割り当てた後、**backup** コマンドを発行する。この例では、制御ファイル・コピー '/oracle/copy/cf.f' をバックアップします。

```
run {
    allocate channel chl type disk;
    backup controlfilecopy '/oracle/copy/cf.f';
}
```

4. 必要に応じて、**list** コマンドを発行して、バックアップ・セットおよびバックアップ・ピースのリストを表示する。

現行の制御ファイルの別のバックアップへの組み込み

バックアップ・オブジェクトを指定した後、**include current controlfile** オプションを指定します。たとえば、次のコマンドは、表領域 FOO をバックアップし、このバックアップに現行の制御ファイルを含めます。

```
run {
    allocate channel c1 type disk;
    backup tablespace foo
        include current controlfile;
}
```

アーカイブ REDO ログのバックアップ

アーカイブ REDO ログは、正常な回復の鍵となるものです。バックアップを定期的に行ってください。

backup コマンドに **delete input** オプションを指定すると、バックアップの済んだアーカイブ REDO ログが削除されます。したがって、アーカイブ・ログをテープにバックアップすることと、古いログは消去してディスク領域を空けることを、1つのステップで実行できます。

一連のアーカイブ REDO ログをバックアップするには

1. RMAN を起動し、ターゲット・データベースに接続する。必要に応じて、リカバリ・カタログ・データベースにも接続します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

出力をログ・ファイルに書き込む場合は、起動時にログ・ファイルを指定します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat log "/oracle/log/mlog.f"
```

2. データベースをマウントまたはオープンする。
3. 1つ以上のチャンネルを割り当てた後、**backup** コマンドを発行する。この例では、アーカイブ REDO ログをすべてテープにバックアップし、ディスクから削除します。

```
run {
    allocate channel chl type 'sbt_tape';
    backup archivelog all          # Backs up all archived redo logs.
        delete input;            # Optionally, delete the input logs
}
```

時間、SCN またはログ順序番号で、アーカイブ REDO ログの範囲を指定することもできます。この例では、8 ~ 29 日前に作成されたすべてのアーカイブ・ログをバックアップします。

```
run {
    allocate channel chl type disk;
    backup archivelog
        from time 'SYSDATE-30' until time 'SYSDATE-7';
}
```

4. **list** コマンドを発行して、バックアップ・セットおよびバックアップ・ピースのリストを表示する。

増分バックアップの作成

一貫性、非一貫性増分バックアップは、データベース、個別の表領域またはデータ・ファイルに対して行うことができます。ここでは、正しく停止したデータベースの増分バックアップの作成手順を説明します。

一貫性増分バックアップの作成

1. RMAN を起動し、ターゲット・データベースに接続する。必要に応じて、リカバリ・カタログ・データベースにも接続します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

2. 一貫性バックアップを作成する場合は、正しく停止されていたことを確認してからデータベースをマウントする（ただしオープンしない）。データベースがオープンまたはマウントされていても、前回オープンされたときに正しく停止されていなければ、非一貫性バックアップとなります。
3. 1 つ以上のチャンネルを割り当てた後、**backup** コマンドを発行する。この例では、レベル 0 のデータベースのバックアップを作成します。

```
run {
    allocate channel chl type disk;
    backup
        incremental level = 0
        database;
}
```

この例では、差分レベル 1 で、SYSTEM 表領域およびデータ・ファイル `sales.f` のバックアップを作成します。この場合、レベル 1 またはレベル 0 の最新のバックアップ後に変更されたデータ・ブロックのみがバックアップされます。

```
run {
    allocate channel chl type disk;
    backup
        incremental level = 1
        tablespace system
        datafile '/oracle/dbs/sales.f';
}
```

この例では、累積レベル 2 で、表領域 TBS_2 のバックアップを作成します。この場合、レベル 1 またはレベル 0 の最新のバックアップ後に変更されたデータ・ブロックのみがバックアップされます。

```
run {
```

```
allocate channel ch1 type disk;
backup
  incremental level = 2 cumulative # specify cumulative option
  tablespace tbs_1;
}
```

4. 必要に応じて、**list** コマンドを発行して、バックアップ・セットおよびバックアップ・ピースのリストを表示する。

イメージ・コピーの作成

ほとんどの場合、バックアップを作成するよりコピーを作成する方がよりよい選択です。コピーは RMAN 専用の形式にならず、追加処理を行わずに使用するのに適しているためです。これに対して、バックアップ・セットは、**restore** コマンドで処理を実行しないと使用できません。したがって、メディア回復は、データ・ファイル・コピーに対しては実行できません。バックアップ・セットに対しては直接実行できません。これはバックアップ・セットに含まれているデータ・ファイルが1つのみで、かつ1つのバックアップ・ピースで構成されている場合も同じです。

注意： 増分コピーは作成できませんが、**level** パラメータを使用して、コピーを今後の増分バックアップ・セットの基礎とすることは可能です。

イメージ・コピーを作成するには、**copy** コマンドを使用します。RMAN は、出力ファイルを常にディスクに書き込みます。コピーが可能なタイプのファイルは次のとおりです。

- データ・ファイル（現行またはコピー）
- アーカイブ REDO ログ
- 制御ファイル（現行またはコピー）

全データベース・ファイルの一貫性コピーの作成

1. RMAN を起動し、ターゲット・データベースに接続する。必要に応じて、リカバリ・カタログ・データベースにも接続します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

2. 一貫性バックアップを作成する場合は、正しく停止されていたことを確認してからデータベースをマウントする（ただしオープンしない）。データベースがオープンまたはマウントされていても、前回オープンされたときに正しく停止されていなければ、非一貫性バックアップとなります。

3. データベースの現行のスキーマのレポートを生成する。

```
RMAN> report schema;
```

```
RMAN-03022: compiling command: report
```

```
Report of database schema
```

File	K-bytes	Tablespace	RB	segs	Name
1	35840	SYSTEM	YES		/oracle/dbs/tbs_01.f
2	978	SYSTEM	YES		/oracle/dbs/tbs_02.f
3	978	TBS_1	NO		/oracle/dbs/tbs_11.f
4	978	TBS_1	NO		/oracle/dbs/tbs_12.f

4. データ・ファイルをすべてコピーし、現行の制御ファイルを組み込む。たとえば、次のように入力します。

```
run {
  allocate channel chl type disk;
  copy
    datafile 1 to '/oracle/copy/df_1.f',
    datafile 2 to '/oracle/copy/df_2.f',
    datafile 3 to '/oracle/copy/df_3.f',
    datafile 4 to '/oracle/copy/df_4.f',
    current controlfile to '/oracle/copy/cf.f';
}
```

5. list copy コマンドを発行して、コピーのリストを表示する。

データ・ファイル、アーカイブ REDO ログおよび制御ファイルのコピー

1. RMAN を起動し、ターゲット・データベースに接続する。必要に応じて、リカバリ・カタログ・データベースにも接続します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

2. 一貫性バックアップを作成する場合は、正しく停止されていたことを確認してからデータベースをマウントする（ただしオープンしない）。データベースがオープンまたはマウントされていても、前回オープンされたときに正しく停止されていなければ、非一貫性バックアップとなります。
3. 必要なデータ・ファイル、アーカイブ REDO ログおよび制御ファイルをコピーする。たとえば、次のように入力します。

```
run {
  allocate channel chl type disk;
  # allocate multiple channels for parallelization. For parallelization,
  issue one
  # copy command rather than multiple copy commands.
  allocate channel ch2 type disk;
```

```
allocate channel ch3 type disk;
copy
  # copy datafiles and datafile copies
  datafile '/oracle/dbs/tbs_8.f' to '/oracle/copy/df_8.f',
  datafilecopy '/oracle/copy/df_2.cp' to '/oracle/dontouch/df_2.f',
  datafilecopy tag = 'weekly_dfl_copy' to '/oracle/copy/df_1.f',

  # copy archived redo logs
  archivelog '/oracle/arc_dest/arcr_1_1.arc' to '/oracle/copy/arcr_1_1.arc',
  archivelog '/oracle/arc_dest/arcr_1_2.arc' to '/oracle/copy/arcr_1_2.arc',

  # copy a control file copy
  controlfilecopy '/oracle/copy/cf.f' to '/oracle/dontouch/cf.f';
}
```

4. **list copy** コマンドを発行して、コピーのリストを表示する。

バックアップおよびコピー処理の監視

バックアップまたはコピー処理を行うサーバー・セッションが、現在何を実行しているかを調べる必要があります。RMAN では、次のチェックを実行することができます。

- [サーバー・セッションとチャネルの関連付け](#)
- [進行状況の監視](#)
- [パフォーマンスの監視](#)

サーバー・セッションとチャネルの関連付け

どのサーバー・セッションがどの RMAN チャネルに対応しているかを調べるには、**command id** パラメータを指定した **set** コマンドを使用します。**command id** パラメータは、指定した文字列を、V\$SESSION 動的パフォーマンス・ビューの CLIENT_INFO 列に入力するものです。サーバー・セッションとチャネルを関連付けるには、V\$SESSION と V\$PROCESS を使用します。

V\$SESSION の CLIENT_INFO 列には、Recovery Manager の各サーバー・セッションの情報が表示されます。このデータは、次のいずれかの形式で表示されます。

- **id=string**
この形式は、RMAN 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用されます。
- **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用されます。

V\$PROCESS の SPID 列には、O/S のプロセス番号が示されます。

関連項目：V\$SESSION および V\$PROCESS の詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。

バックアップ時のプロセスとチャネルの関連付け

1. RMAN を起動し、ターゲット・データベースに接続する。必要に応じて、リカバリ・カタログ・データベースにも接続します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

2. チャネルを割り当て、**command id** パラメータを設定した後、必要なオブジェクトをバックアップする。たとえば、次のように入力します。

```
run {
  allocate channel t1 type disk;
  allocate channel t2 type disk;
  set command id to 'rman';
  backup
    incremental level 0
    filesperset 5
    tablespace 'SYSTEM';
  # optionally, issue a host command to access the O/S prompt
  host;
  sql 'ALTER SYSTEM ARCHIVE LOG ALL';
}
```

3. RMAN ジョブの実行中に、SQL*Plus セッションを開始し、V\$SESSION と V\$PROCESS のビューに問合せを行う。たとえば、次のように入力します。

```
SELECT sid, spid, client_info
FROM v$process p, v$session s
WHERE p.addr = s.paddr
AND client_info LIKE '%id=rman%';
```

SID	SPID	CLIENT_INFO
8	21973	id=rman
16	22057	id=rman
17	22068	id=rman,ch=t1
18	22070	id=rman,ch=t2

関連項目：set command id の構文は、11-134 ページの「[set_run_option](#)」を参照してください。

進行状況の監視

バックアップ、コピーおよび復元の進行状況を監視するには、ビュー V\$SESSION_LONGOPS に問合せを行います。

バックアップ、復元またはコピーを実行する各サーバー・セッションは、復元の各特定の部分処理に要する全作業量に対する進行状況をレポートします。たとえば、2つのチャンネルを使用して復元を行い、チャン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・セッションによって、1つのセットの経過についての進行状況のレポートが行われます。そのセットが完全に復元された後、RMAN は、次に復元するセットについて、進行状況のレポートを開始します。

この情報は、次の SQL 文を使用して問い合わせます。

```
SELECT sid, serial#, context, sofar, totalwork  
       round(sofar/totalwork*100,2) "% Complete",  
FROM v$session_longops  
WHERE opname LIKE 'RMAN:%'  
AND opname NOT LIKE 'RMAN: aggregate%';
```

関連項目：V\$SESSION_LONGOPS の詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。進行状況およびパフォーマンス情報の詳細は、V\$BACKUP_SYNC_IO および V\$BACKUP_ASYNC_IO を参照してください。

パフォーマンスの監視

バックアップおよび復元のパフォーマンスを監視するには、V\$BACKUP_SYNC および V\$BACKUP_ASYNC_IO に問合せを行います。これらのビューの内容、およびこれらを使用したバックアップ・パフォーマンスの調整方法の詳細は、『Oracle8i チューニング』を参照してください。

関連項目：これらのビューの詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。

バックアップおよびコピーの例

バックアップおよびコピーを行う際の使用例を次に示します。

- [バックアップの必要なデータ・ファイルのレポート](#)
- [データベースのバックアップ時のファイルのスキップ](#)
- [バックアップの複数ディスク・ドライブへの分散](#)
- [大規模データベースの複数のファイル・システムへのバックアップ](#)
- [バックアップ・セットのサイズ指定](#)
- [バックアップにおけるデータ・ファイルの多重化](#)

- アーカイブ REDO ログのバックアップ
- アーカイブ REDO ログの複数のコピーのバックアップおよび削除
- 差分増分バックアップの実行
- 累積増分バックアップの実行
- バックアップ・セットの二重化
- バックアップ・セットの平行化
- NOARCHIVELOG モードでのバックアップ
- 平行・サーバー環境でのバックアップ
- O/S コピーのカatalog化
- バックアップおよびコピーのメンテナンス
- バックアップおよびコピー時のエラー処理

バックアップの必要なデータ・ファイルのレポート

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 6 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 6 database;
```

次のコマンドは、5 日間以上バックアップ（全体または増分）していない、表領域 SYSTEM の全データ・ファイルをレポートします。

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

データベースのバックアップ時のファイルのスキップ

次の例（データベースは ARCHIVELOG モードで実行されていると仮定します）は、データベースの一般的なバックアップ方法（オフラインの表領域はスキップする）を示しています。

```
run {
  allocate channel dev1 type 'sbt_tape';
  backup database
    skip readonly
    skip offline;
}
```

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の 1 回のみです。**skip readonly** オプションを使うと、読取り専用の表領域をスキップできます。**skip offline** オプションを使用すると、**backup** はオフライン・データ・ファイルへのアクセスを

試行しなくなります。このオプションは、オフライン・データ・ファイルが使用できない場合に使用してください。

バックアップの複数ディスク・ドライブへの分散

通常、テープへのバックアップを実行する場合、形式の指定は必要ありません。デフォルトの %U 変換変数によって、すべてのテープ・バックアップに一意のファイル名が付与されるためです。ディスクへのバックアップの際、バックアップを複数のディスク・ドライブに分散してパフォーマンスを向上させる場合は、形式を指定することができます。この場合、ディスク・ドライブ 1 つにつき 1 つの **disk** チャネルを割り当て、形式文字列を **allocate channel** コマンドに指定します。形式の指定は、各ファイル名が別のディスクに割り当てられるよう行います。

たとえば、次のように発行します。

```
run {
    allocate channel disk1 type disk format '/disk1/%d_backups/%U';
    allocate channel disk2 type disk format '/disk2/%d_backups/%U';
    backup database;
}
```

大規模データベースの複数のファイル・システムへのバックアップ

この例では、40 GB のデータベースをディスクにバックアップするものとします。RMAN は、ロー・ディスクへのバックアップを行わないため、複数のファイル・システムにバックアップを分散する必要があります。そこで、各バックアップ・セットをほぼ同じサイズ (10GB) にし、4 つのファイル・システムにバックアップすることにしました。各バックアップ・セットに 5 つのバックアップ・ピースを組み込むため、各バックアップ・ピースを 2GB 以下としようと考えています。

allocate channel コマンドの、**format** パラメータを使用し、各チャネルがそれぞれ別のファイル・システムに書き込みを行うよう設定することにした。置換変数を使用して、各バックアップ・ピースに一意の名前が付与されるようにします。たとえば、次の RMAN スクリプトを使用すると、バックアップが 4 つのファイル・システム (/fs1、/fs2、/fs3、/fs4) に分散され、これらのディレクトリに 4 つのバックアップ・セットが作成されます。また、各バックアップ・セットがほぼ同じサイズになるよう、データ・ファイルがグループ化されます。

```
run {
    allocate channel fs1 type disk format='/fs1/%u.%p';
    allocate channel fs2 type disk format='/fs2/%u.%p';
    allocate channel fs3 type disk format='/fs3/%u.%p';
    allocate channel fs4 type disk format='/fs4/%u.%p';

    setlimit channel fs1 kbytes=2000000; #limit file size to 2Gb
    setlimit channel fs2 kbytes=2000000;
    setlimit channel fs3 kbytes=2000000;
    setlimit channel fs4 kbytes=2000000;
}
```

```

    backup database;
}

```

バックアップ・セットのサイズ指定

バックアップを作成する際、RMAN は、バックアップが必要なファイルの合計数を、割り当てられたチャンネル数で割り、各バックアップ・セットに配分するファイル数を算出します。この計算結果を上書きし、各バックアップ・セットに組み込むファイル数を指定するには、**filesperset** パラメータおよび **setsize** パラメータを使用します。

filesperset の使用方法

filesperset パラメータを指定すると、RMAN は **filesperset** の値と、自動的に算出された値（ファイル数 / 割り当てられたチャンネル数）を比較し、いずれか低い方を選択します。これにより、チャンネルがすべて使用されます。*backupSpec* 句で指定または暗黙指定したファイル数が、**filesperset** より大きい場合（たとえば、**filesperset**=4 のときに、8 つのファイルすべてがバックアップを必要としている場合）、RMAN は複数のバックアップ・セットを作成し、ファイルとバックアップ・セットの正当な比率を維持します。

filesperset パラメータを指定しない場合、RMAN は算出された値（ファイル数 / 割り当てられたチャンネル数）を、デフォルト値である 64 と比較し、いずれか低い方を選択します。したがって、この場合もチャンネルがすべて使用されます。ほとんどのアプリケーションにとって、この 64 というデフォルト値は高い値です。これよりも低い値を指定するか、または **setsize** パラメータを使用して、バックアップ・セットのサイズを制限します。

RMAN は、割り当てられたチャンネルがすべて作業状態になるようバックアップ・セットを常に作成しようとします。例外は、バックアップするファイルよりもチャンネル数が多い場合です。たとえば、RMAN が 1 つのデータ・ファイルをバックアップするときに、**filesperset**=1 で、3 つのチャンネルが割り当てられている場合は、必然的に 2 つのチャンネルがアイドル状態となります。

次の例では、バックアップをパラレル化し、1 つのバックアップ・セットに 4 つ以上のデータ・ファイルまたは 2 つ以上のアーカイブ REDO ログが含まれないように指定しています。

```

run {
    allocate channel ch1 type disk;
    allocate channel ch2 type disk;
    allocate channel ch3 type disk;
    allocate channel ch4 type disk;
    backup
        datafile 1,2,3,4,5,9,10,11,12,15
            filesperset = 3
        archivelog all
            filesperset = 1;
}

```

setsize の使用方法

setsize パラメータは、バックアップ・セットの最大サイズを、1K (1024 バイト) 単位で指定します。したがって、バックアップ・セットを 2MB に制限するには、**setsize** を 2000 に指定します。RMAN 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。この機能は、メディア・マネージャ構成で、1 つのバックアップ・セットを 1 つのテープよりも大きくしないようにする場合に便利です。

バックアップ・セットは、複数のテープ・ボリュームに分散せず、1 つのテープ・ボリュームに収まるよう構成してください。そのようにしないと、複数のボリュームのバックアップ・セットで構成されるテープに障害が発生した場合、バックアップ・セット全体を失うこととなります。

アーカイブ REDO ログのバックアップの作成は、**filesperset** パラメータを使用するよりも **setsize** パラメータを使用した方が簡単です。この例では、アーカイブ REDO ログをテープにバックアップしますが、サイズは 2MB に設定して、各バックアップ・セットが 1 つのテープ・ボリュームに収まるようにします。

```
run {
  allocate channel ch1 type 'sbt_tape';
  allocate channel ch2 type 'sbt_tape';
  backup setsize = 2000
    archivelog all;
}
```

データ・ファイルがストライプされている場合、または別々のディスクに分散している場合で、かつ次のような処理を行うときも、**setsize** パラメータを使用する方が、データ・ファイルのバックアップが簡単になります。

- 複数のデータ・ファイルの多重化が必要になる高帯域幅のテープ・ドライブを使用して、テープ・ドライブへの書込みを行う。
- データベースをオープンしてバックアップを作成し、複数のディスク・スピンドルに I/O の負荷を分散して、オンライン処理のパフォーマンスを確保する。

たとえば、データ・ファイルが 4 つに分割されている場合は、**filesperset** の設定を 4 にすると、各バックアップ・セットは 4 つのディスク装置からデータの読取りを行うようになり、結果として I/O の負荷が分散されます。

バックアップにおけるデータ・ファイルの多重化

FOO という名称のデータベースをバックアップすると仮定します。条件は次のとおりです。

- バックアップする際、テープ・ドライブを 3 台使用できる。
- 26 のデータ・ファイルがこのデータベースに含まれている。
- データ・ファイルは多重化してバックアップするが、その際、バックアップ・セット 1 つにつき、4 つのデータ・ファイルが含まれるようにする。

filesperset は 4 に設定します。テープ・ドライブへの書き込み速度を一定に維持するにはこれで十分であると仮定しています。この例では、バックアップ・セットにデータ・ファイルをどのようにグループ化するかは、考慮しないものとします。

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 filesperset 4
    database format 'FOO.FULL.%n.%s.%p';
}
```

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、Oracle は 7 つのバックアップ・セットを作成します。バックアップ・ピースのファイル名は、次のような形式になります。*db_name* はデータベースの名称、*set_num* はバックアップ・セットの番号、*piece_num* はバックアップ・ピースの番号をそれぞれ示します。

```
FOO.FULL.db_name.set_num.piece_num
# for example, a file may have the following name:
FOO.FULL.prod1.3.1
```

このジョブより前にリカバリ・カタログに記録されたバックアップ・セットはないものと仮定すると、*set_num* の範囲は 1 ~ 7、*piece_num* は 1 以上となります。

SBT API バージョン 2.0 では、メディア・ベンダーによるバックアップ・ピースの最大サイズ指定が可能です。これを指定すると、RMAN が自動的にその制限に従うことに注意してください。

アーカイブ REDO ログのバックアップ

アーカイブ REDO 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。時間、SCN またはログ順序番号で、アーカイブ REDO ログの範囲を指定することが可能です。

注意： 時間で範囲を指定する場合は、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。5-2 ページの「[NLS 環境変数の設定](#)」の項を参照してください。

アーカイブ・ログの範囲を指定しても、RMAN はその範囲内のすべての REDO のバックアップをとる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、範囲内のログの 1 つが欠けていることもあります。RMAN

は、検出したログをバックアップする際、警告を発行しません。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、まず最初にアーカイブする必要があります。

次の例では、1998 年 11 月 18 日の午後 8:57 から午後 9:06 までの間にアーカイブされたログをバックアップします。

```
NLS_LANG=american
NLS_DATE_FORMAT='Mon DD YYYY HH24:MI:SS'
run {
    allocate channel dev1 type 'sbt_tape';
    backup
    archivelog all
        from time 'Nov 13 1998 20:57:13'
        until time 'Nov 13 1998 21:06:05';
}
```

次の例では、順序番号 288 から 301 までのアーカイブ・ログをすべてバックアップし、バックアップの完了後、これらのアーカイブ REDO 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 from logseq 288 until logseq 301 thread 1
        delete input;
}
```

ここ 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。この例では、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 REDO ログをアーカイブしています。

```
run {
    allocate channel dev1 type 'sbt_tape';
    sql "ALTER SYSTEM ARCHIVE LOG CURRENT";
    backup archivelog from time 'SYSDATE-1';
}
```

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・システム固有のマニュアルを参照してください。

アーカイブ REDO ログの複数のコピーのバックアップおよび削除

この例では、`/oracle/arch/dest_1/*` および `/oracle/arch/dest_2/*` というディレクトリに、REDO ログが自動的にアーカイブされるよう、初期化パラメータを設定します。したがって、ログ順序番号 1 つあたり、アーカイブ REDO ログのコピーが同じ内容で 2 つ存在することになります。アーカイブ REDO ログの各コピーをバックアップし、その後でオリジナルを削除することにします。

backup archivelog all コマンドには、" ログ順序番号 1 つにつき、1 つのコピーをバックアップする " という意味があるため、RMAN が、同じログ順序番号を有する 2 つのコピーを同一のバックアップ・セットに組み込むことはありません。ここでさらに、**delete input** オプションを指定すると、RMAN はバックアップされたアーカイブ REDO ログのうち、指定されたコピーのみ削除します。

この場合の最も簡単な解決策は、各アーカイブ REDO ログのコピーを両方ともバックアップし、その後でこれらを削除することです。*archivelogRecordSpecifier* の **like** パラメータを使用して、使用する宛先を明示します。**like** パラメータを使用することにより、両方のアーカイブの宛先におけるファイル名を合致させることができます。ここでは、次のとおり実行します。

```
run {
  allocate channel t1 type 'sbt_tape';
  allocate channel t2 type 'sbt_tape';
  backup
    filesperset=20
    format='al_%d/%t/%s/%p'
    (archivelog all like '/oracle/arch/dest1/%' channel t1 delete input)
    (archivelog all like '/oracle/arch/dest2/%' channel t2 delete input);
}
```

この例では、第 1 の宛先のアーカイブ REDO ログをある 1 つのテープ・セットに、第 2 の宛先のログをこれとは別のテープ・セットにバックアップします。また、この例では、テープ・ドライブを 2 台利用できるものと仮定しています。テープ・サブシステムによっては、この 2 つの RMAN チャンネルを 1 つのデータ・ストリームにまとめた上で、1 台のテープ・ドライブに書き込みを行うものもあります。このようにならないようメディア管理ベンダーを構成する必要があります。

差分増分バックアップの実行

差分増分バックアップには、同レベル以下の最新のバックアップ以降に変更されたブロックのみが組み込まれます。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 のバックアップにしてください。

```
run {
  allocate channel dev1 type 'sbt_tape';
  backup incremental level 0
    database;
}
```

次の増分バックアップ（レベル 1 以上）には、最新のレベル 1 のバックアップ以降に変更されたブロックがすべて組み込まれます。前回のレベル 1 のバックアップが使用できない場合、RMAN は基礎となるレベル 0 のバックアップ以後に変更されたブロックをすべてコピーします。

```
run {
  allocate channel dev1 type 'sbt_tape';
```

```
        backup incremental level 1
        database;
    }
```

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、レベル 0 のバックアップを作成した後で、新たに増分バックアップを作成する必要があります。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;
}
```

増分バックアップは NOARCHIVELOG モードで実行できます。

累積増分バックアップの実行

レベル n の累積増分バックアップには、レベル $n-1$ 以下の最後のバックアップ以降に変更されたブロックのみが含まれます。累積バックアップは、差分バックアップよりも大きい格納領域を必要とします。ただし、必要なバックアップの数はどのレベルの場合でも 1 つのみとなるため、復元操作を重要視する場合には累積バックアップが望ましい選択です。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 のバックアップである必要があることに注意してください。

レベル 2 の累積バックアップには、レベル 1 の最後のバックアップ以後に変更されたブロックがすべて含まれます。前回のレベル 1 が使用不能であった場合に限り、レベル 0 の基礎バックアップ以後に変更されたブロックがすべてコピーされます。差分レベル 2 のバックアップは、累積バックアップとは異なり、レベル 1 とレベル 2 のバックアップのどちらが新しいかを判断し、新しい方のバックアップ以降に変更されたブロックをすべてコピーします。

```
run {
    allocate channel dev1 type 'sbt_tape';
    backup incremental level 2 cumulative # blocks changed since level 0 or level 1
        database;
}
```

バックアップ・セットの二重化

災害、メディア障害、人為ミスに備え、バックアップ・セットのコピーは複数作成するようにしてください。Oracle では、最高 4 つの同一内容のバックアップ・セットを同時に作成することができます。

set duplex コマンドは、**backup** コマンドにのみ適用するコマンドです。このコマンドは、RMAN が作成する各バックアップ・ピースのコピー数を指定します。**set duplex** コマンド

は、このコマンドの発行後に割り当てられたチャンネルすべてに適用されます。セッション時、明示的に使用禁止（OFF）に設定するか、または変更するまで有効です。

注意： 前回割り当てられたチャンネルがある場合、**set duplex** コマンドはエラーを生成します。

たとえば、次のように入力できます。

```
run {
    set duplex=3;
    allocate channel chl type 'sbt_tape';
    backup datafile 1;
}
```

このコマンドにより、datafile 1 と同じ内容のバックアップが 3 つ作成されます。各バックアップ・ピースには、%U デフォルト形式により、一意の名称が付与されます。

二重化バックアップを実行するためには、BACKUP_TAPE_IO_SLAVES 初期化パラメータを TRUE に設定する必要があることに注意してください。そのようにしない場合はエラーが表示されます。RMAN は、要求されたバックアップ・コピーの数に応じて、スレーブを構成します。

関連項目： BACKUP_TAPE_IO_SLAVES 初期化パラメータの詳細は、『Oracle8i 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・セットの平行化

複数のバックアップ・セットを作成し、複数のチャンネルを割り当てる処理は、RMAN が自動的に平行化します。複数のバックアップ・セットの書き込みも平行に行われます。割当てが行われたサーバー・セッションは、指定されたデータ・ファイル、制御ファイルおよびアーカイブ REDO ログのバックアップ処理を分割します。1 つのバックアップ・セットを複数のチャンネルに分散することはできません。

RMAN は、1 つのデバイスに 1 つのバックアップ・セットを自動的に割り当てます。**channel** パラメータを使用すれば、backupSpec の全バックアップ・セットを特定のチャンネルに書き込むよう Oracle を設定することができます。

たとえば、この RMAN では、どのチャンネルをどの処理に割り当てるかを指定して、バックアップ処理を平行化しています。

```
run {
    allocate channel chl type 'SBT_TAPE';
    allocate channel ch2 type disk;
    allocate channel ch3 type disk;
    backup
        # channel chl backs up datafiles to tape
        (datafile 1,2,3,4
```

```
channel ch1)
# channel ch2 backs up control file copy to disk
(controlfilecopy '/oracle/copy/cf.f'
channel ch2)
# channel ch3 backs up archived redo logs to disk
(archiveolog from time 'SYSDATE-14'
channel ch3);
}
```

NOARCHIVELOG モードでのバックアップ

次のスクリプトは、一貫性全体データベース・バックアップに適したモードに設定した上で、データベースをバックアップするものです。このスクリプトは、停止、起動、停止、起動というように処理を行った後、二重化バックアップを実行することに注意してください。

```
# Shut down the database cleanly using immediate priority. This type of shutdown lets
# current calls to the database complete, but prevents further logons or calls.
# If the database is not up now, you will get a message saying so but RMAN will not
# treat this situation as an error.
```

```
shutdown immediate;
```

```
# Start up the database in case it crashed or was not shutdown cleanly prior to
# starting this script. This will perform a crash recovery if it is needed. Oracle
# uses the default INIT.ORA file. Alternatively, use this form: startup force dba
# pfile=<filename>. Use the DBA option because you are going to shut down again right
# away and do not want to let users in during the short interval. Use the FORCE
# option because it cannot hurt and might help in certain situations.
```

```
startup force dba;
shutdown immediate;
```

```
# Here, we know that the database is cleanly closed and is now ready for a cold
# backup. RMAN requires that the database be started and mounted to perform a backup,
# so do that now.
```

```
startup mount;
run {
    # duplex the backup
    set duplex = 2;

    # allocate channel t1 type 'SBT_TAPE';
    # allocate channel t2 type 'SBT_TAPE';
    allocate channel t1 type disk;
    allocate channel t2 type disk;

    set limit channel t1 kbytes 2097150;
    set limit channel t2 kbytes 2097150;
```

```

        backup
        incremental level 0
        filesperset 5
        database;
    }

# now that the backup is complete, open the database.
alter database open;

```

特定の表領域をスキップすることも可能です。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も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。またディスクにバックアップする前に、宛先（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。

パラレル・サーバー環境でのバックアップ

次のスクリプトは、データ・ファイルのバックアップ作業とアーカイブ REDO ログのバックアップ作業を、パラレル・サーバー環境の2つのノードに分散するものです。

```

run {
    allocate channel node_1 type disk connect 'sys/sys_pwd@node_1';
    allocate channel node_2 type disk connect 'sys/sys_pwd@node_2';

    backup filesperset 1
        (tablespace system, rbs, data1, data2
         channel node_1)
        (tablespace temp, reccat, data3, data4
         channel node_2);
    backup filesperset 20
        (archivelog
         until time 'SYSDATE'
         thread 1
         delete input
         channel node_1);
        (archivelog
         until time 'SYSDATE'
         thread 2
         delete input
         channel node_2);
}

```

関連項目：OPS バックアップの詳細は、『Oracle8i Parallel Server 概要および管理』を参照してください。

O/S コピーのカatalog化

O/S ユーティリティを使用すると、データ・ファイルのコピーを作成した後、リカバリ・カタログでカatalog化することができます。カatalog化できるのは、ディスクへのコピーのみ

であることに注意してください。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、O/S コマンドを使用して作成する必要があります。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、まず ALTER TABLESPACE BEGIN BACKUP を発行します。その結果生成されるイメージ・コピーは、次の方法でカタログ化できます。

```
catalog datafilecopy '?/dbs/tbs_33.f';
```

バックアップおよびコピーのメンテナンス

バックアップおよびコピーを保存する必要がある期間は、次の要因によって異なります。

- バックアップを行う頻度
- 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。

```
# delete a specific datafile copy
change datafilecopy '?/dbs/tbs_35.f' delete;

# delete archived redo logs older than 31 days
change archivelog until time 'SYSDATE-31' delete;
```

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にしてください。

allocate channel for maintenance 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 maintenance type 'sbt_tape';
change backuppiece 'testdb_87fa39e0' delete;
release channel;
```

バックアップおよびコピー時のエラー処理

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、バックアップまたはイメージ・コピーに格納されます。**nochecksum** オプションを使用すると、チェックサムが算出されなくなります。ただし、ブロックにチェックサムがすでに含まれていた場合、そのチェックサムは妥当性チェックが行われた後、バックアップに格納されます。妥当性チェックに失敗した場合、そのブロックにはバックアップ内で破損のマークが付されます。

set maxcorrupt for datafile コマンドは、**backup** または **copy** で許可される、データ・ファイル内の破損ブロック数を指定するものです。**maxcorrupt** パラメータに指定した値より多い破損ブロックがあるデータ・ファイルの場合、このコマンドは正常に終了しません。**check logical** オプションを指定すると、RMAN は論理的な破損と物理的な破損を検出します。

backup コマンドは、データ・ファイルにアクセスできない場合は終了するようデフォルト設定されています。終了を防ぐさまざまなパラメータを指定できます。

指定するオプション	RMAN がスキップするファイル
skip inaccessible オプション	アクセスできないデータ・ファイル。データ・ファイルは、読取りが不可能な場合のみアクセス不能と判断されることに注意してください。オフラインのデータ・ファイルでも、ディスク上に残っているために読取りが可能になっている場合があります。また、削除されるか移動されたために読取りが不可能になり、アクセス不能とされるものもあります。
skip offline オプション	オフラインのデータ・ファイル
skip readonly オプション	読取り専用状態のデータ・ファイル

```
run {
  allocate channel dev1 type 'sbt_tape';
  set maxcorrupt for datafile 1,2,3 to 5;
  backup database
    skip inaccessible
    skip readonly
    skip offline;
}
```

Recovery Manager による復元と回復

この章では、Recovery Manager を使用して、復元操作および回復操作を実行する方法について説明します。項目は次のとおりです。

- データ・ファイル、制御ファイルおよびアーカイブ REDO ログの復元
- データ・ファイルの回復
- 復元および回復の使用例

データ・ファイル、制御ファイルおよびアーカイブ REDO ログの復元

データ・ファイル、制御ファイルまたはアーカイブ REDO ログをバックアップ・セットまたはイメージ・コピーから復元するには、RMAN の **restore** コマンドを使用します。RMAN は、バックアップをディスクまたはテープから復元します。ただし、イメージ・コピーはディスクのみから復元されます。

ファイルを復元するときは、次の操作が必要です。

- バックアップまたはコピーの復元の前に、少なくとも 1 つのチャンネルを割り当てること。復元操作をパラレル化するには、複数のチャンネルを指定します。
- **from copy** オプションを使用する場合は、**type disk** のチャンネルを 1 つ割り当てること。
- ファイルを復元するときには、適切な **disk** または '**sbt_tape**' チャンネルを割り当てること。適切なタイプのデバイスを割り当てないと、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。

ファイルの復元先は次のどちらかになります。

- デフォルトの場所。同じ名前のファイルを上書きします。
- **set newname** コマンドで指定した新規の場所。データ・ファイルを新規の場所に復元する場合、Oracle はそのデータ・ファイルをデータ・ファイル・コピーとみなし、制御ファイルおよびリカバリ・カタログにそのように記録します。

この項では、次の項目について説明します。

- [データベースの復元](#)
- [表領域およびデータ・ファイルの復元](#)
- [制御ファイルの復元](#)
- [アーカイブ REDO ログの復元](#)
- [不完全回復準備のための復元](#)

関連項目 : **restore** コマンドの構文は、11-113 ページの「[restore](#)」を参照してください。**set newname** コマンド構文は、11-134 ページの「[set_run_option](#)」を参照してください。

データベースの復元

ターゲット・データベースの復元では、次のことができます。

- メディア障害の発生時にデータベースをデフォルトの場所に復元する。
- メディア障害の発生時にデータベースを新しいホストに移動する。
- ターゲット・データベースのバックアップを使用して、テスト・データベースを作成する。

データベースをデフォルトの場所に復元するには、**restore database** コマンドを発行します。ターゲット・データベースを新しいホストに移動するには、**set newname** を使用して必要に応じてデータ・ファイルを改名します。ターゲット・データベースのバックアップを使用してテスト・データベースを作成するには、**duplicate** コマンドを使用します（完全なインスタクションについては、第 10 章の「[Recovery Manager による複製データベースの作成](#)」を参照）。

この章では、次の項目について説明します。

- デフォルトの場所へのデータベースの復元
- 同じファイル・システムの新しいホストへのターゲット・データベースの移動
- 異なるファイル・システムの新しいホストへのターゲット・データベースの移動

デフォルトの場所へのデータベースの復元

復元ジョブで **set newname** コマンドを指定しない場合、データベースをクローズ状態またはオフライン状態にしておく必要があります。このようにしないと、ファイルをオンラインにして、datafile 3 を復元したときの結果（次に示します）と同じ出力が表示されます。

```

RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure during compilation of command
RMAN-03013: command type: restore
RMAN-03006: non-retryable error occurred during execution of command: IRESTORE
RMAN-07004: unhandled exception during command execution on channel chl
RMAN-10035: exception raised in RPC: ORA-19573: cannot obtain exclusive enqueue
for datafile 3
ORA-19600: input file is datafile-copy 102 (/vobs/oracle/dbs/df.3)
ORA-19601: output file is datafile 3 (/vobs/oracle/dbs/tbs_11.f)
RMAN-10031: ORA-19573 occurred during call to DBMS_BACKUP_RESTORE.COPYDATAFILECOPY

```

データベース全体を復元するときには、データベースはクローズ状態であることが必要です。ターゲット・データベースがマウントされている場合は、該当するデータ・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。

データベースをデフォルトの場所に復元する

1. RMAN を起動し、ターゲット・データベース、およびオプションでリカバリ・カタログに接続する。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

2. データベースがオープンになっている場合は、停止した後にマウントする。

```
shutdown immediate;
startup mount;
```

3. チャンネルを割り当てた後に、データベースを復元する。

```
run {  
    allocate channel ch1 type disk;  
    allocate channel ch2 type disk;  
    allocate channel ch3 type disk;  
    restore database;  
}
```

同じファイル・システムの新しいホストへのターゲット・データベースの移動

メディア障害の結果、あるホストから別のホストにバックアップを復元することにより、データベースの移動が必要な場合があります。テスト用に複製データベースを作成する一方、元のデータベースも維持する必要がある場合は、この手順は使用しないで **duplicate** コマンドを発行する必要がありますので、注意してください。(第10章の「[Recovery Manager による複製データベースの作成](#)」を参照)

復元されたデータベースには本番データベースのオンライン REDO ログが含まれていないので、各スレッドの最新のアーカイブ REDO ログの最小の SCN に到達するまで不完全回復を実行し、次に RESETLOGS オプションでデータベースをオープンすることが必要です。

この使用例は、次のことを前提とします。

- HOST_A から HOST_B にバックアップを復元する。
- テープ・バックアップから復元する。
- HOST_A のデータベースは、HOST_B と同じ DB_NAME ではない。
- HOST_B のファイル名とディレクトリ・パスは、HOST_A と同じである。

ターゲット・データベースがリカバリ・カタログを使用するかどうかで、復元の手順は異なります。

注意： あるホストで作成したイメージ・コピーを異なるホストに復元するには、RMAN は使用できません。O/S ユーティリティを使用してファイルを転送し、**catalog** コマンドを使用して RMAN メタデータを更新します。

リカバリ・カタログを使用して HOST_A から HOST_B にデータベースを復元する

1. HOST_A の init.ora ファイルを、O/S ユーティリティを使用し、HOST_B にコピーする。
2. HOST_B のターゲット・インスタンスおよび HOST_A のリカバリ・カタログに接続する。たとえば、次のように入力します。

```
% rman target sys/change_on_install@host_b rman/rman@rcat
```

3. インスタンスをマウントせずに起動する。

```
startup nomount;
```

4. 制御ファイルを復元し、マウントする。次のサブコマンドを持つ **run** コマンドを実行します。

- a. 少なくとも 1 つのチャンネルを割り当てる。
- b. 制御ファイルを復元する。
- c. 制御ファイルをマウントする。

```
run {
    allocate channel chl type disk;
    restore controlfile;
    alter database mount;
}
```

5. 複数の REDO スレッドが存在する可能性があるので、change-based の回復を使用する。回復終了の SCN を取得するには、各スレッドの最新アーカイブ REDO ログのなかで最小の SCN を検索します。

必要な SCN を決めるには、SQL*Plus を起動し、次の問合せをします。

```
SELECT min(scn)
FROM (SELECT max(next_change#) scn
      FROM v$sarchived_log
      GROUP BY thread#);
```

6. 次のサブコマンドを使用した **run** コマンドを実行する。
 - a. 前のステップで取得した値を使用して、リカバリ終了の SCN を設定する。
 - b. 少なくとも 1 つのチャンネルを割り当てる。
 - c. データベースを復元する。
 - d. データベースを回復する。
 - e. RESETLOGS オプションを使用してデータベースをオープンする。

```
run {
    set until scn = 500; # use appropriate SCN for incomplete recovery
    allocate channel chl type 'sbt_tape';
    restore database;
    recover database;
    sql "ALTER DATABASE OPEN RESETLOGS";
}
```

リカバリ・カタログなしで HOST_A から HOST_B にデータベースを復元する

1. HOST_A の init.ora ファイルを、O/S ユーティリティを使用して、HOST_B にコピーする。
2. O/S ユーティリティを使用して HOST_A の制御ファイルのイメージ・コピーを作成し、それを O/S ユーティリティを使用して HOST_B に転送する。
3. **nocatalog** で HOST_B のターゲット・インスタンスに接続する。たとえば、次のように入力します。

```
% rman target sys/change_on_install@host_b nocatalog
```

4. データベースをマウントする。

```
startup mount;
```

5. 複数の REDO スレッドが存在する可能性があるので、change-based の回復を使用する。回復終了の SCN を取得するには、各スレッドの最新アーカイブ REDO ログのなかで最小の SCN を検索します。

必要な SCN を決めるには、SQL*Plus を起動し、次の問合せをします。

```
SELECT min(scn)
FROM (SELECT max(next_change#) scn
      FROM v$sarchived_log
      GROUP BY thread#);
```

6. 次のサブコマンドを使用した **run** コマンドを実行する。
 - a. 前のステップで取得した値を使用して、リカバリ終了の SCN を設定する。
 - b. 少なくとも 1 つのチャンネルを割り当てる。
 - c. データベースを復元する。
 - d. データベースを回復する。
 - e. RESETLOGS オプションを使用してデータベースをオープンする。

```
run {
    set until scn 500; # use appropriate SCN for incomplete recovery
    allocate channel chl type 'sbt_tape';
    alter database mount;
```



```

restore database;
recover database;
sql "ALTER DATABASE OPEN RESETLOGS";
}

```

異なるファイル・システムの新しいホストへのターゲット・データベースの移動

ファイル・システムが異なるマシンにデータベースを移動する手順は、基本的には 9-33 ページの「[オープン状態のデータベースにおけるアクセス不可能なデータ・ファイルの回復](#)」で説明する手順と同じです。違いは、各データ・ファイルを **set newname** を使用して改名する必要がある点です。

たとえば、次のように仮定します。

- HOST_A のデータベースには 10 個のデータ・ファイルが存在する。
- テープ・バックアップから復元する。
- いくつかのデータ・ファイルは /disk1 に復元し、他のものは HOST_B の /disk_2 に復元する。

リカバリ・カタログを使用して HOST_B にデータベースを復元する

1. 9-4 ページの「[同じファイル・システムの新しいホストへのターゲット・データベースの移動](#)」の手順（リカバリ・カタログを使用する場合）を実行するが、**run** コマンドを実行する前に停止する。パス名を指定する `init.ora` ファイルの `*_DEST` および `*_PATH` パラメータはすべて確実にリセットしてください。
2. **run** コマンドを、そのかわりとして実行する。
 - a. SQL*Plus 問合せで取得した終了 SCN を設定する。
 - b. 少なくとも 1 つのチャンネルを割り当てる。
 - c. 各データ・ファイルに新しいファイル名を指定する。
 - d. データベースをマウントする。
 - e. データベースを復元する。
 - f. データ・ファイルを切り替える。
 - g. データベースを回復する。
 - h. RESETLOGS オプションを使用してデータベースをオープンする。

```

run {
  set until scn 500; # use appropriate SCN for incomplete recovery
  allocate channel ch1 type disk;
  set newname for datafile 1 to '/disk1/%U'; # rename each datafile manually
  set newname for datafile 2 to '/disk1/%U';
}

```

```
set newname for datafile 3 to '/disk1/%U';
set newname for datafile 4 to '/disk1/%U';
set newname for datafile 5 to '/disk1/%U';
set newname for datafile 6 to '/disk2/%U';
set newname for datafile 7 to '/disk2/%U';
set newname for datafile 8 to '/disk2/%U';
set newname for datafile 9 to '/disk2/%U';
set newname for datafile 10 to '/disk2/%U';
alter database mount;
restore database;
switch datafile all; # points the control file to the renamed datafiles
recover database;
sql "ALTER DATABASE OPEN RESETLOGS";
}
```

リカバリ・カタログなしで HOST_B にデータベースを復元する

- 9-4 ページの「[同じファイル・システムの新しいホストへのターゲット・データベースの移動](#)」の手順（リカバリ・カタログを使用しない場合）を実行するが、ステップ 6 の **run** コマンドを実行する前に停止する。パス名を指定する `init.ora` ファイルの `*_DEST` および `*_PATH` パラメータはすべて確実にリセットします。
- 次のサブコマンドを使用した **run** コマンドを実行する。
 - SQL*Plus 問合せで取得した終了 SCN を設定する。
 - 少なくとも 1 つのチャネルを割り当てる。
 - 各データ・ファイルに新しいファイル名を指定する。
 - データベースを復元する。
 - データ・ファイルを切り替える。
 - データベースを回復する。
 - RESETLOGS オプションを使用してデータベースをオープンする。

```
run {
  set until scn 500; # use appropriate SCN for incomplete recovery
  allocate channel chl type disk;
  set newname for datafile 1 to '/disk1/%U'; # rename each datafile manually
  set newname for datafile 2 to '/disk1/%U';
  set newname for datafile 3 to '/disk1/%U';
  set newname for datafile 4 to '/disk1/%U';
  set newname for datafile 5 to '/disk1/%U';
  set newname for datafile 6 to '/disk2/%U';
  set newname for datafile 7 to '/disk2/%U';
  set newname for datafile 8 to '/disk2/%U';
  set newname for datafile 9 to '/disk2/%U';
  set newname for datafile 10 to '/disk2/%U';
```

```

restore database;
switch datafile all; # point control file to renamed datafiles
recover database;
sql "ALTER DATABASE OPEN RESETLOGS";
}

```

表領域およびデータ・ファイルの復元

データ・ファイルが消失または破損したがアクセス可能である場合、データ・ファイルを前の場所に復元することができます。この場合表領域をオフラインにして表領域を復元するコマンドを発行します。元の位置がアクセス不可能の場合は、表領域をオフラインにし、関連するデータ・ファイルを新しい位置に復元します。

データ・ファイルをデフォルトの場所に復元できない場合は、復元の前に **set newname** コマンドを使用します。この場合、Oracle では復元するデータ・ファイルをデータ・ファイル・コピーとみなし、それを現行データ・ファイルとするために **switch** コマンドが実行されます。そしてファイル名のエントリを作成するか、すでにある場合はそれを上書きします。

RMAN の **switch** コマンドは、ALTER DATABASE RENAME DATAFILE 文と等価です。switch コマンドを使用すると、現行のデータ・ファイルの位置を変更できることにご注意ください。また、switch コマンドの実行によって、コピーが消費されること、つまりリカバリ・カタログおよび制御ファイルにある対応するレコードが削除されることにも注意してください。

switch コマンドのターゲットを指定しないと、このファイル番号に対する以前の **set newname** で設定したファイル名が switch コマンドのターゲットとして使用されます。**switch datafile all** を指定した場合、このジョブで **set newname** を発行されたデータ・ファイルはすべて、それらの新しいファイル名に切り替えられます。

後で回復を行う目的で **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、復元後で、かつ回復前に switch コマンドを実行して、復元したデータ・ファイルを現行のデータ・ファイルにしてください。

関連項目 : switch コマンドの構文は、11-145 ページの「[switch](#)」を参照してください。

表領域をデフォルトの場所に復元する

1. RMAN を起動し、ターゲット・データベースおよびオプションでリカバリ・カタログに接続する。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

2. チャネルの割当て後に、次の操作をする。
 - 回復が必要な表領域をオフラインにする。
 - 表領域を復元する。

たとえば、表領域 USER_DATA を復元するには、次のコマンドを発行します。

```
run {
    sql 'ALTER TABLESPACE user_data OFFLINE TEMPORARY';
    allocate channel chl type disk;
    restore tablespace user_data;
}
```

3. 復元した表領域にメディア回復を行う。必要な手順の詳細は、9-33 ページの「[オープン状態のデータベースにおけるアクセス不可能なデータ・ファイルの回復](#)」を参照してください。

表領域を新しい位置に復元する

1. RMAN を起動し、ターゲット・データベースおよびオプションでリカバリ・カタログに接続する。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

2. チャネルの割当て後に、次の操作をする。
 - 表領域をオフラインにする。
 - オフライン表領域の破損したデータ・ファイルの復元先として、アクセス可能な場所を指定する。
 - 新しい位置にデータ・ファイルを復元する。
 - 復元したデータ・ファイルに switch コマンドを実行し、制御ファイルに対して現行ファイルと設定する。

表領域 TBS_1 のデータ・ファイルをディスク上の新しい位置に復元するには、次のコマンドを入力します。

```
run {
    allocate channel chl type disk;
    sql 'ALTER TABLESPACE user_data OFFLINE TEMPORARY';
    # restore the datafile to a new location
    set newname for datafile '/disk1/oracle/tbs_1.f' to '/disk2/oracle/tbs_1.f';
    restore tablespace tbs_1;
    # make the control file recognize the restored file as current
    switch datafile all;
}
```

3. 復元表領域に対してメディア回復を行う。必要な手順の詳細は、9-33 ページの「[オープン状態のデータベースにおけるアクセス不可能なデータ・ファイルの回復](#)」を参照してください。

制御ファイルの復元

メディア障害によって制御ファイルが破損し、かつ多重化コピーがない場合は、バックアップの復元が必要になります。**restore controlfile** コマンドを発行して、制御ファイルをパラメータ・ファイルで指定した最初の CONTROL_FILES の場所に復元します。RMAN は、パラメータ・ファイルで指定したすべての CONTROL_FILES の場所に自動的に制御ファイルをコピーします。

デフォルト以外の位置に制御ファイルを復元するときは、**restore controlfile 'filename'** により宛先を指定します。ファイルがすでに存在する場合、ファイルに上書きします。制御ファイルを新しい位置に復元するときは、**replicate controlfile from 'filename'** を使用して CONTROL_FILES の宛先に制御ファイルをコピーします。RMAN は、制御ファイルを自動的にコピーしません。

replicate controlfile コマンドの使用方法は、**copy controlfile** コマンドの場合と同じです。ユーザーが制御ファイルを名前指定すると、RMAN はターゲット・データベースの CONTROL_FILES 初期化パラメータで指定した場所にファイルをコピーします。

リカバリ・カタログを使用して制御ファイルをデフォルトの位置に復元する

1. RMAN を起動し、ターゲット・データベースおよびリカバリ・カタログ・データベースに接続する。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

2. データベースをマウントせずにインスタンスを開始します。

```
startup nomount;
```

3. 次の操作を行います。

- a. なんらかの理由である日付より前に作成した制御ファイルを復元する必要がある場合は、その日付で **set until** コマンドを発行する。その他の場合は、次のステップに進みます。
- b. 1 つ以上のチャンネルを割り当てる。
- c. 制御ファイルを復元する。
- d. データベースをマウントする。

```
run {
    # To restore a control file created before a certain date, issue the
    following
    # set command using a valid date for 'date_string'. You can also specify an
    SCN
    # or log sequence number.
    # set until time = 'date_string';
    allocate channel chl type 'sbt_tape';
    restore controlfile;
```

```
        alter database mount;
    }
```

RMAN は、CONTROL_FILES 初期化パラメータで指定した場所に自動的に制御ファイルをコピーします。

4. 制御ファイルの復元後にデータ・ファイルのメディア回復を実行する必要がある場合は、9-17 ページの「[完全回復の実行](#)」または 9-23 ページの「[不完全回復の実行](#)」を参照してください。

リカバリ・カタログを使用せずに制御ファイルを新しい位置に復元する

特定のバックアップに関する情報を含む制御ファイルは、そのバックアップによってバックアップされた制御ファイルとは異なりますので注意してください。たとえば、**backup database** を発行した場合、この全体データベース・バックアップに含まれるバックアップ制御ファイルには、この全体データベース・バックアップの記録は入っていません。次の制御ファイルのバックアップに、この全体データベース・バックアップについての情報が含まれます。

1. RMAN を起動し、ターゲット・データベースに接続する。たとえば、次のように入力します。

```
% rman target / nocatalog
```

2. データベースをマウントします。

```
startup mount;
```

3. 次の操作を行います。

- a. ある日付より前に作成した制御ファイルを復元する必要がある場合は、その日付を指定して **set until** コマンドを発行する。その他の場合は、次のステップに進みます。
- b. 1 つ以上のチャンネルを割り当てる。
- c. 現行制御ファイルを誤って上書きすることを防ぐために、バックアップ制御ファイルを一時ファイルに復元する。
- d. データベースを停止する。
- e. 制御ファイルを復元した一時ファイルから、パラメータ・ファイルの CONTROL_FILES パラメータで指定したすべての場所にコピーする。
- f. データベースをマウントする。

```
run {
    # To restore a control file created before a certain date, issue the
    following
    # set command using a valid date for 'date_string'. You can also specify an
    SCN
```

```

# or log sequence number.
# set until time = 'date_string';
allocate channel chl type 'sbt_tape';
# restore control file to new location
restore controlfile to '/oracle/dbs/cfl.ctl';
shutdown immediate;
# replicate the control file manually to locations in parameter file
replicate controlfile from '/oracle/dbs/cfl.ctl';
startup mount;
}

```

4. 制御ファイルの復元後にデータ・ファイルのメディア回復を実行する必要がある場合は、9-17 ページの「[完全回復の実行](#)」または 9-23 ページの「[不完全回復の実行](#)」を参照してください。

関連項目 : `replicate controlfile` コマンドの構文は、11-101 ページの「[replicate](#)」を参照してください。

アーカイブ REDO ログの復元

RMAN は、ターゲット・データベースの `LOG_ARCHIVE_FORMAT` パラメータと、`LOG_ARCHIVE_DEST` または `LOG_ARCHIVE_DEST_1` のどちらかのパラメータを組み合わせた名前で、アーカイブ REDO ログを復元します。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。

デフォルト以外の宛先を指定するには **set archivelog destination** コマンドで指定します。データベースの復元を実行中に、このコマンドを使用して、さまざまな場所へアーカイブ・ログを準備できます。回復中、RMAN は新たに復元されたアーカイブ・ログの所在を認識しています。このためこれらのログがパラメータ・ファイルで指定した場所になくてもかまいません。

たとえば、`init.ora` ファイルに指定された場所とは異なる宛先を指定して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を検出するので、`init.ora` パラメータ指定された場所のファイルは検索されません。

必要な場合は、アーカイブ REDO ログに複数の復元宛先を指定できます。ただし、それらの宛先を同時に指定することはできません。たとえば、次のように発行できます。

```

run {
  allocate channel chl type disk;
  # Set a new location for logs 1 through 10.
  set archivelog destination to '/disk1/oracle/temp_restore';
  restore archivelog from logseq 1 until logseq 10;
  # Set a new location for logs 11 through 20.
  set archivelog destination to '/disk1/oracle/arch';
  restore archivelog from logseq 11 until logseq 20;
  # Set a new location for logs 21 through 30.
  set archivelog destination to '/disk2/oracle/temp_restore';
}

```

```
restore archivelog from logseq 21 until logseq 30;
...
recover database;
}
```

アーカイブ REDO ログを複数の場所に復元する場合、1 つの **recover** コマンドを発行するのみで済みます。RMAN は、復元されたアーカイブ REDO ログを自動的に検索し、データ・ファイルにそれらを適用します。

必要なアーカイブ REDO ログの復元

1. RMAN を起動し、ターゲット・データベースおよびオプションでリカバリ・カタログに接続する。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

オプションで、接続時にメッセージ・ログ・ファイルを指定します。

```
% rman target / catalog rman/rman@rcat log = rman_log
```

2. データベースがオープンになっている場合は、停止してからマウントする。

```
shutdown immediate; startup mount
```

3. **run** コマンドのなかで、次の操作を実行する。

- a. 必要な場合、**set archivelog destination** コマンドを使用して、復元アーカイブ REDO ログに新しい位置を指定する。その他の場合は、次のステップに進みます。

- b. チャネルを割り当てる。

- c. アーカイブ REDO ログを復元する。

たとえば、このジョブではすべてのバックアップ・アーカイブ REDO ログを復元します。

```
run {
  # Optionally, set a new location for the restored logs.
  set archivelog destination to '/oracle/temp_restore';
  allocate channel chl type disk;
  restore archivelog all;
}
```

関連項目 : **set archivelog destination** コマンドの構文は、11-134 ページの「[set_run_option](#)」を参照してください。

不完全回復準備のための復元

set until コマンドを使用して、回復の終了点を指定します。このコマンドは、同じ **run** コマンド内の後続の **restore**、**switch** および **recover** の各コマンドのどれにも影響します。

不完全回復準備のためにデータベースを復元する方法

1. RMAN を起動し、ターゲット・データベースおよびオプションでリカバリ・カタログに接続する。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

オプションで、接続時にメッセージ・ログ・ファイルを指定します。

```
% rman target / catalog rman/rman@rcat log = rman_log
```

2. データベースがオープンになっている場合は、停止してからマウントする。

```
shutdown immediate;
startup mount;
```

3. **run** コマンドのなかで、次の操作を実行する。

- a. 特定の時刻、SCN またはログ順序番号に到達するまで回復する必要があるかどうかを判断し、適切な **set until** コマンドを発行する。
- b. チャンネルを割り当てる。
- c. データベースを復元する。

たとえば、このジョブでは不完全回復に備えてデータベースを 1998 年 12 月 15 日午前 9 時まで復元します。

```
run {
    set until time 'Dec 15 1998 09:00:00';
    allocate channel chl type 'sbt_tape';
    restore database;
}
```

データ・ファイルの回復

メディア回復とは、現行時刻または他のなんらかの指定時刻までファイルを更新するために、復元ファイルに REDO ログまたは増分バックアップを適用することです。回復または増分バックアップを適用できるのは、データ・ファイルのみです。データ・ファイル・コピーには適用できません。

次の場合に、メディア回復を実行します。

- メディア障害によってデータ・ファイルが破損し、現行時刻までの回復が必要である。
- データベース全体を以前のある時刻まで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 コマンドを実行した。

RMAN は、メディア回復を実行する上で、必要に応じてアーカイブ REDO ログのバックアップ・セットを復元します。デフォルトでは、RMAN は `init.ora` ファイルに指定したログのアーカイブ先にアーカイブ REDO ログを復元します。別の場所を指定するには、**set archivelog destination** コマンドを使用します。

RMAN が増分バックアップの適用か REDO の適用かを選択できる場合は、常に増分バックアップが選択されます。オーバーラップしたレベルの増分バックアップが使用できる場合は、RMAN が自動的に最長期間にわたる増分バックアップを選択します。

メディア回復を実行するには、できる限りリカバリ・カタログを使用可能にしてください。使用できない場合、RMAN は、ターゲット・データベース制御ファイルの情報をを使用して回復を実行します。

注意： 制御ファイルの回復が必要な場合は、リカバリ・カタログを使用可能にする必要があります。リカバリ・カタログもターゲット・データベース制御ファイルも使用できない場合、RMAN は操作を行うことができません。

この項では、次の項目について説明します。

- [メディア回復の準備](#)
- [完全回復の実行](#)
- [不完全回復の実行](#)

関連項目：増分バックアップの概要は、4-38 ページの「[増分バックアップ](#)」を参照してください。

メディア回復の準備

いつ、どのような方法でメディア回復を行うかは、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。

メディア回復の必要性の判断

1. SQL*Plus を起動し、ターゲット・データベースに接続する。たとえば、次のコマンドを発行して PRO1 に接続します。

```
% sqlplus sys/change_on_install@prod1;
```

2. コマンド行で次の SQL 問合せを実行して、データベースの状態を判断する。

```
SELECT parallel, status FROM v$instance;
```

```
PAR STATUS
-----
NO OPEN
```

STATUS 列が OPEN であれば、データベースはオープン状態です。しかし、それでもどの表領域およびどのデータ・ファイルを復元または回復する必要があるかはわかりません。

3. 次の問合せを実行して、データ・ファイルのヘッダーをチェックし、下の表に従って解決策を実行してください。

```
SELECT file#, status, error, recover, tablespace_name, name
FROM v$datafile_header
WHERE error IS NOT NULL
OR recover = 'YES';
```

エラー列	RECOVER 列	解決策
NULL	NO	エラーの原因がハードウェアまたは O/S の一時的な問題ではない場合は、データ・ファイルを復元するか、そのデータ・ファイルのコピーに切り替えてください。
NULL	YES	データ・ファイルを回復します。 recover コマンドは、まず適切な増分バックアップがあればそれを適用し、次に REDO ログを適用します。RMAN は、増分バックアップおよび REDO ログを必要だけ復元します。
NULL ではない		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。

注意： 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のみなので、データ・ファイルの復元が必要になる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、そのデータ・ファイルのヘッダー・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。

完全回復の実行

完全回復を実行するときは、データベース全体またはデータベースのサブセットを回復します。たとえば、表領域の大部分の完全回復を実行し、次に残りの表領域を回復することもできます。データ・ファイルが読み書き可能または NORMAL モードでオフラインにされた表領域であっても、違いはありません。

完全回復の使用方法は、データベースがオープン状態かクローズ状態で決まります。

データベースの状態	処理
クローズ状態	次のどちらかの操作をします。 <ul style="list-style-type: none">1つの操作でデータベース全体を回復する。別々の操作でそれぞれの表領域を回復する。
オープン状態	次のどちらかの操作をします。 <ul style="list-style-type: none">データベースをクローズし、回復する。個々の表領域をオフラインにし、それらを回復する。

skip オプションが役立つのは、一時データのみの表領域の回復を避ける場合や、いくつかの表領域の回復を延期する場合です。**skip** オプションは、メディア回復を開始する前に指定した表領域をオフラインにし、メディア回復完了までそれらをオフラインの状態のままにします。

アーカイブ REDO ログまたは増分バックアップ・セットを復元する必要がある場合でなければ、**recover** コマンドの前に、**channel** コマンドを少なくとも1回は実行しておく必要があります。復元が必要なバックアップ・セットに適切なタイプのデバイスを割り当ててください。適切なタイプの記憶デバイスが使用できない場合は、**recover** コマンドは失敗します。

データベースの回復

データベースの完全回復を実行する手順は、制御ファイルが使用可能かどうかによって変わります。

制御ファイルに障害がないときのデータベースの回復

1. RMAN を起動し、ターゲット・データベースおよびオプションでリカバリ・カタログに接続する。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

2. データベースがオープンになっている場合は、これを停止してからマウントする。

```
shutdown immediate;  
startup mount;
```

3. チャネルを割り当てた後、データベースを復元し、回復する。この例では、TEMP 表領域をスキップしています。

```
run {  
    allocate channel chl type disk;  
    restore database;  
    recover database  
        skip tablespace temp;  
}
```

4. 正常に回復できたか出力を調べる。RMAN が必要なデータ・ファイルを復元したら、出力に RMAN-08055 があるか探します。

```

RMAN-08024: channel chl: restore complete
RMAN-03023: executing command: partial resync
RMAN-08003: starting partial resync of recovery catalog
RMAN-08005: partial resync complete

RMAN-03022: compiling command: recover

RMAN-03022: compiling command: recover(1)

RMAN-03022: compiling command: recover(2)

RMAN-03022: compiling command: recover(3)
RMAN-03023: executing command: recover(3)
RMAN-08054: starting media recovery
RMAN-08515: archivelog filename=/oracle/arc_dest/arcr_1_40.arc thread=1
sequence=40
RMAN-08515: archivelog filename=/oracle/arc_dest/arcr_1_41.arc thread=1
sequence=41
RMAN-08055: media recovery complete

RMAN-03022: compiling command: recover(4)
RMAN-08031: released channel: chl

```

バックアップ制御ファイルを使用したデータベースの回復

バックアップ制御ファイルとリカバリ・カタログを使用して復元操作を実行する場合、RMAN は制御ファイルが復元バックアップの構造を反映するよう自動的に調整します。

1. RMAN を起動し、ターゲット・データベースおよびオプションでリカバリ・カタログに接続する。リカバリ・カタログを使用すると、RMAN は制御ファイルを更新します。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

2. データベースをマウントせずにインスタンスを開始します。

```
startup nomount;
```

3. 1 つ以上のチャンネルを割り当てた後、次の操作をする。
 - a. **restore controlfile** コマンドを使用して、CONTROL_FILES 初期化パラメータに指定したすべての場所に、制御ファイルを復元する。
 - b. データベースをマウントする。
 - c. データベースを復元し、回復する。

- d. RESETLOGS オプションを使用してデータベースをオープンする。

```
run {
    allocate channel chl type 'sbt_tape';
    restore controlfile;
    alter database mount;
    restore database;
    recover database;
    sql "ALTER DATABASE OPEN RESETLOGS";
}
```

4. データベースの情報をリセットする。

```
reset database;
```

5. ただちにデータベースをバックアップする。データベースは新しいインカーネーションであるため、RESETLOGS 以前のバックアップは使用できません。たとえば、次のように入力します。

```
run {
    allocate channel chl type 'sbt_tape';
    backup database;
}
```

表領域の回復

表領域回復の手順は、データベースがオープン状態かクローズ状態か、また本来の表領域の位置にアクセス可能かによって変わります。

データベースがクローズ状態のときにアクセス可能な表領域を回復する

1. RMAN を起動し、ターゲット・データベースおよびオプションでリカバリ・カタログに接続する。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

2. チャンネルを割り当てた後、表領域を復元し、それを回復する。この例では、表領域 TBS_3 を回復します。

```
run {
    allocate channel chl type disk;
    restore tablespace tbs_3;
    recover tablespace tbs_3;
}
```

3. 正常に回復したか出力を調べる。

データベースがクローズ状態のときにアクセス不可能な表領域を回復する

1. RMAN を起動し、ターゲット・データベースおよびオプションでリカバリ・カタログに接続する。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

2. チャンネルを割り当てた後、次の操作をする。
 - a. アクセス可能な位置を指定して、破損したデータ・ファイルのファイル名を変更する。
 - b. 新しい位置にバックアップしたデータ・ファイルを復元する。
 - c. 復元したデータ・ファイルの切替えを行い、制御ファイルがこれを現行ファイルとみなすよう設定する。
 - d. 表領域を回復する。

```
run {
    allocate channel chl type disk;
    set newname for datafile '/disk1/oracle/tbs_1.f' to '/disk2/oracle/tbs_
1.f';
    restore tablespace tbs_1;
    switch datafile all;
    recover tablespace tbs_1;
}
```

データベースがオープン状態のときにアクセス可能な表領域を回復する

データ・ファイルが消失または破損したがアクセス可能である場合は、データ・ファイルをデフォルト位置に復元する必要があります。

1. RMAN を起動し、ターゲット・データベースおよびオプションでリカバリ・カタログに接続する。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

2. 次の操作をする。
 - a. 回復が必要な表領域をオフラインにする。
 - b. チャンネルを割り当てる。
 - c. オプションで、回復に必要なアーカイブ REDO ログの復元先を指定する。RMAN はこの場所にアーカイブ REDO ログを復元するため、どこかのアーカイブ REDO ログを使用すべきかを認識しています。
 - d. 表領域を復元し、回復する。
 - e. 表領域をオンラインにする。

```
run {
```

```
sql 'ALTER TABLESPACE user_data OFFLINE TEMPORARY';
allocate channel chl type disk;
set archivelog destination to '/oracle/temp/arcl_restore';
restore tablespace user_data;
recover tablespace user_data;
sql 'ALTER TABLESPACE user_data ONLINE';
}
```

データベースがオープン状態のときにアクセス不可能な表領域を回復する

メディア障害のために表領域またはデータ・ファイルがアクセス不可能な場合、データ・ファイルを新しい位置に復元するか、または既存のデータ・ファイル・コピーに切り替える必要があります。

1. RMAN を起動し、ターゲット・データベースおよびオプションでリカバリ・カタログに接続する。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

2. チャンネルを割り当てた後、次の操作をする。
 - a. 回復が必要な表領域をオフラインにする。
 - b. アクセス可能な位置を指定して、破損したデータ・ファイルのファイル名を変更する。
 - c. 新しい位置にバックアップ・データ・ファイルを復元する。
 - d. 復元したデータ・ファイルの切替えを行い、制御ファイルがこれを現行ファイルとみなすよう設定する。
 - e. 表領域を回復する。
 - f. 表領域をオンラインにする。

```
run {
  sql 'ALTER TABLESPACE user_data OFFLINE IMMEDIATE';
  allocate channel chl type disk;
  set newname for datafile '/disk1/oracle/tbs_1.f' to '/disk2/oracle/tbs_
1.f';
  restore tablespace tbs_1;
  switch datafile all;
  recover tablespace tbs_1;
  sql 'ALTER TABLESPACE tbs_1 ONLINE';
}
```


不完全回復の実行

RMAN では、指定した時刻、SCN、ログ順序番号までのデータベース全体の回復を実行できます。このタイプの回復を**不完全回復 (incomplete recovery)**といいます。データベース全体の回復の場合、**データベースの Point-in-Time 回復 (DataBase Point-In-Time Recovery: DBPITR)**ということもあります。

不完全回復は、完全回復と異なる点がいくつかあります。最も重要な違いは、不完全回復ではデータベースを RESETLOGS オプションでオープンする必要があることです。このオプションを使用すると、オンライン REDO ログには新しいタイムスタンプおよび SCN が記録されます。その結果、古いアーカイブ REDO ログの適用でデータ・ファイルを破損する危険性がなくなります。

不完全回復の後に RESETLOGS オプションでオープンする必要があるため、すべてのデータ・ファイルを回復することが必要です。データ・ファイルのなかには RESETLOGS の前には回復できないもの、および RESETLOGS の後には回復できないものがあります。たとえば、Oracle ではデータ・ファイルがオフラインの場合、RESETLOGS オプションでオープンすることができません。例外は、データ・ファイルが NORMAL モードでオフラインにされたまたは読取り専用の場合です。RESETLOGS の後は、読取り専用または NORMAL モードでオフラインにされた表領域のファイルをオンラインにできます。これらのファイルあるいは表領域には REDO を適用する必要がないからです。

データベースの Point-in-Time 回復 (DataBase Point-In-Time Recovery: DBPITR)を実行する最も容易な方法は、**set until** コマンドを使用する方法です。このコマンドは、同じ **run** コマンド内の後続する **restore**、**switch**、および **recover** の各コマンドに対して、必要な時刻を設定します。**set until** コマンドを **restore** の後で、かつ **recover** の前に指定した場合、データベースを必要な時刻まで回復できないので注意してください。復元されたファイルに設定時刻よりも最近のタイムスタンプがすでにあるからです。したがって、通常 **set until** コマンドは **restore** コマンドまたは **switch** コマンドの前に指定するのが最善の方法です。

関連項目 : **set until** コマンドの構文は、11-147 ページの「**untilClause**」を参照してください。

リカバリ・カタログを使用した不完全回復の実行

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。時刻を指定した回復を行う場合、RMAN を起動する前に時刻の書式を指定する環境変数の設定が必要ですので注意してください (5-2 ページの「**NLS 環境変数の設定**」を参照)。たとえば、次のように入力します。

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

指定時刻までデータベースを回復する

1. RMAN を起動し、ターゲット・データベースおよびオプションでリカバリ・カタログに接続する。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

オプションで、接続時にログ・ファイルを指定します。

```
% rman target / catalog rman/rman@rcat log = rman_log
```

2. データベースがオープンになっている場合は、停止してからマウントする。

```
shutdown immediate;  
startup mount;
```

3. どの時刻まで回復する必要があるかを定める。たとえば、午前 9 時 15 分に、ユーザーが午前 9 時 2 分に表領域を誤って削除したことを発見した場合、削除が生じる直前の午前 9 時まで回復できます。その時刻以降にデータベースに加えた変更は消失します。

4. **run** コマンドのなかで、次の操作を実行する。

- a. NLS_LANG および NLS_DATE_FORMAT で指定した日付書式を使用して、回復終了時刻を設定する。
- b. チャンネルを割り当てる。
- c. データベースを復元する。
- d. データベースを回復する。
- e. RESETLOGS オプションを使用してデータベースをオープンする。

たとえば、このジョブでは 11 月 15 日午前 9 時まで不完全回復を実行します。

```
run {  
    set until time 'Nov 15 1998 09:00:00';  
    allocate channel chl type 'sbt_tape';  
    restore database;  
    recover database;  
    sql 'ALTER DATABASE OPEN RESETLOGS';  
}
```

5. データベースをリセットする。

```
reset database;
```

6. ただちにデータベースをバックアップする。データベースは新しいインカーネーションであるため、RESETLOGS 以前のバックアップは使用できません。たとえば、次のように入力します。

```
run {  
    allocate channel chl type 'sbt_tape';  
    backup database;  
}
```

指定 SCN までデータベースを回復する

1. RMAN を起動し、ターゲット・データベースおよびオプションでリカバリ・カタログに接続する。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

オプションで、接続時にメッセージ・ログ・ファイルを指定します。

```
% rman target / catalog rman/rman@rcat log = rman_log
```

2. データベースがオープンになっている場合は、停止してからマウントする。

```
shutdown immediate;
startup mount;
```

3. どの SCN まで回復が必要かを定める。たとえば、表領域 TBS_1 のバックアップを作成し、そのしばらく後にユーザーが TBS_3 にあるデータ・ファイルを誤って上書きした場合、**list** コマンドを発行して TBS_1 の SCN を決定し、次に昨日の全体データベース・バックアップを復元し、その SCN までデータベースを回復することができます。

4. run コマンドのなかで、次の操作を実行する。

- a. 回復終了 SCN を設定する。
- b. チャンネルを割り当てる。
- c. データベースを復元する。
- d. データベースを回復する。
- e. RESETLOGS オプションを使用してデータベースをオープンする。

たとえば、このジョブでは SCN1000 まで不完全回復を実行します。

```
run {
    set until scn 1000;
    allocate channel ch1 type 'sbt_tape';
    restore database;
    recover database;
    sql 'ALTER DATABASE OPEN RESETLOGS';
}
```

5. データベースをリセットする。

```
reset database;
```

6. ただちにデータベースをバックアップする。データベースは新しいインカーネーションであるため、RESETLOGS 以前のバックアップは使用できません。たとえば、次のように入力します。

```
run {
    allocate channel chl type 'sbt_tape';
    backup database;
}
```

指定ログ順序番号までデータベースを回復する

1. RMAN を起動し、ターゲット・データベースおよびオプションでリカバリ・カタログに接続する。たとえば、次のように入力します。

```
% rman target / catalog rman/rman@rcat
```

オプションで、接続時にメッセージ・ログ・ファイルを指定します。

```
% rman target / catalog rman/rman@rcat log = rman_log
```

2. データベースがオープンになっている場合は、停止してからマウントする。

```
shutdown immediate;
startup mount;
```

3. どのログ順序番号まで回復が必要か決める。たとえば、V\$LOG_HISTORY に問い合せて、アーカイブしてある REDO ログを表示する。

RECID	STAMP	THREAD#	SEQUENCE#	FIRST_CHAN	FIRST_TIM	NEXT_CHANG
1	344890611	1	1	20037	24-SEP-98	20043
2	344890615	1	2	20043	24-SEP-98	20045
3	344890618	1	3	20045	24-SEP-98	20046
4	344890621	1	4	20046	24-SEP-98	20048
5	344890624	1	5	20048	24-SEP-98	20049
6	344890627	1	6	20049	24-SEP-98	20050
7	344890630	1	7	20050	24-SEP-98	20051
8	344890632	1	8	20051	24-SEP-98	20052
8 rows selected.						

4. run コマンドのなかで、次の操作を実行する。

- a. 回復終了のログ順序番号を設定する。
- b. チャネルを割り当てる。
- c. データベースを復元する。
- d. データベースを回復する。
- e. RESETLOGS オプションを使用してデータベースをオープンする。

たとえば、このジョブではスレッド 1 のログ順序番号 6 まで不完全回復を実行します。

```
run {
    set until logseq 6 thread 1;
    allocate channel chl type 'sbt_tape';
    restore database;
    recover database;
    sql 'ALTER DATABASE OPEN RESETLOGS';
}
```

5. データベースをリセットする。

```
reset database;
```

6. ただちにデータベースをバックアップする。データベースは新しいインカーネーションであるため、RESETLOGS 以前のバックアップは使用できません。たとえば、次のように入力します。

```
run {
    allocate channel chl type 'sbt_tape';
    backup database;
}
```

リカバリ・カタログなしの不完全回復の実行

リカバリ・カタログなしで DBPITR を実行できますが、次の指示を確実に守る必要があります。

- すべての RMAN バックアップ・コマンドを実行した後に、個々の制御ファイルのバックアップをとる。
- すべてのバックアップの RMAN 出力を保存する。

制御ファイルの個別のバックアップ RMAN データベース・バックアップをとってから、制御ファイルのバックアップを作成する。これは、いま作成したデータベース・バックアップについての情報を収めたバックアップ制御ファイルが必要になるからです。datafile_1 をバックアップする際に **include current controlfile** を指定した場合や、データベースのバックアップに制御ファイルのバックアップが含まれている場合でも、そのバックアップ・セットにあるバックアップ制御ファイルは自己参照することができません。以下のコマンドについて検討してみます。

```
backup database;
```

このコマンドは、制御ファイルのバックアップを含むバックアップ・セットを作成します。バックアップ制御ファイルには、自分自身を含むバックアップ・セットのレコードはありません。その結果、このバックアップ制御ファイルを復元し、それをマウントしても、バックアップ・セットからファイルを復元できません。制御ファイルにはこのファイルのレコードがないためです。

制御ファイルを個別にバックアップするには、**run** コマンドでなかで次の順序のコマンドを発行します。

```
backup database;
backup current controlfile; # obtain a useful control file backup.
```

これらのコマンドで2つのバックアップ・セットを作成します。各バックアップ・セットごとにバックアップ制御ファイルがあります。2番目のコマンドで作成した制御ファイルのバックアップは非常に重要なバックアップです。つまり、データベース・バックアップに関連するすべてのレコードを含む制御ファイル・バックアップです。

リカバリ・カタログなしで DBPITR を実行する

1. **nocatalog** オプションを指定して、RMAN を起動し、ターゲット・データベースに接続する。

```
% rman target / nocatalog
```

2. データベースがオープンになっている場合は、停止してからマウントする。

```
startup force mount;
```

3. バックアップ制御ファイルを一時的な場所に復元する。前述したように個別の制御ファイル・バックアップが作成されていれば、RMAN はこのバックアップを使用します。**backup database** と **backup current controlfile** の間の時間間隔にある時間を指定した場合、RMAN は不適切な制御ファイルを選択することになります。

以前述べたように RMAN 出力が保存されていれば、RMAN が選択したバックアップ制御ファイルが正しいものであることを検証できます。**backup current controlfile** コマンドで **tag** オプションを使用し、次に復元でこのタグを使用して RMAN に必要な制御ファイルの選択を強制する方法も可能です。

たとえば、次のコマンドを発行して制御ファイルを一時的な場所に復元します。

```
run {
  set until time 'Jun 18 1998 16:32:36';
  allocate channel chl type disk;
  # restore a backup controlfile to a temporary location.
  restore controlfile to '/tmp/cf.tmp';
}
```

RMAN が復元した制御ファイルが、すべてのバックアップを終えた後に **backup current controlfile** コマンドで作成した制御ファイルであることを検証します。**backup current controlfile** コマンドでタグを指定した場合、**restore controlfile** コマンドで **from tag** オプションを指定する方法も可能です。

4. 現行の制御ファイルをバックアップした後にそれを前のステップで復元したバックアップ制御ファイルに置き換える。

```
run {
  allocate channel chl type disk;
  # save a copy of the current controlfile just to be safe
  copy current controlfile to '/tmp/original.cf';
}
```

```

shutdown immediate;
startup nomount;
replicate controlfile from '/tmp/cf.tmp';
alter database mount;
}

```

5. 次の操作を実行する。

- a. 回復の終了時刻、SCN またはログ順序番号（ARCHIVELOG モードで実行する場合）を指定する。
- b. データベースを復元し、回復する。データベースを NOARCHIVELOG モードで実行している場合 **recover** コマンドで **noredo** オプションを指定します。データベースを ARCHIVELOG モードで実行している場合は、**noredo** オプションは除きます。
- c. RESETLOGS オプションを使用してデータベースをオープンする。

```

run {
  set until time 'Jun 18 1998 16:32:36';
  restore database;
  recover database noredo;
  sql 'ALTER DATABASE OPEN RESETLOGS';
}

```

6. データベースをリセットする。

```
reset database;
```

7. ただちにデータベースをバックアップする。

```

run {
  allocate channel chl type disk;
  backup database;
}

```

復元および回復の使用例

復元および回復操作を実行するための使用例を次に挙げます。

- データ・ファイル・コピーを使用した新しいホストへの復元
- 複数データベースが同じ名前を共有する場合の復元
- バックアップ・セットからの制御ファイルの変則的な復元の実行
- オープン状態のデータベースにおけるアクセス不可能なデータ・ファイルの回復
- ディスクおよびテープのバックアップを使用してのアクセス不可能なデータ・ファイルの回復
- データベース全体のメディア障害に対する回復の実行

- [RESETLOGS 以前のバックアップの回復](#)
- [NONARCHIVELOG モードでのデータベースの回復](#)

データ・ファイル・コピーを使用した新しいホストへの復元

データベースをデータ・ファイル・コピーを使用して新しいホストに移動するには、コピーを新しいマシンに手動で転送する必要があります。この例では、リカバリ・カタログを使用しているものとして説明を行います。

1. ターゲット・データベースおよびリカバリ・カタログに接続してから、**list** コマンドを発行して、データ・ファイル・コピーおよびそれらの主キーのリストを表示する。

```
list copy;
```

2. O/S ユーティリティを使用して、データ・ファイル・コピーを新しいホストにコピーする。たとえば、UNIX ユーザーは次のとおりに入力します。

```
% cp -r /oracle/copies /net/new_host/oracle/dbs
```

3. 旧ホストのデータ・ファイル・コピーをカタログから削除する。たとえば、次のように入力します。

```
change datafile copy 1,2,3,4,5,6,7,9,10 uncatalog;
```

4. 転送したデータ・ファイル・コピーをその新しいファイル名でカタログに登録する。たとえば、次のように入力します。

```
catalog datafilecopy '/oracle/dbs/tbs_1.f', '/oracle/dbs/tbs_2.f',  
'/oracle/dbs/tbs_3.f', '/oracle/dbs/tbs_4.f', '/oracle/dbs/tbs_5.f',  
'/oracle/dbs/tbs_6.f', '/oracle/dbs/tbs_7.f', '/oracle/dbs/tbs_8.f',  
'/oracle/dbs/tbs_9.f', '/oracle/dbs/tbs_10.f';
```

5. 9-4 ページの「[同じファイル・システムの新しいホストへのターゲット・データベースの移動](#)」または 9-7 ページの「[異なるファイル・システムの新しいホストへのターゲット・データベースの移動](#)」に説明されている復元操作および回復操作を実行する。チャネルは、'*sbt_tape*' ではなくて **type disk** を指定します。

複数データベースが同じ名前を共有する場合の復元

データベース識別子は、データベース作成時に計算する 32 ビットの数値です。他のデータベースと同じ名前を共有するデータベースを復元する場合は、その識別が必要です。RMAN の **set dbid** コマンドを使用し、そのデータベース識別子によってデータベースを指定します。

復元が必要なデータベースの DBID の取得

RMAN の出力を保存している場合は、データベース識別子を決定するためにこの情報を参照してください。ユーザーがデータベースに接続する際には、RMAN が必ずこの情報を表示します。

```
% rman target /
```

```
Recovery Manager: Release 8.1.5.0.0
```

```
RMAN-06005: connected to target database: RMAN (DBID=1231209694)
```

RMAN の出力を保存していないが復元操作のためにデータベースの DBID の値が必要な場合は、RC_DATABASE または RC_DATABASE_INCARNATION リカバリ・カタログ・ビューによってデータベース識別子を取得してください。

この使用例ではリカバリ・カタログに登録されたデータベースの名前は一意ではないと想定しているので、正しい DBID の決定にはなんらかの一意の情報を使用する必要があります。復元の必要があるデータベースに対応付けられたデータ・ファイルまたはオンライン REDO ログの名前がわかっており、かつこのファイル名がリカバリ・カタログに登録されたすべてのデータベース間で一意である場合は、次の問合せの中で *filename_of_log_or_df* のファイル名をこの完全修飾ファイル名に置き換えます。次の問合せのどちらかを実行して、DBID を決定します。

```
SELECT distinct db_id
FROM db, dbinc, dfatt
WHERE db.db_key = dbinc.db_key
      AND dbinc.dbinc_key = dfatt.dbinc_key
      AND dfatt.fname = 'filename_of_log_or_df';
```

```
SELECT distinct db_id
FROM db, dbinc, orl
WHERE db.db_key = dbinc.db_key
      AND dbinc.dbinc_key = orl.dbinc_key
      AND orl.fname = 'filename_of_log_or_df';
```

DBID を使用したバックアップ制御ファイルの復元

次の条件のすべてを満たすときは、**set dbid** コマンドのみを使用して制御ファイルを復元します。

- 制御ファイルが消失し、バックアップからの復元が必要である。
- リカバリ・カタログを使用している。
- リカバリ・カタログに登録された複数データベースはデータベース名を共有している。

これらの条件が満たされない場合は、RMAN-20005: target database name is ambiguous が戻されます。制御ファイルの復元を試みたときに表示されるメッセージです。

RMAN は復元する制御ファイルを正しく識別しますので、**set dbid** コマンドを使用する必要はありません。

RMAN が **set dbid** を受け入れるのは、ターゲット・データベースに接続していない場合のみです。つまり、**set dbid** は **target** コマンドの前に発行する必要があります。ターゲット・データベースがマウントされていると、RMAN はユーザー指定の DBID がデータベース自体の DBID に一致するか検証し、一致しない場合はエラーを通知します。ターゲット・データベースがマウントされていない場合、RMAN はユーザー指定の DBID を使用して制御ファイルを復元します。制御ファイルの復元が完了した後、データベースをマウントすることで、データベースの残りの部分を復元できます。

データベースの ID を設定するには、次のコマンドを入力します。この場合の *target_dbid* は整数値です。

```
set dbid = target_dbid;
```

制御ファイルをデフォルトの位置に復元するには、次のとおりに入力します。

```
run {
    allocate channel dev1 type 'sbt_tape';
    restore controlfile;
    alter database mount;
}
```

バックアップ・セットからの制御ファイルの変則的な復元の実行

次の状況で、RMAN バックアップから制御ファイルを復元するには、変則的な手順を使用する必要があります。

- リカバリ・カタログに同じ名前を持つデータベースが 1 つ以上登録されており、データベースの復元に RMAN の 8.0.5 以前のバージョンを使用する場合（この問題の説明は、9-30 ページの「[複数データベースが同じ名前を共有する場合の復元](#)」を参照）。
- リカバリ・カタログを使用しておらず、かつ唯一の制御ファイル・バックアップが RMAN バックアップ・セット内にある場合。

RMAN バックアップ・セット内を除き他に制御ファイルのバックアップがなく、かつ復元操作の実行に制御ファイルが必要な場合、次の PL/SQL プログラムを使用してバックアップ・セットから制御ファイルを取り出す必要があります。

ターゲット・データベースに SYSDBA として接続している間に、SQL*Plus からこのプログラムを実行します。

```
DECLARE
    devtype varchar2(256);
    done    boolean;
BEGIN
    devtype := dbms_backup_restore.deviceallocate('devtype', params=> '');
    # Replace 'devtype' with the device type you used when creating the backup: disk or
    # sbt_tape. If you used an sbt_tape device and specified a 'parms' option on the RMAN
```

```
# allocate channel command, then put that parms data in the 'params' operand here.

dbms_backup_restore.restoresetdatafile;

dbms_backup_restore.restorecontrolfileto('/tmp/foo.cf');
# This path specifies the location for the restored control file. If there are multiple
# control files specified in the init.ora file, copy the control file to all specified
# locations before mounting the database.

dbms_backup_restore.restorebackuppiece('handle',done=>done);
# Replace 'handle' with the your backup piece handle. This example assumes that the
# backup set contains only one backup piece. If there is more than one backup piece in
# the backup set (which only happens if the RMAN command set limit kbytes is used), then
# repeat the restorebackuppiece statement for each backup piece in the backup set.

END;
/
```

制御ファイルの復元が正常に終了した後は、データベースをマウントして復元操作および回復操作を実行できます。

オープン状態のデータベースにおけるアクセス不可能なデータ・ファイルの回復

この使用例では、データベースはオープン状態ですがデータ・ファイルにアクセスできない状態を想定しています。次の SQL 問合せを実行して、その状態を調べます。

```
SELECT * FROM v$recover_file;

FILE# ONLINE  ERROR          TIME
-----
19 ONLINE  FILE NOT FOUND
```

次に RMAN を起動し、ターゲット・データベースおよびリカバリ・カタログ・データベースに接続します。

```
% rman target / catalog rman/rman@rcat
```

report コマンドを発行して、データ・ファイルの表領域およびファイル名を決定します。

```
RMAN> report schema;

RMAN-03022: compiling command: report
Report of database schema
File K-bytes    Tablespace          RB segs Name
-----
1      47104 SYSTEM              YES   /oracle/dbs/tbs_01.f
2       978 SYSTEM              YES   /oracle/dbs/tbs_02.f
3       978 TBS_1                NO    /oracle/dbs/tbs_11.f
```

4	978 TBS_1	NO	/oracle/dbs/tbs_12.f
5	978 TBS_2	NO	/oracle/dbs/tbs_21.f
6	978 TBS_2	NO	/oracle/dbs/tbs_22.df
7	500 TBS_1	NO	/oracle/dbs/tbs_13.f
8	500 TBS_2	NO	/oracle/dbs/tbs_23.f
9	500 TBS_2	NO	/oracle/dbs/tbs_24.f
10	500 TBS_3	NO	/oracle/dbs/tbs_31.f
11	500 TBS_3	NO	/oracle/dbs/tbs_32.f
12	500 TBS_4	NO	/oracle/dbs/tbs_41.f
13	500 TBS_4	NO	/oracle/dbs/tbs_42.f
14	500 TBS_5	YES	/oracle/dbs/tbs_51.f
15	500 TBS_5	YES	/oracle/dbs/tbs_52.f
16	5120 SYSTEM	YES	/oracle/dbs/tbs_03.f
17	2048 TBS_1	NO	/oracle/dbs/tbs_14.f
18	2048 TBS_2	NO	/oracle/dbs/tbs_25.f
19	2048 TBS_3	NO	/oracle/dbs/tbs_33.f
20	2048 TBS_4	NO	/oracle/dbs/tbs_43.f
21	2048 TBS_5	YES	/oracle/dbs/tbs_53.f

メディア障害について調査する直前にデータ・ファイルをオンラインにする必要があるの
で、データ・ファイルを新しい位置に復元し、そのデータ・ファイルのコピーに切り替えま
す。

```
run {
  sql 'ALTER TABLESPACE tbs_3 OFFLINE IMMEDIATE';
  allocate channel chl type disk;
  set newname for datafile '/oracle/dbs/tbs_33.f' to '/oracle/temp/tbs_33.f';
  restore tablespace tbs_3;
  switch datafile all;
  recover tablespace tbs_3;
  sql 'ALTER TABLESPACE tbs_3 ONLINE';
}
```

ディスクおよびテープのバックアップを使用してのアクセス不可能なデータ・ファイルの回復

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・ファイルのコピーに切り替える必要があります。次の復元例では、4つのデータ・ファイルを収めた表領域 TBS_1 を復元し、回復します。これらのファイルのコピーのいくつかはディスク上にあり、いくつかのバックアップはテープ上にあるので、例ではディスクおよびテープの双方から **restore** ができるように、ディスク・チャンネルとメディア管理チャンネルを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';
set newname for datafile '/disk7/oracle/tbs12.f'
  to '/disk9/oracle/tbs12.f';
set newname for datafile '/disk7/oracle/tbs13.f'
  to '/disk9/oracle/tbs13.f';
set newname for datafile '/disk7/oracle/tbs14.f'
  to '/disk9/oracle/tbs14.f';
restore tablespace tbs_1;
switch datafile all;      # makes the renamed datafile the current datafile
recover tablespace tbs_1;
sql "ALTER TABLESPACE tbs_1 ONLINE";
}

```

データベース全体のメディア障害に対する回復の実行

次の使用例は、以下のことを想定しています。

- データベース全体、すべての制御ファイル、オンライン REDO ログおよびリカバリ・カタログが消失した。
- メディアの問題を修復してから、データベースを元の場所に復元する。
- テープドライブが4台ある。
- リカバリ・カタログを使用している。

データベースの復元の前に、次の操作が必要です。

- O/S コマンドあるいはユーティリティを使用して、最新のバックアップから `init.ora` ファイル、パスワード・ファイル（使用している場合）、およびリカバリ・カタログを復元する。
- ディスク上にあるがリカバリ・カタログに登録されていないアーカイブ REDO ログ、データ・ファイル・コピー、あるいはバックアップ・セットをカタログに登録する。復元の終了点のログ順序番号までのアーカイブ REDO ログは、リカバリ・カタログにカタログする必要があります。カタログに登録しないと、Recovery Manager は、それらをどこから検索すべきかを認識できません。リカバリ・カタログを頻繁に再同期化し、復元を行った元の最新コピーがある場合、カタログへの登録が必要なアーカイブ REDO ログはそれほど多くはありません。

次の使用例では、使用可能のもののうち、最新のアーカイブログ（スレッド 1 内のログ 124）まで、データベースを復元し、回復します。

- データベースをマウントしないでインスタンスを開始し、接続を DBA ユーザーのみに限定する。
- `init.ora` パラメータ `CONTROL_FILES` により指定した場所に、制御ファイルを復元する。
- 制御ファイルをマウントする。

- リカバリ・カタログにないアーカイブ REDO ログをカタログに登録する。
- データベース・ファイルを元の場所に復元する。ボリューム名が変更されている場合は、復元の前に **set newname** コマンドを使用して改名し、復元後に切替えを実行してデータ・ファイルの新しい位置の制御ファイルを更新します。
- データ・ファイルの回復には、増分バックアップと REDO の組み合わせまたは REDO のみを使用します。RMAN は、指定したログ順序番号に到達すると回復を停止します。
- RESETLOGS モードでデータベースをオープンする。他には適用できるアーカイブ REDO ログがないことが確信できる場合のみ、この最後のステップを実行します。
- オラクル社は、RESETLOGS の後でデータベースをバックアップすることをお勧めします（ただし、この例には含まれません）。

```
% rman target sys/sys_pwd@prod1 catalog rman/rman@rcat
startup nomount dba;

run {
  # If you need to restore the files to new locations, tell Recovery Manager
  # to do this using 'set newname'.
  # set newname for datafile 1 to '/dev/vgd_1_0/rlvt5_500M_1';
  # set newname for datafile 2 to '/dev/vgd_1_0/rlvt5_500M_2';
  # set newname for datafile 3 to '/dev/vgd_1_0/rlvt5_500M_3';
  # set newname for datafile 4 to '/dev/vgd_1_0/rlvt5_500M_4';
  # etc...

  # The set until command is used in case the database
  # structure has changed in the most recent backups, and you wish to
  # recover to that point-in-time. In this way Recovery Manager restores
  # the database to the same structure the database was at that time.
  set until logseq 124 thread 1;

  allocate channel t1 type 'SBT_TAPE';
  allocate channel t2 type 'SBT_TAPE';
  allocate channel t3 type 'SBT_TAPE';
  allocate channel t4 type 'SBT_TAPE';

  restore controlfile;
  alter database mount;

  # Catalog any archivelogs that are not in the recovery catalog
  # catalog archivelog '/oracle/db_files/prod1/arch/arch_1_123.rdo';
  # catalog archivelog '/oracle/db_files/prod1/arch/arch_1_124.rdo';
  # etc...
  restore database;

  # Update the control file by telling it the new location of the datafiles
  # only if you used 'set newname for datafile' above.
  # switch datafile all;
  recover database;
```

```
# Complete this last step only if no more archived logs need to be applied.
sql 'ALTER DATABASE OPEN RESETLOGS';
}
```

RESETLOGS 以前のバックアップの回復

次のことを想定します。

- リカバリ・カタログを使用する。
- PROD1 のバックアップを 1998 年 7 月 2 日に作成した。
- 1998 年 7 月 10 日にこのデータベースの不完全回復を行い、RESETLOGS オプションでオープンした。これにより新しいインカーネーションが作成された。
- 必要とする重大データが 1998 年 7 月 8 日午前 8 時にデータベースから削除されたことを 7 月 25 日に発見した。
- PROD1 を前のインカーネーションにリセットし、7 月 2 日のバックアップを復元し、7 月 8 日午前 7 時 55 分まで回復することに決定した。

```
# obtain primary key of old incarnation
list incarnation of database prod1;
```

List of Database Incarnations

DB Key	Inc Key	DB Name	DB ID	CUR	Reset SCN	Reset Time
1	2	PROD1	1224038686	NO	1	02-JUL-98
1	582	PROD1	1224038686	YES	59727	10-JUL-98

```
# reset database to old incarnation
reset database to incarnation 2;
# recover it
run {
    set until time 'Jul 8 1998 07:55:00';
    allocate channel dev1 type disk;
    shutdown abort;
    startup nomount;
    restore controlfile;
    alter database mount;
    restore database;
    recover database;
    sql 'alter database open resetlogs';
}
# make this new incarnation the current incarnation
reset database;
```

NONARCHIVELOG モードでのデータベースの回復

増分バックアップを使用して、NOARCHIVELOG モードで実行しているデータベース・モードを回復することができます。次の使用例を想定してみます。

- PROD1 のデータベースを NOARCHIVELOG モードで実行している。
- リカバリ・カタログを使用している。
- 月曜日にデータベース PROD1 のレベル 0 バックアップを作成した。
- 水曜日と金曜日の午前 7 時に、レベル 1 増分バックアップの差分を作成する。
- データベースは日曜日にメディア障害の被害を受け、これによってデータ・ファイルの半分が消失する。

この場合、金曜日が最新の増分バックアップの日付であることから、金曜日までの不完全メディア回復を実行してください。RMAN は回復中にアーカイブログを探す前に必ず増分バックアップを探すことに注意してください。

ユーザーが **recover** コマンドで **noredo** オプションを指定すれば、RMAN は必要な不完全メディア回復を自動的に実行できます。**noredo** オプションの指定がない場合、RMAN は金曜日の増分バックアップを適用した後アーカイブ REDO ログを探し、それらを探すことができない場合はエラー・メッセージを発行します。

PROD1 およびカタログ・データベースに接続後、次のコマンドを使用してデータベースを回復します。

```
run {
  allocate channel dev1 type 'sbt_tape';
  restore database;
  recover database noredo;
  sql 'ALTER DATABASE OPEN RESETLOGS';
}
```

Recovery Manager による複製データベースの作成

この章では、Recovery Manager を使用して、テスト目的の複製データベースを作成する方法を説明します。項目は次のとおりです。

- 複製データベース作成の概要
- 複製データベースのローカルまたはリモート・ホストへの作成
- 複製データベースの使用例

複製データベース作成の概要

RMAN の **duplicate** コマンドを使用すると、元のデータベースを保持しながら、ターゲット・データベースを使用してテスト・データベースを作成することができます。このコマンドは、ターゲット・データベースのファイルのイメージ・コピーまたはバックアップ・セットを使用し、新たなデータベースを生成します。本番データベースを週 7 日、1 日 24 時間実行する必要がある場合は、複製データベースが特に役に立ちます。

複製操作の一環として、RMAN は次の操作を管理します。

- ターゲット・データベースのデータ・ファイルを複製データベースに復元し、すべての使用可能なアーカイブ REDO ログおよび増分バックアップを使用して不完全回復を実行する。
- 不完全回復の後、複製データベースを RESETLOGS オプションでオープンし、REDO ログを作成する。
- 複製データベースに一意のデータベース識別子を新たに生成する。

ターゲット・データベースの復元では、次のことが可能です。

- 読取り専用の表領域を **skip readonly** 句でスキップする。読取り専用の表領域は、デフォルトでは含まれています。省略したものを、後から追加することができます。
- 複製データベースを新しいホストに作成する。新しいホストでもディレクトリ構造が同じ場合は、**nofilenamecheck** を使用し、ターゲット・データ・ファイルのファイル名を複製データ・ファイルのファイル名に再使用できます。
- 複製データベースを作成するときに **set until** オプションを使用し、複製データベースを現行以外の時刻に回復する。デフォルトでは、**duplicate** コマンドはターゲット・データベースの最新のバックアップを使用してデータベースを作成してから、増分バックアップおよびアーカイブ REDO ログのバックアップに含まれる一貫性をもつ最新の時点まで回復を実行します。
- リカバリ・カタログを指定しないで **duplicate** コマンドを使用する。
- 複製データベースをターゲット・データベースと同じリカバリ・カタログに登録する。このオプションが可能なのは、複製のとき複製データベースにデータベース識別子を新たに割り当てからです。O/S ユーティリティを使用してターゲット・データベースをコピーした場合、コピー・データベースのデータベース識別子が元のままなので、同じリカバリ・カタログには登録できません。

関連項目 : **duplicate** コマンドの構文は、11-69 ページの「**duplicate**」を参照してください。

制限事項

RMAN の複製機能には次の制限があります。これらのことは実行できません。

- オフライン表領域の複製。ただし、複製後に手動でこれらの表領域を追加できます。
- ターゲット・バックアップのなかに存在しないものがある場合に、データベースを複製すること。RMAN は、次の複製を試みます。
 - データ・ファイルがオンラインであるかどうかにかかわらず、オンライン表領域に含まれるすべてのデータ・ファイル。
 - IMMEDIATE オプションでオフラインにしたすべての表領域。

表領域またはデータ・ファイルのバックアップに有効なものがなければ、コマンドは失敗します。

- 複製データ・ファイルの名前がターゲット・データ・ファイルと同じで、かつデータベースが同じホストにある場合は、**nofilenamecheck** を使用する。**nofilenamecheck** を指定すると、RMAN はデータ・ファイルを上書きするか、または次のエラーを表示します。

```
RMAN-10035: exception raised in RPC: ORA-19504: failed to create file
"/oracle/dbs/tbs_01.f"
ORA-27086: skgfglk: unable to lock file - already in use
SVR4 Error: 11: Resource temporarily unavailable
Additional information: 8
RMAN-10031: ORA-19624 occurred during call to
DBMS_BACKUP_RESTORE.RESTOREBACKUPPIECE
```

データベースに同じファイル名が必要で、かつデータベースが異なるホストにある場合は、**nofilenamecheck** オプションを使用する必要があります。

- ターゲット・データベースをオープンしていないときに複製する。
- あるホストにあるイメージ・コピーを使用して新しいホストへの複製データベースを作成する。イメージ・コピーのかわりに、テープ・バックアップを使用する必要があります。

複製データベース用のファイルの作成

データベースを複製するときは、次の操作を実行する必要があります。

- [制御ファイルの作成](#)
- [オンライン REDO ログの作成](#)
- [データ・ファイルの改名](#)

制御ファイルの作成

duplicate コマンドは、複製データベースの `init.ora` ファイルにリストされている名前を使用して、制御ファイルを作成します。複製データベース制御ファイルの名前を選択するときは、ターゲット・データベースの `init.ora` で設定されている本番ファイルを上書きしないようにしてください。

オンライン REDO ログの作成

複製オンライン REDO ログの名前を作成する場合、次の選択肢があります。これらを優先順に示します。

表 10-1 REDO ログのファイル名作成の優先順序

順序	方法	結果
1	duplicate コマンドの logfile 句を指定する。	指定のとおり REDO ログを作成する。
2	<code>LOG_FILE_NAME_CONVERT</code> 初期化パラメータを設定する。	ターゲット・ファイル名を、たとえば <code>log.*</code> から <code>duplog.*</code> に、変換する。 注意: このパラメータを使用した場合、REDO ログはそのサイズが一致する限り、あらかじめ存在していても問題はありません。このパラメータはログ作成時に reuse パラメータを使用するからです。
3	前述のどれも実行しない。	ターゲット・ファイル名を再使用する。この方法を使用するときは、 nofilenamecheck オプションを指定する必要があります。

RMAN が REDO ログをどのように改名するかは、優先順序によって決まります。たとえば、ユーザーが **logfile** 句と `LOG_FILE_NAME_CONVERT` パラメータの両方を指定した場合、RMAN は **logfile** 句を使用します。すべてのオプションを指定した場合、RMAN は **logfile** 句を使用し、他のものは無視します。

警告： ターゲット・データベースと複製データベースが同一ホスト内に存在する場合、ターゲット・データベースが使用中のオンライン REDO ログの名前を使用しないようにしてください。また、複製データベースが別のホストにあり **nofilenamecheck** を使用していない場合も、ターゲット・データベースが使用中の REDO ログの名前を使用しないようにしてください。

データ・ファイルの改名

複製データ・ファイルに異なる名前を付ける必要がある場合は、パラメータまたはコマンドを使用して、それを指定する必要があります。データ・ファイルの改名には、表に示す選択肢があります。これらを優先順に示します。

表 10-2 データ・ファイルのファイル名作成の優先順序

順序	方法	結果
1	set newname コマンドを発行する。	新しいデータ・ファイルのファイル名を作成する。このコマンドは、ファイルの改名が必要になるたびに、再発行する必要があります。
2	set auxname コマンドを発行する。	新しいデータ・ファイルのファイル名を作成する。この設定は、 set auxname ... to null コマンドで無効にされるまで、有効です。
3	DB_FILE_NAME_CONVERT 初期化パラメータを設定する。	たとえば、ターゲット・ファイル名を tbs.* から dupdbs.* に、変換する。 set newname および set auxname で改名されることがないファイルに対して、このパラメータが使用できます。
4	前述のどれも実行しない。	ターゲット・ファイル名を再使用する。この方法を使用するときは、 nofilenamecheck オプションを指定する必要があります。

RMAN がどのようにファイルに名前をつけるかは、優先順序によって決まります。たとえば、すべてのコマンドおよび初期化パラメータを指定した場合、RMAN は **set newname** を使用します。**set auxname** コマンドと DB_FILE_NAME_CONVERT を指定した場合、TMAN は **set auxname** を使用します。最初の 3 つのオプションのいずれも指定しない場合、RMAN は複製ファイルに対して元のターゲット・ファイル名を使用します。

読取り専用表領域のスキップ **skip readonly** を指定した場合、RMAN はこれらの表領域のデータファイルを複製しません。以下のビューまたは表には、スキップされるデータ・ファイルおよび表領域が次の値で表示されます。

表またはビュー	列	値
V\$DATAFILE	STATUS	OFFLINE
V\$DATAFILE	ENABLED	READ ONLY
V\$DATAFILE	NAME	MISSINGxxx
SYS.DBA_DATA_FILES	STATUS	AVAILABLE
SYS.DBA_TABLESPACES	STATUS	READ ONLY

NORMAL モードでオフラインにされた表領域のスキップ 表領域を NORMAL モード・オプションでオフラインにしたとき、RMAN はこのような表領域のデータ・ファイルを複製しません。複製の後で、手動でこのような表領域の追加または削除を行うことができます。

以下のビューまたは表には、スキップされるデータ・ファイルおよび表領域が次の値で表示されます。

表またはビュー	列	値
V\$DATAFILE	STATUS	OFFLINE
V\$DATAFILE	ENABLED	DISABLED
V\$DATAFILE	NAME	MISSINGxxx
SYS.DBA_DATA_FILES	STATUS	AVAILABLE
SYS.DBA_TABLESPACES	STATUS	OFFLINE

IMMEDIATE オプションで表領域をオフラインにした場合、RMAN は表領域をスキップせずに複製します。オンライン表領域の場合と同じく、RMAN では複製にも有効なバックアップが必要です。

ファイル名チェックの禁止 `set newname`、`set newname` または `DB_FILE_NAME_CONVERT` はターゲット・データベースですでに使用中の名前を生成することができます。この場合、`nofilenamecheck` を指定して、エラーを避けます。たとえば、ホスト A には 2 つのデータベースがあるとします。datafile1 の名前は `/oracle/data/file1.f` で、datafile2 の名前は `/oracle/data/file2.f` です。ホスト B に複製するとき次のコマンドを発行します。

```
run {
  set newname for datafile 1 to /oracle/data/file2.f; # rename datafile 1 as file2.f
  set newname for datafile 2 to /oracle/data/file1.f; # rename datafile 2 as file1.f
  allocate ...
  duplicate target database to newdb;
}
```

すべてのデータ・ファイルに対して `set newname` コマンドを発行しても、複製ファイル名がまだターゲット・データベースで使用中なので、`duplicate` コマンドは正常に実行されません。ターゲットの datafile 1 は `/oracle/datafile2.f` を使用しておらず、ターゲットの datafile 2 も `/oracle/datafile1.f` を使用していませんが、複製データベースの 1 つがターゲット・ファイル名を使用しているために、`nofilenamecheck` を指定してエラーを避ける必要があります。

注意： すべてのデータ・ファイルが DB_FILE_NAME_CONVERT によって変換される場合、つまりデータ・ファイルのすべてが同じ接尾辞または接頭辞をもつ場合、**set newname** と **set auxname** のどちらも使用せずに、DB_FILE_NAME_CONVERT のみを使用してください。

複製用の補助インスタンスの準備

RMAN により複製データベースを作成するには、次の要件があります。

- 補助インスタンスの Oracle パスワード・ファイルの作成
- 補助インスタンスのパラメータ・ファイルの作成
- 補助インスタンスの起動
- Net8 と補助インスタンスとの接続性の確認
- ターゲット・データベースのオープン
- RMAN コマンド行インタフェースの起動
- 必要なバックアップおよびアーカイブ REDO ログの確認
- 補助チャネルの割当て

補助インスタンスの Oracle パスワード・ファイルの作成

Oracle パスワード・ファイルの作成と維持の詳細は、『Oracle8i 管理者ガイド』を参照してください。

補助インスタンスのパラメータ・ファイルの作成

補助インスタンスの `init.ora` ファイルを作成し、次のパラメータ要件を設定する。

パラメータ	指定
DB_NAME	duplicate コマンドで使用する名前と同じ名前。
CONTROL_FILES	10-4 ページの「 制御ファイルの作成 」を参照。

オプションとして、次のパラメータを設定します。

パラメータ	指定
DB_FILE_NAME_CONVERT	10-5 ページの「データ・ファイルの改名」を参照。
LOG_FILE_NAME_CONVERT	10-4 ページの「オンライン REDO ログの作成」を参照。

SYSDBA として Net8 への接続を可能にするパラメータも含めて、その他のパラメータを必要に応じて設定します。同じホスト、または異なるファイル・システムを持つ新しいホストに複製するとき、パス名を指定するすべてのパラメータに特に注意してください。

次に示すのは、複製データベース用の init.ora パラメータの例です。

```
DB_NAME=newdb
CONTROL_FILES=(/oracle/dup_prod/cf/cf1.f,/oracle/dup_prod/cf/cf2.log)
DB_FILE_NAME_CONVERT=(/oracle/prod/db,/oracle/dup_prod/db)
LOG_FILE_NAME_CONVERT=( "/oracle/prod/log", "/oracle/dup_prod/log")
```

関連項目 : Net8 の詳細は、『Net8 管理者ガイド』を参照してください。

補助インスタンスの起動

RMAN による複製データベースの作成を開始する前に、SQL*Plus を使用して補助インスタンスに接続し、NOMOUNT モード（必要ならばパラメータ・ファイルを指定して）で起動します。この例では、aux_pwd は SYSDBA 認可レベルのユーザーのパスワードであり、aux_str は補助インスタンスのネット・サービス名です。

```
SQL> connect sys/aux_pwd@aux_str
SQL> startup nomount pfile='/oracle/aux/dbs/initAUX.ora';
```

補助インスタンスにはまだ制御ファイルがないので、インスタンスは NOMOUNT モードでしか起動できません。制御ファイルの作成、補助インスタンスのマウントまたはオープンの実行しないでください。

Net8 と補助インスタンスとの接続性の確認

補助インスタンスは Net8 経由でアクセス可能な状態になっていることが必要です。先に進む前に、SQL*Plus を使用して補助インスタンスへの接続を確立できることを確認してください。補助インスタンスには SYSDBA 権限で接続する必要があり、パスワード・ファイルが必要ですので注意してください。

ターゲット・データベースのオープン

RMAN による複製データベースの作成を開始する前に、ターゲット・データベースがオープン状態でない場合は、ターゲット・データベースをオープンします（必要ならばパラメータ・ファイルを指定します）。たとえば、次のように入力します。


```
SQL> startup pfile='/oracle/dbs/initPROD1.ora';
```

RMAN コマンド行インタフェースの起動

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。

- [O/S コマンド行への接続](#)
- [RMAN プロンプトでの接続](#)

O/S コマンド行への接続 補助インスタンスには SYSDBA 権限で接続する必要があり、パスワード・ファイルが必要です。補助インスタンス、ターゲット・インスタンス、およびリカバリ・カタログに接続するには、Recovery Manager の起動時に次の情報を提供します。

```
% rman target sys/target_pwd@target_str catalog rman/cat_pwd@cat_str auxiliary \
> sys/aux_pwd@aux_str
```

このとき、

<i>target_pwd</i>	SYSDBA として接続するためにターゲット・データベースの orapwd ファイルに指定してあるパスワード。
<i>target_str</i>	ターゲット・データベースのネット・サービス名。
<i>cat_pwd</i>	リカバリ・カタログの orapwd ファイルに指定している、ユーザー RMAN のパスワード。
<i>cat_str</i>	リカバリ・カタログ・データベースのネット・サービス名。
<i>aux_pwd</i>	SYSDBA として接続するために補助データベースの orapwd ファイルに指定しているパスワード。
<i>aux_str</i>	補助データベースのネット・サービス名。

RMAN プロンプトでの接続 補助インスタンスへの接続を行わずに RMAN コマンド行インタフェースを起動し、RMAN プロンプトで **connect auxiliary** コマンドを使用して補助インスタンスに接続することもできます。

```
% rman
RMAN> connect auxiliary sys/aux_pwd@aux_str
RMAN> connect target sys/target_pwd@target_str
RMAN> connect catalog rman/cat_pwd@cat_str
```

必要なバックアップおよびアーカイブ REDO ログの確認

ターゲット・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があることを確認してください。すべてのバックアップがない場合には、複製操作は失敗します。データベースのバックアップは全体データベース・バックアップである必要はありません。個々のデータ・ファイルの全体および増分バックアップを混合して使用できます。

必要な時刻、SCN またはログ順序番号まで回復するために必要なすべてのアーカイブ REDO ログのバックアップがすべて存在することを確認してください。

補助チャネルの割当て

duplicate コマンドを発行する前に、同じ **run** コマンドのなかで少なくとも 1 つの補助チャネルを割り当ててください。チャネル・タイプ (**disk** または *'sbt_tape'*) は、ターゲット・データベースのバックアップが存在するメディアと一致する必要があります。バックアップがディスク上にある場合、割り当てるチャネルが多いほど、短時間で実行できます。テープ・バックアップの場合、チャネル数は操作に使用することができるデバイスの数に制限してください。

```
run {
  # to allocate a channel of type 'sbt_tape' issue:
  allocate auxiliary channel ch1 type 'sbt_tape';

  # to allocate three auxiliary channels for disk issue (specifying whatever channel
  # id that you want):
  allocate auxiliary channel aux1 type disk;
  allocate auxiliary channel aux2 type disk;
  allocate auxiliary channel aux3 type disk;
  . . .
}
```

複製データベースのローカルまたはリモート・ホストへの作成

複製データベースを作成するとき、次のオプションを選択できます。

- [同じディレクトリ構造を持つリモート・ホストへのデータベースの複製](#)
- [異なるディレクトリ構造のリモート・ホストへのデータベースの複製](#)
- [ローカル・ホストへの複製データベースの作成](#)

同じディレクトリ構造を持つリモート・ホストへのデータベースの複製

一番簡単なケースは、データベースを別のホストに複製し、同じディレクトリ構造を使用する場合です。この場合、`init.ora` ファイルを変更したり、複製データベースのデータ・ファイルに新しいファイル名を設定したりすることは必要ありません。

同じファイル・システムを持つ別のホスト上にデータベースを複製する方法

1. O/S ユーティリティを使用して、パラメータ・ファイルをターゲット・ホストのディレクトリ構造内の場所から、複製ホストのディレクトリ構造内の同じ場所にコピーする。
2. SQL*Plus を使用して、複製インスタンスを、マウントせずに開始する。たとえば、次のように入力します。

```
SQL> startup nomount pfile=initDUPDB.ora;
```

3. SQL*Plus を使用して、ターゲット・データベースがまだオープン状態でない場合、それをオープンする。たとえば、次のように入力します。

```
SQL> startup pfile=initPROD1.ora;
```

4. 補助インスタンスは Net8 経由でアクセスできる必要がある。先に進む前に、SQL*Plus を使用して補助インスタンスへの接続が確立できることを確認してください。補助インスタンスには SYSDBA 権限で接続する必要があり、パスワード・ファイルが必要です。ご注意ください。
5. RMAN を使用して、ターゲット・データベース、複製データベース、および（使用している場合）リカバリ・カタログ・データベースに接続する。この例では、O/S 認証を使用し、リカバリ・カタログなしで接続を確立しています。

```
% rman target / auxiliary sys/sys_pwd@dupdb
```

次の例では、ユーザー SCOTT には SYSDBA 権限があり、ターゲットにはネット・サービス名を使用しています。

```
% rman auxiliary scott/tiger@dupdb target sys/sys_pwd@prod
```

次の例では、3 つのデータベースへの接続を確立しています。すべてネット・サービス名を使用しています。

```
% rman catalog rman/rman@rcat target sys/sys_pwd@prod1 auxiliary  
scott/tiger@dupdb
```

6. 次の操作を実行する。
 - 少なくとも補助チャネルを 1 つ割り当てる。
 - **nofilenamecheck** パラメータを指定する。

たとえば、次のように入力します。

```
run {  
    allocate auxiliary channel ch1 type 'sbt_tape';  
    duplicate target database to dupdb  
    nofilenamecheck;  
}
```

7. RMAN により、アーカイブ REDO ログおよび増分バックアップで利用できるものはすべて使用して不完全回復を実行し、次に RESETLOGS オプションでデータベースをオープンしてオンライン REDO ログを作成する。

異なるディレクトリ構造のリモート・ホストへのデータベースの複製

異なるファイル・システムを持つホスト上に複製データベースを作成する場合、`init.ora` ファイルのパラメータをいくつか変更したり、複製データベースのデータ・ファイルに新しいファイル名を生成したりする必要があります。

`LOG_FILE_NAME_CONVERT` または `logfile` 句を使用して、オンライン REDO ログのファイル名を変換します。データ・ファイル名の変換には、`DB_FILE_NAME_CONVERT`、`set newname` コマンド、または `set auxname` コマンドを使用します。

関連項目：データ・ファイルのファイル名の変換に関しては、10-5 ページの表 10-2 を参照してください。

DB_FILE_NAME_CONVERT および LOG_FILE_NAME_CONVERT でデータベースを複製する方法

1. O/S ユーティリティを使用して、パラメータ・ファイルをターゲット・ホストのディレクトリ構造内の場所から、複製ホストのディレクトリ構造内の同じ場所にコピーする。確実に次の設定を行ってください。
 - パス名を指定するすべての `*_DEST` および `*_PATH` 初期化パラメータ。
 - `DB_FILE_NAME_CONVERT`。このパラメータは、すべてのターゲット・データ・ファイルを獲得し、たとえば `tbs_*` から `dup_tbs_*` への変換のように、適切に名前を変換する。
 - `LOG_FILE_NAME_CONVERT`。このパラメータは、すべてのオンライン REDO ログを獲得し、たとえば `log_*` から `dup_log_*` への変換のように、適切に名前を変換する。
2. SQL*Plus を使用して、複製インスタンスを、マウントせずに開始する。たとえば、次のように入力します。

```
SQL> startup nomount pfile=initDUPDB.ora;
```

3. SQL*Plus を使用して、ターゲット・データベースがまだオープン状態でない場合、これをオープンする。たとえば、次のように入力します。

```
SQL> startup pfile=initPROD1.ora;
```

4. 補助インスタンスは Net8 経由でアクセスできる必要がある。先に進む前に、SQL*Plus を使用して補助インスタンスへの接続を確立できることを確認してください。補助インスタンスには SYSDBA 権限で接続する必要があり、パスワード・ファイルが必要です。ので注意してください。
5. RMAN を使用して、ターゲット・データベース、複製データベース、および（使用している場合）リカバリ・カタログ・データベースに接続する。この例では、O/S 認証を使用し、リカバリ・カタログなしで接続を確立しています。

```
% rman target / auxiliary sys/sys_pwd@dupdb
```

次の例では、ユーザー SCOTT には SYSDBA 権限があり、ターゲットにはネット・サービス名を使用しています。

```
% rman auxiliary scott/tiger@dupdb target sys/sys_pwd@prod
```

次の例では、3 つのデータベースへの接続を確立しています。すべてがネット・サービス名を使用しています。

```
% rman catalog rman/rman@rcat target sys/sys_pwd@prod1 auxiliary  
scott/tiger@dupdb
```

6. **duplicate** コマンドを発行する。たとえば、次のように入力します。

```
run {  
    allocate auxiliary channel chl type 'sbt_tape';  
    duplicate target database to dupdb;  
}
```

7. RMAN により、アーカイブ REDO ログおよび増分バックアップで利用できるものをすべて使用して不完全回復を実行し、次に RESETLOGS オプションでデータベースをオープンしてオンライン REDO ログを作成する。

DB_FILE_NAME_CONVERT および logfile 句でデータベースを複製する方法

パラメータ LOG_FILE_NAME_CONVERT を使用した複製データベース作成と同じ手順で進めます。ただし、次の点は変えます。

- ステップ 1 では、LOG_FILE_NAME_CONVERT パラメータを設定しない。
- ステップ 6 で、**run** コマンドを次のとおりに変更して、複製データベースの REDO ログ・メンバーを指定する。

```
run {  
    allocate auxiliary channel chl type 'sbt_tape';  
    duplicate target database to dupdb  
    logfile  
        '/oracle/dbs/log1.f' size 200K,  
        '/oracle/dbs/log2.f' size 200K;  
}
```

set newname コマンドを使用してデータベースを複製する方法

1. O/S ユーティリティを使用して、パラメータ・ファイルをターゲット・ホストのディレクトリ構造内の場所から、複製ホストのディレクトリ構造内の同じ場所にコピーする。パス名を指定するすべての *_DEST および *_PATH 初期化パラメータを設定します。
2. SQL*Plus を使用して、複製インスタンスを、マウントせずに開始する。

```
SQL> startup nomount pfile=initDUPDB.ora;
```

3. SQL*Plus を使用して、ターゲット・データベースがまだオープン状態でない場合は、これをオープンする。

```
SQL> startup pfile=initPROD1.ora;
```

4. 補助インスタンスは Net8 経由でアクセスできる必要がある。先に進む前に、SQL*Plus を使用して補助インスタンスへの接続を確立できることを確認してください。補助インスタンスには SYSDBA 権限で接続する必要があり、パスワード・ファイルが必要です。ので注意してください。
5. RMAN を使用して、ターゲット・データベース、複製データベース、および（使用している場合）リカバリ・カタログ・データベースに接続する。この例では、O/S 認証を使用し、リカバリ・カタログなしで接続を確立しています。

```
% rman target / auxiliary sys/sys_pwd@dupdb
```

次の例では、ユーザー SCOTT には SYSDBA 権限があり、ターゲットにはネット・サービス名を使用しています。

```
% rman auxiliary scott/tiger@dupdb target sys/sys_pwd@prod
```

次の例では、3 つのデータベースへの接続を確立しています。すべてがネット・サービス名を使用しています。

```
% rman catalog rman/rman@rcat target sys/sys_pwd@prod1 auxiliary  
scott/tiger@dupdb
```

6. 次の操作を実行する。
 - 少なくとも補助チャンネルを 1 つ割り当てる。
 - ターゲット・データベースで使用したのと同じ REDO ログ・メンバーおよびグループの番号を指定する。
 - 複製データベースのデータ・ファイルに新しいファイル名を指定する。

たとえば、次のように入力します。

```
run {  
    # allocate at least one auxiliary channel of type disk or tape  
    allocate auxiliary channel dupdb1 type 'sbt_tape';  
    . . .  
    # set new filenames for the datafiles  
    set newname for datafile 1 TO '$ORACLE_HOME/dbs/dupdb_data_01.f';  
    set newname for datafile 2 TO '$ORACLE_HOME/dbs/dupdb_data_02.f';  
    . . .  
    # issue the duplicate command  
    duplicate target database to dupdb  
    # create at least two online redo log groups  
    logfile  
        group 1 ('$ORACLE_HOME/dbs/dupdb_log_1_1.f',
```

```

        '$ORACLE_HOME/dbs/dupdb_log_1_2.f') size 200K,
group 2 ('$ORACLE_HOME/dbs/dupdb_log_2_1.f',
        '$ORACLE_HOME/dbs/dupdb_log_2_2.f') size 200K;
}

```

7. RMAN により、アーカイブ REDO ログおよび増分バックアップで利用できるものをすべて使用して不完全回復を実行し、次に RESETLOGS オプションでデータベースをオープンしてオンライン REDO ログを作成する。

set auxname コマンドを使用してデータベースを複製する方法

1. O/S ユーティリティを使用して、パラメータ・ファイルをターゲット・ホストのディレクトリ構造内の場所から、複製ホストのディレクトリ構造内の同じ場所にコピーする。パス名を指定するすべての *_DEST および *_PATH 初期化パラメータを設定する。

2. SQL*Plus を使用して、複製インスタンスを、マウントせずに開始する。

```
SQL> startup nomount pfile=initDUPDB.ora;
```

3. SQL*Plus を使用して、ターゲット・データベースがまだオープン状態でない場合は、これをオープンする。

```
SQL> startup pfile=initPROD1.ora;
```

4. 補助インスタンスは Net8 経由でアクセスできる必要がある。先に進む前に、SQL*Plus を使用して補助インスタンスへの接続を確立できることを確認してください。補助インスタンスには SYSDBA 権限で接続する必要があり、パスワード・ファイルが必要です。ご注意ください。

5. RMAN を使用して、ターゲット・データベース、複製データベース、および（使用している場合）リカバリ・カタログ・データベースに接続する。この例では、O/S 認証を使用し、リカバリ・カタログなしで接続を確立しています。

```
% rman target / auxiliary sys/sys_pwd@dupdb
```

次の例では、ユーザー SCOTT には SYSDBA 権限があり、ターゲットにはネット・サービス名を使用しています。

```
% rman auxiliary scott/tiger@dupdb target sys/sys_pwd@prod
```

次の例では、3 つのデータベースへの接続を確立しています。すべてがネット・サービス名を使用します。

```
% rman catalog rman/rman@rcat target sys/sys_pwd@prod1 auxiliary
scott/tiger@dupdb
```

6. データ・ファイルの補助名を設定する。たとえば、次のように入力します。

```
# set auxiliary names for the datafiles
set auxname for datafile 1 to '/oracle/auxfiles/aux_1.f';
```

```
set auxname for datafile 2 to '/oracle/auxfiles/aux_2.f';
...
set auxname for datafile n to '/oracle/auxfiles/aux_n.f';
```

7. 次の操作を実行する。

- 少なくとも補助チャネルを1つ割り当てる。
- ターゲット・データベースで使したものと同じ REDO ログ・メンバーおよびグループの番号を指定する。

```
run {
  # allocate at least one auxiliary channel of type disk or tape
  allocate auxiliary channel dupdb1 type 'sbt_tape';
  . . .
  # issue the duplicate command
  duplicate target database to dupdb
  . . .
  # create at least two online redo log groups
  logfile
    group 1 ('$ORACLE_HOME/dbs/dupdb_log_1_1.f',
             '$ORACLE_HOME/dbs/dupdb_log_1_2.f') size 200K,
    group 2 ('$ORACLE_HOME/dbs/dupdb_log_2_1.f',
             '$ORACLE_HOME/dbs/dupdb_log_2_2.f') size 200K;
}
```

8. RMAN により、アーカイブ REDO ログおよび増分バックアップで使えるものをすべて使用して不完全回復を実行し、次に RESETLOGS オプションでデータベースをオープンしてオンライン REDO ログを作成する。
9. データ・ファイルが誤って上書きされないよう、データ・ファイルの補助名を未指定にする。たとえば、次のように入力します。

```
# un-specify auxiliary names for the datafiles
set auxname for datafile 1 to null;
set auxname for datafile 2 to null;
...
set auxname for datafile n to null;
```

ローカル・ホストへの複製データベースの作成

ターゲット・データベースと同じホストに複製データベースを作成するときは、異なるディレクトリ構造を持つリモート・ホストに複製する場合と同じ手順で進めます（10-12 ページの「異なるディレクトリ構造のリモート・ホストへのデータベースの複製」）。

ターゲットと同じ \$ORACLE_HOME にデータベースを複製できますが、別のホスト上での変換に使用すると同じ方法を使用して、ファイル名を変換する必要がありますので注意してください。

警告： プライマリ・データベースと同じ \$ORACLE_HOME にデータベースを複製するときは、`nofilenamecheck` オプションは使用しないでください。このオプションを使用した場合、ターゲット・データベースを上書きするか、または複製コマンドにエラーが発生し失敗します。

複製データベースの使用例

次に、複製データベース作成の使用例を示します。

- [新しいファイル名の手作業での設定](#)
- [複製データベースとターゲット・データベースの再同期化](#)
- [最新でない複製データベースの作成](#)

新しいファイル名の手作業での設定

この例は次の状況を想定しています。

- リカバリ・カタログ・データベース RCAT を使用する。
- ターゲット・データベースは HOST1 にあり、9 つのデータ・ファイルがある。
- ターゲット・データベースをリモート・ホスト HOST2 の DUPDB データベースに複製する必要がある。
- HOST1 と HOST2 は異なるファイル・システムを使用している。
- すべてのデータ・ファイルを HOST2 の `/oracle/dbs` サブディレクトリに格納し、各データ・ファイルには `tbs_*` の接頭辞を使用する必要がある。
- O/S ユーティリティを使用して、パラメータ・ファイルを HOST1 から HOST2 の適切なファイルにコピーした。
- パス名を指定するすべての `*_DEST` および `*_PATH` 初期化パラメータをリセットした。
- ターゲット・データベースのすべてのデータ・ファイルのディスク・コピーまたはバックアップ・セットをディスクに格納してあり、それらを O/S ユーティリティを使用し、手動で HOST2 に移動した。
- オンライン REDO ログ・グループは 2 つにする。それぞれサイズ 200K のメンバーが 2 つある。

```
connect target;
connect catalog rman/rman@rcat;
connect auxiliary sys/change_on_install@dupdb;
run {
    allocate auxiliary channel dupdb1 type disk;
    allocate auxiliary channel dupdb2 type disk;
```

```
allocate auxiliary channel dupdb3 type disk;
allocate auxiliary channel dupdb4 type disk;
set newname for datafile 1 TO '$ORACLE_HOME/dbs/tbs_01.f';
set newname for datafile 2 TO '$ORACLE_HOME/dbs/tbs_02.f';
set newname for datafile 3 TO '$ORACLE_HOME/dbs/tbs_03.f';
set newname for datafile 4 TO '$ORACLE_HOME/dbs/tbs_04.f';
set newname for datafile 5 TO '$ORACLE_HOME/dbs/tbs_05.f';
set newname for datafile 6 TO '$ORACLE_HOME/dbs/tbs_06.f';
set newname for datafile 7 TO '$ORACLE_HOME/dbs/tbs_07.f';
set newname for datafile 8 TO '$ORACLE_HOME/dbs/tbs_08.f';
set newname for datafile 9 TO '$ORACLE_HOME/dbs/tbs_09.f';
duplicate target database to dupdb logfile
  group 1 ('$ORACLE_HOME/dbs/log_1_1.f',
           '$ORACLE_HOME/dbs/log_1_2.f') size 200K reuse,
  group 2 ('$ORACLE_HOME/dbs/log_2_1.f',
           '$ORACLE_HOME/dbs/log_2_2.f') size 200K reuse;
}
```

複製データベースとターゲット・データベースの再同期化

この例では、10-17 ページの「[新しいファイル名の手作業での設定](#)」と同じことを前提としています。さらに、常に複製データベースをターゲット・データベースの現行状態にするため、複製データベースを毎日更新する必要があることを前提としています。

```
# start RMAN and then connect to the databases
connect target /
connect catalog rman/rman@rcat
connect auxiliary sys/change_on_install@dupdb

# set auxiliary names for the datafiles only once
set auxname for datafile 1 TO '$ORACLE_HOME/dbs/tbs_01.f';
set auxname for datafile 2 TO '$ORACLE_HOME/dbs/tbs_02.f';
set auxname for datafile 3 TO '$ORACLE_HOME/dbs/tbs_03.f';
set auxname for datafile 4 TO '$ORACLE_HOME/dbs/tbs_04.f';
set auxname for datafile 5 TO '$ORACLE_HOME/dbs/tbs_05.f';
set auxname for datafile 6 TO '$ORACLE_HOME/dbs/tbs_06.f';
set auxname for datafile 7 TO '$ORACLE_HOME/dbs/tbs_07.f';
set auxname for datafile 8 TO '$ORACLE_HOME/dbs/tbs_08.f';
set auxname for datafile 9 TO '$ORACLE_HOME/dbs/tbs_09.f';

# Create the duplicate database. Issue the same command daily
# to re-create the database, thereby keeping the duplicate
# in sync with the target.
run {
  # allocate auxiliary channels
  allocate auxiliary channel dupdb1 type disk;
  allocate auxiliary channel dupdb2 type disk;
  allocate auxiliary channel dupdb3 type disk;
  allocate auxiliary channel dupdb4 type disk;
```

```
duplicate target database to dupdb
logfile
group 1 ('$ORACLE_HOME/dbs/log_1_1.f',
         '$ORACLE_HOME/dbs/log_1_2.f') size 200K reuse,
group 2 ('$ORACLE_HOME/dbs/log_2_1.f',
         '$ORACLE_HOME/dbs/log_2_2.f') size 200K reuse;
}
```

最新でない複製データベースの作成

この例は次の状況を想定しています。

- ターゲット・データベース PROD1 と複製データベース DUPDB は別のホストにあるが、ファイル構造まったく同じである。
- 複製データベースのファイルをターゲット・データベースのファイルとまったく同じ名前にする。
- リカバリ・カタログは使用しない。
- 複製データベースを 1 週間前まで回復する必要がある。

```
connect target sys/change_on_install@prod1
connect auxiliary sys/sysdba@dupdb
run {
    set until time 'sysdate-7';
    allocate auxiliary channel dupdb1 type 'sbt_tape';
    allocate auxiliary channel dupdb2 type 'sbt_tape';
    duplicate target database to dupdb
    nofilenamecheck;
}
```


第 III 部

Recovery Manager リファレンス

Recovery Manager のコマンド構文

この章では、Recovery Manager のコマンドおよび句についてアルファベット順に説明します。

このリファレンス・マニュアルでの表記法

この項では、本書での表記法について説明します。

- テキスト
- 構文図および表記法
- コマンド例

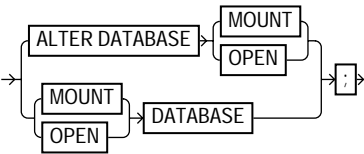
テキスト

このリファレンスのテキストは、次の表記表に従います。

大文字	大文字のテキストは、SQL のコマンドおよびキーワード、ファイル名、表およびビューの列の見出し、初期化パラメータを明示します。
太字	太字テキストは、Recovery Manager のキーワードを明示します。
イタリック	イタリック体テキストは、用語の定義、Recovery Manager のパラメータおよびキーワードではないオプション（たとえば、 <i>integer</i> ）の名前、および Recovery Manager のパラメータのサンプル値（たとえば、 datafile tbs_01.f ）を明示します。

構文図および表記法

構文図 このリファレンスでは、構文図によって Recovery Manager のコマンドを説明します。構文図では、次の例に示すように線と矢印で構文構造を表します。



この項では、構文図の構成要素を説明し、Recovery Manager コマンドの書式の例を示します。構文図の構成要素には、次のものがあります。

キーワード キーワードは、Recovery Manager の構文で特殊な意味を持ちます。構文図で、キーワードは、四角形の箱で囲み、大文字フォントで表します。説明文のなかでは、RMAN キーワードは、太字の小文字で表します。たとえば、**backup database** と表します。RMAN 文においてキーワードは、構文図に示すとおりを使用する必要があります。ただし、大文字と小文字は区別しません。

RMAN のコマンドは、自由形式です。キーワードは、少なくとも 1 つの空白文字で区切る必要があります。その他の制約はありません。コマンドは、複数行になることがあります。

パラメータ パラメータは、構文図でブレースホルダになります。パラメータは、構文図で楕円形の枠内に表します。説明文のなかで、RMAN パラメータは、小文字のイタリック体で表します。たとえば、'*filename*' と表します。通常、パラメータは、データベース・オブジェクト名 (*tablespace_name*)、Oracle データ・タイプ名 (*date_string*)、あるいは句 (*datafileSpec*) を示します。構文図にパラメータがあったときには、RMAN 文の適切なオブジェクトまたは型に置き換えます。たとえば、**duplicate target database to** コマンドを記述する場合、構文図の *database_name* パラメータを、作成する複製データベース名、たとえば *dupdb* に置き換えます。

パラメータの値には、引用符で囲む必要があるものと、任意のものがあります。構文図では一重引用符を使用しますが、二重引用符を使用することもできます。たとえば、'*filename*' とも、"*filename*" とも指定できます。**sql** コマンドの場合は、二重引用符の使用を推奨します。

次のリストに、構文図に表されるパラメータとそれらを置き換える値の例を示します。

パラメータ	説明	例
引用符に囲まれた文字列。 <i>'filename'</i> 、 <i>'tablespace_name'</i> 、 <i>'channel_name'</i> 、 <i>'channel_parms'</i> など。	一重引用符または二重引用符で囲んだ文字列。たとえば、' <i>filename</i> ' または " <i>filename</i> "。引用符で囲んだ文字列には、空白文字、句読点、RMAN キーワードおよび SQL キーワードも使用できます。	"?/dbs/cf.f" 'dev1'
引用符なしの文字列。 <i>channel_id</i> 、 <i>tag_name</i> 、 <i>date_string</i> など。	先頭がアルファベット文字で始まり、空白および句読点のない文字列。	chl
整数	数字のみの文字列。	67843

コマンド例

このリファレンスには、RMAN コマンドの多くの例が示されています。RMAN の要素の使用方法が、これらの例でわかります。次に、**backup** コマンドの例を示します。

```
run {
    allocate channel chl type disk;
    backup database;
}
```

例と説明文では、異なるフォントを使用していることに注意してください。

コマンド入力

コマンドあるいは句の説明は、次の項を含んでいます。

構文	文の要素である、キーワードとパラメータについて説明します。 注意: キーワードおよびパラメータは、常に有効とは限りません。構文上の制約については、必ず各文の「キーワードおよびパラメータ」の項を参照してください。
目的	文の基本的な使用例について説明します。
要件	コマンドを正しく使用するための要件および制約について説明します。
キーワードおよびパラメータ	キーワードおよびパラメータの目的について説明します。制約および使用上の注意についても、この項で説明します。
例	文の各句および各オプションの使用例を示します。 使用上の注意: 例の次のオプションの項では、文の使用方法和使用時機について詳しく説明します。

RMAN コマンドの概要

次の表に RMAN コマンドの機能の概要を示します。リリース 8.0 のコマンドは、すべてリリース 8.1 の RMAN 実行可能ファイルで動作することに注意してください。

表 11-1 Recovery Manager コマンド

コマンド	目的
11-9 ページの「 allocate 」	チャネル (channel) を確立します。つまり、RMAN とデータベース・インスタンスを接続します。
11-9 ページの「 allocateForMaint 」	change コマンドなどのメンテナンス・コマンドの発行の準備として、チャネルを割り当てます。
11-9 ページの「 alterDatabase 」	データベースをマウントまたはオープンします。
11-17 ページの「 archivelogRecordSpecifier 」	バックアップ、復元およびメンテナンスの操作やリカバリ・カタログへの問合せに使用する、アーカイブ REDO ログ・ファイルの範囲を指定します。
11-21 ページの「 backup 」	データベース、表領域、データ・ファイル、あるいはアーカイブ REDO ログ・ファイルのバックアップをとります。
11-32 ページの「 catalog 」	<p>リカバリ・カタログおよび制御ファイルに、データ・ファイル・コピー、アーカイブ REDO ログまたは制御ファイルのコピーに関する情報を書き込みます。</p> <p>データ・ファイル・コピーをレベル 0 バックアップとしてカタログに登録します。これによって、コピーを増分バックアップ計画の一部分として使用することができます。</p> <p>RMAN の導入前あるいは RMAN 以外の方法で作成されたファイル・コピーの存在を記録します。</p>
11-35 ページの「 change 」	<p>バックアップ・ピース、イメージ・コピー、あるいはアーカイブ REDO ログを、available または unavailable にマークします。</p> <p>バックアップ・ピース、イメージ・コピー、あるいはアーカイブ REDO ログをオペレーティング・システムから削除し、リカバリ・カタログ・レコードを deleted 状態に更新します。</p> <p>バックアップ・ピース、データ・ファイル・コピー、あるいはアーカイブ REDO ログが使用可能であるかチェックして、使用可能でない場合は、expired にマークします。</p>

表 11-1 Recovery Manager コマンド (続き)

コマンド	目的
11-39 ページの「 cmdLine 」	<p>ターゲット・データベース、リカバリ・カタログあるいは補助データベースに接続します。</p> <p>リカバリ・カタログなしで RMAN を使用していることを指定します。</p> <p>コマンド・ファイルを指定します。コマンド・ファイルとは、RMAN コマンドを含むユーザー定義ファイルです。</p> <p>RMAN がコマンドの処理結果を記録するファイルを指定します。</p> <p>コマンド・ファイルの内容を上書きではなく、追加します。</p> <p>デバッグするアウトプットを作成し、その場所を指定します。</p>
11-42 ページの 「 completedTimeSpec 」	バックアップまたはコピーを行う時間範囲を指定する句です。
11-44 ページの「 connect 」	RMAN とターゲット・データベース、補助データベース (TSPITR に使用する複製または補助インスタンス)、あるいはリカバリ・カタログ・データベースとの接続を確立します。
11-46 ページの 「 connectStringSpec 」	<p>ターゲット・データベース、リカバリ・カタログ・データベース、あるいは補助データベースに接続するため、ユーザー名、パスワードおよびネット・サービス名を指定します。接続は、ユーザーの認証およびデータベースの識別のために必要です。</p>
11-48 ページの「 copy 」	ファイルのイメージ・コピーを作成します。
11-52 ページの「 createCatalog 」	リカバリ・カタログのためのスキーマを作成します。
11-54 ページの「 createScript 」	ストアド・スクリプトを作成し、後で参照するためにリカバリ・カタログに格納します。
11-57 ページの「 crosscheck 」	ディスクまたはテープに格納したバックアップ・セットがまだ存在するか、判断します。
11-60 ページの「 datafileSpec 」	データ・ファイルをファイル名または絶対ファイル番号で指定します。
11-62 ページの「 debug 」	RMAN のデバッグ機能をオン、オフします。
11-63 ページの「 deleteExpired 」	crosscheck コマンドによって期限切れとマークされたバックアップ・セットを削除し、その参照をリカバリ・カタログおよび制御ファイルから削除します。
11-63 ページの「 deleteScript 」	ストアド・スクリプトをリカバリ・カタログから削除します。
11-66 ページの「 deviceSpecifier 」	バックアップまたはコピーに使用する記憶装置のタイプを指定します。
11-68 ページの「 dropCatalog 」	リカバリ・カタログからスキーマを削除します。
11-69 ページの「 duplicate 」	ターゲット・データベースのバックアップを使用して、テスト目的に使用できる複製データベースを作成します。

表 11-1 Recovery Manager コマンド (続き)

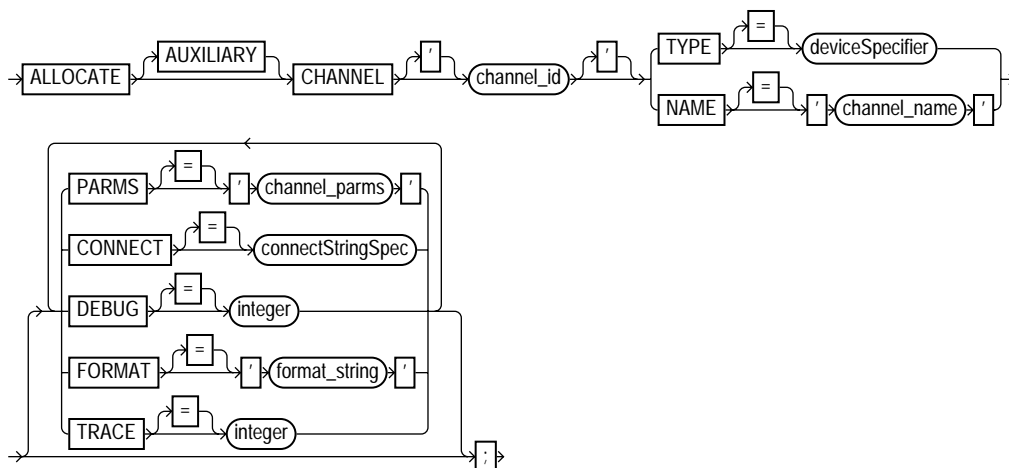
コマンド	目的
11-74 ページの「 host 」	RMAN のなかから O/S コマンド行のサブ・シェルを呼び出します。
11-76 ページの「 list 」	リカバリ・カタログまたはターゲット制御ファイルに記録された特定グループのバックアップ・セットまたはコピーの詳細レポートを作成します。
11-85 ページの「 listObjList 」	データベースまたは 1 つ以上の表領域、制御ファイル、データ・ファイルまたはアーカイブ REDO ログを指定します。
11-87 ページの「 printScript 」	ストアド・スクリプトを RMAN メッセージ・ログ・ファイルに出力します。コマンド行にログ・ファイル名と log 引数を指定します (11-44 ページの「 connect 」を参照)。
11-89 ページの「 recover 」	復元したバックアップ・セットまたはコピーに、REDO ログまたは増分バックアップを適用して、その内容を特定のある時点の状態に更新します。
11-94 ページの「 register 」	ターゲット・データベースをリカバリ・カタログに登録し、RMAN からアクセスできるようにします。
11-96 ページの「 release 」	ターゲット・データベースへ接続した状態で、順次 I/O デバイスを解放します。
11-97 ページの「 releaseForMaint 」	allocate channel コマンドの for delete または for maintenance オプションで指定された順次 I/O デバイスを解放します。
11-98 ページの「 replaceScript 」	リカバリ・カタログに格納されている既存のスクリプトを置換します。既存のものがいない場合、 replace script は、スクリプトを作成します。
11-101 ページの「 replicate 」	制御ファイルを複数の宛先にコピーします。
11-103 ページの「 report 」	リカバリ・カタログの内容の詳細分析を行います。
11-111 ページの「 reset 」	リカバリ・カタログに新規データベースのインカーネーション・レコードを作成します。
11-113 ページの「 restore 」	バックアップ・セットまたはディスク上のコピーによって、現行の位置にファイルを復元します。
11-119 ページの「 resync 」	完全再同期化 (resynchronization) を行います。それによってスナップショット制御ファイルを作成し、リカバリ・カタログをターゲット・データベースの現行の制御ファイルまたはスナップショット制御ファイルと比較して、欠落または変更されている情報があれば更新します。
11-122 ページの「 rmanCmd 」	スタンドアロン・コマンドを実行します。コマンド行インタプリタ (CLI)、たとえば RMAN プロンプトからのコマンドの実行です。
11-125 ページの「 run 」	ジョブ・コマンドをコンパイルし、実行します。すなわち、 run の中カッコのなかの 1 つ以上の文を実行します。
11-128 ページの「 send 」	ベンダー固有の引用符付き文字列を 1 つ以上の特定チャネルに送信します。

表 11-1 Recovery Manager コマンド (続き)

コマンド	目的
11-130 ページの「set」	<p>ターゲット・データ・ファイルの補助ファイル名を指定します。この操作は、TSPITR を実行する場合に役立ちます。</p> <p>メッセージ・ログに記録された実行済み RMAN コマンドを表示します。</p> <p>データベースの DB 識別子を指定します。</p> <p>スナップショット制御ファイルのファイル名を設定します。</p>
11-134 ページの「set_run_option」	<p>データ・ファイルの新規ファイル名を指定します。</p> <p>ブロック破損許容数の上限を指定します。</p> <p>デフォルトのアーカイブ REDO ログの出力先を変更します。</p> <p>バックアップの二重化を指定します。</p> <p>サーバー・プロセスとチャンネルの対応を判断します。</p> <p>指定されたチャンネルの各入力データ・ファイルから読み取るバッファ数を制限します。</p> <p>指定のチャンネルで一定時間に backup 操作がオープンすることができる入力ファイル数を制限します。</p> <p>指定したチャンネルのバックアップ・ピースのサイズを制限します。</p>
11-138 ページの「shutdown」	<p>RMAN を終了せずに、ターゲット・データベースを停止します。このコマンドは、SQL*Plus の SHUTDOWN コマンドと同等です。</p>
11-141 ページの「sql」	<p>Recovery Manager から SQL 文を実行します。</p>
11-143 ページの「startup」	<p>RMAN 環境からデータベースを起動します。このコマンドは、SQL*Plus の STARTUP コマンドと同等です。</p>
11-145 ページの「switch」	<p>データ・ファイルのコピーを現行データ・ファイルとして使用する、すなわち制御ファイルが指すデータ・ファイルを変更します。</p>
11-149 ページの「upgradeCatalog」	<p>リカバリ・カタログ・スキーマを、旧バージョンから、RMAN 実行可能ファイルに必要なバージョンにアップグレードします。</p>
11-151 ページの「validate」	<p>バックアップ・セットを調べて、データの損傷がないかレポートします。RMAN は指定したバックアップ・セットのすべてのバックアップ・ピースを走査し、チェックサムを調べて、必要な場合は内容の復元が正常にできることを検証します。</p>

allocate

構文



目的

チャンネル (channel) を確立します。すなわち、RMAN とデータベース・インスタンスとの接続を行います。各接続では、ターゲット・インスタンスの Oracle サーバー・セッションが開始されます。このサーバー・セッションで、バックアップ・セットまたはコピーのバックアップ、復元および回復の作業を行います。

各チャンネルは、一度に 1 つのバックアップ・セット (**backup**、**restore** または **recover** の場合) または 1 度に 1 つのイメージ・コピー (**copy** の場合) で使用されます。RMAN は、ジョブ終了時点で自動的にチャンネルを解放します。

割り当てられたチャンネル数によって、ジョブでの並列度を制御します。複数チャンネルを同時に割り当て、単一ジョブで複数のバックアップ・セットまたはコピーをパラレルに読み書きできます。複数接続を確立すると、各接続は別個のバックアップ・セットまたはファイル・コピーに対して使用されます。

allocate channel でオペレーティング・システム・リソースを割り当てるかどうかは、オペレーティング・システムにより異なります。プラットフォームによっては、オペレーティング・システム・リソースは、コマンドが発行されるときに割り当てられます。別のプラットフォームでは、オペレーティング・システム・リソースはファイルを読み書きのためにオープンするまで割り当てられません。

注意： `type disk` を指定した場合、サーバー・セッションの作成以外の目的には、O/S リソースは割り当てられません。

要件

- `allocate` は、`run` コマンドの中カッコ内のみで実行してください。
- `backup`、`duplicate` (11-69 ページの「`duplicate`」を参照)、`copy`、`restore`、`recover` または `validate` コマンドを実行する前に、チャンネルの割当てが必要です。
- バックアップを二重化する場合は、`set duplex` コマンド (11-134 ページの「`set_run_option`」を参照) を実行してから、`backup` コマンドに対するチャンネル割当てを実行してください。

キーワードおよびパラメータ

<code>auxiliary</code>	<p>RMAN と補助データベース・インスタンスとの接続を指定します。補助インスタンスは、<code>duplicate</code> コマンドの実行または TSPITR の実行の場合に使用します。補助データベースは、元のデータベースと同じホストにも、異なるホストにも作成することができます。このオプションを指定した場合、補助データベースのマウントが必要です。オープンしておく必要はありません。</p> <p>関連項目： データベースの複製の方法は、11-69 ページの「<code>duplicate</code>」を参照してください。複製データベースへの接続方法については、11-44 ページの「<code>connect</code>」を参照してください。</p>
<code>channel</code> <code>channel_id</code>	<p>RMAN とターゲット・データベース・インスタンスとの接続を指定します。各接続では、データベース・インスタンスの Oracle サーバー・セッションが開始されます。このサーバー・セッションで、バックアップ・セットまたはコピーの作成、復元および回復の作業を行います。</p> <p><code>channel</code> キーワードの後に、チャンネル ID、すなわちチャンネルの名前を指定します。Oracle は、<code>release channel</code> コマンドの実行および I/O エラーのレポートに、この識別子を使用します。</p>
<code>type</code> <code>deviceSpecifier</code>	<p>記憶デバイス (11-66 ページの「<code>deviceSpecifier</code>」を参照) のタイプを指定します。</p> <p>注意： <code>type</code> パラメータを指定しない場合、<code>name</code> パラメータにより、特定の順次 I/O デバイスを指定する必要があります。V\$BACKUP_DEVICE ビューへの問合せで、使用することができるデバイス・タイプと名前がわかります。</p>
<code>name</code> <code>'channel_name'</code>	<p>順次 I/O デバイスの名前を指定します。デバイス名を指定しないと、システムは各記憶デバイスの使用可能デバイスを使用します。このパラメータと <code>type</code> パラメータと一緒に使用しないでください。</p> <p>現在のところ、<code>name</code> パラメータをサポートするプラットフォームはありません。</p>

parms 'channel_parms'	<p>割り当てるデバイスに対するデバイス固有のパラメータを指定します。type disk を指定した場合、この固有の文字列は使用しないでください。</p> <p>parms と type 'sbt_tape' とを一緒に使用した場合、デバイスに対する環境変数を指定できます。構文に沿った例を次に示します。</p> <pre>PARMS="ENV=(var1=value1,var2=value2,var3=value3 . . .)" PARMS="BLKSIZE=integer"</pre> <p>たとえば、次のように指定できます。</p> <pre>PARMS="BLKSIZE=16384,ENV=(NSR_SERVER=tape_server,NSR_CLIENT=oracleclnt, NSR_GROUP=oracle_tapes)"</pre> <p>引用符付き文字列の最大長は 1000 バイトです。</p>
connect <i>connectStringSpec</i>	<p>RMAN がバックアップまたは復元の操作を行うデータベース・インスタンスへの接続文字列を指定します (11-46 ページの「connectStringSpec」を参照)。OPS 構成で別々のインスタンスにバックアップまたは復元の操作を分散する必要がある場合に、このパラメータを使用します。</p> <p>このパラメータの指定がなく、かつ auxiliary オプションも指定していない場合、RMAN はコマンド行パラメータ (11-39 ページの「cmdLine」を参照) で指定したターゲット・データベース・インスタンスか、または connect コマンドを発行したときに接続されたインスタンスに対してすべての操作を実行します。一般的には、connect パラメータと auxiliary オプションを一緒に使用することはできません。</p>
debug integer	<p>このチャンネルで実行するコピー、バックアップおよび復元の操作に関するデバッグ情報を、ターゲット・データベースまたは補助データベースごとに、トレース・ファイルにログを出力するよう Oracle に指定します。オラクル社カスタマ・サポートの指示があるときのみこのパラメータを使用してください。オラクル社カスタマ・サポートがどの整数を使用するかお知らせします。</p>
format 'format_string'	<p>このチャンネルで作成するバックアップ・ピースの名前に使用するフォーマットを指定します。フォーマットの指定がない場合、RMAN はデフォルトの %U を使用します。これにより一意の識別子を保証できます。使用できる format パラメータについては、backup コマンドを参照してください。</p> <p>複数のディスク・チャンネルを割り当て、各チャンネルに別々のファイル・システムへの書き込みをさせる場合に、このパラメータは役に立ちます。format パラメータを backup コマンドで指定すると、allocate channel で指定した format パラメータに上書きします。</p>
trace integer	<p>指定する整数の意味は、メディア管理ソフトウェアにより異なります。一般的には、このパラメータによって、メディア管理ソフトウェアがどのぐらいの量の診断トレース・データを作成するかを制御します。</p>

例

バックアップのために単一チャンネルを割り当て このコマンドで全体データベース・バックアップ用に 1 つのテープ・チャンネルを割り当てます。

```
run {
    allocate channel dev1 type 'sbt_tape';
```

```
        backup database;
    }
```

バックアップ・セットを複数チャンネルに分散 ディスクにバックアップする場合、複数のディスク・ドライブに分散したバックアップが実行できます。ディスク・ドライブごとに1つの type disk チャンネルを割り当て、ファイル名ごとに異なるディスクになるようフォーマット文字列を指定します。

```
run{
    allocate channel disk1 type disk format '/disk1/%d_backups/%U';
    allocate channel disk2 type disk format '/disk2/%d_backups/%U';
    allocate channel disk3 type disk format '/disk3/%d_backups/%U';
    backup database;
}
```

バックアップ・セットを二重化 バックアップ・セットを二重化する場合、チャンネルを割り当てる前に、**set duplex** コマンド（11-134 ページの「[set_run_option](#)」を参照）を指定します。次の例では、datafile 1 の同一のバックアップが4つ作成されます。

```
run {
    set duplex = 4;
    allocate channel dev1 type 'sbt_tape';
    backup datafile 1;
}
```

補助チャンネルの割当て 複製データベース（11-69 ページの「[duplicate](#)」を参照）を作成する場合には、**auxiliary** オプションを使用してチャンネルを割り当てます。

```
run {
    allocate auxiliary channel c1 type disk;
    allocate auxiliary channel c2 type disk;
    duplicate target database to ndbnewh
        logfile
            '/oracle/dbs/log_1.f' size 200K,
            '/oracle/dbs/log_2.f' size 200K
    skip readonly
    nofilenamecheck;
}
```

関連項目

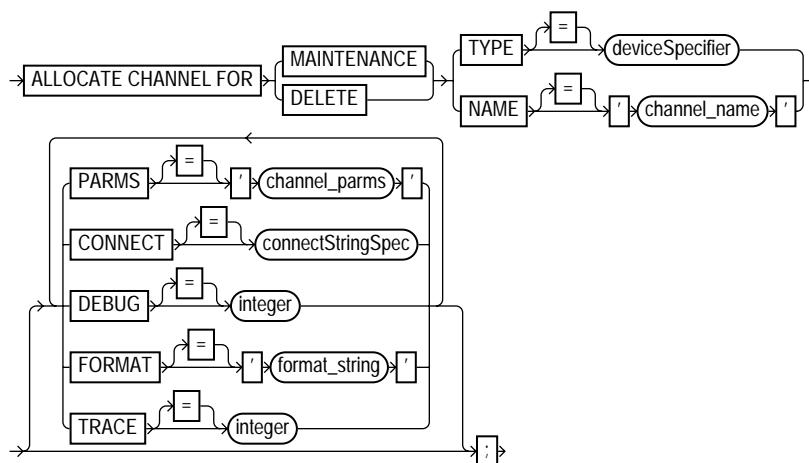
11-13 ページの「[allocateForMaint](#)」

11-69 ページの「[duplicate](#)」

11-39 ページの「[cmdLine](#)」

allocateForMaint

構文



目的

change または **crosscheck** コマンドを発行するための準備として、チャンネルを割り当てます。**maintenance** オプションおよび **delete** オプション（両者は意味的に全く同じです）を使用して、状態を変更しようとしているファイルに適したデバイス・タイプを指定します。

要件

- このコマンドは、RMAN プロンプトから実行してください。
- 次のコマンドを発行する前に、**allocate channel for delete** または **allocate channel for maintenance** コマンドを発行してください。
 - **change backupset ... delete**
 - **change backuppiece ... delete**
 - **change backupset ... crosscheck**
 - **change backuppiece ... crosscheck**
 - **crosscheck**
- チャンネル ID は指定しないでください。

- 新しいチャンネルを割り当てる前に、現在割り当てられているチャンネルを解放する必要があります。すなわち、複数のメンテナンス・チャンネルを割り当てることはできません。

キーワードおよびパラメータ

11-9 ページの「[allocate](#)」を参照してください。

例

バックアップ・ピースの削除 この例では、メディア管理カタログからバックアップ・ピースを削除します。

```
allocate channel for maintenance type 'sbt_tape';
change backuppiece '/oracle/dbs/01aj3q5012' delete;
release channel;
```

ファイルを使用不可にマーク この例では、主キーが 828 のバックアップ・セットをメディア管理カタログとのクロスチェックにより確認しています。

```
allocate channel for maintenance type disk;
change backupset 828 crosscheck;
release channel;
```

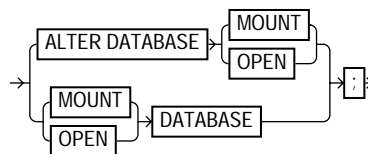
関連項目

11-9 ページの「[allocate](#)」

11-35 ページの「[change](#)」

alterDatabase

構文



目的

データベースをマウントまたはオープンします。

要件

- このコマンドは、**run** コマンドの中カッコ内か、または RMAN プロンプトで実行できます。
- Oracle インスタンスが起動されている場合に限り、このコマンドを実行できます。

キーワードおよびパラメータ

alter database	データベースをマウントまたはオープンします。
mount	データベースをマウントします。オープンはしません。
open	データベースをオープンします。
mount database	データベースをマウントします。オープンはしません。このオプションは、SQL 文 ALTER DATABASE MOUNT と同等です。 関連項目 : ALTER DATABASE の構文は、『Oracle8i SQL リファレンス』を参照してください。
open database	データベースをマウントし、かつオープンします。このオプションは、SQL 文 ALTER DATABASE OPEN と同等です。 関連項目 : ALTER DATABASE の構文に関しては、『Oracle8i SQL リファレンス』を参照してください。

例

バックアップ後のデータベースのオープン この例では、データベースをマウントし、全体データベース・バックアップを実行してからデータベースをオープンします。RMAN プロンプトで、次のように入力します。

```
startup mount;
run {
    allocate channel chl type disk;
    backup database;
    # now that the backup is complete, open the database.
    alter database open;
}
```

制御ファイル復元後のデータベースのマウント 制御ファイルをデフォルトの位置に復元し、データベースをマウントするには、次のように入力します。

```
startup nomount;
run {
    allocate channel chl type 'sbt_tape';
    restore controlfile;
}
# mount the database with the restored control file.
alter database mount;
```

関連項目

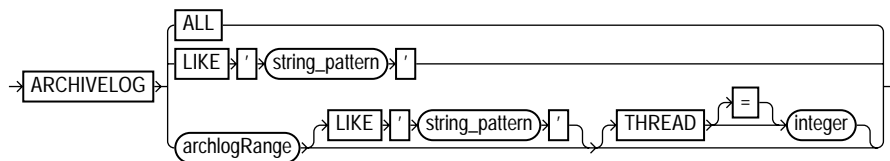
11-141 ページの「[sql](#)」

11-143 ページの「[startup](#)」

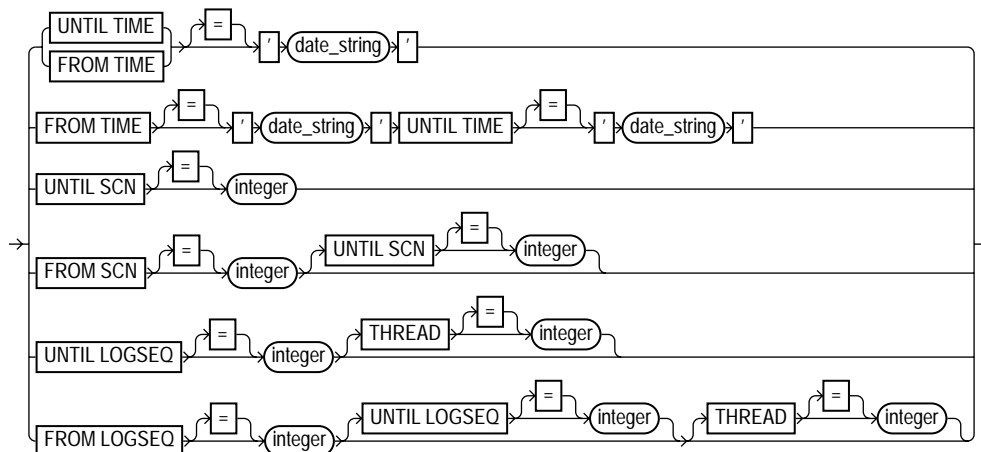
11-138 ページの「[shutdown](#)」

archivelogRecordSpecifier

構文



archlogRange::=



目的

バックアップ、復元およびメンテナンスの操作で使用するアーカイブ REDO ログ・ファイルの範囲を指定する句です。

要件

この句は、次のコマンドでのみ使用できます。

- **backup**
- **change**
- **crosscheck**

- [deleteExpired](#)
- [list](#)
- [restore](#)

アーカイブ REDO ログ・ファイルの範囲を指定した場合、RMAN がすべての REDO データをその範囲内に含めるという保証はありません。たとえば、最新のアーカイブ・ログ・ファイルの最終時点が範囲の最後より前の場合、あるいは範囲内のアーカイブ・ログ・ファイルが欠落している場合などです。RMAN は、検索したアーカイブ REDO ログは選択しますが、ファイルの欠落について警告はしません。

キーワードおよびパラメータ

アーカイブ・ログのタイムスタンプ、SCN およびログ順序番号を決めるには、VSARCHIVED_LOG データ・ディクショナリ・ビューを検索します。時刻の書式を指定する際に、NLS_LANG および NLS_DATE_FORMAT 環境変数を使用する方法は、『Oracle8i リファレンス・マニュアル』を参照してください。

all	各ログ順序番号ごとに必ず 1 つのアーカイブ REDO ログを含めるよう指定します。たとえば、 backup archivelog all を実行し、ログが複数の宛先にアーカイブされている場合でも、RMAN がバックアップするのは、各ログ順序番号の 1 つのコピーであって、各ログ順序番号の全てのアーカイブ・コピーではありません。
like 'string_pattern'	アーカイブ REDO ログ・ファイルのパス名を指定します。OPS モードに対して操作を行っている場合に、RMAN がアクセスするファイル・システムを指定するために、このパラメータを使用します。 関連項目 : Oracle Parallel Server 構成に関しては、『Oracle8i Parallel Server 概要および管理』を参照してください。
until time 'date_string'	一連のアーカイブ REDO ログ・ファイルの終了時刻を指定します。文字列による時刻指定は、現在有効な NLS 日付書式に従って行う必要があります。 from time パラメータを指定しない場合、順序番号に使用する開始時刻は、使用できる一番古いアーカイブ REDO ログの時刻になります。ログの最初のエン트리と最後のエントリのタイムスタンプを調べるには、VSARCHIVED_LOG データ・ディクショナリ・ビューに問い合わせます。 関連項目 : 時刻の書式を指定する NLS_LANG および NLS_DATE_FORMAT 環境変数の使用方法は、『Oracle8i リファレンス・マニュアル』を参照してください。
from time 'date_string'	一連のアーカイブ REDO ログ・ファイルの開始日を指定します。 until time パラメータを指定しない場合、RMAN は from time パラメータで指定した日付から始まるすべての使用可能なログ・ファイルを組み込みます。 ログの最初のエン트리と最後のエントリのタイムスタンプを調べるには、VSARCHIVED_LOG データ・ディクショナリ・ビューを使用します。 関連項目 : 時刻の書式を指定する NLS_LANG および NLS_DATE_FORMAT 環境変数の使用方法は、『Oracle8i リファレンス・マニュアル』を参照してください。

until SCN <i>integer</i>	一連のアーカイブ REDO ログ・ファイルに対して終了 SCN を指定します。 from SCN パラメータを指定しない場合、RMAN は一番古い SCN を使用して順序を開始します。
from SCN <i>integer</i>	一連のアーカイブ REDO ログ・ファイルに対する開始 SCN を指定します。 until SCN パラメータを指定しない場合、RMAN は from SCN パラメータで指定した SCN から始まるすべての使用可能なログ・ファイルを組み込みます。
until logseq <i>integer</i>	一連のアーカイブ REDO ログ・ファイルの終了ログ順序番号を指定します。 from logseq パラメータを指定しない場合、RMAN は一番低いログ順序番号を使用して順序を開始します。
from logseq <i>integer</i>	一連のアーカイブ REDO ログ・ファイルの開始ログ順序番号を指定します。 until logseq パラメータを指定しない場合、RMAN は from logseq パラメータで指定した日付から始まるすべての使用可能なログ・ファイルを組み込みます。
thread <i>integer</i>	選択する必要があるアーカイブ REDO ログ・ファイルが含まれているスレッドを指定します。データベースを OPS モードで使用する場合に限り、このパラメータの指定が必要です。 アーカイブ REDO ログ・レコードのスレッド番号を調べるには、V\$ARCHIVED_LOG データ・ディクショナリ・ビューを使用します。

例

時刻によるレコードの指定 この例では、2 週間よりも古いアーカイブ REDO ログのバックアップをすべて削除します。

```
change archivelog until time 'SYSDATE-14' delete;
```

SCN によるレコードの指定 この例では、SCN の範囲内に入るバックアップ・アーカイブ REDO ログ・ファイルをテープから復元します。

```
run {
    allocate channel dev1 type 'sbt_tape';
    restore archivelog
        from SCN 500 until SCN 700;
    release channel dev1;
}
```

ログ順序番号によるレコードの指定 この例では、スレッド 1 の順序番号 288 から 301 までのすべてのアーカイブ・ログをバックアップし、そ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は、ログを削除しません。

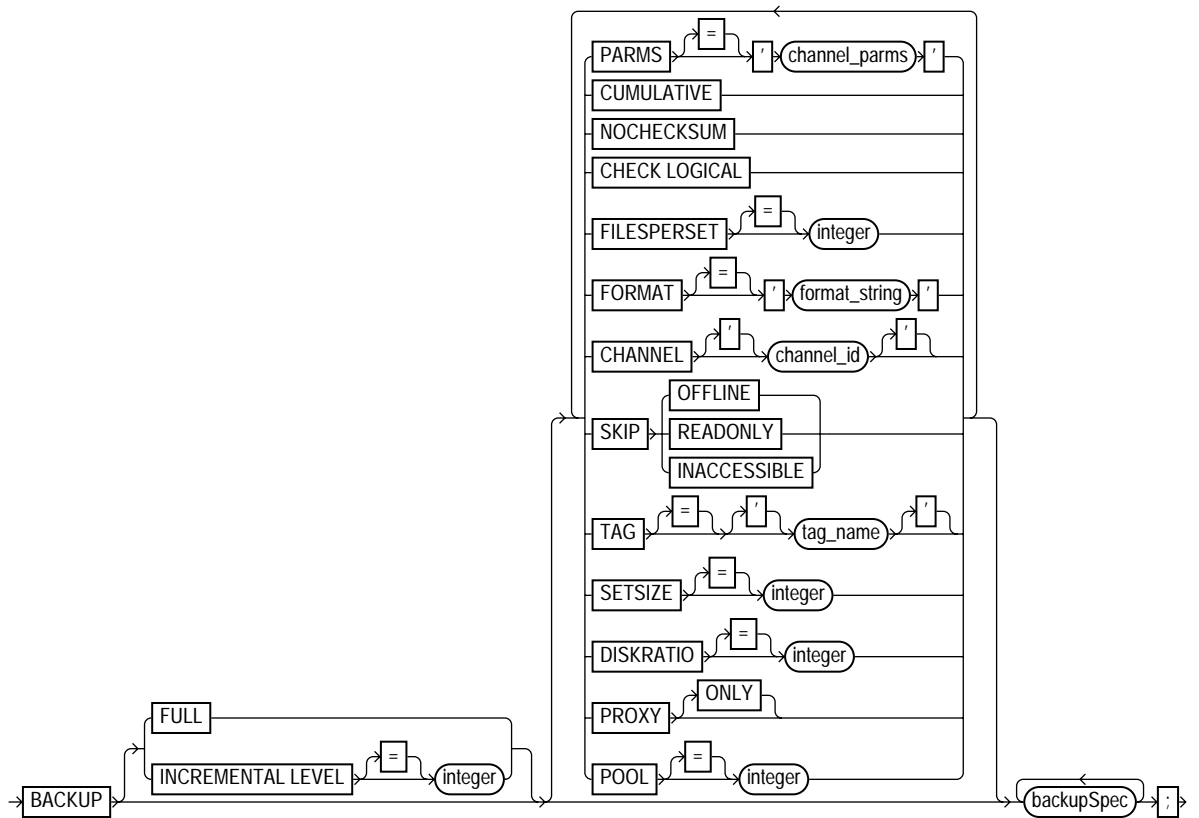
```
run {
    allocate channel dev1 type 'sbt_tape';
    backup archivelog
        from logseq 288 until logseq 301 thread 1
        # delete original archived redo logs after backup completes
        delete input;
}
```

関連項目

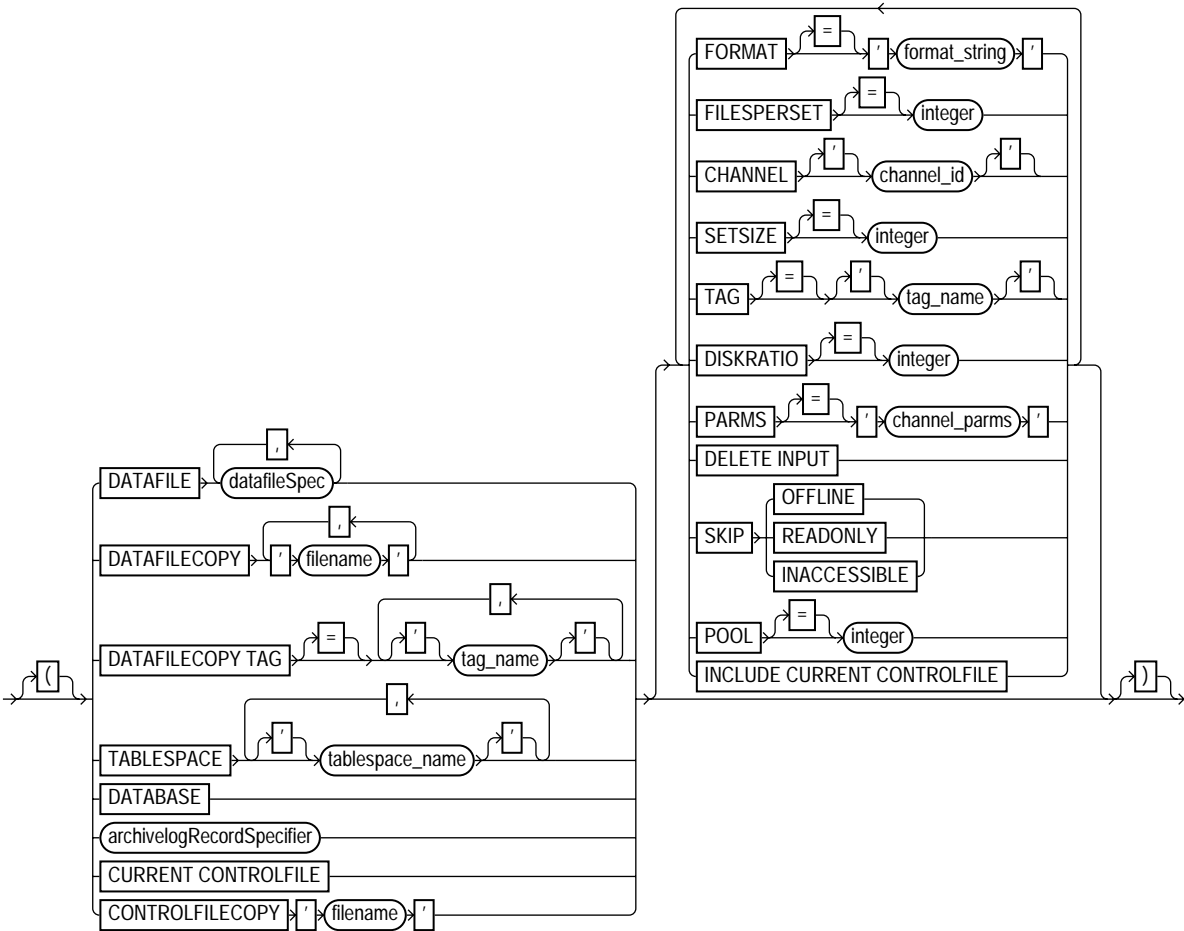
- 11-21 ページの「[backup](#)」
- 11-35 ページの「[change](#)」
- 11-57 ページの「[crosscheck](#)」
- 11-63 ページの「[deleteExpired](#)」
- 11-76 ページの「[list](#)」
- 11-113 ページの「[restore](#)」

backup

構文



backupSpec ::=



目的

データベース、表領域、データ・ファイル、制御ファイルまたはアーカイブ REDO ログ・ファイルのバックアップを実行します。バックアップを実行する場合、バックアップが必要なファイルを指定します。RMAN は入力ファイルをバックアップ・セットに書き込みます。バックアップ・セットは RMAN 固有の論理構造になっています。各バックアップ・セットには少なくとも 1 つのバックアップ・ピースが含まれています。プロキシ・コピーの作成にも、**backup** コマンドが使用されます。プロキシ・コピーとはメディア・マネージャによって作成されたバックアップのことです。

Oracle が作成するバックアップ・セットの数および RMAN が 1 つのバックアップ・セットに入れる入力ファイルの数は、ユーザーが制御します。ファイルの読取り中またはバック

アップ・ピースの書込み中に I/O エラーが発生した場合、Oracle はジョブを異常終了させます。

関連項目 : ファイルのバックアップ方法は、[第 8 章の「Recovery Manager でのバックアップおよびコピーの作成」](#)を参照してください。

要件

backup コマンドを使用する場合、次の操作が必要です。

- データベースをマウントまたはオープンする。RMAN では、データベースが ARCHIVELOG モードになっている場合、[非一貫性バックアップ \(inconsistent backup\)](#) が実行できます。ただし、復元操作に使用するバックアップに一貫性をもたせるため、REDO ログを適用する必要があります。
- 現在の制御ファイルを使用する。
- **backup** コマンドの実行は、**run** コマンドの中カッコ内で行う。
- **backup** コマンドの実行ごとにチャンネルを割り当てる。
- 各バックアップ・ピースに一意の名前を付ける。
- 有効なメディアにバックアップする。**type disk** を指定した場合、ディスクにバックアップすることが必要です。バックアップを作成することができるのは、Oracle データ・ファイルを格納できるデバイスです。すなわち、CREATE TABLESPACE *tablespace_name* DATAFILE '*filename*' の文が有効であれば、'*filename*' は有効なバックアップ・パス名です。**type 'sbt_tape'** を指定した場合、メディア管理ソフトウェアがサポートするメディアであれば、どのメディアにもバックアップできます。

また、以下のことは実行できません。

- NOARCHIVELOG モードでオープン・バックアップを実行する。
- 1 つのバックアップを複数のチャンネルに分散する。
- 1 つの入力ファイルを複数のバックアップに分散する。
- アーカイブ REDO ログ・ファイルとデータ・ファイルを 1 つのバックアップに含める。
- **backup** コマンドに、チャンネルを **allocate** した後で、**set duplex** コマンド (11-134 ページの「[set_run_option](#)」を参照) を実行する。**set duplex** コマンドは、チャンネルを割り当てる前に実行する必要があります。そうしない場合エラーになります。
- バックアップ・セットに入れるバックアップ・ピースの数を指定する。
- 100 以上のバックアップ・ピースを入れたバックアップ・セットを作成する。
- 各バックアップの一意なタグ名を自動的に作成する。一意なタグ名をそのつど作成するには、O/S ユーティリティで、バックアップ・スクリプトを作り、その実行前に編集することが必要です。

キーワードおよびパラメータ

full	<p>すべてのブロックをバックアップ・セットにコピーします。これまでに使用されたことがないデータ・ファイル・ブロックはスキップされます。full か incremental の指定がない場合、RMAN はデフォルトにより全体バックアップを実行します。アーカイブ REDO ログまたは制御ファイルをバックアップする場合、サーバー・セッションはブロックをスキップしません。</p> <p>全体バックアップはその後の増分バックアップに何の影響も与えません。したがって、全体バックアップは増分バックアップ計画の一部分とはみなされません。</p>
incremental level <i>n</i> <i>integer</i>	<p>最後のレベル <i>n</i> 増分バックアップ以降に変更されたブロックのみをコピーします。この場合の <i>n</i> は、1 から 4 までの整数です。たとえば、レベル 2 増分バックアップでは、RMAN は最近作成したレベル 2、レベル 1、レベル 0 増分バックアップ以降に使用されたすべてのブロックをバックアップします。</p> <p>このタイプの増分バックアップは、累積バックアップと区別するために、差分バックアップと呼ばれます。レベル 0 の増分バックアップは内容的には全体バックアップと同じですが、全体バックアップと違って、増分バックアップ計画の一部分とみなされます。</p> <p>Oracle システムは、レベルが 0 より大きい増分バックアップの作成を試みると、チェックを実行します。このチェックによって、増分バックアップがその後の recover コマンドで使用できることを確認します。チェックの内容は次のとおりです。</p> <ul style="list-style-type: none"> ■ backup コマンドに指定されているデータ・ファイルには、レベル 0 バックアップ、またはレベル 0 データ・ファイル・コピーが存在すること。これらが unavailable (11-35 ページの「change」を参照) にマークされていないこと。 ■ レベル 0 以降に増分バックアップが作成されている場合、これから作成する増分バックアップが必要とするバックアップを使用できること。 <p>incremental を指定した場合、<i>backupSpec</i> で datafile、datafilecopy、tablespace または database のいずれかを指定します。RMAN は、制御ファイル、アーカイブ REDO ログまたはバックアップ・セットの増分バックアップをサポートしません。</p>
parms ' <i>channel_parms</i> '	<p>O/S 固有の情報を含む引用符付き文字列を指定します。RMAN は、バックアップ・ピースを作成するたびに、OSD レイヤーに文字列を渡します。</p>
cumulative	<p><i>n</i>-1 以下のレベルの最新バックアップ以降に使用されたことがあるデータ・ブロックをコピーします。たとえば、レベル 2 の累積バックアップでは、RMAN は直近のレベル 1 または 0 以降に使用されたすべてのブロックをバックアップします。</p>

nochecksum	<p>ブロックに対するチェックサムを抑止します。<i>checksum</i> とは、データ・ブロックの内容によって計算した数字のことです。DB_BLOCK_CHECKSUM 初期化パラメータが TRUE の場合、Oracle は各ブロックのチェックサムを計算し、ブロックをディスクに書き込む前に、ブロック内に値を格納します。その後 Oracle ではディスクからブロックを読み出すとき、ブロックが同じチェックサムを生成するかどうかを確認します。同じチェックサムが生成されない場合は、ブロックは破損しています。</p> <p>nochecksum オプションを指定しなければ、各ブロックのチェックサムが計算され、値がバックアップに格納されます。データ・ファイルをバックアップから復元するときは、チェックサムを確認します。また復元されるデータ・ファイルにもチェックサムを書きこみます。データベースがすでにブロック・チェックサムをメンテナンスしている場合は、このフラグは何の効果もありません。この場合、常にチェックサムが確認されバックアップに格納されます。</p> <p>関連項目 : DB_BLOCK_CHECKSUM 初期化パラメータの詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。</p>
filesperset integer	<p>1 バックアップ・セットに含める入力ファイルの最大数を指定します。</p> <p>RMAN は、バックアップが必要なファイルの数を割り当てられたチャネルの数で割り、各バックアップ・セットに含めるファイル数を決めます。filesperset パラメータを指定すると、RMAN は filesperset の値を前述の計算値と比較して低い方の値を使用します。その結果、すべてのチャネルが確実に使用されることになります。<i>backupSpec</i> 句で指定または自動的に選択されたファイル数が filesperset の値よりも大きい場合、たとえば filesperset = 4 でバックアップが必要なファイル数が 8 の場合、RMAN は複数のバックアップ・セットを作成して、バックアップ・セット当たりのファイル数の適切な比率をメンテナンスします。</p> <p>filesperset を指定しない場合、RMAN は計算した値（ファイル数 / 割り当てたチャネル数）をデフォルト値の 64 と比較して低い方の値をとり、すべてのチャネルを確実に使用します。デフォルト値 64 はほとんどのアプリケーションで高い値になります。バックアップ・セットのサイズを制限するには、小さい値にするか、あるいは setsize パラメータを使用します。</p> <p>RMAN は、割り当てられたすべてのチャネルを使用することができるよう、十分なバックアップ・セットを設定するよう常に試みます。バックアップするファイルの数よりも多くのチャネルがある場合は、このルールの例外となります。たとえば、3 つのチャネルが割り当てられ、filesperset = 1 のときに、RMAN が 1 つのデータ・ファイルをバックアップする場合、2 つのチャネルが必然的にアイドル状態になります。</p> <p>注意 : setsize パラメータの方が filesperset よりも使用しやすい場合があります。具体的には、アーカイブ・ログ・バックアップを行う場合およびデータ・ファイルがストライピングされているか、個別のディスクに分散している場合です。両方のパラメータを指定した場合、両方とも有効です。RMAN は、filesperset を上限として、setsize パラメータによってバックアップ・セットのサイズを決めます。</p>

format
'format_string'

バックアップ・ピースに使用するファイル名を指定します。各バックアップ・ピースが一意的な名前であり、かつプラットフォーム上の順次ファイル名として有効であればどのような名前でも指定できます。ディスクにバックアップする場合、一意の有効なファイル名であれば、どのようなディスク・ファイル名も使用できます。**format** パラメータを指定しない場合、RMAN はバックアップ・ピースをポート固有のディレクトリ (UNIX の \$ORACLE_HOME/dbs) に格納します。

format パラメータは、次に示す場所で指定します。

- **backupSpec** 句
- **backup** コマンド
- **allocate channel** コマンド

前述の場所のうち、2 箇所以上に指定した場合、RMAN は前述の順序で、**format** パラメータを検索します。

一意のファイル名を生成するために、次の置換変数をフォーマット文字列で使うことができます。

%c	1 セットの二重バックアップ・ピースのなかにあるバックアップ・ピースのコピー番号を指定します。 set duplex コマンドを発行していない場合、この変数は正規バックアップ・セットに対しては 1、プロキシ・コピーに対しては 0 になります。 set duplex が発行済みの場合、変数はコピー番号を示します。1、2、3 または 4。
%p	バックアップ・セットのなかのバックアップ・ピース番号を指定します。この値は、バックアップ・セットごとに 1 から始まり、バックアップ・ピースが作成されるごとに 1 ずつ増えます。
%s	バックアップ・セット番号を指定します。この番号は制御ファイルにあるカウンタで、バックアップ・セットごとに加算されます。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になっています。バックアップ制御ファイルを復元した場合には、重複した値になることがあります。また、CREATE CONTROLFILE によって、カウンタの値は 1 に初期化されます。
%d	データベース名を指定します。
%n	データベース名を指定します。データベース名は合計 8 文字の桁数になるまで、右側に 'x' が埋め込まれます。たとえば、PROD1 というデータベース名の場合、PROD1xxx がデータベース名です。
%t	バックアップ・セットのタイムスタンプを指定します。4 バイトの内部的な基準日時以降の経過秒数を表す値です。バックアップ・セットの一意の名前を作るには、%s と %t の組み合わせを使用できます。
%u	バックアップ・セット番号とバックアップ・セット作成時刻の圧縮形で表わされる 8 文字の名前を指定します。
%U	生成されたバックアップ・ファイル名の一意性を保証できる %u_%p_%c の各置換変数を組み合わせた値を指定します。書式の指定がない場合、RMAN はデフォルトにより %U を使用します。

channel <i>channel_id</i>	<p>バックアップ・セットを作成するときに使用するチャンネルの名前を指定します。任意の名前を使用できます。たとえば、<i>ch1</i> あるいは <i>dev1</i> です。Oracle は、release channel コマンドの実行および I/O エラーの報告に、このチャンネル ID を使用します。このパラメータを指定しないと、RMAN はジョブ実行中に使用できるチャンネルに動的にバックアップ・セットを割り当てます。</p> <p>注意: <i>backupSpec</i> 句でもこのパラメータを指定できます。</p>
skip	<p>バックアップ・セットに含めないデータ・ファイルまたはアーカイブ REDO ログを指定します。</p> <p>注意: <i>backupSpec</i> 句でもこのオプションを指定できます。</p>
offline	<p>オフライン・データ・ファイルをバックアップ・セットに含めないことを指定します。</p>
readonly	<p>読取り専用データ・ファイルをバックアップ・セットに含めないことを指定します。</p>
inaccessible	<p>I/O エラーのために読み込むことができないデータ・ファイルまたはアーカイブ REDO ログをバックアップ・セットに含めないことを指定します。</p> <p>データ・ファイルは読み込めない場合に限り、アクセス不可能とみなされません。オフライン・データ・ファイルでも、ディスクに存在している間は読取り可能なものもあります。削除または移動したために読取りができなくなり、アクセス不可能となるオフライン・データ・ファイルもあります。</p>
tag tag_name	<p>バックアップ・セットにタグ名を指定します。一般的に、タグは <i>monday_evening_backup</i> または <i>weekly_full_backup</i> など意味のある名前にします。タグは 30 文字以下にしてください。タグは再使用できます。たとえば、バックアップ・セット 100 がある週に <i>monday_evening_backup</i> のタグを使用し、翌週にはバックアップ・セット 105 が同じタグを使用することができます。</p> <p><i>backupSpec</i> レベルでもタグを指定できます。どのレベルでタグを指定するかによって、次のようになります。</p> <ul style="list-style-type: none"> ■ コマンド・レベルで指定した場合は、このコマンドによって作成さ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。 ■ <i>backupSpec</i> レベルで指定した場合は、異なるバックアップ指定で作成されたバックアップ・セットには、それぞれ異なるタグが与えられます。 ■ 両方のレベルに指定すると、<i>backupSpec</i> のタグが優先します。 <p>注意: 各バックアップに異なるタグ名を自動的に割り当てることはできません。各バックアップに新しいタグを付ける最も簡単な方法は、バックアップ・スクリプトを作成し、実行の前に常に O/S ユーティリティで編集することです。</p>

setsize <i>integer</i>	<p>バックアップ・セットの最大サイズを 1K (1024 バイト) 単位で指定します。バックアップ・セットを 3MB に制限するには、setsize = 3000 と指定します。RMAN 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。これは、メディア管理ソフトウェアを使用する場合にどのバックアップ・セットも 1 つのテープに収まるよう設定する場合に役に立つ方法です。</p> <p>アーカイブ REDO ログ・バックアップを行うとき、setsize パラメータは filesperset よりも使用しやすい手段です。バックアップ・セットを構成するときは、複数のテープ・ボリュームにまたがらないで 1 テープ・ボリュームに合うようにしてください。そうしないと、複数ボリュームのバックアップ・セットの 1 テープにエラーが生じた場合、すべてのテープのデータを失うことになります。</p> <p>注意 : setsize と filesperset の両方を指定した場合、両方とも有効です。RMAN は、filesperset を上限とみなした上で、setsize パラメータによってバックアップ・セットのサイズを決めます。</p>
diskratio <i>integer</i>	<p>指定した数のドライブに分散したバックアップ・セットへのデータ・ファイル (かつデータ・ファイルのみ) の割当てを RMAN に指示します。たとえば、システムで 10 台のディスクを使用しており、ディスクは 1 秒に 10 バイトのデータを送る能力があり、テープ・ドライブではストリーミングの維持のため 1 秒に 50 バイトのデータを要求している場合を想定します。この場合、diskratio を 5 に設定して、バックアップの負荷を 5 台のディスクに分散します。</p> <p>データ・ファイルがストライプされているが別個のディスクにあり、かつ次のどちらかの要件をもつ場合にも、diskratio パラメータは有用です。</p> <ul style="list-style-type: none"> ■ 高帯域のテープ・ドライブを使用しており、テープ・ドライブのストリーミングを維持するには、複数のデータ・ファイルの多重化が必要である。 ■ データベースがオープン中にバックアップを行うが、オンライン操作に帯域幅を確保しておくために、複数のディスクに I/O 負荷を分散する必要がある。 <p>バックアップ・パフォーマンスの調整のために filesperset を指定し、diskratio は指定しない場合、diskratio のデフォルトは filesperset と同じ値になります。どちらの指定もない場合、diskratio のデフォルトは 4 になります。</p> <p>注意 : テープ・ストリーミングを維持するのに必要な最少ディスク台数以上に I/O を分散しないでください。それ以上に分散しても、パフォーマンスは改善しないばかりでなく、単一ファイルの復元時間も増大します。</p>
proxy	<p>プロキシ・コピー機能を使用して、指定したファイルをバックアップします。これにより、メディア管理ソフトウェアは記憶デバイスとディスク上の Oracle データ・ファイルとの間のデータ転送を制御することになります。(RMAN ではなく) メディア・マネージャがデータ移動の方法と時期を決めます。</p> <p>プロキシ・コピーを行うファイルごとに、RMAN はメディア・マネージャに問い合せて、コピーができるか判断します。メディア・マネージャがプロキシ・コピーを行えない場合、RMAN は従来のバックアップ・セットを使用してファイルをバックアップします。</p> <p>only プロキシ・コピーができない場合、従来のバックアップ・セットを作るのではなく、Oracle にエラー・メッセージを発行させます。</p>
pool <i>integer</i>	<p>バックアップを格納するメディア・プールを指定します。メディア管理ソフトウェアのマニュアルを調べて、pool オプションがサポートされているか確認してください。</p>

backupSpec	<p><i>backup_specification_list</i> には、1 つ以上の <i>backupSpec</i> 句のリストが入っています。<i>backupSpec</i> 句には、最低限 1 つ以上の、バックアップするオブジェクト (1 つ以上) のリスト、<i>backup_object_list</i> が含まれます。</p> <p>各 <i>backupSpec</i> 句は、1 つ以上のバックアップ・セットを生成します。<i>backup_object_list</i> に指定または自動的に選択されたデータ・ファイルの数が filesperset の制限を超えている場合は、<i>backupSpec</i> 句で複数のバックアップ・セットが作成されます。</p>
datafile <i>datafileSpec</i>	<p>1 つ以上のデータ・ファイル (11-60 ページの「<i>datafileSpec</i>」を参照) のリストを指定します。</p> <p>注意: 表領域 SYSTEM の第 1 ファイル <i>datafile 1</i> をバックアップする場合、RMAN はバックアップ・セットに制御ファイルを自動的に組み込みます。</p>
datafile copy <i>'filename'</i>	<p>1 つ以上のデータ・ファイル・イメージ・コピーのファイル名を指定します。</p>
datafile copy tag <i>tag_name</i>	<p>タグで識別された、1 つ以上のデータ・ファイル・コピーのリストを指定します。このタグの付いたデータ・ファイル・コピーが複数存在する場合、Oracle は特定のデータ・ファイルの最新のデータ・ファイルのみをバックアップします。</p>
tablespace <i>tablespace_name</i>	<p>1 つ以上の表領域の名前を指定します。RMAN は、表領域の一部になっているデータ・ファイルをすべてバックアップします。</p> <p>Oracle は表領域名を内部的にデータ・ファイルのリストに変換します。</p>
database	<p>制御ファイルとデータベースのすべてのデータ・ファイルを指定します。Oracle は表領域名を内部的にデータ・ファイルのリストに変換します。</p>
<i>archivelogRecord-Specifier</i> 句	<p>アーカイブ REDO ログの範囲を指定します。11-17 ページの「<i>archivelogRecordSpecifier</i>」を参照してください。</p>
current controlfile	<p>現行の制御ファイルを指定します。</p>
controlfile copy <i>'filename'</i>	<p>制御ファイル・コピーのファイル名を指定します。</p>
parms <i>'channel_parms'</i>	<p>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、この固有の文字列は使用しないでください。</p> <p>parms を type 'sbt_tape' と一緒に使用すれば、デバイスに対する環境変数を指定できます。有効な構文のモデルを次に示します。</p> <pre>PARMS="ENV=(var1=value1,var2=value2,var3=value3 . . .)" PARMS="BLKSIZE=integer"</pre> <p>たとえば、次のように指定できます。</p> <pre>PARMS="BLKSIZE=16384,ENV=(NSR_SERVER=tape_server,NSR_CLIENT=oracleclnt,NSR_GROUP=oracle_tapes)"</pre> <p>引用符付き文字列の最大長は 1000 バイトです。</p>

format <i>'format_string'</i>	バックアップ・ピースに使用するファイル名を指定します。コマンド・レベルの format パラメータの説明を参照してください。
filesperset <i>integer</i>	1 つのバックアップ・セットに含めるデータ・ファイルの最大数を指定します。コマンド・レベルの filesperset の説明を参照してください。
channel <i>channel_id</i>	<i>backupSpec</i> 句のバックアップ・セットを作成するときに使用するチャンネルの名前を指定します。コマンド・レベルの channel の説明を参照してください。
setsize <i>integer</i>	バックアップ・セットの最大サイズを 1K (1024 バイト) 単位で指定します。コマンド・レベルの setsize パラメータの説明を参照してください。
tag <i>tag_name</i>	バックアップ・セットに対してタグを作成します。コマンド・レベルの tag パラメータの説明を参照してください。
diskratio <i>integer</i>	バックアップに関連するディスク台数を指定します。コマンド・レベルの diskratio パラメータの説明を参照してください。
delete <i>input</i>	バックアップ・セットが正しく作成された時点で入力ファイルを削除します。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のみです。すべての入力ファイルに対して change ... delete コマンドを発行するのと同じです。 注意: backup コマンドは各ログ順序番号について 1 つのコピーのみをバックアップします。したがって、 delete input オプションが指定されても、RMAN はバックアップしたファイルのコピーのみを削除します。
skip	offline 、 readonly または inaccessible のデータ・ファイルをスキップします。コマンド・レベルの skip オプションの説明を参照してください。
pool	バックアップを格納するメディア・プールを指定します。コマンド・レベルの pool の説明を参照してください。
include current controlfile	現行の制御ファイルのスナップショットを作成し、作成するバックアップ・セットに含めます。

例

表領域およびデータ・ファイルのバックアップ このコマンドは、表領域とデータ・ファイルをバックアップするために、2 つの *backupSpec* 句を使用します。

```
run {
  allocate channel dev1 type disk;
  allocate channel dev2 type disk;
  backup
    (tablespace system,sales1,sales2,sales3
     filesperset 20
     skip readonly
     channel dev1)
```

```
(datafile 1, 4, 5
channel dev2);
}
```

データベースの累積増分バックアップの実行 この例では、直近のレベル 0 またはレベル 1 のバックアップ以降にデータベース上で変更された、すべてのブロックをバックアップします。

```
run {
  allocate channel dev1 type 'sbt_tape';
  backup
    incremental level 2 cumulative
    # do not include inaccessible datafiles in the backup
    skip inaccessible
    database;
}
```

バックアップ・セットを二重化 バックアップ・セットを二重化する場合、チャンネルを割り当てる前に、**set duplex** コマンドを指定します。

```
run {
  # generate four identical backup sets of datafile 1
  set duplex=4;
  allocate channel dev1 type 'sbt_tape';
  backup datafile 1;
}
```

破損のチェック この例では、datafile 3 をバックアップするが、物理的な破損または論理的な破損のあるブロックが 2 つ以上あれば許容しないことを指定しています。

```
run {
  set maxcorrupt for datafile 3 to 2;
  allocate channel dev1 type 'sbt_tape';
  backup check logical
    datafile 3;
}
```

関連項目

11-9 ページの「[allocate](#)」

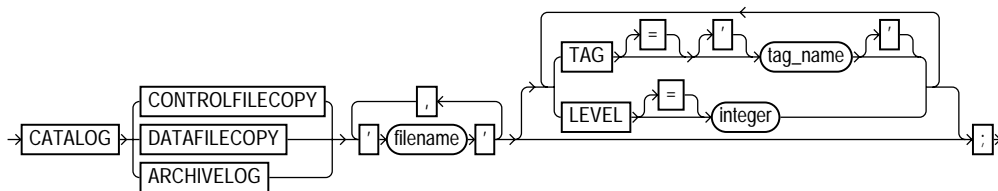
11-17 ページの「[archivelogRecordSpecifier](#)」

11-87 ページの「[printScript](#)」

11-134 ページの「[set_run_option](#)」

catalog

構文



目的

次の目的に **Catalog** コマンドを使用します。

- リカバリ・カタログおよび制御ファイルに、O/S データ・ファイル・コピー、アーカイブ REDO ログまたは制御ファイルのコピーに関する情報を追加する。
- データ・ファイル・コピーをレベル 0 バックアップとしてカタログに追加する。これによって、コピーを増分バックアップの一部として使用することができます。
- RMAN のインストール以前に作成した Oracle8 のデータベースのバックアップをカタログに追加する。
- Oracle8 または Oracle8i への移行以前に作成した読取り専用または NORMAL モードでオフラインにされたファイルの Oracle7 バックアップをカタログに追加する。

要件

- **catalog** コマンドは、RMAN プロンプトのみで実行してください。
- RMAN をリカバリ・カタログと一緒に使用してください。
- O/S バックアップのカタログへの追加には、次の条件が必要です。
 - ディスク上でアクセスできること。
 - 単一ファイルの完全イメージ・コピーであること。
 - データベース全体、表領域、データ・ファイル、制御ファイルまたはアーカイブ REDO ログの一貫性がある、または一貫性のないバックアップであること。一貫性がない場合、BEGIN BACKUP および END BACKUP 文を使用して作成されている必要があります。制御ファイルのバックアップの場合、ALTER DATABASE BACKUP CONTROLFILE 文を使用して作成されている必要があります。

RMAN は、このような O/S バックアップをすべてデータ・ファイルのコピーとみなします。

次の操作の実行に **catalog** を使用することはできません。

- Oracle7 で作成したアーカイブ REDO ログおよび制御ファイルのコピーのカatalogへの追加。ただし、これらのファイルが Oracle バージョン 8 以降に移行したときに NORMAL モードでオフラインにされたまたは読取り専用であった表領域に属するものは、Catalogに追加できます。
- バックアップ・ピースまたはバックアップ・セットを再度Catalogに追加すること。

キーワードおよびパラメータ

controlfilecopy 'filename'	リカバリ・Catalogおよび制御ファイルへの追加またはこれらの更新が必要な制御ファイル・コピーのファイル名を指定します。
datafilecopy 'filename'	リカバリ・Catalogおよび制御ファイルへの追加またはこれらの更新が必要なデータ・ファイル・コピーのファイル名を指定します。
archivelog 'filename'	リカバリ・Catalogおよび制御ファイルへの追加またはこれらの更新が必要なアーカイブ・ログ・コピーのファイル名を指定します。
tag tag_name	入力ファイルのタグ、たとえば <i>Sunday_PM_Backup</i> を指定します。
level integer	ファイル・コピーを指定レベル、(一般的にはレベル 0) の増分バックアップとして記録することを指示します。増分バックアップはデータ・ファイル・コピーをベースとなるレベル 0 バックアップとして使用できます。

例

アーカイブ REDO ログをCatalogに追加 この文では、アーカイブ REDO ログ log1、log2 および log3 をCatalogします。

```
catalog archivelog 'log1', 'log2', 'log3';
```

ファイル・コピーを増分バックアップとしてCatalogに追加 次の例では、データ・ファイル・コピー tbs_2.c を増分レベル 0 バックアップとしてCatalogします。

```
catalog datafile '/oracle/copy/tbs_2.c' level 0;
```

O/S コピーをCatalogに追加 次の例では、RMANhost コマンドを使用してデータ・ファイルの O/S コピーを作り、次にそのコピーをCatalogします (サンプル出力も示します)。

```
host 'cp $ORACLE_HOME/dbs/sales.f $ORACLE_HOME/dbs/sales.bak';
catalog datafilecopy '$ORACLE_HOME/dbs/sales.bak';
```

```
RMAN-03022: compiling command: catalog
```

```
RMAN-03023: executing command: catalog
RMAN-08050: cataloged datafile copy
RMAN-08513: datafile copy filename=/oracle/dbs/sales.bak recid=121 stamp=342972501
RMAN-03023: executing command: partial resync
RMAN-08003: starting partial resync of recovery catalog
RMAN-08005: partial resync complete
```

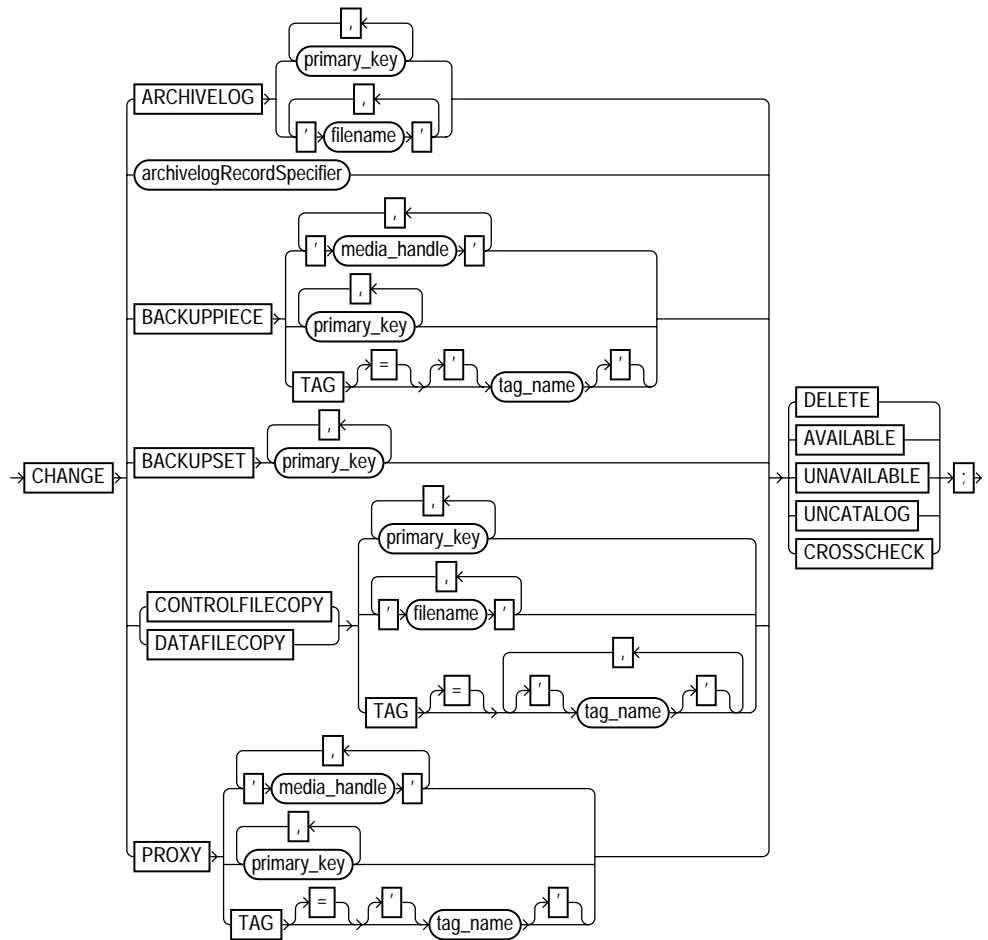
関連項目

11-76 ページの「[list](#)」

11-103 ページの「[report](#)」

change

構文



目的

RMAN メタデータにおけるバックアップとコピーの状態を変更すること。このコマンドは次の目的に使用します。

- バックアップまたはコピーを **unavailable** または **available** の状態にマークすること。
- バックアップまたはコピーを O/S から削除し、その状態を **deleted** に更新すること。
- バックアップ、イメージ・コピーおよびアーカイブ REDO ログが使用可能であるかチェックして、使用可能でない場合は、**expired** にマークします。

要件

- **change** コマンドは、RMAN プロンプトに対して、または **run** コマンドの中で実行する。
- RMAN メタデータに記録されており、かつ現行データベースのインカーネーションに属するファイルにのみ、このコマンドを使用する。
- 次のオプションには、リカバリ・カタログが必要です。
 - **available**
 - **unavailable**
 - **uncatalog**
 - **crosscheck** (バックアップ・セットとバックアップ・ピースにのみ必要)
- 次のコマンドを発行するには、その前に、**allocate channel for delete** または **allocate channel for maintenance** コマンドを発行する必要があります。
 - **change backupset ... delete**
 - **change backuppiece ... delete**
 - **change backupset ... crosscheck**
 - **change backuppiece ... crosscheck**

キーワードおよびパラメータ

状態変更が必要なレコードの主キーを取得するには、**list** コマンドを発行するか、あるいはリカバリ・カタログ・ビューを検索します。

archivelog	<i>primary_key</i> または ' <i>filename</i> ' のどちらかで、REDO ログを指定します。
archivelogRecord-Specifier 句	アーカイブ REDO ログの範囲を指定します。11-17 ページの「 archivelogRecordSpecifier 」を参照してください。
backuppiece	<i>primary_key</i> 、' <i>media_handle</i> '、または <i>tag_name</i> により、バックアップ・ピースを指定します。
backupset <i>primary_key</i>	<i>primary_key</i> によりバックアップ・セットを指定します。

controlfilecopy	<i>primary_key</i> 、' <i>filename</i> '、または <i>tag_name</i> により、制御ファイル・コピーを指定します。制御ファイル・コピーのクロスチェックを行う場合、主キーではなくファイル名を指定する必要があります。
datafilecopy	<i>primary_key</i> 、' <i>filename</i> '、または <i>tag_name</i> により、データ・ファイル・コピーを指定します。
proxy	<i>primary_key</i> 、' <i>filename</i> '、または <i>tag_name</i> により、プロキシ・コピーを指定します。
delete	バックアップまたはコピーの状態を deleted に変更し、ファイルを O/S から物理的に削除します。
available	バックアップまたはコピーを available の状態にマークし、状態を list 出力に表示します。
unavailable	バックアップまたはコピーを unavailable の状態にマークし、状態を list 出力に表示します。このオプションは、ファイルが検索できないか、またはオフサイトに移行した場合に使用します。 unavailable にマークされたファイルは、 restore コマンドや recover コマンドでは使用できません。後でファイルが検索された場合、またはメイン・サイトに戻された場合は、 available オプションを使用して、この変更を反映します。
uncatalog	<p>データ・ファイル・コピーまたは REDO ログ（バックアップ・ピースおよびバックアップ・セットは除く）への参照をリカバリ・カタログから除去します。ファイルが change ... delete コマンド以外の手段で削除されたときに、このコマンドを使用して RMAN に通知します。バックアップ・ピースまたはバックアップ・セットに対して uncatalog オプションを使用すると、RMAN はエラー・メッセージを戻します。</p> <p>警告: バックアップ制御ファイルから再同期化すると、カタログから除去したレコードは再びメタデータに登録されます。</p> <p>注意: deleted の状態のレコードをただちに除去する必要がある場合、<code>\$ORACLE_HOME/admin</code> ディレクトリにある <code>prgrmanc.sql</code> スクリプトを実行します。6-13 ページの「バックアップおよびコピーの削除と、RMAN メタデータにおけるそれらの状態の更新」を参照してください。</p>
crosscheck	<p>指定したバックアップおよびコピーの存在をチェックします。RMAN は、バックアップ・ピースを検索することができない場合、expired の状態にマークします。RMAN は、その他の検索できないファイル（イメージ・コピーおよびアーカイブ REDO ログ）はすべて deleted にマークします。</p> <p>ファイルがディスクにある場合、RMAN はファイル・ヘッダーを検索します。その他のデバイス・タイプの場合、RMAN はメディア・マネージャを検索して、ファイルがメディア管理カタログに存在するか確認します。</p> <p>注意: RMAN はアーカイブ REDO ログをコピーとみなします。したがって、1 つ以上のログが使用不可になった場合は、change archivelog all crosscheck コマンドを発行してください。アーカイブ・ログが再度使用不可になった場合には、catalog archivelog を発行して再カタログする必要があります。</p> <p>注意: 制御ファイル・コピーのクロスチェックを行う場合は、主キーではなくてファイル名を指定する必要があります。</p>

例

バックアップ・ピースの削除 この例では、メディア・マネージャによって格納されたバックアップ・ピースを削除し、メタデータにおけるそのバックアップ・ピースの状態を **deleted** に変更します。(**list** 出力では **deleted** のレコードは表示しません。したがって、このようなレコードにはリカバリ・カタログ・ビューでアクセスする必要があります。)

```
allocate channel for delete type 'sbt_tape';
change backuppiece '$ORACLE_HOME/dbs/testdb_87fa39e0' delete;
release channel;
```

バックアップ・セットを unavailable としてマークする この例では、バックアップ・セットを **unavailable** の状態にマークします。メンテナンス・チャンネルの割当ては不要です。

```
change backupset 100 unavailable;
```

ファイルのクロスチェック この例では、登録済みのアーカイブ REDO ログがすべて存在しているかを確認します。存在しないものとすると、RMAN はメタデータを **expired** に変更します。

```
allocate channel for maintenance type disk;
change archivelog all crosscheck;
release channel;
```

関連項目

11-13 ページの「[allocateForMaint](#)」

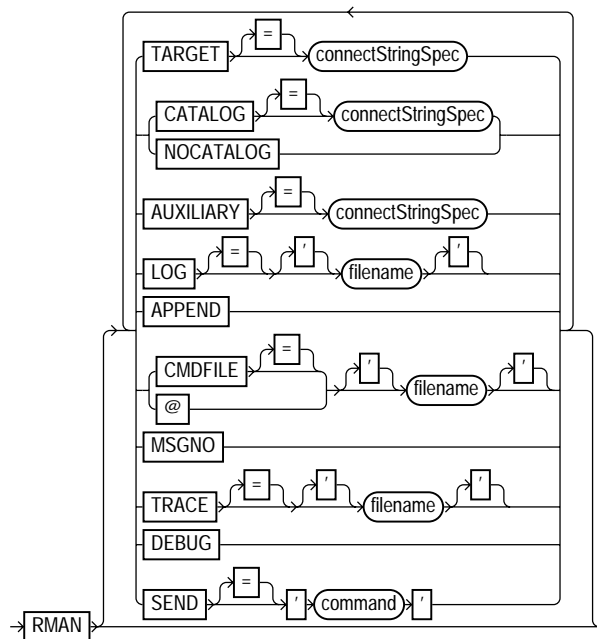
11-17 ページの「[archivelogRecordSpecifier](#)」

11-57 ページの「[crosscheck](#)」

11-63 ページの「[deleteExpired](#)」

cmdLine

構文



目的

O/S コマンド・プロンプトから RMAN を起動します。これらの引数は、次の目的に使用します。

- ターゲット・データベース、リカバリ・カタログ・データベース、あるいは補助データベースに接続する。

注意： プラットフォームによっては、システム内の他のユーザーがパスワードを参照できてしまうため、RMAN のコマンド行から接続することもできます。**connect** コマンドは、この問題を避ける代替手段です。

- リカバリ・カタログなしで RMAN を使用することを指定する。

- コマンド・ファイルを実行する。コマンド・ファイルとは、RMAN コマンドを含むユーザー定義ファイルのことです。
- RMAN がコマンドの処理結果を記録するファイルを指定する。
- コマンド・ファイルの内容を上書きではなく、追加する。
- デバッグ用のアウトプットを作成し、その場所を指定する。
- メディア・マネージャにコマンドを送信する。
- RMAN に **list** 出力のメッセージ番号を出力させる。

要件

これらのコマンドは、RMAN プロンプトではなく O/S コマンド行で使用する必要があります。

キーワードおよびパラメータ

target <i>connectStringSpec</i>	ターゲット・データベースへの接続文字列を指定します。たとえば、 target sys/change_on_install@inst1 のように指定します。11-46 ページの「 connectStringSpec 」を参照してください。
catalog <i>connectStringSpec</i>	リカバリ・カタログを格納するデータベースへの接続文字列を指定します。たとえば、 catalog rman/rman@inst2 のように指定します。11-46 ページの「 connectStringSpec 」を参照してください。
nocatalog	リカバリ・カタログなしで RMAN を使用することを示します。リカバリ・カタログなしで RMAN を開始する場合には、この引数を使用する必要があります。
auxiliary <i>connectStringSpec</i>	補助データベースへの接続文字列を指定します。たとえば、 auxiliary sys/change_on_install@dupdb のように指定します。11-46 ページの「 connectStringSpec 」を参照してください。
log filename	Recovery Manager が RMAN 出力を記録するファイルを指定します。RMAN 出力とは、処理したコマンドおよびその結果です。この引数を指定しないと、Recovery Manager はメッセージ・ログ・ファイルを標準出力に書き出します。
append	新規出力をメッセージ・ログ・ファイルの終わりに追加させます。このパラメータを指定せず、かつメッセージ・ログ・ファイルと同じ名前のファイルがすでにある場合、RMAN はそのファイルに上書きします。
cmdfile filename	ユーザー定義の RMAN コマンド・リストのファイルを実行します。ファイル名の最初の文字がアルファベットの場合は、ファイル名を囲む引用符を省略できます。 コマンド・ファイルの内容は、RMAN プロンプトに入力したものと同じにする必要があります。たとえば、次のファイルの内容によって、RMAN はターゲット・データベースおよびリカバリ・カタログ RCAT に接続します。 connect target; connect catalog rman/rman@rcat; RMAN は、コマンド・ファイルの実行後に終了します。

@filename	cmdfile と同じです。
msgno	RMAN にメッセージ番号を印刷させます。すなわち、 list コマンドの出力に対して、RMAN-xxxx の形で印刷させます。デフォルトでは、 list は RMAN-xxxx 接頭部を印刷しません。
trace filename	RMAN がデバッグ情報を出力するファイルの名前を指定します。デバッグ出力の生成には、 debug オプションの指定も必要です。 debug を指定しても、同時に trace も指定しないと、RMAN はデバッグ出力を標準出力に書き出すか、あるいはメッセージ・ログが指定されていればメッセージ・ログに書き出します。
debug	デバッグ機能を起動します。オラクル社カスタマ・サポートの指示のもとに問題の診断を行うときにのみ、このオプションを使用します。
send'command'	ベンダー固有のコマンド文字列を割当てられたチャネルすべてに送信します。この機能のサポートの有無は、メディア管理ソフトウェアのマニュアルを参照してください。 関連項目 : 特定チャネルへの文字列の送り方は、11-128 ページの「 send 」を参照してください。

例

リカバリ・カタログなしでの接続 この例では、リカバリ・カタログなしでターゲット・データベース PROD1 に接続します。

```
% rman target sys/sys_pwd@prod1 nocatalog
```

補助インスタンスへの接続 この例では、ターゲット・データベース PROD1、リカバリ・カタログ・データベース RCAT および補助インスタンス AUX1 に接続します。

```
% rman target sys/sys_pwd@prod1 catalog rman/rman@rcat auxiliary sys/aux_pwd@aux1
```

コマンド・ファイルの指定 この例では、ターゲット・データベース PROD1、リカバリ・カタログ・データベース RCAT に接続し、その後でコマンド・ファイルを実行します。
b_whole_10.rcv:

```
% rman target sys/sys_pwd@prod1 catalog rman/rman@rcat @'/oracle/dbs/b_whole_10.rcv'
```

APPEND モードでのメッセージ・ログの指定 この例では、リカバリ・カタログなしでターゲット・データベース PROD1 に接続し、メッセージ・ログに RMAN がメッセージを追加するよう指定しています。

```
% rman target sys/sys_pwd@prod1 nocatalog log = $ORACLE_HOME/dbs/log/msglog.f append
```

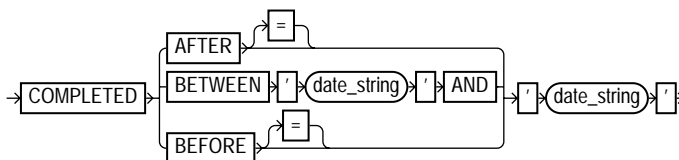
関連項目

11-44 ページの「**connect**」

11-122 ページの「**rmanCmd**」

completedTimeSpec

構文



目的

バックアップまたはコピーを完了させる時刻を指定する句です。

要件

日付文字列は次のものである必要があります。

- 現行有効な NLS 日付書式仕様に従って書式化していること。
- DATE 値を戻す SQL 式で作成していること。たとえば、'SYSDATE-30'。

この句は、次のコマンドと一緒に使用します。

- [crosscheck](#)
- [deleteExpired](#)
- [list](#)

注意： `from time` および `until time` パラメータは、`completedTimeSpec` を使用するコマンドでは無効になりました。これらのパラメータを指定したスクリプトを実行すると、ジョブは失敗します。

キーワードおよびパラメータ

after 'date_string'	その時刻より後に完了したバックアップを指定するための時刻です。
between 'date_string' および 'date_string'	その時間範囲の間に完了したバックアップを指定するための時間範囲です。
before 'date_string'	その時刻より前に完了したバックアップを指定するための時刻です。

例

時間範囲内のバックアップのクロスチェック この例では、前月に作成したデータベースのバックアップ・セットをクロスチェックします。

```
crosscheck backup of database between 'SYSDATE-62' and 'SYSDATE-31';
```

期限切れのバックアップの削除 この例では、最近 2 週間に作成した datafile1 の期限切れのバックアップ・セットを削除します。

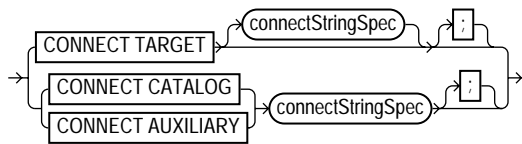
```
delete expired backup of datafile 1 after 'SYSDATE-14';
```

コピーのリスト作成 この例では、1998 年 12 月 13 日より前に作成した /oracle/dbs/tbs_22.f のイメージ・コピーのリストを作成します。

```
list copy of datafile '/oracle/dbs/tbs_22.f' before 'Dec 13 1998 20:31:10';
```

connect

構文



目的

RMAN とターゲット・データベース、補助データベース、またはリカバリ・カタログ・データベースとの接続を確立します。

注意： プラットフォームによっては、O/S コマンド行から接続すると、システムの他のユーザーがパスワードを参照できます。**connect** コマンドを使用すると、この問題を回避できます。

関連項目： コマンド行による接続オプションに関しては、11-39 ページの「**cmdLine**」を参照してください。

要件

connect コマンドを使用できるのは、RMAN がプロンプトを表示しており、かつ未接続状態の場合です。

キーワードおよびパラメータ

connect target <i>connectStringSpec</i>	RMAN とターゲット・データベースとの接続を確立します。11-46 ページの「 connectStringSpec 」を参照してください。
connect catalog <i>connectStringSpec</i>	RMAN とリカバリ・カタログ・データベースとの接続を確立します。11-46 ページの「 connectStringSpec 」を参照してください。
connect auxiliary <i>connectStringSpec</i>	RMAN と補助インスタンスとの接続を確立します。補助インスタンスは、 duplicate コマンドで使用するか、または TSPITR の間に使用することができます。11-46 ページの「 connectStringSpec 」を参照してください。

例

リカバリ・カタログなしでの接続 この例では、RMAN を起動し、次に NET8 サービス名 prod1 のターゲット・データベースに接続します。

```
% rman nocatalog
RMAN> connect target sys/change_on_install@prod1;
```

リカバリ・カタログを使用する接続 次の例では、RMAN を起動し、次に O/S 認証機能を使用してターゲット・データベース PROD1 に接続し、パスワード・ファイルを使用してリカバリ・カタログ・データベース RCAT に接続します。

```
% rman
RMAN> connect target /; connect catalog rman/rman@rcat;
```

ターゲット・データベース、リカバリ・カタログ・データベース、複製データベースへの接続 次の例では、3 つの異なるデータベースに、それぞれユーザー名とパスワードを指定して接続します。

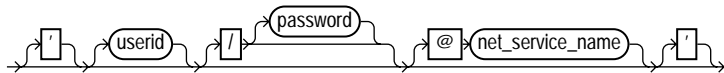
```
% rman
RMAN> connect target sys/sysdba@prod1;
RMAN> connect catalog rman/rman@rcat;
RMAN> connect auxiliary sys/sysdba@dupdb;
```

関連項目

11-39 ページの「[cmdLine](#)」

connectStringSpec

構文



目的

ターゲット・データベース、リカバリ・カタログ・データベースまたは補助データベースに接続するため、ユーザー名、パスワードおよびネット・サービス名を指定する句です。接続は、ユーザーの認証およびデータベースの識別のために必要です。

要件

- ターゲット・データベースおよび補助データベースの SYSDBA 権限があること。
- リカバリ・カタログ・データベースに接続するときは、ユーザー SYS としては接続しないでください。

キーワードおよびパラメータ

/	<p>ターゲット・データベースに接続するときにユーザー ID またはパスワードを指定しないと、スラッシュ (/) が、O/S 認証機能を使用して、SYS としての接続を確立します。たとえば、ターゲット・データベースに接続するには、次のように入力します。</p> <pre>% rman target /</pre> <p>注意: スラッシュは ORACLE_SID 環境変数によって、どのデータベースに接続する必要があるかを認識します。ORACLE_SID は補助データベースまたはターゲット・データベースのどちらかを指しますが、両方同時に指すことはありません。スラッシュのみでは、リカバリ・カタログ・データベースには接続できません。</p>
userid	<p>指定したユーザーに対してデータベース接続を確立します。パスワードを指定しないと、RMAN はプロンプトを表示して対話方式でパスワードを取得します。文字は端末には表示されません。</p> <p>ターゲット・データベースに接続するときは SYSDBA 権限が必要です。ただし、リカバリ・カタログ・データベースに SYS として接続することはできません。</p> <p>注意: 接続文字列に空白を含めることはできませんが、"/" および "@" などの文字を含めることはできます。</p>

<code>/password</code>	パスワードを使用して指定したユーザーに対して接続を確立します。ターゲット・データベースがオープンされていない場合は、パスワード・ファイルの存在を必要とします。
<code>@net_service_name</code>	オプションの Net8 ネット・サービス名を使用してデータベースへの接続を確立します。サービス名は、 <code>tnsnames.ora</code> ファイルに指定された有効にしてください。

例

リカバリ・カタログなしでの接続 この例では、パスワードおよび Net8 サービス名 PROD1 を使用してターゲット・データベースに接続します。

```
% rman target sys/change_on_install@prod1 nocatalog
```

対話方式でのパスワード入力 この例では、コマンド行からパスワードを入力せずに、ユーザー SYS としてターゲット・データベースに接続します。

```
% rman target sys

Recovery Manager: Release 8.1.5.0.0

target database Password:
```

O/S 認証機能を使用した接続 この例では、RMAN を起動し、次に O/S 認証機能を使用してターゲット・データベースに接続し、パスワード・ファイルを使用してリカバリ・カタログ・データベース RCAT に接続します。

```
% rman
RMAN> connect target /
RMAN> connect catalog rman/rman@rcat
```

ターゲット・データベース、リカバリ・カタログおよび補助インスタンスへの接続 この例では、3 つの異なるデータベースに、それぞれユーザー名、パスワードおよびサービス名を指定して、コマンド行から接続します。

```
% rman target sys/sysdba@prod1 catalog rman/rman@rcat auxiliary sys/sysdba@dupdb
```

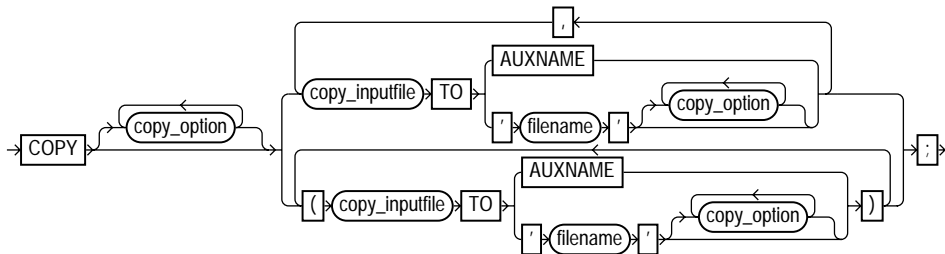
関連項目

11-39 ページの「[cmdLine](#)」

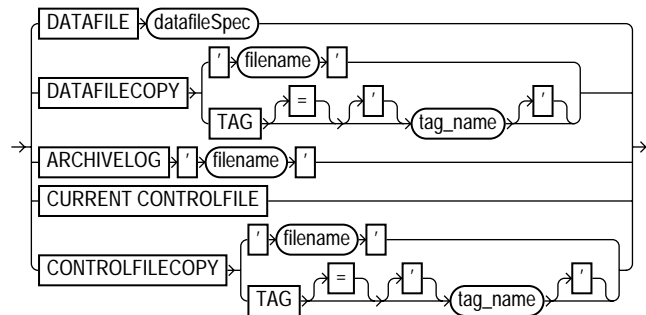
11-44 ページの「[connect](#)」

copy

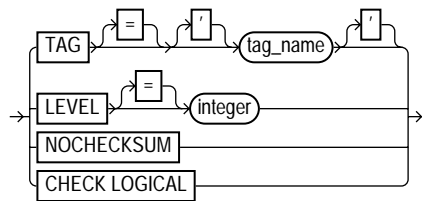
構文



copy_inputfile::=



copy_option::=



目的

ファイルのイメージ・コピーを作成します。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。

- データ・ファイル（現行ファイルまたはコピー）
- アーカイブ REDO ログ
- 制御ファイル（現行ファイルまたはコピー）

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、バックアップ・セットを使用可能にするには、その前に **restore** コマンドを使用する必要があります。したがってメディア回復は、データ・ファイル・コピー上に行うことができますが、1 データ・ファイルのみのバックアップでしかも 1 バックアップ・ピースしかない場合であってもバックアップ・セットに直接行うことはできません。

要件

- コマンドの実行は、**run** コマンドの中カッコ内で行います。
- **copy** コマンドの前に、**allocate type disk** オプションを少なくとも 1 つ指定します。
- 増分コピーは実行できません。

キーワードおよびパラメータ

<i>copy_option</i>	入力ファイルまたは出力ファイルまたは両方に関するオプション・パラメータを指定します。
tag <i>tag_name</i>	入力ファイルまたは出力ファイルのコピーのタグを指定します。
level <i>integer</i>	入力ファイルまたは出力ファイルのコピーをその後の増分バックアップ・セットの基礎として使用することで、それを増分バックアップ計画に組み込みます。一般的には、 level 0 を指定します。 level オプションを指定しなければ、データ・ファイルのコピーは増分バックアップ計画には何ら影響しません。
nochecksum	ブロックに対するチェックサムを抑止します。このオプションを指定しなければ、Oracle は各ブロックのチェックサムを計算します。RMAN はコピーを復元するときにチェックサムを検証します。データベースですでにブロック・チェックサムがメンテナンスされている場合は、このフラグは効果がありません。

	check logical	<p>物理的な破損チェックを通過したデータ・ブロックおよび索引ブロックについて論理的な破損がないかどうかをテストします。たとえば、行ピースまたは索引エントリの破損がないかを調べます。RMAN は論理的な破損を発見すると、<code>alert.log</code> およびサーバー・セッション・トレース・ファイルにそのブロックのログを書き込みます。</p> <p>あるファイルで検出された物理的な破損と論理的な破損の合計が、maxcorrupt の設定を下回る場合、RMAN コマンドは完了し、Oracle は V\$BACKUP_CORRUPTION および V\$COPY_CORRUPTION に破損ブロック範囲を移入します。maxcorrupt をこえている場合は、コマンドはビューへの移入を行わずに終了します。</p> <p>注意: copy および backup の場合の、maxcorrupt の設定は、ファイルに許容される物理的な破損と論理的な破損の総数を意味します。</p>
<i>copy_inputfile</i> 句	入力ファイル、すなわちコピーが必要なファイルのタイプを指定します。	
	datafile <i>datafileSpec</i>	<p>1 つ以上のデータ・ファイルのリストを入力ファイルとして指定します。 11-60 ページの「datafileSpec」を参照してください。</p> <p>注意: ファイル名を指定する場合、それは制御ファイルにリストされている現行のデータ・ファイルの名前であることが必要です。</p>
	datafilecopy	<p>1 つ以上のデータ・ファイルのリストを入力ファイルとして指定します。 データ・ファイル・コピーを <i>'filename'</i> または tag = tag_name によって指定します。制御ファイルにリストされている現行のデータ・ファイル名は使用できません。既存のコピーは、以前の copy コマンドで作成されたものでも、あるいは外部 O/S ユーティリティで作成されたものでもかまいません。</p>
	archivelog <i>'filename'</i>	<p>入力するアーカイブ REDO ログのファイル名を指定します。アーカイブ・ログは、Oracle アーカイブ・セッションにより作成されたものでも、以前の 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 REDO ログをファイル名で指定します。</p>
	current controlfile	<p>現行の制御ファイルを指定します。</p>
	controlfilecopy <i>'filename'</i>	<p>制御ファイル・コピーのファイル名を指定します。1 つ以上の制御ファイル・コピーを指定するために、tag = tag_name を設定することもできます。</p> <p>注意: 制御ファイル・コピーはバックアップ制御ファイルとしてマークされていますので、制御ファイル・コピーをマウントするには、メディア回復が必要です。このコマンドは、ALTER DATABASE BACKUP CONTROLFILE TO '...' 文と同等です。</p>
to 'filename'	出力ファイル・コピーのファイル名を指定します。	
to auxname	入力データ・ファイルを、以前 set auxname コマンドで指定したファイル名にコピーするよう、Oracle に指定します。	

例

データ・ファイルのコピー この例では、データ・ファイル `tbs_01.f` を出力ファイル `temp3.f` に **nochecksum** オプションでコピーし、レベル0のバックアップとしてマークします。

```
run {
  allocate channel dev1 type disk;
  copy
    nochecksum
    datafile '$ORACLE_HOME/dbs/tbs_01.f'
      to '$ORACLE_HOME/copy/temp3.f'
    level 0;
}
```

制御ファイルのコピー この例では、現行の制御ファイルをコピーし、*weekly_cf_copy* のタグを付けます。

```
run {
  allocate channel dev1 type disk;
  copy
    current controlfile
      to '$ORACLE_HOME/copy/cf1.f'
    tag = 'weekly_cf_copy';
}
```

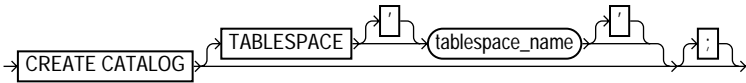
関連項目

11-9 ページの「[allocate](#)」

11-21 ページの「[backup](#)」

createCatalog

構文



目的

リカバリ・カタログとして使用するスキーマを作成します。一般的には、このスキーマはリカバリ・カタログ用のデータベースに作成します。

注意： 8.1.5 以前のリリースでは、リカバリ・カタログ・スキーマはリカバリ・カタログ・データベースに接続し、`catrman.sql` スクリプトを実行して作成します。

要件

- このコマンドは、RMAN プロンプトから実行してください。
- リカバリ・カタログの所有者は、`RECOVERY_CATALOG_OWNER` のロールのみでなく、リカバリ・カタログ・テーブルを作成する表領域の権限も付与されていること。
- `catalog` コマンド行オプション（11-39 ページの「[cmdLine](#)」を参照）または `connect catalog` コマンドのいずれかで、RMAN をリカバリ・カタログに接続していること。
- `SYS` スキーマでリカバリ・カタログを作成しないこと。

関連項目： `RECOVERY_CATALOG_OWNER` の役割の詳細は、『Oracle8i 管理者ガイド』を参照してください。

キーワードおよびパラメータ

tablespace <i>tablespace_name</i>	リカバリ・カタログ・スキーマを格納する表領域を指定します。カタログの所有者は割当て権限を付与されている必要があります。表領域を指定しないと、RMAN はリカバリ・カタログを <code>SYSTEM</code> 表領域に格納します。
---	---

例

カタログ・スキーマの作成 この例では、ユーザー RMAN を作成し、RMAN に RECOVERY_CATALOG_OWNER の役割の権限を付与し、次にデータベース RCAT のスキーマ RMAN.RCVCAT にリカバリ・カタログを作成しています。

```
% sqlplus sys/change_on_install@rcat;

SQL> CREATE USER rman IDENTIFIED BY rman
      2> DEFAULT TABLESPACE rcvcat QUOTA UNLIMITED ON rcvcat;
SQL> GRANT recovery_catalog_owner TO rman;
SQL> exit

% connect rman/rman@rcat;
RMAN> create catalog tablespace rcvcat;
```

関連項目

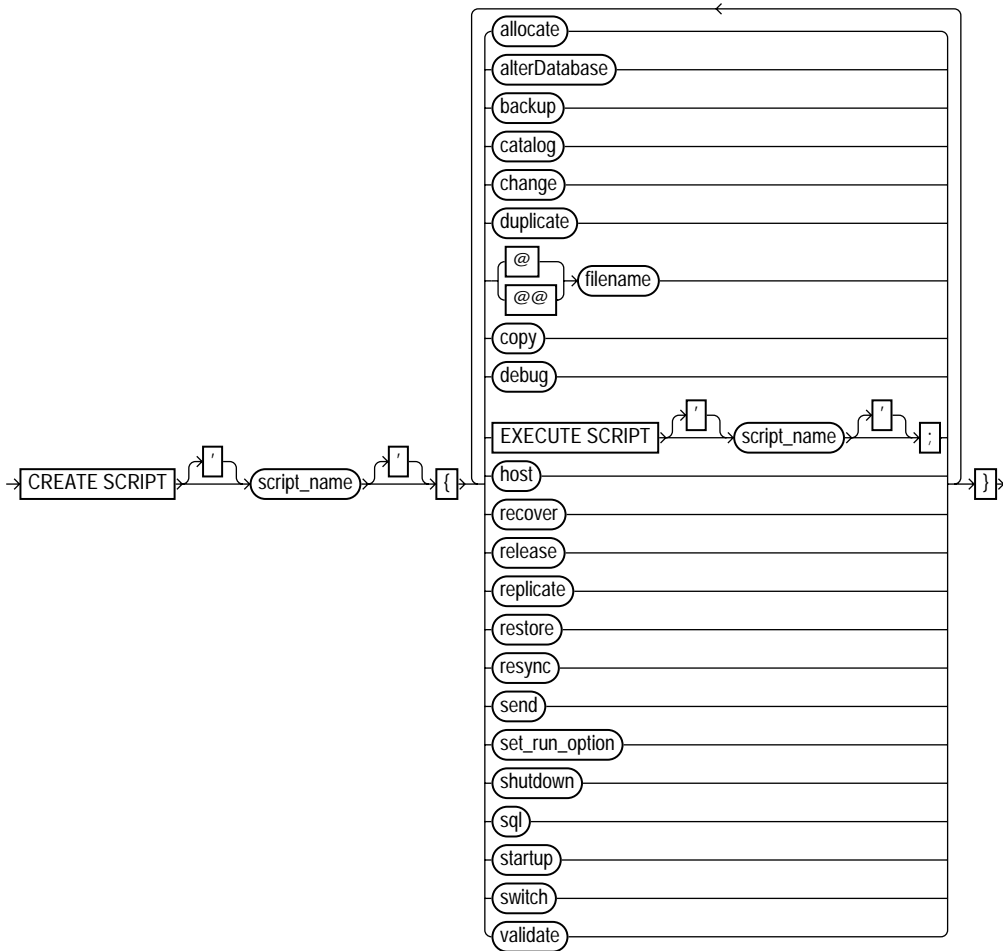
11-44 ページの「[connect](#)」

11-68 ページの「[dropCatalog](#)」

11-149 ページの「[upgradeCatalog](#)」

createScript

構文



目的

スクリプトを作成し、後で参照するためにリカバリ・カタログに格納します。ストアド・スクリプトは頻繁に実行する RMAN コマンドを収集した共通リポジトリを提供します。スクリプトは、**run** コマンドで有効なコマンドを使用してください。スクリプトはそのままでは実行されません。スクリプトを実行するには、**execute script** コマンド (11-125 ページの「**run**」を参照) を使用します。

要件

次の制限事項に注意してください。

- **create script** は、RMAN プロンプトから実行すること。
- リカバリ・カタログに接続していること。
- スストアド・スクリプトのなかで **run** コマンドは実行できません。**run** コマンドで **execute script** コマンドを実行すると、RMAN はスクリプトの内容を **run** の中カッコの間に入れます。そのため、スクリプトのなかでチャンネルをすでに割り当てている場合は、**run** コマンドのレベルではチャンネルを割り当てないでください。

キーワードおよびパラメータ

ストアド・スクリプトで使える各コマンドの説明は、11-21 ページの「**backup**」など、該当の項目を参照してください。スクリプトのなかで実行する場合は、**@@** コマンドは特殊な動作を行うことに注意してください。**execute script** コマンドの詳細は、11-125 ページの「**run**」を参照してください。

<i>script_name</i>	指定した名前のストアド・スクリプトを作成します。 create script 'script_name' (...) コマンドのカッコのなかで使える文は、 run コマンドのなかで使えるものと同じです。中カッコのなかの文は、 <i>job_command_list</i> を構成します。 注意: スストアド・スクリプトを実行するには、 run コマンドの中カッコのなかで execute script コマンドを使用します。
<i>@filename</i>	絶対パス名、(たとえば @\$ORACLE_HOME/dbs/cmd/cmd1.f) が指定された O/S ファイルに格納した一連の RMAN コマンドを実行します。文字列の前後に引用符を使用したり、@ とファイル名の間に空白を残さないでください。RMAN は、指定したファイルの内容が @ コマンドに代入されたかのように、処理します。 注意: ファイルには Recovery Manager コマンドとして完全なもののみを入れる必要があります。ファイルに不完全なコマンドがあると、構文エラーになります。

@@filename

一連の RMAN コマンドが入った O/S ファイルの相対ファイル名、(たとえば `cmd1.f`) を指定します。@@ によって指定されたコマンド・ファイルは、親スクリプトと同じディレクトリにあるとみなされます。文字列の前後に引用符を使用したり、@@ とファイル名の間に空白を残さないでください。RMAN は、指定したファイルの内容が @@ コマンドに代入されたかのように、処理します。

注意: ファイルには Recovery Manager コマンドとして完全なもののみを入れる必要があります。ファイルに不完全なコマンドがあると、構文エラーになります。

例

スクリプトの作成 この例では、データベースおよび REDO ログをバックアップするスクリプト `B_WHOLE_10` を作成し、次に実行します。

```
create script b_whole_10 {
    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
        database;
    sql 'ALTER SYSTEM ARCHIVE LOG CURRENT';
    backup
        filesperset 20
        archivelog all
        delete input;
}
```

RMAN-03022: compiling command: create script

RMAN-03023: executing command: create script

RMAN-08085: created script b_whole_10

```
run { execute script b_whole_10; }
```

関連項目

11-65 ページの「[deleteScript](#)」

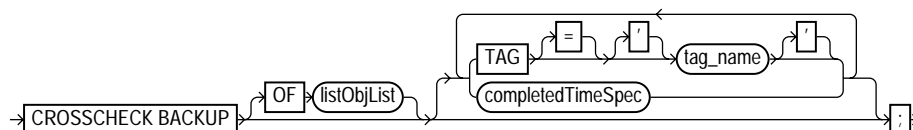
11-87 ページの「[printScript](#)」

11-98 ページの「[replaceScript](#)」

11-125 ページの「[run](#)」

crosscheck

構文



目的

ディスクまたはテープに格納されたバックアップが存在するかを確認します。バックアップとは、バックアップ・セットまたはメディア管理ソフトウェアにより管理されるプロキシ・コピーのことです。

crosscheck コマンドは **available** または **expired** とマークされたバックアップのみをチェックします。チェックは、チャンネルが **type disk** のときはディスクのバックアップ・ピースを調べることであり、またチャンネルが **type 'sbt_tape'** のときはメディア・マネージャに問い合わせることにより行います。指定したチャンネル上に作成されたバックアップのみを処理します。

RMAN は、検索することができないバックアップ・ピースを削除せずに、そのメタデータ・レコードを **expired** の状態に更新します。バックアップ・ピースのなかに、メディア・マネージャの構成ミスなどのために、誤って **expired** にマークされたものがある場合は、ファイルが実際にはメディア・マネージャに存在することを確認してから、**crosscheck backup** コマンドを再度実行し、ファイルを **available** の状態に復元します。

要件

- **crosscheck backup** は、RMAN プロンプトから実行してください。
- **crosscheckbackup** を発行する前に、メンテナンス・チャンネルを割り当ててください。
- リカバリ・カタログを使用してください。

キーワードおよびパラメータ

of <i>listObjList</i>	操作するオブジェクトのリストを <i>listObjList</i> 句で指定したオブジェクト・タイプのものに限定します。オブジェクトの指定がない場合、コマンドは database controlfile archivelog all のすべてのオブジェクトをチェックします。11-85 ページの「 listObjList 」を参照してください。
tag <i>tag_name</i>	バックアップ・セットに対してタグを指定します。

completedTimeSpec バックアップ完了のための時刻範囲を指定します。11-85 ページの「*completedTimeSpec*」を参照してください。

例

すべてのバックアップのクロスチェック 次の例では、ディスクにあるすべてのバックアップの状態を問い合せています。例にはサンプル出力も含まれています。

```

RMAN> allocate channel for maintenance type disk;

RMAN-03022: compiling command: allocate
RMAN-03023: executing command: allocate
RMAN-08030: allocated channel: delete
RMAN-08500: channel delete: sid=15 devtype=DISK

RMAN> crosscheck backup;

RMAN-03022: compiling command: XCHECK
RMAN-03023: executing command: XCHECK
RMAN-08517: backup piece handle=/vobs/oracle/dbs/01a8t4pq_1_1 recid=1 stamp=3448
RMAN-08074: crosschecked backup piece: found to be 'AVAILABLE'
RMAN-08517: backup piece handle=/vobs/oracle/dbs/05a9cfs3_1_1 recid=5 stamp=3453
RMAN-08074: crosschecked backup piece: found to be 'AVAILABLE'
RMAN-08517: backup piece handle=/vobs/oracle/dbs/06a9cfv8_1_1 recid=6 stamp=3453
RMAN-08074: crosschecked backup piece: found to be 'AVAILABLE'
RMAN-08517: backup piece handle=/vobs/oracle/dbs/07a9ck4t_1_1 recid=7 stamp=3453
RMAN-08074: crosschecked backup piece: found to be 'AVAILABLE'
RMAN-08517: backup piece handle=/vobs/oracle/dbs/08a9cl23_1_1 recid=8 stamp=3453
RMAN-08074: crosschecked backup piece: found to be 'EXPIRED'
RMAN-08517: backup piece handle=/vobs/oracle/dbs/09a9cl2b_1_1 recid=9 stamp=3453
RMAN-08074: crosschecked backup piece: found to be 'AVAILABLE'

RMAN> release channel;

RMAN-03022: compiling command: release
RMAN-03023: executing command: release
RMAN-08031: released channel: delete

```

日付範囲を指定したクロスチェック 次の例では、datafile 3 のある 6ヶ月間の状況をメディア・マネージャに問い合せています。RMAN が使用している日付書式は、NLS_DATE_FORMAT パラメータに指定した 'DD-MON-YY' であることに注意してください。

```

allocate channel for maintenance type 'sbt_tape';
crosscheck backup of datafile 3 device type 'sbt_tape' completed between '01-JAN-98' and
'01-JUL-98';
release channel;

```


関連項目

11-13 ページの「[allocateForMaint](#)」

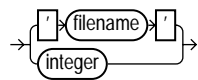
11-35 ページの「[change](#)」

11-76 ページの「[list](#)」

11-103 ページの「[report](#)」

datafileSpec

構文



目的

データ・ファイルをファイル名または絶対ファイル番号で指定する句です。

キーワードおよびパラメータ

<i>'datafile'</i>	絶対パスまたは相対ファイル名のどちらかでデータ・ファイルを指定します。相対ファイル名を指定した場合、ファイル名はターゲット・データベースによってポート固有の方法で修飾されます。
<i>integer</i>	データ・ファイルを絶対ファイル番号で指定します。ファイル番号は、VSDATAFILE、VSDATAFILE_COPY または VSDATAFILE_HEADER ビューから取得するか、または report schema コマンドの出力から取得します。

例

ファイル名でデータ・ファイルを指定 この例では、ファイル名を指定して、データ・ファイル `/oracle/dbs/tbs_12` をディスクにコピーしています。

```
run {
  allocate channel chl type disk;
  copy datafile '/oracle/dbs/tbs_12.f'
    to '/oracle/copy/tbs_1.copy';
}
```

絶対ファイル番号でデータ・ファイルを指定 この例では、ファイル番号を指定して、データ・ファイル `/oracle/dbs/tbs_31.f` をディスクにコピーしています。

```
RMAN> report schema;

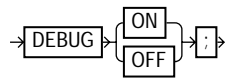
RMAN-03022: compiling command: report
Report of database schema
File K-bytes    Tablespace          RB segs Name
-----
1           47104 SYSTEM              YES    /vobs/oracle/dbs/tbs_01.f
```

2	978 SYSTEM	YES	/vobs/oracle/dbs/tbs_02.f
3	978 TBS_1	NO	/vobs/oracle/dbs/tbs_11.f
4	978 TBS_1	NO	/vobs/oracle/dbs/tbs_12.f
5	978 TBS_2	NO	/vobs/oracle/dbs/tbs_21.f
6	978 TBS_2	NO	/vobs/oracle/dbs/tbs_22.dfb
7	500 TBS_1	NO	/vobs/oracle/dbs/tbs_13.f
8	500 TBS_2	NO	/vobs/oracle/dbs/tbs_23.f
9	500 TBS_2	NO	/vobs/oracle/dbs/tbs_24.f
10	500 TBS_3	NO	/vobs/oracle/dbs/tbs_31.f

```
run {  
    allocate channel chl type disk;  
    copy datafile 10  
    to '/oracle/copy/tbs_31.copy';  
}
```

debug

構文



目的

RMAN のデバッグ機能をオン / オフします。この機能は、オラクル社カスタマ・サポートのガイドに従って使用してください。この機能の目的は、RMAN の問題を診断することです。

要件

このコマンドは、**run** コマンドの中カッコの中、または RMAN プロンプトで実行してください。

キーワードおよびパラメータ

on	デバッグ機能をアクティブ化します。
off	デバッグ機能を非アクティブ化します。

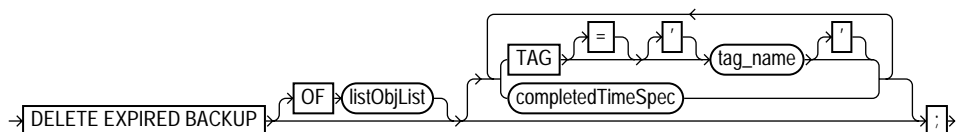
例

デバッグ機能をアクティブ化 この例では、データ・ファイル 3 のバックアップ中にはデバッグ出力が表示されますが、データ・ファイル 4 のバックアップでは表示されません。

```
run {
  allocate channel c1 type disk;
  debug on;
  backup datafile 3;
  debug off;
  backup datafile 4;
}
```

deleteExpired

構文



目的

RMAN メタデータのバックアップ・セットを **expired** の状態から **deleted** の状態に更新します。このコマンドが機能するのは、**crosscheck** コマンドにより **expired** にマークされたバックアップ・ピースのリカバリ・カタログ・レコードのみです。バックアップ・セットの状態データを取得するには、**list** コマンドを使用するか、またはリカバリ・カタログ・ビューを検索します。

注意： **delete expired backup** コマンドを実行したときに、なんらかの理由で **expired** にマークされたバックアップ・セットが存在すると、RMAN はその物理ファイルを削除します。

要件

- リカバリ・カタログを使用する必要があります。
- **delete expired backup** は、RMAN プロンプトでのみ実行可能です。
- **delete expired backup** の前に、**allocatechannel for delete** コマンドまたは **allocate channel for maintenance** コマンドを実行してください (11-13 ページの「**allocateForMaint**」を参照)。

キーワードおよびパラメータ

of <i>listObjList</i>	オブジェクトのリストを <i>listObjList</i> 句で指定したオブジェクト・タイプのものに限定します。オブジェクトを指定しないと、コマンドは database controlfile archivelog all のすべての期限切れオブジェクトを削除します。11-85 ページの「 listObjList 」を参照してください。
tag <i>tag_name</i>	バックアップ・セットに対してタグを指定します。
<i>completedTimeSpec</i>	バックアップ完了のための時刻範囲を指定します。11-85 ページの「 completedTimeSpec 」を参照してください。

例

期限切れのバックアップの削除 次の例では、表領域 USER_DATA について **expired** になって 1ヶ月以上経過しているバックアップがあるかどうかメディア・マネージャをチェックし、該当するレコードを **deleted** の状態に更新しています。

```
allocate channel for delete chl type 'sbt_tape';
crosscheck backup of tablespace user_data completed before 'SYSDATE-31';
delete expired backup of tablespace user_data' completed before 'SYSDATE-31';
release channel;
```

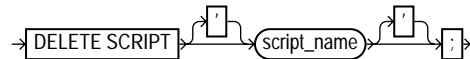
関連項目

11-13 ページの「[allocateForMaint](#)」

11-57 ページの「[crosscheck](#)」

deleteScript

構文



目的

ストアド・スクリプトをリカバリ・カタログから削除します。

要件

- **delete script** は、RMAN プロンプトから実行してください。
- リカバリ・カタログを使用する必要があります。

キーワードおよびパラメータ

<i>script_name</i>	指定したスクリプトを削除します。スクリプト名は、以前の create script または replace script コマンド（11-54 ページの「 createScript 」を参照）で指定した名前である必要があります。
--------------------	--

例

スクリプトの削除 この例では、スクリプト `b_whole_10` を削除します。

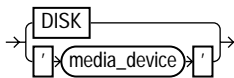
```
delete script 'b_whole_10';
```

関連項目

- 11-54 ページの「[createScript](#)」
- 11-87 ページの「[printScript](#)」
- 11-98 ページの「[replaceScript](#)」
- 11-125 ページの「[run](#)」

deviceSpecifier

構文



目的

バックアップまたはコピーに使用する記憶装置のタイプを指定する句です。

キーワードおよびパラメータ

disk	ディスク記憶装置を指定します。
'media_device'	記憶装置として使用する順次 I/O デバイスまたはアクセス方法を指定します。順次 I/O デバイス・タイプの構文と意味はプラットフォームによって異なります。現行、唯一使用できる値は、sbt_tape です。すなわち、サード・パーティのテープ・サブシステム・インタフェースです。

例

テープ・チャンネルの割当て この例では、メディア管理デバイスにメンテナンス・チャンネルを割り当てます。

```
allocate channel for maintenance type 'sbt_tape';
```

ディスクへのバックアップ この例では、データベースをディスクにバックアップします。

```
run {
  allocate channel ch1 type disk;
  backup database;
}
```

ディスクからテープへの復元 この例では、ディスクおよびテープからのバックアップを使用して、データベースを復元します。

```
run {
  allocate channel dl type disk;
  allocate channel tl type 'sbt_tape'
  restore database;
  recover database;
}
```


関連項目

- 11-9 ページの「[allocate](#)」
- 11-13 ページの「[allocateForMaint](#)」
- 11-76 ページの「[list](#)」
- 11-97 ページの「[releaseForMaint](#)」
- 11-103 ページの「[report](#)」
- 11-113 ページの「[restore](#)」

dropCatalog

構文



目的

リカバリ・カタログからスキーマを削除します。

警告： このコマンドは、リカバリ・カタログからすべての情報を削除します。リカバリ・カタログのバックアップを取っておかないと、このリカバリ・カタログで管理しているデータベースのバックアップは、このコマンドの実行後すべて使用できなくなります。

要件

- このコマンドは、RMAN プロンプトから実行してください。
- **catalog** コマンド行オプション（11-39 ページの「[cmdLine](#)」を参照）または **connect catalog** コマンドのいずれかで、RMAN をリカバリ・カタログに接続していること。
- スキーマの削除を確認するため、コマンドは 2 回入力してください。

例

カタログの削除 この例では、リカバリ・カタログからスキーマを削除します（確認のため、コマンドを 2 回入力する必要があります）。

```
RMAN> drop catalog
```

```
RMAN-06435: recovery catalog owner is rman
```

```
RMAN-06436: enter DROP CATALOG command again to confirm catalog removal
```

```
RMAN> drop catalog
```

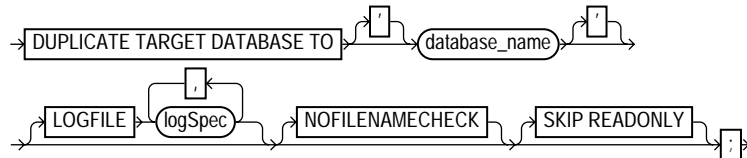
関連項目

11-52 ページの「[createCatalog](#)」

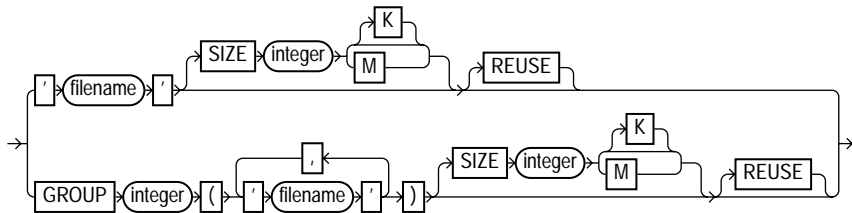
11-149 ページの「[upgradeCatalog](#)」

duplicate

構文



logSpec::=



目的

ターゲット・データベースのバックアップを使用して、複製データベースを作成します。複製データベースによって、バックアップおよび復元の手順のテストに安全な環境が提供されます。

関連項目：データベースの複製の方法は、[第 10 章の「Recovery Manager による複製データベースの作成」](#)を参照してください。

要件

- このコマンドの実行は、**run** コマンドの中カッコ内で実行してください。
- このコマンドでは、リカバリ・カタログを使用する必要はありません。
- 1 つ以上の **allocate auxiliary channel** コマンドを実行した後に、**duplicate** コマンドを実行してください。イメージ・コピーまたはディスクにあるバックアップから複製する場合は、割り当てるチャンネルが多いほど、複製操作が速くなります。
- ターゲット・データベースと補助インスタンスの両方に接続する必要があります。
- 補助インスタンスは NOMOUNT オプションで起動する必要があります。

- ターゲット・データベースはオープンの状態である必要があります。
- ターゲット・データベースと複製データベースが同じホストにある場合、CONTROL_FILES パラメータを設定して、複製データベースの制御ファイルがターゲットの制御ファイルを上書きしないようにしてください。パラメータの説明は、『Oracle8i リファレンス・マニュアル』を参照してください。
- ターゲット・データベースと複製データベースが同じホストを共有する場合、すべての * _PATH および * _DEST 初期化パラメータを適切に設定して、ターゲット・データベース・ファイルが複製データベース・ファイルに上書きされないようにしてください。パラメータの説明は、『Oracle8i リファレンス・マニュアル』を参照してください。
- ターゲット・データベースと複製データベースが別のホストにある場合は、次のいずれかの処置をとる必要があります。
 - ディスク・コピーおよびバックアップをターゲット・ホストから複製ホストに移動して、再度カタログに登録すること。
 - 複製ホストから、ターゲット・ホストのすべてのバックアップおよびコピーに、確実にリモート・アクセスできること。
- データ・ファイルおよびオンライン REDO ログに対して、新規ファイル名を指定するか、またはターゲット・ファイル名を変換してください。ファイル名を指定しないと、RMAN はターゲットのデータ・ファイル名を再使用します。この場合、**nofilenamecheck** を使用する必要があります。
- **duplicate** コマンドは、各ターゲット・ファイルの該当するバックアップまたはコピーを自動的に復元します。ターゲット・データ・ファイルにバックアップがない場合、複製操作は正常に終了しません。

キーワードおよびパラメータ

database_name 複製データベースの名前を指定します。データベース名の指定が必要な理由は、データベースを起動したのに、それがマウントされていないければ、データベース名を取得できないからです。名前は、複製データベースの `init.ora` ファイルにある名前と一致させる必要があります。これが一致していないと、Oracle は制御ファイルを作成するときにエラーを表示します。

注意: ターゲット・データベースと複製データベースに同じデータベース名を使用することができます。RMAN は複製データベースに新しい DBID を生成するからです。

logfile <i>logSpec</i>	<p>オンライン REDO ログを指定します。構文は、CREATE DATABASE 文の LOGFILE オプションの構文と同じです。</p> <p>logfile 句を指定しないと、RMAN は LOG_FILE_NAME_CONVERT が設定されていれば、それを使用します。logfile も LOG_FILE_NAME_CONVERT も指定しないと、RMAN は複製ファイルに元のターゲット REDO ログのファイル名を使用します。この場合、nofilenamecheck オプションを使用する必要があります。</p> <p>関連項目 : CREATE DATABASE 文に関しては、『Oracle8i SQL リファレンス』を参照してください。</p>
<i>'filename'</i>	オンライン REDO ログのファイル名を指定します。
size <i>integer</i>	<p>ファイルのサイズをキロバイト (K) またはメガバイト (M) で指定します。ファイルがすでに存在する場合は、このパラメータを省略することができます。</p>
reuse	<p>Oracle は既存のファイルを再使用します。ファイルがすでに存在する場合、Oracle はそのサイズが size パラメータと一致するか検証します。ファイルが存在しない場合、Oracle はファイルを作成します。ファイルがすでに存在する場合は、size パラメータを省略することができます。</p> <p>reuse オプションは、size と一緒に使用したときにのみ意味があります。size パラメータを省略した場合、Oracle はファイルがすでに存在しているものとみなします。</p>
group <i>integer</i> (<i>'filename', ...</i>)	1 つ以上のメンバーを持つ REDO ログ・グループを指定します。カッコ内に指定したファイル名はグループのメンバーを示します。
nofilenamecheck	<p>複製ファイルと同じ名前を共有するターゲット・データ・ファイルが使用されているかどうかのチェックを RMAN にさせないようにします。このオプションを指定した場合に、複製操作が使用中のデータを上書きしないか確認するのは、ユーザーの責任です。</p> <p>このオプションが必要になるのは、ディスク構成、ディレクトリ構造およびファイル名がターゲット・データベースのホストと同じであるが、それとは別のホストに複製データベースを作成する場合です。たとえば、HOST1 の /dbs ディレクトリに次のものが存在する、小規模なデータベースを考えてください。</p> <pre> /oracle/dbs/system_prod1.dbf /oracle/dbs/users_prod1.dbf /oracle/dbs/tools_prod1.dbf /oracle/dbs/rbs_prod1.dbf /oracle/dbs/users2_prod1.dbf </pre> <p>HOST2 にデータベースを複製する必要があるとします。また、HOST2 は HOST1 と同じファイル・システム /oracle/dbs/ * で、かつ複製データベースには HOST1 と同じファイル名を使用する必要があるとします。この場合、nofilenamecheck を使用することで、すべてのファイルの改名は行わないでください。RMAN は別のホストを認識できないので、ファイル名のチェックをすべきでないことを、自動的に判断できません。</p>

skip readonly

読取り専用の表領域にあるデータ・ファイルは複製データベースから除外します。

注意: スキップした読取り専用の表領域のレコードは、DBA_TABLESPACES にまだ存在しています。この機能によって、後で読取り専用の表領域をアクティブ化できます。たとえば、読取り専用の表領域のデータをディスクではなく CD-ROM に格納して、その後 CD-ROM をマウントしてデータを参照することができます。

例

新しいファイル名の手動による設定 この例では、ターゲット・データベースが HOST1 にあり、ファイル構造 /oracle/dbs/* の host2 の NEWDB にデータベースを複製することを想定しています。HOST1 にあるファイル名は不規則に命名されており、またいくつものサブ・ディレクトリに分散しているので、**set newname** コマンドを使用して、一貫性をもつように再命名します。**duplicate** コマンドは、テープに格納してあるバックアップ・セットを使用して、ターゲット・データベースを NEWDB データベースに複製します。

```
connect target;
connect catalog rman/rman@mancat;
connect auxiliary sys/change_on_install@newdb;
run {
  allocate auxiliary channel newdb1 type 'sbt_tape';
  allocate auxiliary channel newdb2 type 'sbt_tape';
  allocate auxiliary channel newdb3 type 'sbt_tape';
  allocate auxiliary channel newdb4 type 'sbt_tape';
  set newname for datafile 1 TO '$ORACLE_HOME/dbs/newdb_data_01.f';
  set newname for datafile 2 TO '$ORACLE_HOME/dbs/newdb_data_02.f';
  set newname for datafile 3 TO '$ORACLE_HOME/dbs/newdb_data_11.f';
  set newname for datafile 4 TO '$ORACLE_HOME/dbs/newdb_data_12.f';
  set newname for datafile 5 TO '$ORACLE_HOME/dbs/newdb_data_21.f';
  set newname for datafile 6 TO '$ORACLE_HOME/dbs/newdb_data_22.f';
  duplicate target database to newdb logfile
    group 1 ('$ORACLE_HOME/dbs/newdb_log_1_1.f',
             '$ORACLE_HOME/dbs/newdb_log_1_2.f') size 200K,
    group 2 ('$ORACLE_HOME/dbs/newdb_log_2_1.f',
             '$ORACLE_HOME/dbs/newdb_log_2_2.f') size 200K reuse;
}
```

ターゲット・ファイル名の再使用 この例では、新しいホストへ復元すること、および次のことを想定しています。

- ターゲット・ホストと複製ホストは同一のファイル構造である。
- 複製ファイルをターゲット・データベース・ファイルとまったく同じ名前にする。
- リカバリ・カタログは使用しない。
- 読取り専用の表領域は複製しない。

- RMAN に、複製ファイルと同じ名前のターゲット・データベースのファイルが使用中か、チェックさせない。

```
connect target
connect auxiliary sys/aux_pwd@newdb
run {
    allocate auxiliary channel ndbnewh1 type disk;
    allocate auxiliary channel ndbnewh2 type disk;
    duplicate target database to ndbnewh
    logfile
        '$ORACLE_HOME/dbs/log_1.f' size 200K,
        '$ORACLE_HOME/dbs/log_2.f' size 200K
    skip readonly
    nofilenamecheck;
}
```

関連項目

- 11-9 ページの「[allocate](#)」
- 11-44 ページの「[connect](#)」
- 11-48 ページの「[copy](#)」
- 11-130 ページの「[set](#)」
- 11-143 ページの「[startup](#)」

host

構文



目的

RMAN のなかから O/S コマンド行のサブ・シェルを起動します。

要件

このコマンドは、**run** コマンドの中カッコの中、または RMAN プロンプトで実行してください。

キーワードおよびパラメータ

host	<p>O/S コマンドを実行できるようにします。次のパラメータを使用します。</p> <ul style="list-style-type: none">■ 'command' あり。この場合、RMAN は指定文字列にあるコマンドを実行し、継続します。■ 'command' なし。この場合、RMAN はコマンド・プロンプトを表示し、ユーザーがサブ・シェルを終了後に再開します。
------	--

例

RMAN 内での O/S コピーの実行 この例では、データベースを停止し、メディア・マネージャを使用してデータ・ファイル tbs_01.f のバックアップをとり、次に UNIX コマンドを使用してディスクに同じファイルのイメージ・コピーを作成しています。データベースは、分裂ブロックを防ぐため、正しく停止する必要があります。

```
shutdown immediate;
run {
  allocate channel ch1 type disk;
  allocate channel ch2 type 'sbt_tape';
  backup datafile '$ORACLE_HOME/dbs/tbs_01.f' channel ch2;
  host 'cp $ORACLE_HOME/dbs/tbs_01.f $ORACLE_HOME/dbs/copy/tbs_01.f';
}
```


コピー・ジョブでの O/S への切替え この例では、datafile3 のイメージ・コピーを作成し、UNIX プロンプトに切替えて、コピーがディレクトリにあることをチェックし、次に **run** ジョブを再開しています。

```
RMAN> run {
2> allocate channel c1 type disk;
3> copy datafile 3 to 'df.3';
4> host;
5> release channel c1;
6> }
```

RMAN-03022: compiling command: allocate
RMAN-03023: executing command: allocate
RMAN-08030: allocated channel: c1
RMAN-08500: channel c1: sid=17 devtype=DISK

RMAN-03022: compiling command: copy
RMAN-03023: executing command: copy
RMAN-08000: channel c1: copied datafile 3
RMAN-08501: output filename=/oracle/dbs/df.3 recid=102 stamp=352745706
RMAN-03023: executing command: partial resync
RMAN-08003: starting partial resync of recovery catalog
RMAN-08005: partial resync complete

RMAN-03022: compiling command: host
% ls df.3
df.3
% exit
exit
RMAN-06134: host command complete

RMAN-03022: compiling command: release
RMAN-03023: executing command: release
RMAN-08031: released channel: c1

関連項目：リストおよびレポートの作成方法は、第 7 章の「[Recovery Manager によるリストおよびレポートの生成](#)」を参照してください。

要件

- **list** は、RMAN プロンプトでのみ実行できます。
- **incarnation** オプションには、リカバリ・カタログの使用が必要です。
- ターゲット・データベースに接続してあること。リカバリ・カタログを使用する場合は、それに接続してあることも必要です。
- **list** コマンドは、**deleted** の状態のレコードを表示しません。**deleted** の状態のレコードを表示するには、リカバリ・カタログ・ビューに問い合わせます。第 12 章の「[リカバリ・カタログ・ビュー](#)」を参照してください。

キーワードおよびパラメータ

incarnation	<p>データベースのインカーネーションに関する情報を表示します。list incarnation の出力テーブルの列見出しの説明は、表 11-8 を参照してください。</p> <p>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のインカーネーション・レコードの主キーが含まれます。RMAN 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、reset database コマンドでこのキーを使用します。of database オプションを指定しなかった場合、コマンドはリカバリ・カタログに登録したすべてのデータベースをリストします。</p> <p>of database データベースの名前を指定します。 <i>database_name</i></p>
copy	<p>データ・ファイルのコピー、アーカイブ REDO ログおよびアーカイブ REDO ログのイメージ・コピーに関する情報を表示します。デフォルトでは、list はデータベースのすべてのファイルのコピーを表示します。使用可能なイメージ・コピーと使用不能イメージ・コピーばかりでなく、復元できないもの、あるいは f のもの、さらには使用不可のものも出力に含まれます。</p> <p>関連項目：list copy 出力テーブルの列見出しの説明は、表 11-6 および表 11-7 を参照してください。</p>
backup	<p>バックアップ・セット、バックアップ・ピースおよびプロキシ・コピーに関する情報を表示します。出力には、各々の一意キーが表示されます。デフォルトでは、全データベース・バックアップがリストされます。出力に含まれるバックアップには、使用可能のものと使用不能のもの、さらに、復元できないもの、期限切れのものまたは使用不能のものもあります。また、親の全バックアップまたはコピーが既にないために復元できない増分バックアップも出力に含まれます。</p> <p>関連項目：list backup 出力テーブルの列見出しの説明は、表 11-2、表 11-3、表 11-4、および表 11-5 を参照してください。出力の KEY 列を使用して、change および delete expired backupset コマンドで利用できる主キーを取得できます。</p>

of <i>listObjList</i>	操作するオブジェクトのリストを <i>listObjList</i> 句で指定したオブジェクト・タイプのものに限定します。11-85 ページの「 <i>listObjList</i> 」を参照してください。オブジェクトを指定しないと、 list はデフォルトの of database になります。
<i>completedTimeSpec</i>	バックアップ完了のための時刻範囲を指定します。11-42 ページの「 <i>completedTimeSpec</i> 」を参照してください。
tag <i>tag_name</i>	コピーまたはバックアップのタグを指定して、データ・ファイルのコピーおよびバックアップを限定します。 tag を指定した場合、指定したタグのコピーまたはバックアップのみがリストされます。
recoverable	<p>使用可能であり、復元操作で使用できると思われる、データ・ファイルのバックアップまたはコピーのみを指定します。復元操作に使用できるようにするには、バックアップは次の 2 つの基準を満たす必要があります。</p> <ul style="list-style-type: none">■ 増分バックアップの場合、この増分バックアップを適用できる、有効な親バックアップが存在する必要があります。■ バックアップが以前のインカーネーションの場合、そのインカーネーション内のファイルに、その後変更が行われていないことが必要です。言いかえると、ファイルはオフラインである必要があります、そのインカーネーション内で再びオンラインになっていないことが必要です。
<i>untilClause</i>	終了時刻、SCN またはログ順序番号を指定します。11-147 ページの「 <i>untilClause</i> 」を参照してください。
device type <i>deviceSpecifier</i>	指定のデバイス・タイプのいずれかにあるバックアップ・セットのみをリストします (11-66 ページの「 <i>deviceSpecifier</i> 」を参照してください。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されます。このオプションは、 list backupset コマンドのみに適用されます。
like <i>string_pattern</i>	ファイル名パターンを指定して、データ・ファイル・コピーおよび REDO ログを限定します。パターンには、Oracle のパターン照合文字 '%' および '_' が使用できます。RMAN は、パターンに一致する名前のファイルのみをリストします。

List Output

表 11-2 に、出力に表示される状態情報を示します。

表 11-2 バックアップ・セットのリスト

列	意味
KEY	<p>このバックアップを識別する一意のキー。</p> <p>注意: リカバリ・カタログのかわりにターゲット・データベース制御ファイルを使用する場合、このフィールドはターゲット・データベース制御ファイルにおいてこのバックアップ・セットを指定する一意の識別子になります (またリカバリ・カタログを使用しないときに識別子の役割をする RECID と同等です)。このキーを change... backupset 文で使用し、バックアップ・セットの状態を変更します。</p>

表 11-2 バックアップ・セットのリスト

RECID	STAMP 列と結合することで、ターゲット・データベース制御ファイルでこのバックアップ・セットを識別する一意のキーとなります。新規制御ファイル・レコードが旧レコードが使用していたスペースを使用している場合、RECID は無効になります。したがって、 resync コマンドを頻繁に発行して、新規レコードをできるだけ早くリカバリ・カタログにコピーしてください。
STAMP	RECID 列と結合すると、ターゲット・データベース制御ファイルでこのバックアップ・セットを識別する一意のキーとなります。
LV	バックアップのレベルを表します。非増分の場合は NULL、増分の場合はレベル 0-4 です。
SET STAMP	<p>SET COUNT 列と結合すると、ターゲット・データベース制御ファイルでこのバックアップ・セットを識別する一意のキーとなります。V\$BACKUP_SET、V\$BACKUP_PIECE、V\$BACKUP_DATAFILE および V\$BACKUP_REDOLOG ビューで制御ファイル・レコードにアクセスするために、これらの値を使用します。</p> <p>SET STAMP の値は、制御ファイルでも（リカバリ・カタログを使用しないとき）、リカバリ・カタログを使用するときでも、有効です。SET STAMP の値は、2 値キーの一部であるため、ユーザーが入力することはありません。オラクル社カスタマ・サポートは、リカバリ・カタログが存在せず、制御ファイル・レコードも消失している状態で、ユーザーのデータベースの復元が必要となった場合に、この値を必要とすることがあります。</p> <p>関連項目: データ・ディクショナリ・ビューの詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。</p>
SET COUNT	<p>SET STAMP 列と結合すると、ターゲット・データベース制御ファイルでこのバックアップ・セットを識別する一意のキーとなります。V\$BACKUP_SET、V\$BACKUP_PIECE、V\$BACKUP_DATAFILE および V\$BACKUP_REDOLOG ビューで制御ファイル・レコードにアクセスするために、これらの値を使用します。</p> <p>SET COUNT の値は、制御ファイルでも（リカバリ・カタログを使用しないとき）、リカバリ・カタログを使用するときでも、常に有効です。SET COUNT の値は、2 値キーの一部であるため、ユーザーが入力することはありません。オラクル社カスタマ・サポートは、リカバリ・カタログが存在せず、制御ファイル・レコードも消失している状態で、ユーザーのデータベースの復元が必要となった場合に、この値を必要とします。</p>
COMPLETION TIME	バックアップ・セットをとった日付と時刻。このフィールドの書式は、NLS_LANG および NLS_DATE_FORMAT の環境設定によって変わります。

表 11-3 バックアップ・ピースのリスト

列	意味
KEY	リカバリ・カタログまたはターゲット・データベース制御ファイルにあるバックアップ・ピースの一意の識別子。 注意 : リカバリ・カタログおよび制御ファイルにある KEY の値は異なります。
PC#	バックアップ・セットにあるバックアップ・ピースのピース番号。
CP#	削除済みのバックアップにあるバックアップ・ピースのコピー番号。たとえば、 set duplex = 4 の場合、CP# は 1 から 4 の範囲になります。 注意 : バックアップが二重化されていない場合、CP# は 1 です。
STATUS	バックアップ・ピースの状態で、次のとおり。使用可能、使用不能、または期限切れ（各状態の説明は、 change コマンドを参照）。
COMPLETION TIME	ピースを作成した日付と時刻。
PIECE NAME	バックアップ・ピースの名前。

表 11-4 制御ファイルに組み込んだもの

列	意味
CKP SCN	バックアップ制御ファイルのチェックポイント SCN。指定した SCN より前に REDO レコードに記録されたデータベース変更はすべて、制御ファイルに反映されます。
CKP TIME	バックアップ制御ファイルのチェックポイントの時刻。指定した時刻より前に REDO レコードに記録されたデータベース変更はすべて、制御ファイルに反映されます。

表 11-5 データ・ファイルが組み込んだもののリスト

列	意味
FILE	バックアップされたファイルの数。
NAME	ファイルがこのバックアップ・セットから復元され、かつ set newname コマンド（11-134 ページの「 set_run_option 」を参照）がなかった場合に、このファイルが復元される場所。
LV	バックアップのレベル。非増分の場合は NULL、増分の場合はレベル 0 - 4。
TYPE	バックアップが FULL か INCR（増分）かを示す。

表 11-5 データ・ファイルが組み込んだもののリスト

CKP SCN	データ・ファイルをバックアップした時点のデータ・ファイルのチェックポイント。SCN より前のデータベース変更はすべて、ファイルに書き込まれます。指定した SCN 以降の変更はファイルに書き込まれていません。
CKP TIME	データ・ファイルをバックアップした時点のデータ・ファイルのチェックポイント。時刻より前のデータベース変更はすべて、ファイルに書き込まれます。指定した時刻以降の変更はファイルに書き込まれていません。

表 11-6 データ・ファイル・コピーのリスト

列	意味
KEY	データ・ファイル・コピーを指す一意の識別子。 change コマンドでこの値を使用して、データ・ファイル・コピーの状態を変更します。 注意: リカバリ・カタログおよび制御ファイルにある KEY の値は異なります。
FILE	コピー元のデータ・ファイルのファイル番号。
S	バックアップ・ピースの状態で、次のとおり。使用可能、使用不能、または期限切れ（各状態の説明は、 change コマンドを参照）。
COMPLETION TIME	コピーを取った日付と時刻。このフィールドの書式は、NLS_LANG および NLS_DATE_FORMAT の環境変数によって変わります。
CKP SCN	データ・ファイルのコピーを取った時点のデータ・ファイルのチェックポイント。SCN より前のデータベース変更は、すべてこのファイルに書き込まれます。
CKP TIME	データ・ファイルのコピーを取った時点のデータ・ファイルのチェックポイント。この時刻より前のデータベース変更は、すべてこのファイルに書き込まれます。
NAME	データ・ファイル・コピーのファイル名。

表 11-7 アーカイブ・ログ・コピーのリスト

列	意味
KEY	アーカイブ REDO ログ・コピーの一意の識別子。change コマンドでこの値を使用して、コピーの状態を変更します。 注意 ：リカバリ・カタログおよび制御ファイルにある KEY の値は異なります。
THRD	REDO ログのスレッド番号。
SEQ	ログ順序番号。
COMPLETION TIME	コピーを取った日付と時刻。このフィールドの書式は、NLS_LANG および NLS_DATE_FORMAT の環境変数によって変わります。
NAME	アーカイブ REDO ログ・コピーのファイル名。

表 11-8 データベースのインカーネーションのリスト

列	意味
DB KEY	INC KEY と結合して、RMAN がリカバリ・カタログのデータベースのインカーネーションを識別する一意のキー。このキーを使用してデータベースの登録解除をします。すなわち、そのデータベースに関連するすべての行をリカバリ・カタログから削除します。
INC KEY	DB KEY と結合して、RMAN がリカバリ・カタログのデータベースのインカーネーションを識別する一意のキー。最新の RESETLOGS 以前の時点にデータベースを回復する必要がある場合に使用する reset database to incarnation コマンドでこのキーを使用します。
DB NAME	DB_NAME パラメータにリストされたデータベース名。
DB ID	データベース作成のとき Oracle が自動的に生成するデータベース識別番号。
CUR	インカーネーションがデータベースの現行インカーネーションであるかどうかを示す。
RESET SCN	インカーネーションが作成されたときの SCN。
RESET TIME	インカーネーションが作成されたときの時刻。

例

コピーのリスト作成 次の例では、リカバリ・カタログに記録されたデータ・ファイル・コピーおよびアーカイブ REDO ログをリストしています。

```
list copy of database archivelog all;
```

List of Datafile Copies

Key	File S	Completion time	Ckp SCN	Ckp time	Name
1262	1	A 18-AUG-98	219859	14-AUG-98	/vobs/oracle/dbs/copy/tbs_01.f

List of Archived Log Copies

Key	Thrd	Seq	S Completion time	Name
789	1	1	A 14-JUL-98	/vobs/oracle/work/arc_dest/arcr_1_1.arc
790	1	2	A 11-AUG-98	/vobs/oracle/work/arc_dest/arcr_1_2.arc
791	1	3	A 12-AUG-98	/vobs/oracle/work/arc_dest/arcr_1_3.arc

バックアップのリスト作成 次の例では、リカバリ・カタログに記録された2つのデータ・ファイルのバックアップをリストしています。

```
list backup of datafile '/oracle/dbs/tbs_01.f', '/oracle/dbs/tbs_02.f';
```

List of Backup Sets

Key	Recid	Stamp	LV Set Stamp	Set Count	Completion Time
1174	12	341344528	0 341344502	16	14-SEP-98

List of Backup Pieces

Key	Pc#	Cp#	Status	Completion Time	Piece Name
1176	1	1	AVAILABLE	14-AUG-98	/vobs/oracle/dbs/0ga5h07m_1_1

Controlfile Included

Ckp SCN	Ckp time
219857	14-AUG-98

List of Datafiles Included

File Name	LV Type	Ckp SCN	Ckp Time
1 /oracle/dbs/tbs_01.f	0 Full	199843	14-AUG-98
2 /oracle/dbs/tbs_02.f	0 Full	199843	14-AUG-98

データベースのインカーネーションのリスト作成 次の例では、リカバリ・カタログに記録されたすべてのデータベースのインカーネーションをリストしています。

```
list incarnation;
```

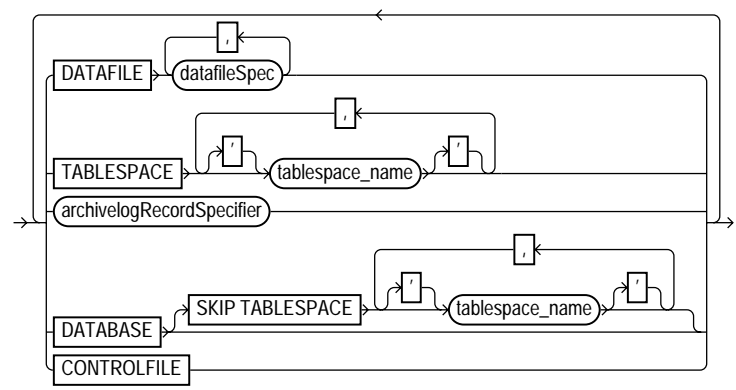
List of Database Incarnations						
DB Key	Inc Key	DB Name	DB ID	CUR	Reset SCN	Reset Time
-----	-----	-----	-----	---	-----	-----
1	2	PROD1	1224038686	NO	1	02-JUL-98
1	582	PROD1	1224038686	YES	59727	10-JUL-98

関連項目

- 11-57 ページの「[crosscheck](#)」
- 11-85 ページの「[listObjList](#)」
- 11-103 ページの「[report](#)」
- 11-151 ページの「[validate](#)」

listObjList

構文



目的

データベース・ファイルおよび REDO ログの指定に使用します。

要件

この句は、次のコマンドで使用します。

- [list](#)
- [crosscheck](#)
- [deleteExpired](#)

キーワードおよびパラメータ

datafile <i>datafileSpec</i>	ファイル番号に対応するファイル名でデータ・ファイルを指定します。この句では、データ・ファイルの少なくとも1つが含まれているデータ・ファイル・イメージ・コピーまたはバックアップ・セットを指定します。11-60 ページの「 datafileSpec 」を参照してください。
tablespace <i>tablespace_name</i>	表領域名を指定します。この句では、指定した表領域からのデータ・ファイルの少なくとも1つが含まれているデータ・ファイル・イメージ・コピーまたはバックアップ・セットを指定します。

<i>archivelogRecord-Specifier</i>	アーカイブ REDO ログの範囲を指定します。11-17 ページの「 archivelogRecordSpecifier 」を参照してください。
database	現行データベースのすべてのファイルのバックアップ・セットまたはイメージ・コピーを指定します。 skip tablespace database 指定から、指定した表領域を除外します。 <i>tablespace_name</i>
controlfile	現行の制御ファイルを指定します。

例

データ・ファイル・コピーのリスト作成 次のコマンドは、データベースにあるファイルすべてのイメージ・コピーを、TEMP 表領域をスキップして、リストします。

```
list copy of database skip tablespace temp;
```

アーカイブ REDO ログのクロスチェック 次の例では、最近 90 日間に作成したアーカイブ REDO ログ・バックアップ・セットの状態について、メディア・マネージャに問い合わせます。

```
allocate channel for maintenance type 'sbt_tape';
crosscheck
  backup
    of archivelog
    from time 'SYSDATE-90';
release channel;
```

期限切れになった制御ファイル・バックアップ・セットの削除 次の例では、制御ファイルの期限切れになったバックアップを削除します。

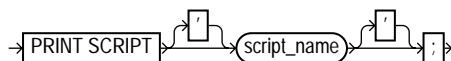
```
delete expired backup of controlfile;
```

関連項目

- 11-17 ページの「[archivelogRecordSpecifier](#)」
- 11-57 ページの「[crosscheck](#)」
- 11-60 ページの「[datafileSpec](#)」
- 11-63 ページの「[deleteExpired](#)」
- 11-76 ページの「[list](#)」

printScript

構文



目的

ストアド・スクリプトを標準出力または RMAN メッセージ・ログに出力します。コマンド行にログ・ファイル名と **log** 引数を指定します。この引数を指定しないと、Recovery Manager はメッセージ出力を標準出力に書き出します。

注意： RC_STORED_SCRIPT_LINE リカバリ・カタログ・ビューへの問合せによっても、ストアド・スクリプトの各行の表示ができます。

要件

- このコマンドは、RMAN プロンプトで実行してください。
- リカバリ・カタログを使用すること。

キーワードおよびパラメータ

<i>script_name</i>	指定した名前のストアド・スクリプトを標準出力またはメッセージ・ログに出力します。すべてのストアド・スクリプトのリストを取得するには、SQL*Plus を使用してカタログ所有者としてリカバリ・カタログに接続し、次の問合せを発行します。 <pre>select * from rc_stored_script;</pre> 注意： スタード・スクリプトを実行するには、 run コマンドの中カッコの内の execute script を使用します。 関連項目： RC_STORED_SCRIPT に関しては、12-26 ページの「 RC_STORED_SCRIPT 」を参照してください。
--------------------	---

例

スクリプトのメッセージ・ログへの出力 この例では、ターゲット・データベース PROD1 およびリカバリ・カタログ・データベース RCAT に接続し、RMAN ログをメッセージ・ロ

グ・ファイルに出力するよう指示しています。その後で backup_db スクリプトを作成し、それを rman_log に出力しています。最後にスクリプトを実行します。

```
rman target sys/change_on_install@prod1 catalog rman/rman@rcat log rman_log
create script backup_db {
    allocate channel d1 type disk;
    backup database;
}
print script backup_db;

run{ execute script backup_db};
```

スクリプトの画面への表示 この例では、ストアド・スクリプトを画面に表示します。

```
print script tbs1_b;

RMAN-03027: printing stored script: tbs1_b
{
allocate channel ch1 type disk;
backup tablespace tbs1;
}
```

関連項目

11-39 ページの「[cmdLine](#)」

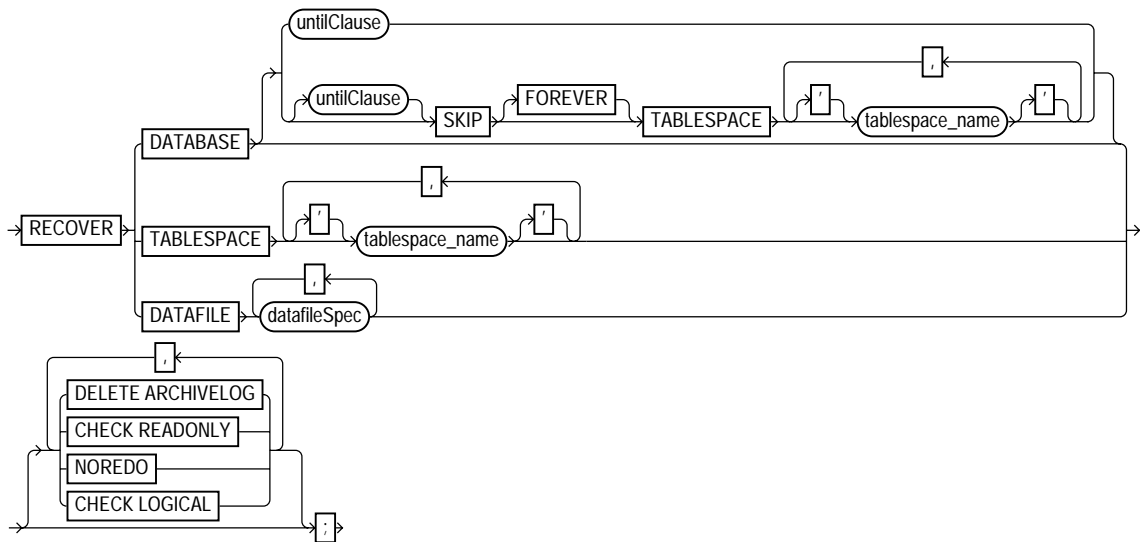
11-54 ページの「[createScript](#)」

11-65 ページの「[deleteScript](#)」

11-125 ページの「[run](#)」

recover

構文



目的

1 つ以上の復元したデータ・ファイルに、REDO ログまたは増分バックアップを適用して、その内容を特定時点の状態に更新します。

RMAN は、オンライン REDO ログを使用して、メディア回復の実行に必要なだけのアーカイブ REDO ログのバックアップ・セットを復元します。RMAN が増分バックアップの適用か REDO の適用かを選択できる場合は、常に増分バックアップを選択します。オーバーラップしたレベルの増分バックアップが使用できる場合、RMAN は自動的に最長期間にわたる増分バックアップを選択します。

関連項目 : データ・ファイルの回復方法は、[第 9 章の「Recovery Manager による復元と回復」](#)を参照してください。

要件

- このコマンドの実行は、**run** コマンドの中カッコ内でのみ実行すること。
- リカバリ・カタログなしでこのコマンドを使用することができますが、これは制御ファイルの回復が不必要な場合に限りです。RMAN は、リカバリ・カタログかターゲット・データベース制御ファイルのどちらかがなければ操作できません。
- 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・データベースをマウントしておいてください。そのデータベースがオープンされている場合は、回復するデータ・ファイルまたは表領域はオフラインにしてください。データベースの回復の場合は、そのデータベースをマウントしておく必要がありますが、オープンしてあってはなりません。
- アーカイブ REDO ログまたは増分データ・ファイル・バックアップ・セットを回復する必要がある場合は、**recover** に先行して、**allocate channel** コマンドを少なくとも 1 回実行しておく必要があります。
- 復元が必要なバックアップに適切なタイプのデバイスを割り当ててください。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。
- 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・ファイルのみです。
- 特定の時点まで回復する場合、**restore** および **recover** コマンドの両方の前に、**set until** コマンドの入力が必要です。**restore** の後で **recover** の前に **set until** コマンドを指定した場合、復元されたファイルのタイムスタンプは指定した時刻の後になるので、必要な時刻までのデータベースへのメディア回復は実行できない場合があります。
- **recover database** コマンドは、ファイルが回復される時点で NORMAL モードでオフラインにされているまたは読取り専用であるファイルの復元はしません。RMAN は、NORMAL モードでオフラインにされたファイルをそれ以上のチェックはしないで、除外します。**check readonly** の指定がある場合、RMAN はディスク上の各読取り専用ファイルをチェックして、指定した時点ですでに現行になっていることを確認します。**check readonly** の指定がなければ、RMAN は読取り専用ファイルを処理しません。
- 不完全回復またはバックアップ制御ファイルを使用した回復の後には、RESETLOGS オプションでオープンしてください。

キーワードおよびパラメータ

database	データベース全体の回復を指定します。オプションの <i>untilClause</i> を指定して、指定の条件に達したときに回復を停止させることができます。
untilClause	recover に対して現行以外の時刻、SCN またはログ順序番号を指定します。11-147 ページの「 <i>untilClause</i> 」を参照してください。
skip [forever] tablespace tablespace_name	<p>回復の必要がない表領域をリストします。一時データのみがある表領域の回復を行わないようにしたり、いくつかの表領域の回復を延期したりするのに役立ちます。skip 句は、メディア回復開始前に、指定された表領域にあるデータ・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。</p> <p>不完全回復を実行する場合、skip は使用できません。そのかわりに、skip forever を使用することによって、RESETLOGS オプションによってデータベースをオープンした後スキップした表領域を削除するようにします。skip forever 句を使用すると、RMAN は、DROP オプションを使用してデータ・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。</p>
tablespace tablespace_name	表領域名によって表領域を指定します。
datafile datafileSpec	<p>回復する 1 つ以上のデータ・ファイルをリストして指定します。引用符付き文字列を使用したファイル名または <i>integer</i> を使用した絶対データ・ファイル番号で、データ・ファイルを指定します (11-60 ページの「<i>datafileSpec</i>」を参照)。</p> <p>RMAN メタデータのリポジトリとして制御ファイルのみを使用している場合、ファイル名は制御ファイルで認識されているデータ・ファイルの名前にする必要があります。</p> <p>リカバリ・カタログを使用している場合、データ・ファイルのファイル名はカタログに記録された最新の名前にする必要があります。たとえば、データ・ファイルが制御ファイルにおいて改名されたとします。このとき、データベースがカタログを再同期化する前にクラッシュしたとします。この場合 recover コマンドではデータ・ファイルの古い名前を指定してください。これはこの名前がカタログに記録されている名前だからです。</p>
delete archivelog	不要になった復元済みのアーカイブ・ログを削除します。RMAN は、 restore コマンド開始前にすでにディスク上に存在したアーカイブ・ログは削除しません。
check readonly	読取り専用ファイルのヘッダーをチェックして、それらを回復から除外する前に、現行であることを確認します。
noredo	REDO ログの適用を抑止します。増分バックアップのみを適用します。このオプションは、増分バックアップを使用して NOARCHIVELOG モードで運用されているデータベースを回復するためのものです。NOARCHIVELOG モードで運用されているデータベースの回復のときに noredo を指定しないと、Oracle は回復を完了できず、エラー・メッセージを発行します。

例

オープン状態のデータベースでの表領域の回復 次の例では、表領域 TBS_1 をオフラインにし、それを復元および回復してから、オンラインに戻しています。

```
run {
    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";
}
```

新しい場所に復元したデータ・ファイルの回復 次の例では、ディスク上のデータ・ファイル・コピーおよびテープのバックアップを使用するために1つのディスク・チャンネルと1つのメディア管理チャンネルを割り当て、表領域 TBS_1 にあるデータ・ファイルの1つを別の場所に復元します。

```
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";
}
```

バックアップ制御ファイルを使用しての不完全回復の実行 ディスク・クラッシュのためにデータベースおよびアーカイブ REDO ログ 1234 の両方が消失したとします。増分バックアップを取っていないので、使用可能なアーカイブ REDO ログでデータベースの回復が必要です。表領域 READONLY1 はログ 1234 以降に変更がなかったなので、回復は必要ありません。

```
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' ;
    # Because you specified a restore destination, you must manually replicate the
    # control file. The restore command replicates automatically when no destination is
    # specified.
    replicate controlfile from '/vobs/oracle/dbs/cf1.f';
    alter database mount;
    restore database skip tablespace temp1, readonly1;
```

```
recover database skip forever 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;  
}
```

関連項目

- 11-9 ページの「[allocate](#)」
- 11-134 ページの「[set_run_option](#)」
- 11-113 ページの「[restore](#)」
- 11-147 ページの「[untilClause](#)」

register

構文

```
→ REGISTER DATABASE > ;
```

目的

ターゲット・データベースをリカバリ・カタログに登録し、RMAN がアクセスできるようにします。RMAN は、ターゲット・データベースの登録に必要な情報をターゲット・データベース自体から取得します。

要件

- このコマンドは、RMAN プロンプトで実行してください。
- データベースは 1 回のみ登録できます。
- リカバリ・カタログを使用すること。
- ターゲット・データベースはマウントしてある必要があります。
- RMAN が同一のデータベース識別子を検出すると、**register database** コマンドはエラーになります。この状況になるのは、データベースを作成するのに **duplicate** コマンドを使用しないで既存のデータベースからファイルをコピーしたときです。

この障害が発生した場合、他のユーザーのスキーマで、別のユーザー ID を使用して **create catalog** (11-52 ページの「[createCatalog](#)」を参照) を実行することにより、別のリカバリ・カタログを作成できます。その後で、新しいスキーマにある、新たに作成されたリカバリ・カタログに同一のデータベース識別子を付けたデータベースを登録します。

注意： 同じデータベース名と識別子を持つ異なるターゲット・データベースを RMAN で使用する場合は、RMAN 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。

例

データベースの登録 この例では、ターゲット・データベースを登録し、既存のデータ・ファイル・コピーをカタログし、次にデータベースを使用するためにオープンします。

```
connect target / catalog rman/rman@rcat;  
startup mount;  
register database;  
catalog datafilecopy '/vobs/oracle/dbs/foo.f';  
sql 'alter database open';
```

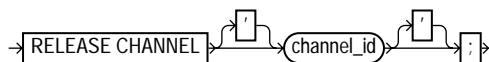
関連項目

11-32 ページの「[catalog](#)」

11-69 ページの「[duplicate](#)」

release

構文



目的

ターゲット・データベース・インスタンスへ接続した状態で、チャンネルを解放します。
allocate channel コマンドで使用した識別子と同じ識別子を付けて、チャンネル名を指定してください。このコマンドはオプションになっています。RMAN は、**run** コマンドが終了すると、割り当てられていたすべてのチャンネルを自動的に解放します。

要件

このコマンドの実行は、**run** コマンドの中カッコ内でのみ実行すること。

キーワードおよびパラメータ

<i>channel_id</i>	allocate channel コマンドで使用したチャンネル ID を指定します。
-------------------	---

例

チャンネルの解放 この例では、テープに datafile1 の複製バックアップ・セットを 3 組作成し、次にテープ・チャンネルを解放しています。RMAN は、次にディスクに datafile 2 の複製バックアップを 3 組作成し、次にディスク・チャンネルを解放します。

```
run {
  set duplex=3;
  allocate channel ch1 type 'SBT_TAPE';
  allocate channel ch2 type disk;
  backup channel ch1 datafile 1;
  release channel ch1;
  backup datafile 2;
}
```

関連項目

11-9 ページの「[allocate](#)」

releaseForMaint

構文

```
→ RELEASE CHANNEL > >
```

目的

allocate channel コマンドの **for delete** または **for maintenance** オプションで指定された順次 I/O デバイスを解放すること。メンテナンス・チャンネルは **run** コマンドのなかで発行される **allocate channel** および **release channel** コマンドに影響されないという点に注意してください。

要件

- このコマンドは、RMAN プロンプトのみで実行してください。
- メンテナンス・チャンネルを解放するためには、それが割当て済みであることが必要です。

例

削除操作後のメンテナンス・チャンネルの解放 この例では、メディア・マネージャにメンテナンス・チャンネルを割り当て、次に解放します。

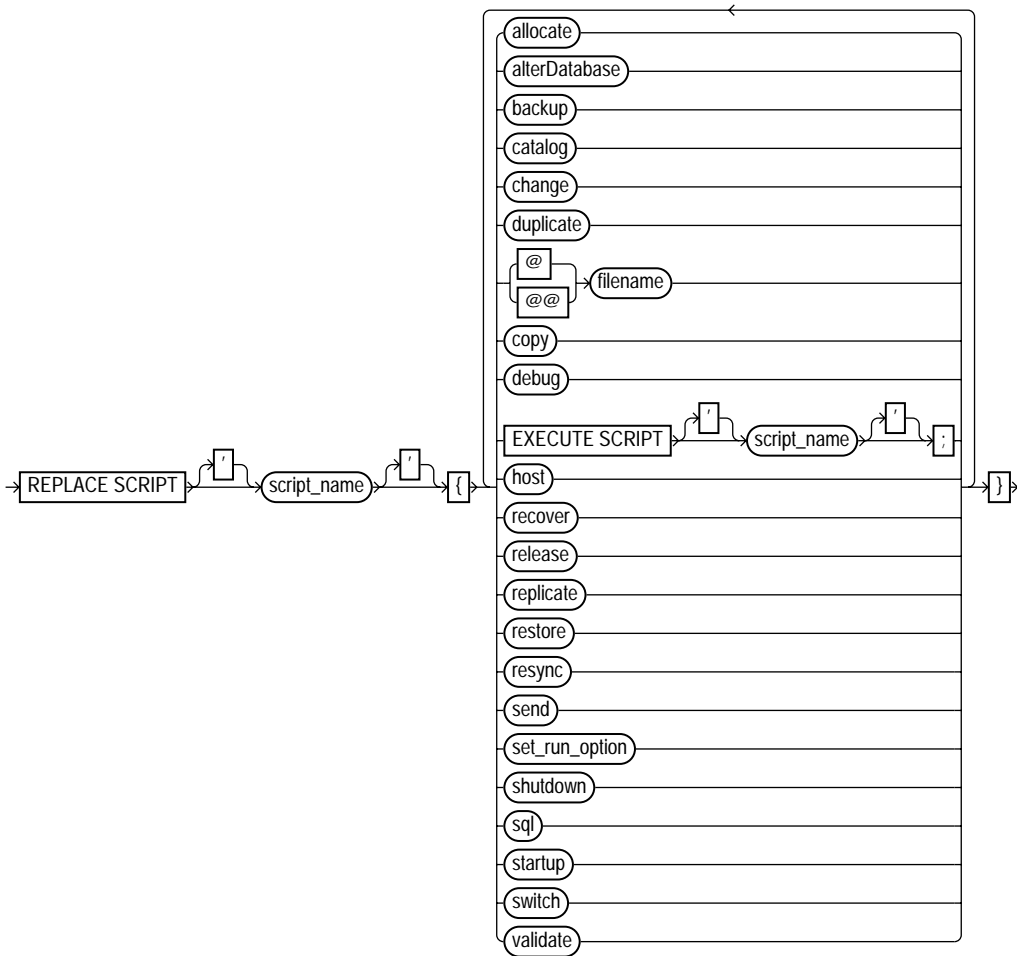
```
allocate channel for delete type 'sbt_tape';
change backuppiece 100 delete;
run {
    allocate channel ch1 type disk;
    backup datafile 1;
    release channel ch1; # releases run channel
}
release channel; # releases maintenance channel
```

関連項目

11-13 ページの「[allocateForMaint](#)」

replaceScript

構文



目的

リカバリ・カタログに格納されている既存のスクリプトを置換します。既存のものがいない場合、**replace script** によりスクリプトが作成されます。

ストアド・スクリプト機能は、主として、実行頻度が高い RMAN コマンドを収集した共通リポジトリを提供するためのものです。スクリプトには、**run** コマンドで有効なコマンドを使用してください。スクリプトは、そのままでは実行されません。スクリプトを実行するには、**execute script** コマンド (11-125 ページの「**run**」を参照) を使用します。

要件

- **replace script** は、RMAN プロンプトでのみ実行すること。
- リカバリ・カタログを使用すること。

キーワードおよびパラメータ

ストアド・スクリプトで使える各コマンドの説明は、11-21 ページの「**backup**」など、該当の項目を参照してください。@ および @@ 引数については、11-54 ページの「**createScript**」を参照してください。**execute script** コマンドについては、11-125 ページの「**run**」を参照してください。

replace script <i>script_name</i>	<p>指定したストアド・スクリプトを新しいコマンドに置き換えます。replace script 'filename' (...) コマンドのカッコ内で使用できる文は、run コマンドのなかで使用できる文と同じです。</p> <p>すべてのストアド・スクリプトのリストを取得するには、SQL*Plus を使用してリカバリ・カタログ・データベースにカタログ所有者として接続し、次の問合せを発行します。</p> <pre>select * from rc_stored_script;</pre> <p>注意: スクリプトを実行するには、run コマンドの中カッコの中で execute script を発行します。</p> <p>関連項目: RC_STORED_SCRIPT の詳細は、12-26 ページの「RC_STORED_SCRIPT」を参照してください。</p>
---	--

例

スクリプトの置換え この例では、**backup_full** という名称のスクリプトを作成し、それを別のスクリプトに置換してから、実行します。

```
create script backup_full {
  allocate channel ch1 type 'SBT_TAPE';
  allocate channel ch2 type 'SBT_TAPE';
  allocate channel ch3 type 'SBT_TAPE';
  backup database;
}
replace script backup_full {
  allocate channel ch1 type disk;
```

```
        backup database;  
    }  
run { execute script backup_full; }
```

関連項目

11-54 ページの「[createScript](#)」

11-65 ページの「[deleteScript](#)」

11-87 ページの「[printScript](#)」

11-125 ページの「[run](#)」

replicate

構文

```
→ REPLICATE CONTROLFILE FROM 'filename' ;
```

目的

制御ファイルをターゲット・データベースの CONTROL_FILES 初期化パラメータで指定した場所にコピーします。

制御ファイルを復元後、**replicate controlfile** 文を使用してデータベースをマウントする準備をします。この操作は、複数の **copy controlfile** 文を実行する場合と同じです。

注意： **restore** コマンドは、復元先の指定がない場合、すべての CONTROL_FILES の場所に制御ファイルを自動的にレプリケートします。

要件

- **replicate controlfile** は、**run** コマンドの中カッコ内のみで実行してください。
- **type disk** オプションを指定した **allocate channel** 文を **replicate controlfile** 文の前に少なくとも 1 回は実行する必要があります。

キーワードおよびパラメータ

<i>'filename'</i>	レプリケートされる制御ファイルの場所を指定します。たとえば、制御ファイルのバックアップを /oracle/temp/cf.bak に復元する場合、 replicate コマンドにもこのファイル名を指定します。
-------------------	---

例

復元された制御ファイルのレプリケート この例では、制御ファイルを復元し、次にそれをレプリケートします。

```
startup nomount;
run {
    set until time 'Jun 18 1998 16:32:36';
    allocate channel chl type disk;
    # restore a backup controlfile to a temporary location.
    restore controlfile to '/tmp/cf.tmp';
    replicate controlfile from '/tmp/cf.tmp';
    startup force mount;
}
```

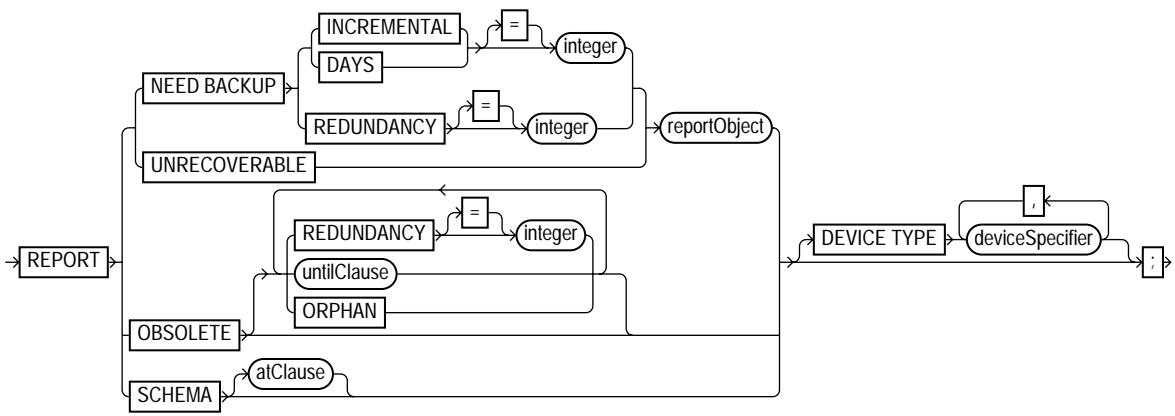
関連項目

11-48 ページの「[copy](#)」

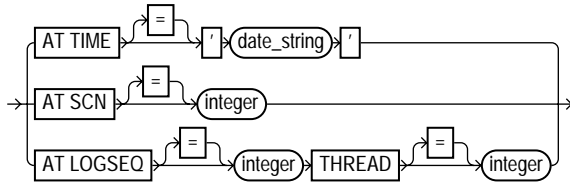
11-113 ページの「[restore](#)」

report

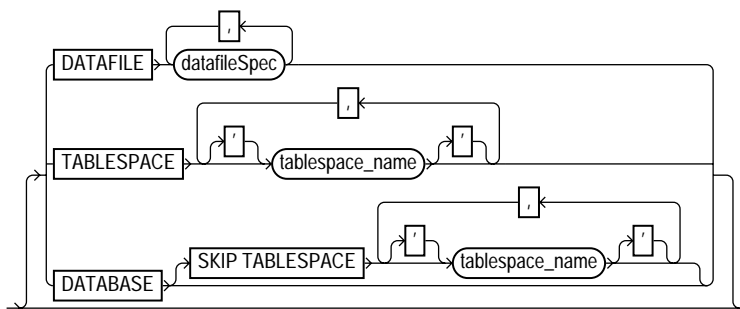
構文



atClause::=



reportObject::=



目的

RMAN メタデータの詳細分析を実行します。Oracle では、**report** コマンドの出力を標準出力またはメッセージ・ログ・ファイルに書き出します（11-44 ページの「[connect](#)」）を参照してください。

次のような質問に回答するには、**report** を使用してください。

- 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取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？

要件

- このコマンドは、RMAN プロンプトのみで実行してください。
- **report schema** コマンドを **at time**、**at scn** または **at logseq** オプションとともに発行する場合は、リカバリ・カタログの使用が必要です。それらのオプションがない場合は、**report** コマンドにリカバリ・カタログは必要ありません。

キーワードおよびパラメータ

need backup	新規バックアップが必要なすべてのデータ・ファイルをリストします。レポートでは、復元操作にユーザーが最新のバックアップを使用することを前提としています。
incremental <i>integer</i>	<p>増分バックアップのしきい値を指定します。指定した数以上の増分バックアップがデータ・ファイルの完全回復に必要な場合、データ・ファイルには新規の全体バックアップが必要になります。report コマンドは、recover コマンドと同様に、選択ができる場合には一番低いレベルの増分バックアップを使用します。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 RMAN が使用する方法と同じです。</p> <p>注意: 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。それらのファイルを表示するには、report need backup redundancy コマンドを使用してください。</p>
days <i>integer</i>	<p>このファイルの回復時に適用する必要がある REDO ログ・ファイルの日数のしきい値を指定します。たとえば、report need backup days 7 database では、その回復に 1 週間分以上のアーカイブ REDO ログが必要なデータ・ファイルを示しています。</p> <p>ターゲット・データベースの制御ファイルがマウントされていて現行である場合、RMAN はこのレポートに次のような最適化を行います。</p> <ul style="list-style-type: none"> ■ 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。 ■ 以前はオフラインであ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過している場合に限りレポートに含まれます。
redundancy <i>integer</i>	<p>データ・ファイルをバックアップが必要ない範囲にあるとみなすために必要なバックアップまたはコピーの最小数を指定します。つまり、このファイルに <i>integer</i> 以下のバックアップまたはコピーしか存在しない場合にデータ・ファイルのバックアップが必要になります。たとえば、redundancy 2 では、データ・ファイルのバックアップまたはコピーが 2 つ未満の場合は新しいバックアップが必要になることを意味します。</p>
unrecoverable	<p>すべての回復不能なデータ・ファイルをリストします。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れば、そのデータ・ファイルは回復不能とみなされます。</p> <p>注意: 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理由にはなりません。このようなデータ・ファイルは、ファイルの作成時以降の REDO ログがまだ存在していれば、CREATE DATAFILE コマンドを使用して回復できるからです。</p>
reportObject 句	レポートに組み込むデータ・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。

	datafile <i>datafileSpec</i>	指定したデータ・ファイルをリストします。RMAN は、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またはデータ・ファイル・コピーについてレポートを作ります。
	tablespace <i>tablespace_name</i>	指定した表領域にあるデータ・ファイルをリストします。RMAN は、指定した表領域にあるデータ・ファイルを少なくとも 1 つ含むバックアップまたはデータ・ファイル・コピーについてレポートを作ります。
	database	現行データベースにあるすべてのファイルのバックアップまたはデータ・ファイル・コピーをリストします。
	skip tablespace <i>tablespace_name</i>	database 指定から、指定した表領域を除外します。
obsolete		RMAN メタデータに記録されている全体バックアップまたはデータ・ファイル・コピーのうち、必要がなくなったため削除できるものをリストします。これ以外のパラメータを指定しないと、 redundancy はデフォルトの 1 になります。このオプションを device type と一緒に使用すると、RMAN は指定したタイプのバックアップおよびコピーのみを行います。
	redundancy <i>integer</i>	バックアップまたはコピーを不要とみなすために必要な最小レベルの冗長性を指定します。データ・ファイル・コピーが不要とみなされるのは、このファイルに少なくとも <i>integer</i> で指定した数以上の最近のバックアップまたはイメージ・コピーがある場合です。データ・ファイルのバックアップ・セットは、バックアップ・セットに含まれる各ファイルに少なくとも <i>integer</i> で指定した数以上の最近のバックアップまたはイメージ・コピーが存在する場合不要とみなされます。たとえば、 redundancy 2 は、データ・ファイルのバックアップまたはコピーに少なくともさらに 2 つ以上最近のものが存在しないと他のバックアップまたはコピーを不要とすることができないことを意味します。
	<i>untilClause</i>	<p>少なくとも <i>n</i> (<i>n</i> は redundancy の値) の最近のバックアップまたはコピーが存在し、それらが指定した時刻、SCN またはログ順序番号以降の変更を含まないものでなければ、バックアップまたはコピーを不要とみなさないことを指定します。たとえば、obsolete redundancy 2 until 'SYSDATE-7' は、少なくとも 2 つの最近のバックアップまたはコピーがあり、かつそれらが 1 週間以上前にチェックポイントを取ったものである場合、そのバックアップまたはコピーは不要とされることを意味します。</p> <p>この句は、データベースが現行以外の時刻、SCN またはログ順序番号に対して回復可能にする必要がある場合に役立ちます。11-147 ページの「<i>untilClause</i>」を参照してください。</p>
	orphan	バックアップまたはコピーが属するデータベースのインカーネーションが現行のインカーネーションの直接の親ではないためにこのようなバックアップまたはコピーを使用できない場合に、それらを不要と指定します。親なしバックアップの説明は、4-22 ページの「 孤立したバックアップのレポート作成 」を参照してください。
スキーマ		指定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します。
<i>atClause</i>		時刻、SCN またはログ順序番号により特定の時点を指定します。

at time <i>date_string</i>	日付を指定します。NLS_LANG および NLS_DATE_FORMAT 環境変数で時刻の書式を指定します。
at scn <i>integer</i>	SCN を指定します。
at logseq <i>integer</i>	指定した REDO スレッドに対するログ順序番号を指定します。この整数は、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を示します。
device type <i>deviceSpecifier</i>	記憶デバイスのタイプを指定します。RMAN は、レポート用に指定したデバイスに存在するバックアップおよびコピーのみを使用可能とみなします。11-66 ページの「 deviceSpecifier 」を参照してください。

Report Output

各レポートのフィールドについて次に説明します。

表 11-9 データベース・スキーマのレポート

列	意味
FILE	絶対データ・ファイル番号。
K-BYTES	キロバイト単位のファイルのサイズ。
TABLESPACE	表領域名。
RB SEGS	表領域にロールバック・セグメントがある場合は YES、ない場合は NO（リカバリ・カタログに接続されている場合のみ）。RMAN をカタログに接続していないと、'****' が表示されます。
NAME	データ・ファイルのファイル名。

表 11-10 不要なバックアップおよびコピーのレポート

列	意味
TYPE	オブジェクトがバックアップ・セットか、バックアップ・ピースか、プロキシ・コピーか、またはデータ・ファイル・コピーかの区分。
KEY	ターゲット・データベースの制御ファイルでこのバックアップを識別する一意のキー。
COMPLETION TIME	バックアップまたはコピーを取った時刻。
FILENAME/HANDLE	バックアップまたはデータ・ファイル・コピーのファイル名またはメディア・ハンドル。

表 11-11 回復不能操作のためにバックアップが必要となったファイルのレポート

列	意味
FILE	回復不能操作のために新しいバックアップを必要とするデータ・ファイルの絶対番号。
TYPE OF BACKUP REQUIRED	FULL または INCREMENTAL。このファイルのすべてのデータの回復可能性を保証するために必要なバックアップのタイプによって決まります。FULL の場合、全体バックアップ、レベル 0 のバックアップ、またはデータ・ファイル・コピーを作成します。INCREMENTAL の場合、全体または増分バックアップです。
NAME	データ・ファイルのファイル名。

表 11-12 n 以下の冗長度のバックアップしかないファイルのレポート

列	意味
FILE	冗長度 n 以下のバックアップしかないデータ・ファイルの絶対データ・ファイル番号。
#BKPS	このファイルに対して存在するバックアップの数。
NAME	ファイルの名前。

表 11-13 n 日以上のアークাইブ・ログが回復のために必要なファイルのレポート

列	意味
FILE	n 日以上のアークাইブ REDO ログが回復のために必要であるデータ・ファイルの絶対ファイル番号。
DAYS	回復のために必要なアークাইブ REDO データの日数。
NAME	データ・ファイルの名前。

表 11-14 n 以上の増分が回復中に必要であるファイルのレポート

列	意味
FILE	n 以上の増分が完全回復のために必要なデータ・ファイルの絶対ファイル番号。
INCREMENTALS	完全回復のために必要な増分バックアップの数。
NAME	データ・ファイルの名前。

例

データベース・スキーマのレポート この例では、1週間前のデータベースにあったすべてのデータ・ファイルおよび表領域の名前をレポートします。

```
report schema at time 'SYSDATE-7';
```

Report of database schema

File	K-bytes	Tablespace	RB	segs	Name
1	47104	SYSTEM	YES		/vobs/oracle/dbs/tbs_01.f
2	978	SYSTEM	YES		/vobs/oracle/dbs/tbs_02.f
3	978	TBS_1	NO		/vobs/oracle/dbs/tbs_11.f
4	978	TBS_1	NO		/vobs/oracle/dbs/tbs_12.f
5	978	TBS_2	NO		/vobs/oracle/dbs/tbs_21.f
6	978	TBS_2	NO		/vobs/oracle/dbs/tbs_22.f
7	500	TBS_3	NO		/vobs/oracle/dbs/tbs_31.f
8	500	TBS_3	NO		/vobs/oracle/dbs/tbs_32.f
9	5120	SYSTEM	YES		/vobs/oracle/dbs/tbs_03.f

増分バックアップが必要なデータ・ファイルのレポート この例では、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5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・ファイルすべてをレポートします。

```
report need backup incremental 5 database;
```

Report of files that need more than 5 incrementals during recovery

File	Incrementals	Name
1	9	/vobs/oracle/dbs/tbs_01.f
2	9	/vobs/oracle/dbs/tbs_02.f
3	9	/vobs/oracle/dbs/tbs_11.f
4	9	/vobs/oracle/dbs/tbs_12.f
5	9	/vobs/oracle/dbs/tbs_21.f
6	9	/vobs/oracle/dbs/tbs_22.f
7	9	/vobs/oracle/dbs/tbs_23.f
8	9	/vobs/oracle/dbs/tbs_03.f

バックアップが必要なデータ・ファイルのレポート 次の例では、表領域 SYSTEM のデータ・ファイルのうち最新のバックアップで復元してから2日以上アーカイブ REDO ログを適用して回復する必要があるものすべてをレポートします。

```
report need backup days 2 tablespace system;
```

Report of files whose recovery needs more than 2 days of archived logs

File	Days	Name
1	3	/vobs/oracle/dbs/tbs_01.f

```
2      3      /vobs/oracle/dbs/tbs_02.f
16     3      /vobs/oracle/dbs/tbs_03.f
```

回復不能なデータ・ファイルのレポート 次の例では、REDO が消失したために既存のバックアップから回復できないデータ・ファイルすべてをレポートします。

```
report unrecoverable;

Report of files that need backup due to unrecoverable operations
File Type of Backup Required Name
-----
4      FULL                               /vobs/oracle/dbs/tbs_12.f
```

不要なバックアップおよびコピーのレポート 次の例では、冗長度 1 とした場合に不要となるバックアップおよびコピーをレポートします。

```
report obsolete;

Report of obsolete backups and copies
Type                Key      Completion Time      Filename/Handle
-----
Backup Set          836      04-DEC-98
Backup Piece        839      04-DEC-98            /vobs/oracle/dbs/05aetj6b_1_1
Backup Set          807      04-DEC-98
Backup Piece        810      04-DEC-98            /vobs/oracle/dbs/03aetj1f_1_1
Backup Set          835      04-DEC-98
Backup Piece        838      04-DEC-98            /vobs/oracle/dbs/04aetj6b_1_1
```

関連項目

- 11-76 ページの「[list](#)」
- 11-147 ページの「[untilClause](#)」
- 11-151 ページの「[validate](#)」

reset

構文



目的

リカバリ・カタログに新しいデータベースのインカーネーション・レコードを作成します。RMAN は、新しいインカーネーションをデータベースの現行のインカーネーションとみなします。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・アーカイブ操作は、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。

要件

- **reset database** は、RMAN プロンプトでのみ実行できます。
- リカバリ・カタログを使用すること。
- RESETLOGS オプションでオープンされたターゲット・データベースで RMAN を使用する前に、**reset database** コマンドの発行が必要です。そうしないと、RMAN は RESETLOGS と古い制御ファイルを間違えて復元してしまった場合との区別ができないので、RMAN はリカバリ・カタログへのアクセスを拒否します。**reset database** コマンドは、ユーザーが RESETLOGS コマンドを発行したことを RMAN に確認させます。

キーワードおよびパラメータ

to incarnation	RMAN が現行とみなすインカーネーションを古いインカーネーションに変更します。このオプションが役立つのは、RESETLOGS の効果を取り消すことが必要になり、データベースの以前のインカーネーションのバックアップを復元するという場合です。
<i>primary_key</i>	データベースのインカーネーションを指す DBINC レコードの主キーを指定します。主キーの値を取得するには、 list incarnation of database コマンドを使用します。 reset database to incarnation コマンドを発行した後、 restore コマンドおよび recover コマンドを発行して、以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、データベースを回復します。

例

RESETLOG 後のデータベースのリセット 次の例では、不完全メディア回復の実行後にデータベースをリセットします。

```
run {
  allocate channel dev1 type disk;
  set until logseq 1234 thread 1;
  restore database skip tablespace readonly;
  recover database;
  sql "ALTER DATABASE OPEN RESETLOGS";
  release channel dev1;
}
reset database;
```

古いインカーネーションのリセット 次のコマンドは、データベース PROD1 の古いインカーネーションを再び現行にします。

```
# obtain primary key of old incarnation
list incarnation of database prod1;
```

List of Database Incarnations

DB Key	Inc Key	DB Name	DB ID	CUR	Reset SCN	Reset Time
-----	-----	-----	-----	---	-----	-----
1	2	PROD1	1224038686	NO	1	02-JUL-98
1	582	PROD1	1224038686	YES	59727	10-JUL-98

```
shutdown immediate;
# reset database to old incarnation
reset database to incarnation 2;
# recover it
run {
  allocate channel dev1 type disk;
  restore controlfile;
  startup mount;
  restore database;
  recover database;
  sql "ALTER DATABASE OPEN RESETLOGS";
  release channel dev1;
}
```

関連項目

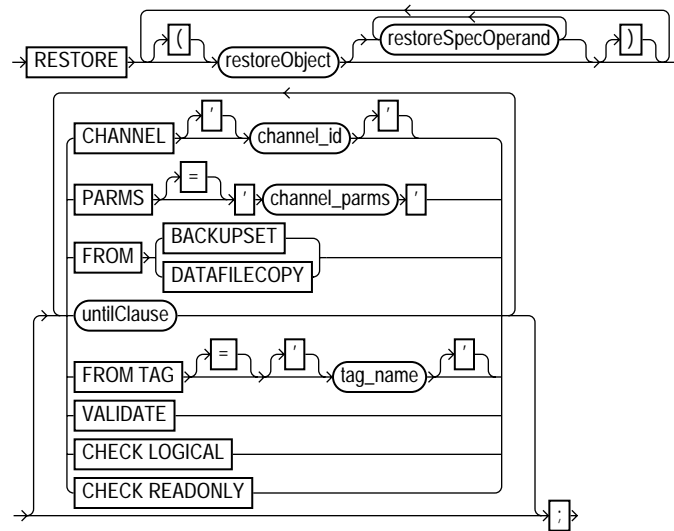
11-76 ページの「[list](#)」

11-113 ページの「[restore](#)」

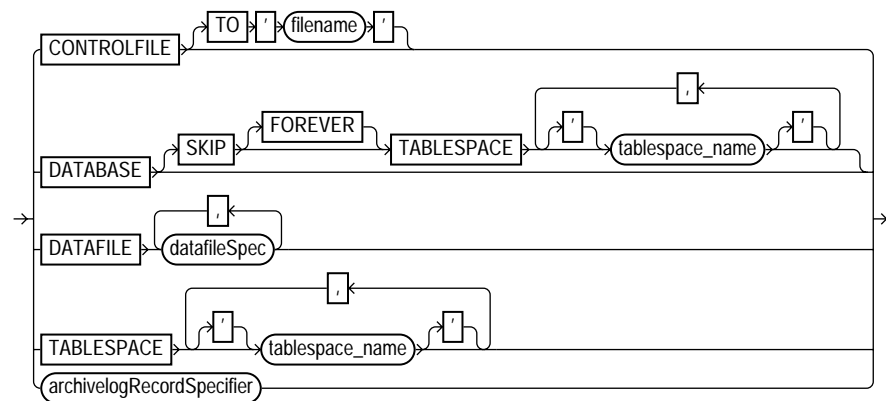
11-89 ページの「[recover](#)」

restore

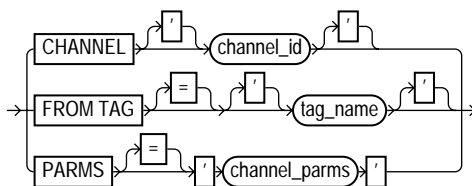
構文



restoreObject::=



restoreSpecOperand::=



目的

バックアップまたはイメージ・コピーによって、現在の位置にファイルを復元し、ファイルと同じ名前の上書きします。RMAN は、ディスクまたはテープからはバックアップを復元しますが、イメージ・コピーはディスクからのみ復元します。

一般的には、メディア障害によって現行のデータ・ファイル、制御ファイルまたはアーカイブ REDO ログ・ファイルが破損したとき、または Point-In-Time 回復を実行する前に復元を行います。

このコマンドは、次の全体バックアップ、レベル 0 増分バックアップ、またはコピーを復元します。

- データベース
- 表領域
- データ・ファイル
- 制御ファイル
- アーカイブ REDO ログ

注意： **recover** コマンドはアーカイブ REDO ログを必要に応じて自動的に復元するので、**restore** コマンドでアーカイブ・ログを復元する必要はほとんどありません。アーカイブ REDO ログを手動で復元する可能性としては、回復のスピード・アップやログの複数宛先への移動があります。

restore コマンドは、レベル 0 より高いレベルの増分バックアップは復元しません。**recover** コマンドを増分バックアップに適用します。

バックアップ制御ファイルおよびリカバリ・カタログを使用して復元操作を実行するとき、RMAN は復元されるバックアップの構造を反映するために制御ファイルを自動的に調整するので注意してください。

関連項目： ファイルのバックアップ方法は、[第 9 章の「Recovery Manager による復元と回復」](#)を参照してください。

要件

- **restore** は、**run** コマンドの中カッコ内のみで実行すること。
- データ・ファイルをその現行の場所に復元するには、データベースがマウントされているか、またはデータ・ファイルがオフラインであることが必要です。データベース全体を復元するには、データベースがマウントされていなければなりません。データ・ファイルを別の場所に復元するには、**set newname** コマンドを使用します。
- **allocate channel** コマンドを **restore** の前に少なくとも 1 つ発行する必要があります。
- **from datafilecopy** オプションを使用する場合、割り当てるチャンネルのタイプは **type disk** にしてください。
- **from backupset** オペランドを使用する場合は、復元が必要となるバックアップ・セット用に、適切なタイプの記憶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検索できず、**restore** コマンドは失敗します。
- RMAN は、**restore** コマンドに割り当てたチャンネルと同じタイプのチャンネルで作成したバックアップのみを復元します。

たとえば、あるデータ・ファイルのいくつかのバックアップを **disk** チャンネルに作成し、他のバックアップは '**sbt_tape**' チャンネルに作成した状態で、**restore** コマンドには **disk** チャンネルのみを割り当てた場合、RMAN は '**sbt_tape**' で作成したバックアップからは復元しません。
- バックアップ制御ファイルを使用してデータベースを復元した後は、**RESETLOGS** オプションを指定してデータベースをオープンすること。
- 復元ジョブでは、データ・ファイルは必ず 1 度のみ指定するようにください。たとえば、次のコマンドでは **datafile 1** が明示的に指定されると同時に **SYSTEM** 表領域によって暗示されているので、コマンドは無効になります。

```
restore
  tablespace system
  datafile 1;
```

キーワードおよびパラメータ

<i>restoreObject</i>	復元するオブジェクトを指定します。
controlfile	現行の制御ファイルを復元し、それを自動的にパラメータ・ファイルのなかのすべての CONTROL_FILES にレプリケートします。新しいパス名を to 'filename' オプションを使用して指定した場合、RMAN は制御ファイルを新しい場所에만復元します。この場合は、 replicate コマンドを使用して手動で制御ファイルをコピーします。
database	オフラインまたは読取り専用のものを除いて、データベースのすべてのデータ・ファイルを復元します。 check readonly オプションを指定した場合、RMAN はすべての読取り専用ファイルのヘッダーを調べて、復元が必要なものは復元します。 backup database コマンドとは異なり、 restore database コマンドは制御ファイルを復元しません。 いくつかの表領域を復元しないためには、オプションの skip ... tablespace 引数を使用します。これは、一時データを含む表領域の復元を避ける必要があるとき役立ちます。
datafile <i>datafileSpec</i>	ファイル名または絶対データ・ファイル番号で指定したデータ・ファイルを復元します。11-60 ページの「 datafileSpec 」を参照してください。
tablespace <i>tablespace_name</i>	指定した表領域にあるデータ・ファイルを復元します。
<i>archivelogRecord-Specifier</i> 句	アーカイブ REDO ログの指定範囲を復元します。11-17 ページの「 archivelogRecordSpecifier 」を参照してください。
<i>restoreSpec-Operand</i>	<i>restoreObject</i> 句に対するオプションを指定します。 注意: これらのパラメータは、 restore コマンドのレベルで同じ名前をもつパラメータに上書きします。
channel <i>channel_id</i>	この復元操作に使用するチャンネルの名前を指定します。チャンネル指定がないと、 restore はいずれか正しいデバイス・タイプで割り当てられた使用可能なチャンネルを使用します。
from tag <i>tag_name</i>	使用可能なものからデフォルトで選択された最新のバックアップまたはファイル・コピーを上書きします。このタグは、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。複数のバックアップ・セットまたはファイル・コピーにマッチングするタグが存在すると、RMAN は最新のものを選択します。
parms <i>channel_parms</i>	O/S 固有の情報を含む引用符付き文字列を指定します。バックアップ・ピースを 1 つ復元するたびに、この文字列が OSD レイヤーに渡されます。
channel <i>channel_id</i>	<i>restoreSpecOperand</i> 句を参照してください。
from tag <i>tag_name</i>	<i>restoreSpecOperand</i> 句を参照してください。

parms	<i>restoreSpecOperand</i> 句を参照してください。
channel_parms	
from	RMAN が復元を、ディスクにある datafilecopy から行うか、それとも backupset から行うかを指定します。デフォルトでは、 restore は最新のバックアップ・セットまたはファイル・コピーを使用します。すなわち、最小のメディア回復で済むファイル・コピーまたはバックアップ・セットを使用します。
untilClause	バックアップ・セットまたはファイル・コピーの選択範囲を Point-in-Time 回復の実行に適したものに限定します。他の基準がない場合は、RMAN 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。11-147 ページの「 untilClause 」を参照してください。
validate	RMAN は、復元が必要なのはどのバックアップ・セット、データ・ファイル・コピーおよびアーカイブ・ログなのかを決定し、次にその内容を検証するためにそれらを走査します。この操作では出力ファイルが作成されません。このオプションを定期的に指定して、指定したファイルの復元に必要なコピーおよびバックアップ・セットに障害がなく使用可能なことを検証してください。
check logical	<p>物理的な破損チェックを通過したデータ・ブロックおよび索引ブロックについて論理的な破損がないかどうかをテストします。たとえば、行ピースまたは索引エントリの破損がないかを調べます。RMAN は論理的な破損を発見すると、<code>alert.log</code> およびサーバー・セッション・トレース・ファイルにそのブロックのログを書き込みます。</p> <p>あるファイルで検出された物理的な破損および論理的な破損の合計数が maxcorrupt の設定値以下の場合、RMAN コマンドは完了し、Oracle は <code>V\$BACKUP_CORRUPTION</code> および <code>V\$COPY_CORRUPTION</code> に破損ブロック範囲を移入します。maxcorrupt を超えている場合は、コマンドはビューへの移入を行わずに終了します。</p> <p>注意: maxcorrupt の設定値は、ファイルに許容される物理的な破損および論理的な破損の合計数を表します。</p>
check readonly	データ・ファイルをチェックして、それが存在すること、読取り可能であること、および適切なチェックポイントであることを確認します。これらの条件が満たされない場合、RMAN はファイルを復元します。この場合、ファイルが読込み専用であるかどうかは問いません。デフォルトでは、RMAN は、ユーザーが restore database コマンドを発行したとき、読込み専用ファイルを復元しません。

例

表領域の復元 この例では、表領域をオフラインにし、それを復元し、次にメディア回復を実行します。

```
run {
    # recover tablespace tbs_1 while the database is open
    allocate channel ch1 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 ch1 ;
}
```

制御ファイルの復元 この例では、制御ファイルをそのデフォルトの場所に復元し、それをすべての CONTROL_FILES の場所にコピーし、データベースをマウントします。

```
startup nomount;
run {
    allocate channel chl type 'sbt_tape';
    restore controlfile;
    alter database mount;
}
```

バックアップ制御ファイルを使用したデータベースの復元 この例では、制御ファイルを新しい場所に復元し、パラメータ・ファイルで指定したすべての制御ファイルの場所にコピーし、次にデータベースを復元するために制御ファイルをマウントします。

```
startup nomount;
run {
    allocate channel chl type 'sbt_tape';
    restore controlfile to '/oracle/dbs/cfl.ct1';
    replicate controlfile from '/oracle/dbs/cfl.ct1';
    alter database mount;
    restore database;
}
```

新しい場所へのアーカイブ REDO ログの復元 この例では、すべてのアーカイブ REDO ログを /oracle/temp_restore ディレクトリに復元します。

```
run {
    set archivelog destination to '/oracle/temp_restore';
    allocate channel chl type disk;
    restore archivelog all;
}
```

関連項目

11-9 ページの「[allocate](#)」

11-89 ページの「[recover](#)」

11-147 ページの「[untilClause](#)」

resync

構文

```
→ RESYNC CATALOG { FROM CONTROLFILECOPY | 'filename' }
```

目的

リカバリ・カタログの完全再同期化（[resynchronization](#)）を実行します。再同期化は、完全でも部分でも実行できます。

完全再同期化では、RMAN は物理スキーマ（[physical schema](#)）の変更されたレコードすべてを更新します。つまり、データ・ファイル、表領域、REDO スレッドおよびオンライン REDO ログのすべてです。データベースがオープン状態の場合、RMAN はロールバック・セグメントについての情報も取得します。

部分再同期化では、RMAN は現行の制御ファイルを読み取って変更情報を更新しますが、物理スキーマ（[physical schema](#)）またはロールバック・セグメントに関するメタデータの再同期化は行いません。

再同期化を行うとき、RMAN は制御ファイルの読み込み一貫性のあるビューを取得するためにスナップショット制御ファイルを作成し、次にカタログをスナップショットの新規情報により更新します。

resync catalog コマンドは、次のクラスのレコードを更新します。

Record Type	説明
Log history	REDO ログ・スイッチが発生したときに作成されたレコード。ログ履歴レコードはオンライン・ログ・スイッチについて表し、ログ・アーカイブについては表しませんので、注意してください。
Archived redo logs	オンライン REDO ログのアーカイブ、既存のアーカイブ REDO ログのコピー、またはアーカイブ REDO ログのバックアップの復元により作成されたアーカイブ・ログに対応付けられたレコード。
Backups	バックアップ・セット、バックアップ・ピース、バックアップ・セット・メンバー、プロキシ・コピーおよびイメージ・コピーに対応付けられたレコード。
Physical schema	データ・ファイルおよび表領域に対応付けられたレコード。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。

次のコマンドは、ターゲット・データベースの制御ファイルがマウントされ、かつリカバリ・カタログ・データベースがコマンド実行時に使用可能のとき、リカバリ・カタログを自動的に更新します。

- **backup**
- **change**
- **copy**
- **crosscheck**
- **deleteExpired**
- **duplicate**
- **restore**
- **switch**
- **recover**
- **list**
- **report**

ユーザーがこれらのコマンドを実行するとき、RMAN は必要に応じて完全または部分再同期化を自動的に実行します。RMAN は、現行の制御ファイルを読み取り、物理スキーマが変更されていないと判断した場合は、物理スキーマに関するメタデータの再同期化は行いません。RMAN が変更を検出した場合には、完全再同期化を実行します。

次の場合は、**resync catalog** を使用して手動で完全再同期化を実行します。

- 再同期化を自動的に実行するコマンドのうちのいずれかをユーザーが発行したときに、リカバリ・カタログが使用不可能である。
- ARCHIVELOG モードで実行している。この場合、ログ・スイッチが発生するか、またはオンライン REDO ログをアーカイブしたときにリカバリ・カタログを自動的に更新しないからです。
- ターゲット・データベースの物理構造に、表領域の追加または削除などの変更を行った。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。

要件

- **resync catalog** コマンドは、RMAN プロンプトまたは **run** コマンドの中カッコのなかで実行してください。
- リカバリ・カタログを使用すること。
- RMAN がリカバリ・カタログ内の物理スキーマ情報を更新するのは、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のみです。ターゲット・

データベースに、バックアップ制御ファイル、新たに作られた制御ファイル、または前に検出した制御ファイルより古い制御ファイルをマウントしている場合は、RMAN はリカバリ・カタログ内の物理スキーマ情報を更新しません。

キーワードおよびパラメータ

from controlfilecopy 'filename'	再同期化に使用する制御ファイルのコピーの名前を指定します。このオプションを使用すると、物理スキーマ情報は更新されません。
--	--

例

構造変更後の再同期化 この例では、表領域 TBS_1 にデータ・ファイル sales.f を追加し、次に物理データベース変更を反映するために、リカバリ・カタログを再同期化します。

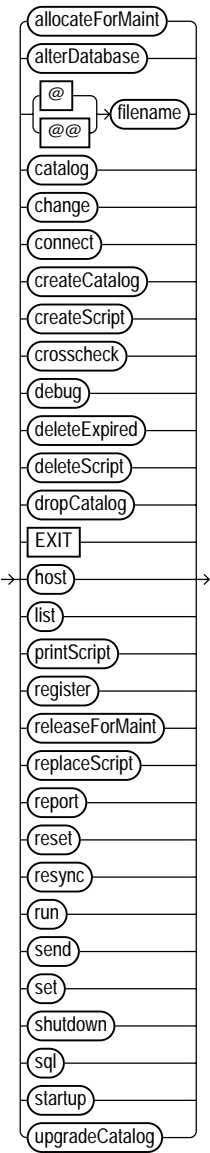
```
startup mount;  
sql "ALTER TABLESPACE tbs_1 ADD DATAFILE 'sales.f' NEXT 10K MAXSIZE 100K";  
resync catalog;
```

ARCHIVELOG モードでの再同期化 この例では、すべてのアーカイブされていない REDO ログをアーカイブした後、ARCHIVELOG データベースの完全再同期化を手動で実行します。

```
sql "ALTER SYSTEM ARCHIVE LOG ALL";  
resync catalog;
```

rmanCmd

構文



目的

stand-alone コマンドを実行すること。このコマンドは、RMAN プロンプトで実行します。

要件

RMAN プロンプトから実行できるコマンドについての情報は、各項目を参照してください。

キーワードおよびパラメータ

@filename	<p>O/S ファイルに格納した一連の RMAN コマンドを、<code>@\$ORACLE_HOME/dbs/cmd/cmd1.rman</code> などのフルパス名を指定して実行します。フルパス名を指定しない場合は、<code>@cmd1.rman</code> などの実行時ディレクトリが前提となります。文字列の前後に引用符を使用したり <code>@</code> とファイル名の間に空白を残さないでください。RMAN は、指定したファイルの内容が <code>@</code> コマンドに代入されたかのように、処理します。</p> <p>注意: ファイルには Recovery Manager コマンドとして完全なもののみを含める必要があります。ファイルに不完全なコマンドがあると、構文エラーになります。</p>
@@filename	<p>スクリプトのなかで使用するのであれば、<code>@filename</code> と同じです。スクリプトのなかで使用する、<code>@@filename</code> は、指定したファイル名が呼び出された元のコマンド・ファイルと同じパス名で、そのファイル名を探すよう RMAN に指示します。</p> <p>たとえば、UNIX での実行時ディレクトリが <code>\$ORACLE_HOME</code> であり、次のように RMAN を起動した場合を想定します。</p> <pre>% rman @\$ORACLE_HOME/rdbms/admin/dba/scripts/cmd1.rman</pre> <p><code>cmd1.rman</code> スクリプトのなかに <code>@@cmd2.rman</code> のコマンドがあるとしたします。この場合、<code>@@</code> コマンドは、ディレクトリ <code>\$ORACLE_HOME/rdbms/admin/dba/scripts</code> にあるファイル <code>cmd2.rman</code> を探すよう RMAN に指示します。</p>
exit	Recovery Manager を終了します。

例

コマンド・ファイルの実行 この例では、ターゲット・データベースおよびリカバリ・カタログに O/S コマンド行から接続し、次にコマンド・ファイル `cmd1.f` を実行します。

```
% rman target / catalog rman/rman@rcvcat
RMAN> @$ORACLE_HOME/dbs/cmd/cmd1.f
```

バックアップのクロスチェック この例では、メンテナンス・チャンネルを割り当て、表領域 FOO のバックアップをクロスチェックし、次に表領域 FOO をバックアップします。

```
allocate channel for maintenance type disk;
crosscheck backup of tablespace foo;
delete expired backup of tablespace foo;
run {
    allocate channel c1 type disk;
    backup tablespace foo;
}
```

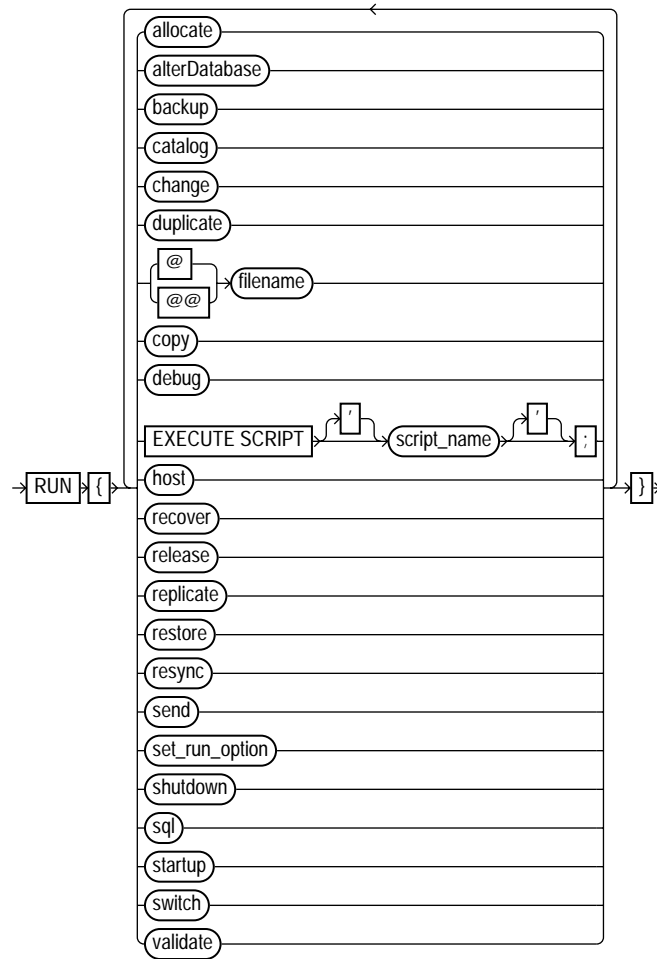
```
release channel;
```

関連項目

11-125 ページの「[run](#)」

run

構文



目的

ジョブ・コマンドをコンパイルし、実行すること。つまり、**run** の中カッコのなかの 1 つ以上の文を実行すること。**run** コマンドは、ジョブ・コマンドのリストを 1 つ以上のジョブ・ステップにコンパイルし、ただちにそれらのジョブ・ステップを実行します。RMAN は、各コマンドを、次のコマンドを処理する前に、コンパイルし、実行します。

要件

- このコマンドは、RMAN プロンプトのみで実行してください。
- ジョブ・コマンドのリストの前には中カッコ、後には閉じ中カッコが必要です。

キーワードおよびパラメータ

RMAN プロンプトから実行できるコマンドについての情報は、各項目を参照してください。

@filename	<p>O/S ファイルに格納した一連の RMAN コマンドを、<code>@\$ORACLE_HOME/dbs/cmd/cmd1.rman</code> などのフルパス名を指定して実行します。フルパス名を指定しない場合は、<code>@cmd1.rman</code> などの実行時ディレクトリが前提となります。文字列の前後に引用符を使用したり、<code>@</code> とファイル名の間に空白を残さないでください。RMAN は、指定したファイルの内容が <code>@</code> コマンドに代入されたかのように処理します。</p> <p>注意: ファイルには Recovery Manager コマンドとして完全なもののみを含める必要があります。ファイルに不完全なコマンドがあると、構文エラーになります。</p>
@@filename	<p>スクリプトのなかで使用するのであれば、<code>@filename</code> と同じです。スクリプトのなかで使用するすると、<code>@@filename</code> は、指定したファイル名が呼出された元のコマンド・ファイルと同じパス名で、そのファイル名を探すよう RMAN に指示します。</p> <p>たとえば、UNIX での実行時ディレクトリが <code>\$ORACLE_HOME</code> であり、次のように RMAN をコマンド行から起動したとします。</p> <pre>% rman @\$ORACLE_HOME/rdbms/admin/dba/scripts/cmd1.rman</pre> <p><code>cmd1.rman</code> スクリプトのなかに <code>@@cmd2.rman</code> のコマンドがあるとします。この場合、<code>@@</code> コマンドは、ディレクトリ <code>\$ORACLE_HOME/rdbms/admin/dba/scripts</code> にあるファイル <code>cmd2.rman</code> を探すよう RMAN に指示します。</p>
execute script script_name	<p>指定したストアド・スクリプトを実行します。すべてのストアド・スクリプトのリストを取得するには、<code>SQL*Plus</code> を使用してリカバリ・カタログ・データベースにカタログ所有者として接続し、次の問合せを発行します。</p> <pre>select * from rc_stored_script;</pre> <p>関連項目: <code>RC_STORED_SCRIPT</code> の詳細は、12-26 ページの「RC_STORED_SCRIPT」を参照してください。スクリプト作成についての情報は、11-54 ページの「createScript」を参照してください。</p>

例

バックアップの作成 この例では、バックアップの実行にシングル・サーバー・プロセスを使用してデータベースをバックアップします。

```
run{
    allocate channel c1 type disk;
    backup database;
}
```

表領域の復元 この例では、表領域 tbs_1 をオフラインにし、それを復元し、次に完全メディア回復を実行します。

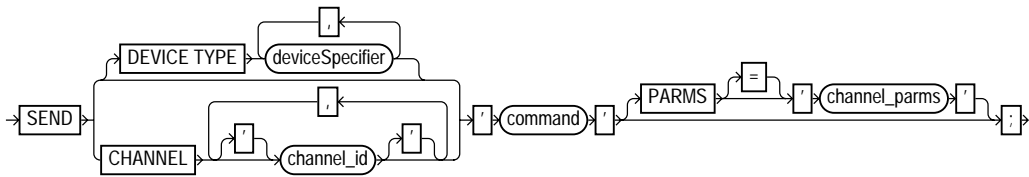
```
run {
    allocate channel ch1 type 'sbt_tape';
    sql "ALTER TABLESPACE tbs_1 OFFLINE IMMEDIATE" ;
    restore tablespace tbs_1 ;
    recover tablespace tbs_1 ;
    sql "ALTER TABLESPACE tbs_1 ONLINE" ;
    release channel ch1 ;
}
```

スクリプトの実行 この例では、ストアド・スクリプト backupdb を実行します。

```
run { execute script backupdb; }
```

send

構文



目的

ベンダー固有の引用符付き文字列を 1 つ以上の特定チャンネルに送信します。どのコマンドがサポートされているかは、メディア管理ソフトウェアのマニュアルを参照してください。

要件

- **send** は、RMAN プロンプトまたは **run** コマンドの中カッコのなかで実行してください。
- **send** を使用するには、メディア・マネージャの使用が必要であり、かつメディア・マネージャがサポートするコマンドでのみ使用する必要があります。Oracle は引用符付き文字列の内容を解釈しないで、そのままメディア管理サブシステムに渡します。

キーワードおよびパラメータ

<i>command</i>	ベンダー固有のメディア管理コマンドを指定します。どのコマンドがサポートされているかは、メディア管理マニュアルを参照してください。
device type <i>deviceSpecifier</i>	記憶デバイスのタイプを指定し、コマンドを指定したタイプのすべてのチャンネルに送信します。11-66 ページの「 deviceSpecifier 」を参照してください。
channel <i>channel_id</i>	どのチャンネルを使用するかを指定します。このキーワードを指定しないと、RMAN はすべての割り当てられているチャンネルを使用します。 channel キーワードの後に、チャンネル ID（すなわち、チャンネルの名前）を指定する必要があります。Oracle では、 release channel コマンドの実行および I/O エラーの報告に、このチャンネル id を使用します。
parms <i>'channel_parms'</i>	割り当てたデバイスに影響するパラメータを指定します。 type disk を指定してある場合、このポート固有の文字列は使用しないでください。 parms と type 'sbt_tape' とを一緒に使用した場合、次の構文の環境変数を指定できます。 PARMS = "ENV = (var1=value1, var2=value2, var3=value3 . . .)" 引用符付き文字列の最大長は 1000 バイトです。

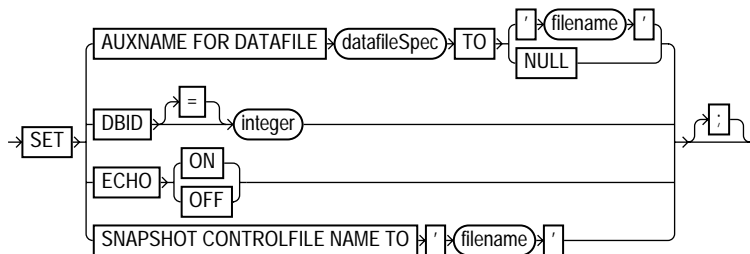
例

メディア・マネージャへのコマンドの送信 この例では、ベンダー固有のコマンドをメディア・マネージャに送信します。

```
send device type 'sbt_tape' 'A223dr';
```

set

構文



目的

次の目的に **set** コマンドを使用します。

- TSPITR で使用する補助データベースのファイル名を指定する。
- **duplicate** コマンドで使用する新しいデータ・ファイルのファイル名を指定する。
- メッセージ・ログに記録された実行済み RMAN コマンドを表示する。
- データベースの DB 識別子を指定する。
- スナップショット制御ファイルのファイル名を設定する。

要件

- **set** は RMAN プロンプトでのみ使用してください。
- **set dbid** は、まだターゲット・データベースに接続していない場合にのみ使用してください。
- **'filename'** オプションに **auxname for datafile** を使用するには、リカバリ・カタログがある必要があります。

キーワードおよびパラメータ

auxname for datafile <i>datafileSpec</i> to 'filename'	<p>補助データベースで使用するターゲット・データベースのデータ・ファイルを新しいファイル名に設定します。TSPITR を実行しているか、または duplicate コマンドを使用している場合、このオプションを設定すると、プロシージャ中に手動で補助ファイル名を指定しなくても、補助データベースで使用するファイル名の事前構成ができます。</p> <p>たとえば、TSPITR 中に、データ・ファイルがロー・ディスクにあってパフォーマンスの理由で補助データ・ファイルをロー・ディスクに復元する必要がある場合は、このコマンドを使用します。一般的には、TSPITR で auxname パラメータを設定するのは、SYSTEM 表領域のデータ・ファイルおよびロールバック・セグメントを収めている表領域を対象とした場合です。本番データベースで使用中のファイルを上書きしないよう注意してください。本質的には、データ・ファイルの auxname とは、TSPITR がデータ・ファイルの一時コピーを作成できる場所です。</p> <p>duplicate コマンドでファイルを改名する場合、set auxname は set newname の代替方法になります。違いは、auxname を一度設定しておけば、別の duplicate コマンドを発行したときにリセットの必要がないことです。設定は、set auxname ... to null を発行するまで有効です。反対に、set newname コマンドでは、ファイルの改名のたびにコマンドの再発行が必要です。</p> <p>関連項目 : RMAN TSPITR を実行する方法は 付録 A の「Recovery Manager による表領域の Point-in-Time 回復の実行」 を参照してください。データベースを複製する方法は、第 10 章の「Recovery Manager による複製データベースの作成」 を参照してください。</p> <p>to null 指定したデータ・ファイルに対する auxname の現行の値を未指定にしておきます。</p>
dbid integer	<p><i>db identifier</i> を指定します。データベースを作成したときに計算される一意で 32 ビットの識別番号です。識別子は、V\$DATABASE データ・ディクショナリ・ビューの DBID 列に表示されます。DBID は、リカバリ・カタログの DB 表にも格納します。</p> <p>set dbid コマンドは、次に示す条件を満たすときに制御ファイルを復元するのに役立ちます。</p> <ul style="list-style-type: none"> ■ 制御ファイルが消失し、バックアップからの復元が必要である。 ■ リカバリ・カタログを使用している。 ■ リカバリ・カタログに登録された複数のデータベースが同じデータベース名を持っている。 ■ "RMAN-20005: target database name is ambiguous" エラー・メッセージが表示された。このメッセージは、制御ファイルの復元を試みたときに表示されるものです。 <p>これらの条件とは違う場合、RMAN は復元する制御ファイルを正しく識別しますので、set dbid コマンドを使用する必要はありません。</p> <p>RMAN が set dbid を受け入れるのは、ターゲット・データベースにまだ接続していない場合のみです。つまり、set dbid は connect target コマンドより先に発行する必要があります。ターゲット・データベースがマウントされていると、RMAN はユーザー指定の DBID がデータベースの DBID に合うかを検証し、合わない場合はエラーを通知します。ターゲット・データベースがマウントされていないと、RMAN はユーザー指定の DBID を使用して制御ファイルを復元します。制御ファイルの復元が完了すれば、データベースをマウントして残ったデータベースを復元できます。</p>

echo [on | off]

RMAN コマンドをメッセージ・ログに表示するかどうかを制御します。コマンド・ファイルからコマンドを読み込むとき、RMAN はそれらのコマンドを自動的にメッセージ・ログに表示します。標準入力からコマンドを読み込む際、**set echo on** コマンドが使用されていないと、RMAN はそれらのコマンドをメッセージ・ログにエコーしません。

コマンドが役立つのは、stdin および stdout がリダイレクトされたときのみです。たとえば、UNIX ではこの方法で RMAN の入力と出力をリダイレクトできます。

```
% rman target sys/sys_pwd@prod1 catalog rman/rman@rcat < input_file > output_file
```

set echo on の指定によって、input_file に含まれているコマンドを output_file に表示することができます。

**snapshot
controlfile name
to 'filename'**

ターゲット・データベースにあるスナップショット制御ファイル名を指定の場所に設定します。RMAN は、スナップショット制御ファイルを使用してリカバリ・カタログを再同期化します。スナップショット制御ファイルについての詳しい情報は、5-3 ページの「[スナップショットの制御ファイル位置の決定](#)」を参照してください。

例

制御ファイルの復元 この例では、ユーザー指定の DBID を使用して制御ファイルを復元します。制御ファイルの復元が完了すれば、データベースをマウントして残ったデータベースを復元できます。

```
set dbid = 862893450;
connect target;
startup nomount;
run {
  allocate channel dev1 type disk;
  # restoring the control file from its default location automatically replicates it
  restore controlfile;
  alter database mount;
}
```

スナップショット制御ファイルの場所の指定 この例では、スナップショット制御ファイルに新しい場所を指定し、次にリカバリ・カタログを再同期化します。

```
set snapshot controlfile name to '/oracle/dbs/snap.cf';
resync catalog;
```

スナップショット制御ファイルの場所の指定 この例では、データ・ファイルの新しいファイル名の指定に **set auxname** を使用して、データベースを異なるディレクトリ構造でリモート・ホストに複製します。

```
# set auxiliary names for the datafiles
set auxname for datafile 1 to '/oracle/auxfiles/aux_1.f';
set auxname for datafile 2 to '/oracle/auxfiles/aux_2.f';
set auxname for datafile 3 to '/oracle/auxfiles/aux_3.f';
set auxname for datafile 4 to '/oracle/auxfiles/aux_4.f';
```

```
run {
  allocate auxiliary channel dupdb1 type disk;
  duplicate target database to dupdb
  logfile
    group 1 ('$ORACLE_HOME/dbs/dupdb_log_1_1.f',
             '$ORACLE_HOME/dbs/dupdb_log_1_2.f') size 200K,
    group 2 ('$ORACLE_HOME/dbs/dupdb_log_2_1.f',
             '$ORACLE_HOME/dbs/dupdb_log_2_2.f') size 200K reuse;
}
# Un-specify the auxiliary names for your datafiles so that they will not be overwritten
# by mistake:
set auxname for datafile 1 to null;
set auxname for datafile 2 to null;
set auxname for datafile 3 to null;
set auxname for datafile 4 to null;
```

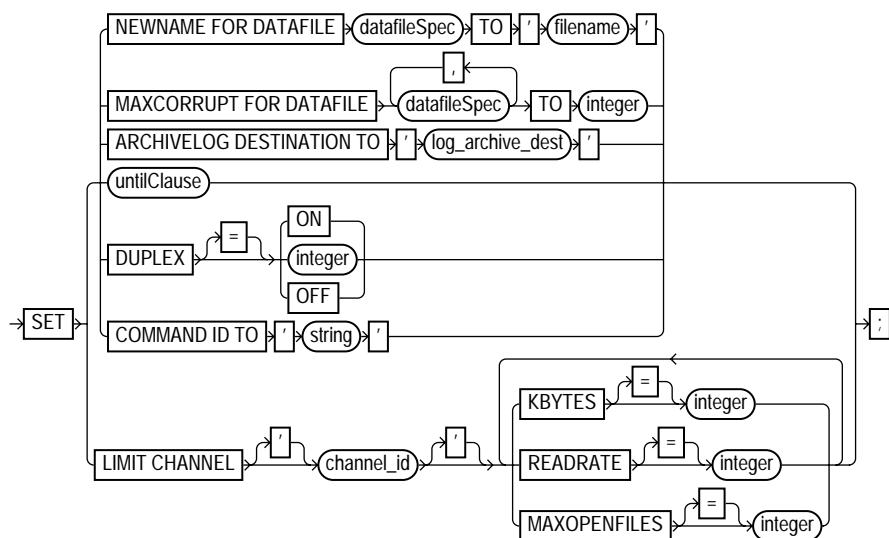
関連項目

11-134 ページの「[set_run_option](#)」

11-69 ページの「[duplicate](#)」

set_run_option

構文



目的

ジョブ終了まで持続する **run** コマンドの属性を設定します。指定した属性は、**set** コマンドの後の **run** コマンド内のすべての文に影響を与えます。次の目的に **set** を使用します。

- データ・ファイルの新規ファイル名を指定する。
- ブロック破損許容数の制限を指定する。
- デフォルトのアーカイブ REDO ログ先を上書きする。
- 回復の終了時刻、SCN またはログ順序番号を指定する。
- バックアップの二重化を指定する。
- サーバー・セッションとチャンネルの対応を判断する。
- 指定されたチャンネルの各入力データ・ファイルから読み取る秒当たりのバッファ数を制限する。
- 指定のチャンネルで一定時間に **backup** コマンドがオープンしておくことが可能な入力ファイル数を制限する。

- 指定したチャンネルのバックアップ・ピースのサイズを制限する。

要件

- このコマンドの実行は、**run** コマンドの中カッコ内でのみ行うこと。
- **set duplex** コマンドは、コマンドの発行後に割り当てるすべてのチャンネルに影響を与え、明示的に使用禁止または変更するまで有効です。**set duplex** コマンドは、すでに割当て済みのチャンネルには影響しません。

キーワードおよびパラメータ

newname for datafile <i>datafileSpec to 'filename'</i>	後続の、指定したデータ・ファイルに影響を与えるすべての restore または switch コマンドのデフォルト名を設定します (11-60 ページの「 datafileSpec 」を参照) データ・ファイル復元操作の前にこのコマンドを発行しない場合、RMAN はファイルをそのデフォルトの場所に復元します。
maxcorrupt for datafile <i>datafileSpec to integer</i>	以前に検出されていなかった物理的なブロック破損の数に制限を設定します。指定したデータ・ファイルまたは一連のデータ・ファイルの中で Oracle は、これらの破損を許容します。(11-60 ページの「 datafileSpec 」を参照)。 backup または copy コマンドが指定数以上の破損を検出した場合、コマンドは正常に終了しません。デフォルトは 0 (ゼロ) で、RMAN が破損ブロックを許容しないことを意味します。 注意: check logical を指定した場合、 maxcorrupt は論理的な破損にも適用されます。
archivelog destination to <i>'log_archive_dest'</i>	後続の restore および recover コマンドで復元されるアーカイブ・ログの名前を構成する際に、ターゲット・データベースの LOG_ARCHIVE_DEST または LOG_ARCHIVE_DEST_1 初期化パラメータを上書きします。RMAN は、 <i>log_archive_dest</i> に指定した宛先にログを復元します。既にディスクに存在しないアーカイブ REDO ログの復元に、このパラメータを使用します。 データベース復元の実行中に、このコマンドを使用して、異なる場所へアーカイブ・ログを移動します。RMAN 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います。アーカイブ・ログが LOG_ARCHIVE_DEST_1 または LOG_ARCHIVE_DEST によって指定された宛先にある必要はありません。たとえば、パラメータ・ファイルで指定した宛先とは異なる宛先を指定してアーカイブ REDO ログのバックアップを復元する場合、後続の復元および回復操作では新たな場所を検出します。RMAN は、アーカイブ REDO ログをバックアップ・セットから復元する前に、それがディスク上にあるかどうかを必ず最初に調べます。
untilClause	後続の restore または recover コマンドで使用する終了時刻、SCN またはログ順序番号を指定します。11-147 ページの「 untilClause 」を参照してください。
duplex	チャンネルが作成する必要がある各バックアップ・ピースの数を指定します。指定可能な値は 1、2、3 または 4 です。 set duplex コマンドは backup コマンドにのみ影響を与えますが、コマンドの発行後は割り当てられるチャンネルすべてに影響をもち、明示的に使用禁止 (off) にするかまたはセッション中に変更するまでは有効です。デフォルトでは、 duplex の値は off です。つまり、RMAN はバックアップ・セットを 1 つ作成します。 on に指定すると、RMAN は同じバックアップ・セットを 2 つ作成します。

command id to <i>'string'</i>	<p>指定した文字列をすべてのチャネルの V\$SESSION.CLIENT_INFO 列に入れます。Oracle サーバー・セッションと RMAN チャネルの対応関係の判断にこの情報を使用します。</p> <p>V\$SESSION.CLIENT_INFO 列には、各 RMAN サーバー・セッションに対する情報があります。データの形式は次の形式のどちらかです。</p> <ul style="list-style-type: none">■ <code>id=string</code>■ <code>id=string, ch=channel_id</code> <p>1 番目の形式は、RMAN ターゲット・データベース接続で使用します。2 番目の形式は、割り当てられたすべてのチャネルについて使用されます。現行ジョブが完了すると、V\$SESSION.CLIENT_INFO 列は消去されます。</p> <p>関連項目： V\$SESSION.CLIENT_INFO の詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。</p>
limit channel <i>channel_id</i>	<p>割り当てたチャネルを使用して実行する backup または copy コマンドのいずれにも適用する制限を指定するパラメータの設定です。</p> <p>kbytes integer このチャ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キロバイトで指定します。</p> <p>readrate integer 各入力データ・ファイルから backup または copy 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。</p> <p>maxopenfiles integer backup コマンドが一定時間にオープンにできる入力ファイルの最大数を制御します。このパラメータは、おおく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、"Too many open files" という O/S エラー・メッセージを出さないために使用します。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。</p>

例

コマンド ID の設定 この例では、コマンド ID を設定し、DATA_1 表領域をバックアップし、O/S に切り替えた後、オンライン REDO ログをアーカイブします。

```
run {
  set command id to 'rman';
  allocate channel t1 type 'SET_TAPE'
  allocate channel t2 type 'SBT_TAPE';
  backup
    incremental level 0
    filesperset 5
    tablespace data_1;
  host;
  sql 'ALTER SYSTEM ARCHIVE LOG ALL';
}
```

バックアップ・セットを二重化 この例では、datafile 1 のバックアップ・セットを 2 つ同時に作成します。

```
run {
    set duplex = ON;
    allocate channel dev1 type disk;
    backup
        filesperset 1
        datafile 1;
}
```

チャネル制限の設定 この例では、3 つのチャネルを割り当て、各チャネルで作成するバックアップ・ピースの最大サイズを設定します。また、データベースの同じバックアップを 3 組作成します。

```
startup mount;
run {
    set duplex=3;
    allocate channel ch1 type 'sbt_tape';
    allocate channel ch2 type 'sbt_tape';
    allocate channel ch3 type 'sbt_tape';

    set limit channel ch1 kbytes 2097150;
    set limit channel ch2 kbytes 2097150;
    set limit channel ch3 kbytes 2907150;

    backup
        filesperset 5
        database;
    alter database open;
}
```

関連項目

11-89 ページの「[recover](#)」

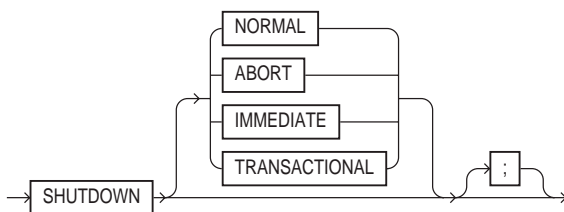
11-113 ページの「[restore](#)」

11-130 ページの「[set](#)」

11-147 ページの「[untilClause](#)」

shutdown

構文



目的

RMAN を終了せずに、ターゲット・データベースを停止します。このコマンドは、SQL*Plus の SHUTDOWN 文を使用するのと同様です。

関連項目：データベースの起動と停止の方法の詳細は、『Oracle8i 管理者ガイド』を参照してください。SHUTDOWN の構文に関しては、『SQL*Plus ユーザーズ・ガイドおよびリファレンス』を参照してください。

要件

- このコマンドは、**run** コマンドの中カッコの内、または RMAN プロンプトで実行してください。
- リカバリ・カタログ・データベースの停止に RMAN の **shutdown** コマンドは使用できません。リカバリ・カタログ・データベースを停止するには、SQL*Plus セッションを開始して SHUTDOWN 文を発行します。
- **normal**、**transactional** および **immediate** のすべてのオプションで、データベースの正しいクローズが実行されます。**abort** オプションではデータベースを正しくクローズしません。Oracle は起動時にインスタンス回復を実行します。
- データベースを NOARCHIVELOG モードで操作している場合、ユーザーはデータベースを正しく停止し、次にバックアップを作成する前に **startup mount** を発行する必要があります。

キーワードおよびパラメータ

normal	<p>データベースを NORMAL モード（デフォルトのオプション）で停止します。これには、次のような意味があります。</p> <ul style="list-style-type: none">■ 文の発行後は新たな接続ができない。■ Oracle は、現行データベースに接続しているすべてのユーザーがデータベースから切断するのを待って、データベースを停止します。■ 次回のデータベース起動時には、インスタンス回復の必要がない。
abort	<p>ターゲット・インスタンスを強制終了します。次の結果になります。</p> <ul style="list-style-type: none">■ すべての現行クライアントの SQL 文は、即時終了する。■ コミットされていないトランザクションは、次の起動までロールバックされない。■ Oracle は、すべての接続ユーザーを切断する。■ Oracle は、次の起動時にデータベースのクラッシュ回復を実行する。
immediate	<p>ターゲット・データベースを即時に停止します。次の結果になります。</p> <ul style="list-style-type: none">■ Oracle が処理中の現行クライアントの SQL 文は、完了する。■ コミットされていないトランザクションはロール・バックされる。■ すべての接続ユーザーは切断される。
transactional	<p>ターゲット・データベースをクライアントへの中断を最小化して停止します。次の結果になります。</p> <ul style="list-style-type: none">■ 現在トランザクションを進めているクライアントは、データベースを停止する前にトランザクションをコミットするか、またはロールバックする。■ どのクライアントもこのインスタンス時に新規のトランザクションを開始できません。新たなトランザクションを開始しようとするクライアントは切断されます。■ すべてのトランザクションがコミットするかまたはロールバックした後、接続中のクライアントは切断される。

例

即時モードでのデータベースの停止 この例では、現行の SQL トランザクションが処理されるのを待ってデータベースを停止し、その後データベースをマウントします。

```
shutdown immediate;  
startup mount;
```

NOARCHIVELOG モードでのデータベースの停止 この例では、NOARCHIVELOG モードで実行中のデータベースをバックアップします。

```
shutdown abort;  
startup dba;
```

```
shutdown;  
startup mount;  
# executing the above commands ensures that database is in proper state for noarchivelog  
# backup  
run {  
    set duplex = 4;  
    allocate channel dev1 type 'sbt_tape';  
    backup database;  
    alter database open;  
}
```

関連項目

11-15 ページの「[alterDatabase](#)」

11-143 ページの「[startup](#)」

sql

構文

→ SQL → ' → command → ' → ; →

目的

Recovery Manager のなかから SQL 文を実行します。有効な SQL 文の構文については、『Oracle8i SQL リファレンス』を参照してください。

要件

- **sql** コマンドは、RMAN プロンプトまたは **run** コマンドの中カッコのなかで実行してください。
- RMAN が PL/SQL に渡す文字列にファイル名がある場合、ファイル名を 2 組の一重引用符で囲み、**sql** パラメータに続く文字列全体を二重引用符で囲む必要があります。たとえば、次のような構文にしてください。

```
sql "CREATE TABLESPACE temp1 DATAFILE ' '$ORACLE_HOME/dbs/temp1.f' ' "
```

SQL パラメータに続く文字列に一重引用符を使用するか、あるいはファイル名に一重引用符を 1 組のみ使用すると、コマンドは正常に終わりません。

- 結果が表示されないため、SELECT 文は実行できません。

キーワードおよびパラメータ

command

SQL 文の実行を指定します。たとえば、次のような SQL 文を RMAN プロンプトで発行します。

```
sql 'ALTER SYSTEM ARCHIVE LOG ALL';
```

この文は、オンライン REDO ログをアーカイブします。

例

オンライン表領域の O/S コピーの作成 この例では、オペレーティング・システムにホストを切り替えてオンライン表領域 TBS_1 の O/S コピーを作成した後に、それをカタログに登録します。

```
sql 'ALTER TABLESPACE tbs_1 BEGIN BACKUP';  
host 'cp $ORACLE_HOME/dbs/tbs_1.f/dbs/tbs_1.f $ORACLE_HOME/copy/temp3.f';
```

```
sql 'ALTER TABLESPACE tbs_1 END BACKUP';
catalog datafilecopy '$ORACLE_HOME/copy/temp3.f';
sql 'ALTER SYSTEM ARCHIVE LOG ALL';
```

引用符付き文字列のなかでのファイル名の指定 この例では、二重引用符付き文字列のコンテキストのなかで 2 組の一重引用符を使用してファイル名を指定します。

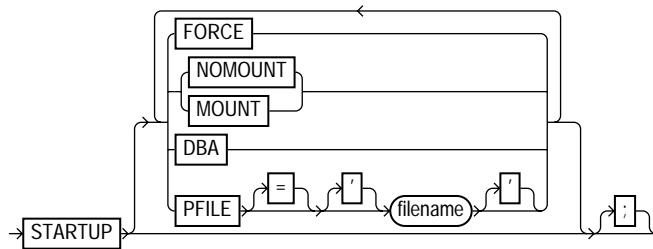
```
sql "ALTER TABLESPACE tbs_1 ADD DATAFILE ' '/oracle/dbs/tbs_7.f' ' NEXT 10K MAXSIZE 100k;"
```

RESETLOGS オプションでのオープン この例では、不完全回復を実行し、データベースを RESETLOGS オプションを指定してオープンします。

```
run {
    set until scn 1000;
    allocate channel c1 type 'sbt_tape';
    restore database;
    recover database;
    sql 'ALTER DATABASE OPEN RESETLOGS';
    reset database;
}
```

startup

構文



目的

RMAN 環境のなかからデータベースを起動します。このコマンドは、SQL*Plus の STARTUP コマンドを使用するのと同様です。次のことができます。

- データベースをマウントせずにインスタンスを起動する。
- インスタンスを起動しデータベースをマウントするが、データベースはクローズ状態にしておく。
- インスタンスを起動し、次のモードでデータベースをマウントしオープンする。
 - 無制限モード（すべてのユーザーがアクセス可能）
 - 制限モード（DBA のみがアクセス可能）

関連項目：データベースの起動と停止の方法は、『Oracle8i 管理者ガイド』を参照してください。SQL*Plus STARTUP 文の構文は、『SQL*Plus ユーザーズ・ガイドおよびリファレンス』を参照してください。

要件

- このコマンドは、RMAN プロンプトまたは **run** コマンドの中カッコのなかで実行してください。
- リカバリ・カタログ・データベースのオープンに RMAN の **startup** コマンドは使用できません。リカバリ・カタログ・データベースを起動するには、SQL*Plus セッションを起動し、STARTUP 文を実行します。

キーワードおよびパラメータ

オプションを指定しないと、RMAN はデータベースをマウントし、オープンします。

force	次の操作のいずれかを実行します。 <ul style="list-style-type: none">■ データベースがオープン状態の場合、このオプションでは、データベースを再オープンする前に、まずデータベースを shutdown abort 文で停止します。■ データベースがクローズ状態の場合は、このオプションはデータベースをオープンします。
nomount	データベースをマウントせずにインスタンスを起動します。
mount	インスタンスを起動してからデータベースをマウントしますが、オープンはしません。 mount も nomount も指定しない場合は、 startup コマンドはデータベースをオープンします。
dba	データベースへのアクセスを RESTRICTED SESSION 権限をもつユーザーに制限します。
pfile = filename	ターゲット・データベースの <code>init.ora</code> ファイルのファイル名を指定します。このパラメータを指定しないと、デフォルトの <code>init.ora</code> のファイル名が使用されます。

例

デフォルトのパラメータ・ファイルを使用してのデータベースのオープン この例では、データベースを開始し、オープンします。

```
startup;
```

パラメータ・ファイルの指定とともにデータベースのマウントをする この例では、**shutdown abort** を強制し、その後、非デフォルトのパラメータ・ファイルの場所を指定して、制限付きアクセスでデータベースをマウントします。

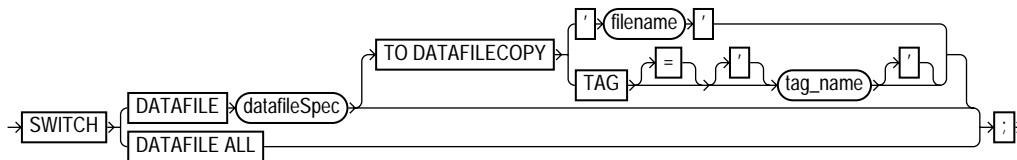
```
startup force mount dba pfile=t_init1.ora;
```

関連項目

- 11-15 ページの「[alterDatabase](#)」
- 11-138 ページの「[shutdown](#)」

switch

構文



目的

データ・ファイルのコピーを現行のデータ・ファイルとすること（すなわち、制御ファイルがポイントするデータ・ファイルであること）を指定します。switchを使用するのは、ALTER DATABASE RENAME DATAFILE 文を使用する場合と同じです。Oracle は制御ファイルではファイルを改名しますが、実際にユーザーのオペレーティング・システム上でファイルを改名するものではありません。この操作によってリカバリ・カタログおよび制御ファイルにあるデータ・ファイル・コピーのレコードが削除されることに注意してください。

要件

- **switch** は、**run** コマンドの中カッコ内のみで実行してください。
- 制御ファイルが復元されたバックアップ制御ファイルである場合、**switch** は、データ・ファイルがまだ制御ファイルに存在しないと、制御ファイルにデータ・ファイルのレコードを追加します。**switch** で追加できるのは、バックアップ制御ファイルが作成された後に作成したデータ・ファイルのみです。

キーワードおよびパラメータ

datafile datafileSpec

改名が必要なデータ・ファイルを指定します。ファイルの切替え後は、制御ファイルは指定したファイルを現行とみなして表示することはありません。たとえば、このコマンドは tbs_1.f から cp1.f までの制御ファイルをポイントします。

```
switch datafile '$/dbs/tbs_1.f' to datafile copy '$/dbs/copies/cp1.f';
```

to オプションを指定しないと、RMAN は、このファイル番号に対して以前の **set newname** コマンド（11-134 ページの「[set_run_option](#)」を参照）で切替え先として指定したファイル名を使用します。

to datafilecopy ' ファイルの切替えに使用する入力コピー・ファイルを指定します。つまり、改名が必要なデータ・ファイル・コピーを指定します。たとえば、次の文を発行したとします。

```
switch datafile 2 to datafilecopy '/oracle/dbs/df2.copy';
```

制御ファイルは、datafile 2 に対応するファイル名として df2.copy をリストします。

データ・ファイル・コピーをタグで指定することもできます。タグが一意でない場合は、最新のコピーを使用します。つまり、メディア回復の必要度が最も低いものを使用します。

タグが一意でない場合、RMAN はタグが最近に作成したコピーを参照するものと理解します。次のコマンドを発行したとします。

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

このコマンドは、datafile 3 を最近作成した月曜夕刻のコピーに切り替えます。

datafile all このジョブでこれまでに発行された **set newname for datafile** コマンド (11-134 ページの「[set_run_option](#)」を参照) の対象となっているすべてのデータ・ファイルを、新しいファイル名に切り替えるよう指定します。

例

復元後の切替え この例では、RMAN 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";
}
```

関連項目

11-87 ページの「[printScript](#)」

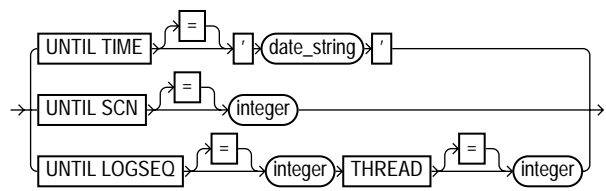
11-113 ページの「[restore](#)」

11-125 ページの「[run](#)」

11-125 ページの「[set](#)」

untilClause

構文



目的

各 RMAN 操作に対して時刻、SCN またはログ順序番号により上限を指定します。

要件

日付文字列は次のようにする必要があります。

- 現行有効な NLS 日付書式仕様に従って書式化していること。
 - DATE 値を戻す SQL 式で作成していること。たとえば、'SYSDATE-30'。
- この句は、次のコマンドと一緒に使用します。

- **recover**
- **report**
- **restore**
- **set_run_option**

キーワードおよびパラメータ

until time <i>'date_string'</i>	時刻を上限として指定します。
until scn <i>integer</i>	SCN を上限として指定します。
until logseq <i>integer</i>	REDO ログ順序番号を上限として指定します。
thread <i>integer</i>	対象となっている REDO ログのスレッド番号を指定します。

例

あるログ順序番号までの不完全回復の実行 この例は、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・ログを使用してデータベースを回復する必要があることを前提とします。

```
run {
  allocate channel ch1 type disk;
  allocate channel ch2 type 'sbt_tape';
  set until logseq 1234 thread 1;
  restore controlfile to '$ORACLE_HOME/dbs/cf1.f' ;
  replicate controlfile from '$ORACLE_HOME/dbs/cf1.f';
  alter database mount;
  restore database;
  recover database;
  sql "ALTER DATABASE OPEN RESETLOGS";
}
```

指定 SCN までの DBPITR の実行 この例では、指定 SCN までデータベースを回復します。

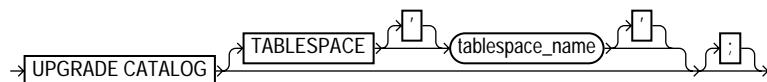
```
startup mount;
run{
  allocate channel ch1 type disk;
  restore database;
  recover database until scn 1000;
  sql "ALTER DATABASE OPEN RESETLOGS";
}
```

不要となったバックアップのレポート この例は、最近 7 日間のうちのいずれかの時点まで回復できることを前提条件としています。この例では、1 週間以上前に作成したバックアップを不要とみなします。

```
report obsolete until time 'SYSDATE-7';
```

upgradeCatalog

構文



目的

リカバリ・カタログ・スキーマを、旧バージョンから、RMAN 実行可能ファイルに必要なバージョンにアップグレードすること。たとえば、RMAN バージョン 8.0 のリカバリ・カタログを RMAN バージョン 8.1 で使用する場合、カタログのアップグレードが必要です。

要件

- RMAN をリカバリ・カタログに接続していること。
- アップグレードの確認のため、**upgrade** コマンドを続けて 2 回入力することが必要です。
- リカバリ・カタログがすでに RMAN 実行可能ファイルが必要とするバージョンより高いバージョンの場合は、エラーになります。ただし、リカバリ・カタログがすでに現行になっていれば、RMAN はコマンドの実行を認めます。その結果、必要があればパッケージの再作成ができます。RMAN は、アップグレード中に生成したエラー・メッセージをメッセージ・ログに表示します。

キーワードおよびパラメータ

tablespace	リカバリ・カタログを格納する表領域を指定します。この指定がない場合、リカバリ・カタログのアップグレードに使用する CREATE TABLE 文では、 tablespace パラメータは使用されません。つまり、カタログはデフォルトの表領域に格納されます。
tablespace_name	

例

カタログのアップグレード この例では、リカバリ・カタログ・データベース RECDB に接続し、次にそれを最新バージョンにアップグレードします。

```
rman catalog rcat/rcat@recdb
```

```
RMAN-06008: connected to recovery catalog database
```

```
RMAN-06186: PL/SQL package rcat.DBMS_RCVCAT version 08.00.04 in RVCAT
             database is too old
```

```
RMAN> upgrade catalog
```

```
RMAN-06435: recovery catalog owner is rcat
```

```
RMAN-06442: enter UPGRADE CATALOG command again to confirm catalog upgrade
```

```
RMAN> upgrade catalog
```

```
RMAN-06408: recovery catalog upgraded to version 08.01.03
```

```
RMAN-06452: DBMS_RCVMAN package upgraded to version 08.01.05
```

```
RMAN-06452: DBMS_RCVCAT package upgraded to version 08.01.03
```

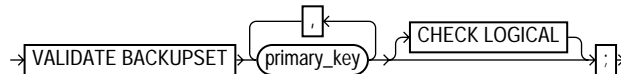
関連項目

11-52 ページの「[createCatalog](#)」

11-68 ページの「[dropCatalog](#)」

validate

構文



目的

バックアップ セットを調べて復元できるかどうかをレポートすること。RMAN は指定したバックアップ・セットのすべてのバックアップ・ピースを走査し、チェックサムを調べて、内容が変更されていないことを検証するので、バックアップは必要に応じて正常に復元できます。**validate backupset** は実際にバックアップ・セットが復元できるかテストしますが、一方 **change crosscheck** は指定したファイルのヘッダーを調べるのみであることに注意してください。

バックアップ・セットのなかの 1 つ以上のバックアップ・ピースが欠落または破損の疑いがあるときは、このコマンドを使用してください。テストするバックアップ・セットの指定には **validate backupset** を使用し、RMAN にどのバックアップを検査するか選択させるときは **restore** コマンドの **validate** オプションを使用します。

要件

- このコマンドの実行は、**run** コマンドの中カッコ内でのみ行うこと。
- **validate backupset** 文を実行する前に、少なくとも 1 つのチャンネルを割り当ててください。

キーワードおよびパラメータ

<i>primary_key</i>	検査するバックアップ・セットを <i>primary_key</i> により指定します。バックアップ・セットの主キーを取得するには、 list 文を実行します。あるいは、リカバリ・カタログを使用している場合は、RC_BACKUP_SET ビューに問い合わせます。
check logical	物理破損チェックを通過したデータ・ブロックおよび索引ブロックについて論理的な破損がないかどうかをテストします。たとえば、行ピースまたは索引エントリの破損がないかを調べます。RMAN は論理的な破損を発見すると、 <i>alert.log</i> およびサーバー・セッション・トレース・ファイルにそのブロックのログを書き込みます。RMAN コマンドは完了し、Oracle は破損ブロックの範囲を V\$BACKUP_CORRUPTION および V\$COPY_CORRUPTION に記録します。 注意： validate は maxcorrupt を使用しません。

例

バックアップ・セットの検査 この例では、主キーが 12 のバックアップ・セットの状態を検査します。

```
run{
    allocate channel ch1 type disk;
    validate backupset 12;
}
# As the output indicates, RMAN determines whether it is possible to restore the
# specified backup set.
RMAN-03022: compiling command: allocate
RMAN-03023: executing command: allocate
RMAN-08030: allocated channel: c1
RMAN-08500: channel ch1: sid=10 devtype=DISK

RMAN-03022: compiling command: validate
RMAN-03023: executing command: validate
RMAN-08016: channel ch1: starting datafile backupset restore
RMAN-08502: set_count=16 set_stamp=341344502 creation_time=14-AUG-98
RMAN-08023: channel ch1: restored backup piece 1
RMAN-08511: piece handle=/oracle/dbs/0ga5h07m_1_1 params=NULL
RMAN-08024: channel ch1: restore complete
RMAN-08031: released channel: ch1
```

関連項目

11-113 ページの「[restore](#)」

リカバリ・カタログ・ビュー

この章では、リカバリ・カタログ・ビューについて説明します。リカバリ・カタログが作成されている場合に限り、リカバリ・カタログ・ビューにアクセスできます。

注意： これらのビューは正規化されていませんが、RMAN での使用に最適化されています。したがってカタログ・ビューを結合した結果、冗長な値が生じる場合があります。

RC_ARCHIVED_LOG

このビューは、すべてのアーカイブ REDO ログについての履歴情報をリストします。これは、制御ファイルにおける VSARCHIVED_LOG 動的パフォーマンス・ビューに対応します。

アーカイブ REDO ログ・レコードは、オンライン REDO ログが正常にアーカイブされるか、または消去された（消去された場合、NAME 列は NULL）後で挿入されます。ログが複数回（最高 5 回まで）アーカイブされると、THREAD#、SEQUENCE#、および FIRST_CHANGE# が同じで、名前が異なる複数のアーカイブ・ログ・レコードが挿入されます。アーカイブ・ログ・レコードは、アーカイブ・ログがバックアップ・セットまたはコピーから復元されたときにも挿入されます。

アーカイブ REDO ログは、制御ファイル内でそのログ履歴レコードが上書きされるか、または RESETLOGS 操作の後では、ログ履歴レコードが存在しない場合があるので注意してください。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	ターゲット・データベースの主キー。この列を使用して、他のほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	数値	NOT NULL	ターゲット・データベースのインカーネーションの主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2(8)	NOT NULL	このレコードが属するデータベースのインカーネーションの DB_NAME。
AL_KEY	数値	NOT NULL	リカバリ・カタログのアーカイブ REDO ログの主キー。リカバリ・カタログに接続している間に list コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	数値	NOT NULL	VSARCHIVED_LOG からのアーカイブ REDO ログの RECID。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
STAMP	数値	NOT NULL	VSARCHIVED_LOG からのアーカイブ REDO ログの STAMP。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
THREAD#	数値	NOT NULL	REDO スレッドの番号。
SEQUENCE#	数値	NOT NULL	ログ順序番号。
RESETLOGS_CHANGE#	数値	NOT NULL	レコードを作成したときの最新のリセットログ SCN。
RESETLOGS_TIME	日付	NOT NULL	レコードを作成したときの最新のリセットログ・タイムスタンプ。
FIRST_CHANGE#	数値	NOT NULL	REDO ログをスイッチしたときの SCN。
FIRST_TIME	日付	NOT NULL	REDO ログをスイッチしたときの時刻。

列	データ型	NULL	説明
NEXT_CHANGE#	数値	NOT NULL	スレッド内の次の REDO ログの最初の SCN。
NEXT_TIME	日付		スレッド内の次の REDO ログの最初のタイムスタンプ。
BLOCKS	数値	NOT NULL	書き込んだブロック数（およびコピーを作成したときのアーカイブ・ログのサイズ）。
BLOCK_SIZE	数値	NOT NULL	バイト単位でのブロック・サイズ。
COMPLETION_TIME	日付	NOT NULL	REDO ログをアーカイブまたはコピーした時刻。
ARCHIVED	VARCHAR2(3)		ログがアーカイブされたかどうかを示す。YES（REDO ログがアーカイブされた）または NO（オンライン・REDO ログが検査された）。Oracle ではオンライン・ログを検査するとき、ファイル・ヘッダーを調べ、各オンライン・ログに対してアーカイブ・ログ・レコードが追加されます。また、ARCHIVED を NO に設定して、オンライン・ログのバックアップをとらないようにします。
STATUS	VARCHAR2(1)	NOT NULL	アーカイブ・ログの状態。A（使用可能）、U（使用不能）または D（削除済み）。

RC_BACKUP_CONTROLFILE

このビューは、バックアップ・セット内の制御ファイルに関する情報をリストします。ファイル番号 0 のバックアップ・データ・ファイル・レコードは、V\$BACKUP_DATAFILE 動的性能ビューのバックアップ制御ファイルを表しますので注意してください。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	ターゲット・データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	数値	NOT NULL	ターゲット・データベースのインカーネーションの主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2(8)	NOT NULL	このレコードが属するデータベースのインカーネーションの DB_NAME。
BCF_KEY	数値	NOT NULL	リカバリ・カタログの制御ファイル・バックアップの主キー。リカバリ・カタログに接続している間に list コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	数値	NOT NULL	V\$BACKUP_DATAFILE の RECID。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。

列	データ型	NULL	説明
STAMP	数値	NOT NULL	V\$BACKUP_DATAFILE の STAMP。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
BS_KEY	数値	NOT NULL	リカバリ・カタログでこのレコードが属するバックアップ・セットの主キー。この列を使用して RC_BACKUP_SET との結合を構成します。
SET_STAMP	数値	NOT NULL	V\$BACKUP_SET からの SET_STAMP の値。SET_STAMP と SET_COUNT が、ターゲット・データベース制御ファイルでこのレコードが属するバックアップ・セットを識別する一意の連結キーとなります。
SET_COUNT	数値	NOT NULL	V\$BACKUP_SET からの SET_COUNT の値。SET_STAMP と SET_COUNT が、ターゲット・データベース制御ファイルでこのレコードが属するバックアップ・セットを識別する一意の連結キーとなります。
RESETLOGS_CHANGE#	数値	NOT NULL	レコードを作成したときの最新のリセットログ SCN。
RESETLOGS_TIME	日付	NOT NULL	レコードを作成したときの最新のリセットログ・タイムスタンプ。
CHECKPOINT_CHANGE#	数値	NOT NULL	制御ファイルのチェックポイント SCN。
CHECKPOINT_TIME	日付	NOT NULL	制御ファイルのチェックポイント時刻。
CREATION_TIME#	日付	NOT NULL	制御ファイルの作成時刻。
BLOCK_SIZE	数値	NOT NULL	ブロック・サイズ (バイト単位)。
OLDEST_OFFLINE_RANGE	数値	NOT NULL	内部使用のみ。
STATUS	VARCHAR2(1)	NOT NULL	バックアップ・セットの状態。A (使用可能)、U (使用不能) または D (削除済み)。

RC_BACKUP_CORRUPTION

このビューは、データ・ファイル・バックアップの破損ブロック範囲をリストします。これは、制御ファイルの V\$BACKUP_CORRUPTION 動的性能ビューに対応します。制御ファイルおよびアーカイブ REDO ログのバックアップには破損が許されないことに注意してください。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	ターゲット・データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	数値	NOT NULL	ターゲット・データベースのインカーネーションの主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。

列	データ型	NULL	説明
DB_NAME	VARCHAR2(8)	NOT NULL	このレコードが属するデータベースのインカーネーションの DB_NAME。
RECID	数値	NOT NULL	V\$BACKUP_CORRUPTION からレコード識別子。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
STAMP	数値	NOT NULL	V\$BACKUP_CORRUPTION から波及したスタンプ。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
BS_KEY	数値	NOT NULL	リカバリ・カタログでこのレコードが属するバックアップ・セットの主キー。この列を使用して RC_BACKUP_SET との結合を構成します。
SET_STAMP	数値	NOT NULL	V\$BACKUP_SET から SET_STAMP の値。SET_STAMP と SET_COUNT が、ターゲット・データベース制御ファイルでこのレコードが属するバックアップ・セットを識別する一意の連結キーとなります。
SET_COUNT	数値	NOT NULL	V\$BACKUP_SET の SET_COUNT の値。SET_STAMP と SET_COUNT が、ターゲット・データベース制御ファイルでこのレコードが属するバックアップ・セットを識別する一意の連結キーとなります。
PIECE#	数値	NOT NULL	ブロックが属するバックアップ・ピース。
BDF_KEY	数値	NOT NULL	リカバリ・カタログにあるデータ・ファイルのバックアップまたはコピーの主キー。このキーを使用して RC_BACKUP_DATAFILE と結合します。リカバリ・カタログに接続している間に list コマンドを発行すると、この値が出力の KEY 列に表示されます。
BDF_RECID	数値	NOT NULL	V\$BACKUP_DATAFILE から RECID の値。
BDF_STAMP	数値	NOT NULL	V\$BACKUP_DATAFILE から STAMP の値。
FILE#	数値	NOT NULL	データ・ファイルの絶対ファイル番号。
CREATION_CHANGE#	数値	NOT NULL	バックアップ作成時の SCN。
BLOCK#	数値	NOT NULL	ファイルの最初の破損ブロックのブロック番号。
BLOCKS	数値	NOT NULL	このブロック番号以降に発見された破損ブロックの数。
CORRUPTION_CHANGE#	数値		破損が検出されたときの SCN。
MARKED_CORRUPT	VARCHAR2(3)		この破損をこれまでに検出していなければ YES、すでに認識済みであれば NO。

RC_BACKUP_DATAFILE

このビューは、バックアップ・セットのデータ・ファイルについての情報をリストします。これは、V\$BACKUP_DATAFILE 動的性能ビューに対応します。バックアップ・データ・ファイルは BDF_KEY によって一意に識別されます。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	ターゲット・データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	数値	NOT NULL	ターゲット・データベースのインカーネーションの主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2(8)	NOT NULL	このレコードが属するデータベースのインカーネーションの DB_NAME。
BDF_KEY	数値	NOT NULL	リカバリ・カタログのデータ・ファイル・バックアップの主キー。リカバリ・カタログに接続している間に list コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	数値	NOT NULL	V\$BACKUP_DATAFILE からのバックアップ・データ・ファイルの RECID。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
STAMP	数値	NOT NULL	V\$BACKUP_DATAFILE からのバックアップ・データ・ファイルの STAMP。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
BS_KEY	数値	NOT NULL	リカバリ・カタログでこのレコードが属するバックアップ・セットの主キー。この列を使用して RC_BACKUP_SET との結合を構成します。
SET_STAMP	数値	NOT NULL	V\$BACKUP_SET からの SET_STAMP の値。SET_STAMP と SET_COUNT が、ターゲット・データベース制御ファイルでこのレコードが属するバックアップ・セットを識別する一意の連結キーとなります。
SET_COUNT	数値	NOT NULL	V\$BACKUP_SET からの SET_COUNT の値。SET_STAMP と SET_COUNT が、ターゲット・データベース制御ファイルでこのレコードが属するバックアップ・セットを識別する一意の連結キーとなります。
BS_RECID	数値	NOT NULL	V\$BACKUP_SET からの RECID。
BS_STAMP	数値	NOT NULL	V\$BACKUP_SET からの STAMP。
BACKUP_TYPE	VARCHAR2(1)	NOT NULL	バックアップのタイプ。D (全体またはレベル 0 増分) または I (増分レベル 1 以上)。
INCREMENTAL_LEVEL	数値		増分バックアップのレベル。NULL または 0 から 4。

列	データ型	NULL	説明
COMPLETION_TIME	日付		バックアップの完了時刻。
FILE#	数値	NOT NULL	データ・ファイルの絶対ファイル番号。
CREATION_CHANGE#	数値	NOT NULL	バックアップ作成時の SCN。
RESETLOGS_CHANGE#	数値	NOT NULL	データ・ファイル・ヘッダーの最新リセットログ SCN。
RESETLOGS_TIME	日付	NOT NULL	データ・ファイル・ヘッダーの最新リセットログ・タイムスタンプ。
INCREMENTAL_CHANGE#	数値	NOT NULL	ブロックを増分バックアップに組み込むかどうかを判断する SCN。ブロック・ヘッダーにある SCN が INCREMENTAL_CHANGE# より大きい、または等しい場合に限って、ブロックは組み込まれます。 増分バックアップが適用される REDO の範囲は、INCREMENTAL_CHANGE# から始まり、CHECKPOINT_CHANGE# で終わります。
CHECKPOINT_CHANGE#	数値	NOT NULL	最新のデータ・ファイル・チェックポイント SCN。
CHECKPOINT_TIME	日付	NOT NULL	最後にデータ・ファイルのチェックポイントを行った時刻。
ABSOLUTE_FUZZY_CHANGE#	数値		絶対ファジー SCN。
DATAFILE_BLOCKS	数値	NOT NULL	データ・ファイルのブロック数。
BLOCKS	数値	NOT NULL	バックアップに書き込まれたデータ・ブロック数。
BLOCK_SIZE	数値	NOT NULL	データ・ブロック・サイズ (バイト単位)。
STATUS	VARCHAR2(1)	NOT NULL	バックアップ・セットの状態。A (すべてのピースが使用可能)、D (すべてのピースが削除済み)、O (いくつかのピースは使用可能だが、その他は使用不能なので、バックアップ・セットは使用できない)。

RC_BACKUP_PIECE

このビューは、バックアップ・ピースについての情報をリストします。これは、V\$BACKUP_PIECE 動的性能ビューに対応します。各バックアップには1つ以上のバックアップ・ピースが含まれます。

同じバックアップ・ピースのコピーを複数もつことができますが、制御ファイルには各コピーのレコード、ビューには各コピーの行ができます。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	ターゲット・データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DB_ID	数値	NOT NULL	データベース識別子。
BP_KEY	数値	NOT NULL	リカバリ・カタログのバックアップ・ピースの主キー。リカバリ・カタログに接続している間に list コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	数値	NOT NULL	V\$BACKUP_PIECE からのバックアップ・ピースの RECID。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
STAMP	数値	NOT NULL	V\$BACKUP_PIECE から波及したバックアップ・ピースの STAMP。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
BS_KEY	数値	NOT NULL	リカバリ・カタログでこのレコードが属するバックアップ・セットの主キー。この列を使用して RC_BACKUP_SET との結合を構成します。
SET_STAMP	数値	NOT NULL	V\$BACKUP_SET からの SET_STAMP の値。SET_STAMP と SET_COUNT が、ターゲット・データベース制御ファイルでこのレコードが属するバックアップ・セットを識別する一意の連結キーとなります。
SET_COUNT	数値	NOT NULL	V\$BACKUP_SET からの SET_COUNT の値。SET_STAMP と SET_COUNT が、ターゲット・データベース制御ファイルでこのレコードが属するバックアップ・セットを識別する一意の連結キーとなります。
BACKUP_TYPE	VARCHAR2(1)	NOT NULL	バックアップのタイプ。D (全体またはレベル 0 増分)、I (増分レベル 1 以上)、L (アーカイブ REDO ログ)。
INCREMENTAL_LEVEL	数値		増分バックアップのレベル。NULL または 0 から 4。
PIECE#	数値	NOT NULL	バックアップ・ピースの番号。最初のピースの値は 1 です。
COPY#	数値	NOT NULL	バックアップ・ピースのコピー番号。

列	データ型	NULL	説明
DEVICE_TYPE	VARCHAR2(255)	NOT NULL	バックアップ・デバイスのタイプ。ディスクまたは SBT_TAPE (順次メディア)。
HANDLE	VARCHAR2(1024)	NOT NULL	バックアップ・ピースのファイル名。これは RMAN が OSD レイヤーに渡す値で、ファイルを識別します。
COMMENTS	VARCHAR2(255)		バックアップ・ピースについてのコメント。
MEDIA	VARCHAR2(80)		このバックアップのメディア・マネージャについての詳細コメント。
MEDIA_POOL	数値		バックアップを格納するメディア・プールの数。
CONCUR	VARCHAR2(3)		バックアップ・メディアが同時アクセスをサポートするかどうかを指定する。YES または NO。
TAG	VARCHAR2(32)		バックアップ・ピースのユーザー指定タグ。
START_TIME	日付	NOT NULL	RMAN がバックアップ・ピースの書き込みを開始した時刻。
COMPLETION_TIME	日付	NOT NULL	バックアップ・ピースの書き込みが完了した時刻。
ELAPSED_SECONDS	数値		バックアップ・ピース作成の実行時間。
STATUS	VARCHAR2(1)	NOT NULL	バックアップ・ピースの状態。A (使用可能)、U (使用不能)、D (削除済み) または X (期限切れ)。

RC_BACKUP_REDOLOG

このビューは、バックアップ・セットのアーカイブ REDO ログについての情報をリストします。これは、制御ファイルの V\$BACKUP_REDOLOG 動的性能ビューに対応します。

オンライン・ログを直接にバックアップすることはできません。まずオンライン・ログをディスクにアーカイブし、次にバックアップを作成します。アーカイブ・ログのバックアップ・セットには、1 つ以上のアーカイブ・ログが含まれます。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	ターゲット・データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	数値	NOT NULL	ターゲット・データベースのインカーネーションの主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2(8)	NOT NULL	このレコードが属するデータベースのインカーネーションの DB_NAME。
BRL_KEY	数値	NOT NULL	リカバリ・カタログのアーカイブ REDO ログの主キー。リカバリ・カタログに接続している間に list コマンドを発行すると、この値が出力の KEY 列に表示されます。

列	データ型	NULL	説明
RECID	数値	NOT NULL	V\$BACKUP_REDOLOG から波及したレコード識別子。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
STAMP	数値	NOT NULL	V\$BACKUP_REDOLOG からの STAMP。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
BS_KEY	数値	NOT NULL	リカバリ・カタログでこのレコードが属するバックアップ・セットの主キー。この列を使用して、RC_BACKUP_SET との結合を構成します。
SET_STAMP	数値	NOT NULL	V\$BACKUP_SET からの SET_STAMP の値。SET_STAMP と SET_COUNT が、ターゲット・データベース制御ファイルでこのレコードが属するバックアップ・セットを識別する一意の連結キーとなります。
SET_COUNT	数値	NOT NULL	V\$BACKUP_SET からの SET_COUNT の値。SET_STAMP と SET_COUNT が、ターゲット・データベース制御ファイルでこのレコードが属するバックアップ・セットを識別する一意の連結キーとなります。
BACKUP_TYPE	VARCHAR2(1)	NOT NULL	バックアップのタイプ。L (アーカイブ REDO ログ)
COMPLETION_TIME	日付	NOT NULL	バックアップが完了した時刻。
THREAD#	数値	NOT NULL	REDO ログのスレッド番号。
SEQUENCE#	数値	NOT NULL	ログ順序番号。
RESETLOGS_CHANGE#	数値	NOT NULL	レコードを作成したときの最新のリセットログ SCN。
RESETLOGS_TIME	日付	NOT NULL	レコードを作成したときの最新のリセットログ・タイムスタンプ。
FIRST_CHANGE#	数値	NOT NULL	REDO ログにスイッチした SCN。
FIRST_TIME	日付	NOT NULL	REDO ログにスイッチしたときの時刻。
NEXT_CHANGE#	数値	NOT NULL	スレッドの次の REDO ログの最初の SCN。
NEXT_TIME	日付	NOT NULL	スレッドの次の REDO ログの最初のタイムスタンプ。
BLOCKS	数値	NOT NULL	バックアップに書き込まれたデータ・ブロック数。
STATUS	VARCHAR2(1)	NOT NULL	バックアップ・セットの状態。A (すべてのピースが使用可能)、D (すべてのピースが削除済み)、O (いくつかのピースは使用可能だが、使用不能なので、バックアップ・セットは使用できない)。

RC_BACKUP_SET

このビューは、データベースのすべてのインカーネーションのバックアップ・セットについての情報をリストします。これは、V\$BACKUP_SET 動的性能ビューに対応します。バックアップ・セット・レコードは、バックアップが正常に完了した後に挿入されます。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	ターゲット・データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DB_ID	数値	NOT NULL	一意のデータベース識別子。
BS_KEY	数値	NOT NULL	リカバリ・カタログのバックアップ・セットの主キー。リカバリ・カタログに接続している間に list コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	数値	NOT NULL	V\$BACKUP_SET からのバックアップ・セットの RECID。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。V\$BACKUP_SET にアクセスするには、RECID と STAMP または SET_STAMP と SET_COUNT の組み合わせのどちらかを使用します。
STAMP	数値	NOT NULL	V\$BACKUP_SET からのバックアップ・セットの STAMP。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。V\$BACKUP_SET にアクセスするには、RECID と STAMP または SET_STAMP と SET_COUNT の組み合わせのどちらかを使用します。
SET_STAMP	数値	NOT NULL	V\$BACKUP_SET からの SET_STAMP の値。SET_STAMP と SET_COUNT が、ターゲット・データベースの制御ファイルにあるこのレコードを識別する一意の連結キーを構成します。V\$BACKUP_SET にアクセスするには、RECID と STAMP または SET_STAMP と SET_COUNT の組み合わせのどちらかを使用します。
SET_COUNT	数値	NOT NULL	V\$BACKUP_SET からの SET_COUNT の値。SET_STAMP と SET_COUNT が、ターゲット・データベースの制御ファイルにあるこのレコードを識別する一意の連結キーを構成します。V\$BACKUP_SET にアクセスするには、RECID と STAMP または SET_STAMP と SET_COUNT の組み合わせのどちらかを使用します。
BACKUP_TYPE	VARCHAR2(1)	NOT NULL	バックアップのタイプ。D (全体またはレベル 0 増分)、I (増分レベル 1 以上)、L (アーカイブ REDO ログ)。
INCREMENTAL_LEVEL	数値	NOT NULL	増分バックアップのレベル。NULL または 0 から 4。
PIECES	数値	NOT NULL	バックアップ・セットのバックアップ・ピースの数。
START_TIME	日付	NOT NULL	バックアップを開始した時刻。

列	データ型	NULL	説明
COMPLETION_TIME	日付	NOT NULL	バックアップが完了した時刻。
ELAPSED_SECONDS	数値		バックアップ実行時間（秒単位）。
STATUS	VARCHAR2(1)	NOT NULL	バックアップ・セットの状態。A（すべてのピースが使用可能）、D（すべてのピースが削除済み）、O（いくつかのピースは使用可能だが、使用不能なので、バックアップ・セットは使用できない）。

RC_CHECKPOINT

このビューは、リカバリ・カタログの再同期化情報をリストします。データベースのチェックポイントについての情報はリストしません。チェックポイントの情報には、ほとんどの場合、RC_RESYNC を使用します。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	ターゲット・データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	数値	NOT NULL	ターゲット・データベースのインカーネーションの主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2(8)	NOT NULL	このレコードが属するデータベースのインカーネーションの DB_NAME。
CKP_KEY	数値	NOT NULL	チェックポイントの主キー。
CKP_SCN	数値	NOT NULL	制御ファイルのチェックポイントの SCN。
CKP_CF_SEQ	数値	NOT NULL	制御ファイルの順序番号。
CKP_TIME	日付		カタログのチェックポイントを行ったときの時刻。
CKP_TYPE	VARCHAR2(7)	NOT NULL	再同期化のタイプ。全体または部分。

RC_CONTROLFILE_COPY

このビューは、ディスク上の制御ファイル・コピーについての情報をリストします。
V\$DATAFILE_COPY 動的性能ビューで制御ファイルのコピーを表示するには、ファイル番号 0 のデータ・ファイル・コピー・レコードを使用する必要がありますので注意してください。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	ターゲット・データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。

列	データ型	NULL	説明
DBINC_KEY	数値	NOT NULL	ターゲット・データベースのインカーネーションの主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2(8)	NOT NULL	このレコードが属するデータベースのインカーネーションの DB_NAME。
CCF_KEY	数値	NOT NULL	リカバリ・カタログの制御ファイル・コピーの主キー。リカバリ・カタログに接続している間に list コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	数値	NOT NULL	V\$DATAFILE_COPY からのレコード識別子。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
STAMP	数値	NOT NULL	V\$BACKUP_COPY からの STAMP。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
NAME	VARCHAR2(1024)	NOT NULL	制御ファイル・コピーのファイル名。
TAG	VARCHAR2(32)		制御ファイル・コピーのタグ。タグを使用しない場合は、NULL。
RESETLOGS_CHANGE#	数値	NOT NULL	レコードを作成したときの最新のリセットログ SCN。
RESETLOGS_TIME	日付	NOT NULL	レコードを作成したときの最新のリセットログ・タイムスタンプ。
CHECKPOINT_CHANGE#	数値	NOT NULL	制御ファイルのチェックポイント SCN。
CHECKPOINT_TIME	日付	NOT NULL	制御ファイルのチェックポイント時刻。
CREATION_TIME	日付	NOT NULL	制御ファイルの作成時刻。
BLOCK_SIZE	数値	NOT NULL	ブロック・サイズ (バイト単位)。
MIN_OFFR_RECID	数値	NOT NULL	内部使用のみ。
OLDEST_OFFLINE_RANGE	数値	NOT NULL	内部使用のみ。
COMPLETION_TIME	日付	NOT NULL	コピーを生成したときの時刻。
STATUS	VARCHAR2(1)	NOT NULL	コピーの状態。A (使用可能) U (使用不能) または D (削除済み)

RC_COPY_CORRUPTION

このビューは、データ・ファイル・コピーの破損ブロック範囲をリストします。これは、V\$COPY_CORRUPTION 動的性能ビューに対応します。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	ターゲット・データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	数値	NOT NULL	ターゲット・データベースのインカーネーションの主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2(8)	NOT NULL	このレコードが属するデータベースのインカーネーションの DB_NAME。
RECID	数値	NOT NULL	V\$COPY_CORRUPTION からレコード識別子。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
STAMP	数値	NOT NULL	V\$COPY_CORRUPTION から波及したスタンプ。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
CDF_KEY	数値	NOT NULL	リカバリ・カタログのデータ・ファイル・コピーの主キー。リカバリ・カタログに接続している間に list コマンドを発行すると、この値が出力の KEY 列に表示されます。この列を使用して、RC_DATAFILE_COPY と結合を構成します。
COPY_RECID	数値	NOT NULL	RC_DATAFILE_COPY からレCID。この値は、制御ファイルから波及します。
COPY_STAMP	数値	NOT NULL	RC_DATAFILE_COPY からレCID。この値は、制御ファイルから波及します。
FILE#	数値	NOT NULL	データ・ファイルの絶対ファイル番号。
CREATION_CHANGE#	数値	NOT NULL	コピー作成時に記録された SCN。
BLOCK#	数値	NOT NULL	ファイルの最初の破損ブロックのブロック番号。
BLOCKS	数値	NOT NULL	このブロック番号以降に発見された破損ブロックの数。
CORRUPTION_CHANGE#	数値		破損が検出されたときの SCN。
MARKED_CORRUPT	VARCHAR2(3)		この破損をこれまでに検出していなければ YES、すでに認識済みであれば NO。

RC_DATABASE

このビューは、リカバリ・カタログに登録されたデータベースについての情報を表示します。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	数値		現行のインカーネーションの主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DBID	数値	NOT NULL	V\$DATABASE からのデータベースの一意の識別子。
NAME	VARCHAR2(8)	NOT NULL	現行のインカーネーションに対応するデータベースの DB_NAME。
RESETLOGS_CHANGE#	数値	NOT NULL	レコードを作成したときの最新のリセットログ SCN。
RESETLOGS_TIME	日付	NOT NULL	レコードを作成したときの最新のリセットログ・タイムスタンプ。

RC_DATABASE_INCARNATION

このビューは、リカバリ・カタログに登録されたすべてのデータベースのインカーネーションについての情報をリストします。Oracle では、ユーザーがデータベースを RESETLOGS オプションでオープンすると、常に新しいインカーネーションが作成されます。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBID	数値	NOT NULL	データベースの一意の識別子。
DBINC_KEY	数値	NOT NULL	インカーネーションの主キー。
NAME	VARCHAR2(8)	NOT NULL	RESETLOGS 操作の時点のデータベースの DB_NAME。
RESETLOGS_CHANGE#	数値	NOT NULL	このインカーネーションを作成したりセットログ SCN。
RESETLOGS_TIME	日付	NOT NULL	このインカーネーションを作成したりセットログ・タイムスタンプ。
CURRENT_INCARNATION	VARCHAR2(3)		現行のインカーネーションの場合は YES、違う場合は NO。
PARENT_DBINC_KEY	数値		このデータベースの前のインカーネーションの DBINC_KEY。データベースの最初に記録されたインカーネーションの場合、値は NULL になります。

RC_DATAFILE

このビューは、リカバリ・カタログに登録されたすべてのデータ・ファイルについての情報をリストします。これは、VSDATAFILE 動的性能ビューに対応します。データ・ファイルは、その表領域が削除された場合には、削除されたものとして表示されます。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	ターゲット・データベースの主キー。
DBINC_KEY	数値	NOT NULL	ターゲット・データベースのインカーネーションの主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2(8)	NOT NULL	このレコードが属するデータベースのインカーネーションの DB_NAME。
TS#	数値	NOT NULL	ターゲット・データベースでの表領域識別子。表領域を削除し、再作成した場合、同じインカーネーションに複数の TS# が存在します。
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	表領域名。表領域を削除し、再作成した場合、同じインカーネーションに複数の名前が存在します。
FILE#	数値	NOT NULL	データ・ファイルの絶対ファイル番号。表領域を削除し、再作成した場合、同じインカーネーションに複数の同一データ・ファイル番号が存在します。
CREATION_CHANGE#	数値	NOT NULL	データ・ファイル作成時の SCN。
CREATION_TIME	日付		データ・ファイル作成の時刻。
DROP_CHANGE#	数値		データ・ファイルを削除したときに記録された SCN。同じファイル番号を持つ新規のデータ・ファイルが発見された場合、DROP_CHANGE# はそのデータ・ファイルの CREATION_CHANGE# に設定されます。その他の場合は、値は RC_CHECKPOINT.CKP_SCN に設定されます。
DROP_TIME	日付		データファイルを削除したときの時刻。同じファイル番号を持つ新規のデータ・ファイルが発見された場合、DROP_TIME はそのデータ・ファイルの CREATION_TIME に設定されます。その他の場合は、値は RC_CHECKPOINT.CKP_TIME に設定されます。
BYTES	数値		データ・ファイルのサイズ (バイト単位)。
BLOCKS	数値		データ・ファイルのブロック数。
BLOCK_SIZE	数値	NOT NULL	データ・ブロックのサイズ。
NAME	VARCHAR2(1024)		データ・ファイルのファイル名。
STOP_CHANGE#	数値		データ・ファイルが通常オフラインまたは読み取り専用の場合、データ・ファイルの SCN。

列	データ型	NULL	説明
READ_ONLY	数値	NOT NULL	STOP_CHANGE# が読取り専用の場合は 1、その他の場合は 0。

RC_DATAFILE_COPY

このビューは、ディスク上のデータ・ファイル・コピーについての情報をリストします。これは、V\$DATAFILE_COPY 動的性能ビューに対応します。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	ターゲット・データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	数値	NOT NULL	ターゲット・データベースのインカーネーションの主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2(8)	NOT NULL	このレコードが属するデータベースのインカーネーションの DB_NAME。
CDF_KEY	数値	NOT NULL	リカバリ・カタログのデータ・ファイル・コピーの主キー。リカバリ・カタログに接続している間に list コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	数値	NOT NULL	V\$DATAFILE_COPY からのデータ・ファイル・コピー・レコード。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
STAMP	数値	NOT NULL	V\$DATAFILE_COPY からのデータ・ファイル・コピーのスタンプ。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
NAME	VARCHAR2(1024)	NOT NULL	データ・ファイル・コピーのファイル名。
TAG	VARCHAR2(32)		データ・ファイル・コピーのタグ。
FILE#	数値	NOT NULL	データ・ファイルの絶対ファイル番号。
CREATION_CHANGE#	数値	NOT NULL	データ・ファイル・コピー作成時の SCN。
RESETLOGS_CHANGE#	数値	NOT NULL	データ・ファイル・ヘッダーの最新リセットログ SCN。
RESETLOGS_TIME	日付	NOT NULL	データ・ファイル・ヘッダーの最新リセットログ・タイムスタンプ。
INCREMENTAL_LEVEL	数値		コピーの増分レベル。0 または NULL。
CHECKPOINT_CHANGE#	数値	NOT NULL	最新のデータ・ファイル・チェックポイント SCN。
CHECKPOINT_TIME	日付	NOT NULL	最新のデータ・ファイル・チェックポイント時刻。

列	データ型	NULL	説明
ABSOLUTE_FUZZY_CHANGE#	数値		認識されている場合、ファイルのいずれかのブロックについている一番大きい値の SCN。
RECOVERY_FUZZY_CHANGE#	数値		ファイルをファジーにしないために、回復操作の到達点とするべき SCN。NULL でない場合、このファイルでデータベースをオープンする前に、少なくとも指定された SCN までの回復が必要です。
RECOVERY_FUZZY_TIME	日付		ファイルをファジーにしないために、回復操作を続けて行うべき時刻。NULL でない場合、このファイルでデータベースをオープンする前に、少なくとも指定された時刻までの回復が必要です。
ONLINE_FUZZY	VARCHAR2(3)		YES または NO。YES に設定されている場合、このコピーはクラッシュまたは immediate を指定したオフラインの後に作成されたものです（あるいはデータベースがオープンされている間に不適切に作成されたコピーのコピーです）。ファイルに一貫性を持たせるには、次のクラッシュ回復マーカまでのすべての REDO を適用する回復が必要です。
BACKUP_FUZZY	VARCHAR2(3)		YES または NO。YES に設定されている場合、このコピーは BEGIN BACKUP/END BACKUP テクニックを使用して作成されています。このコピーに一貫性を持たせるには、ALTER TABLESPACE END BACKUP コマンドを使用するときに REDO ストリームに置かれたマーカまで、すべての REDO を適用する回復が必要です。
BLOCKS	数値	NOT NULL	データ・ファイル・コピーにあるブロックの数（およびコピーを作成したときのデータ・ファイルのサイズ）。
BLOCK_SIZE	数値	NOT NULL	ブロック・サイズ（バイト単位）。
COMPLETION_TIME			コピーが完了した時刻。
STATUS	VARCHAR2(1)	NOT NULL	コピーの状態。A（使用可能）、U（使用不能）または D（削除済み）。

RC_LOG_HISTORY

このビューは、オンライン REDO ログについての履歴情報をリストします。これは、VSLOG_HISTORY 動的性能ビューに対応します。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	ターゲット・データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	数値	NOT NULL	ターゲット・データベースのインカーネーションの主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2(8)	NOT NULL	このレコードが属するデータベースのインカーネーションの DB_NAME。
RECID	数値	NOT NULL	VSLOG_HISTORY からの REDO ログ履歴の RECID。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
STAMP	数値	NOT NULL	VSLOG_HISTORY からの REDO ログ履歴のスタンプ。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
THREAD#	数値	NOT NULL	オンライン REDO ログのスレッド番号。
SEQUENCE#	数値	NOT NULL	REDO ログのログ順序番号。
FIRST_CHANGE#	数値	NOT NULL	REDO ログにスイッチしたときに生成した SCN。
FIRST_TIME	日付	NOT NULL	REDO ログにスイッチしたときのタイムスタンプ。
NEXT_CHANGE#	数値	NOT NULL	スレッドの次の REDO ログの最初の SCN。
CLEARED	VARCHAR2(3)		REDO ログが ALTER DATABASE CLEAR LOGFILE 文で消去された場合は '?'、その他の場合は NULL。この文を使用すると、ログを最初にアーカイブしなくても初期化できます。

RC_OFFLINE_RANGE

このビューはデータ・ファイルのオフライン範囲をリストします。これは、VSOFFLINE_RANGE 動的性能ビューに対応します。

データ・ファイルにオフライン範囲が作成されるのは、まずその表領域が NORMAL モードによるオフラインまたは読取り専用に変更され、そしてその後にオンラインまたは読み書き可能に変更されたときです。データ・ファイル自体がオフラインに変更された場合、あるいは表領域が IMMEDIATE モードでオフラインに変更される場合には、オフライン範囲は作成されません。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	ターゲット・データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	数値	NOT NULL	ターゲット・データベースのインカーネーションの主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2(8)	NOT NULL	このレコードが属するデータベースのインカーネーションの DB_NAME。
RECID	数値	NOT NULL	VSOFFLINE_RANGE からのオフライン範囲のレコード識別子。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
STAMP	数値	NOT NULL	VSOFFLINE_RANGE からのオフライン範囲のスタンプ。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
FILE#	数値	NOT NULL	データ・ファイルの絶対ファイル番号。
CREATION_CHANGE#	数値	NOT NULL	データ・ファイル作成時の SCN。
OFFLINE_CHANGE#	数値	NOT NULL	データ・ファイルをオフラインにしたときに記録された SCN。
ONLINE_CHANGE#	数値	NOT NULL	オンライン・チェックポイント SCN。
ONLINE_TIME	日付	NOT NULL	オンライン・チェックポイント時刻。
CF_CREATE_TIME	日付		制御ファイル作成の時刻。

RC_PROXY_CONTROLFILE

このビューでは、プロキシ・コピーで作成した制御ファイルのバックアップについて説明します。プロキシ・コピーでは、メディア・マネージャがデータのバックアップと復元の操作を行います。各行で、1 つの制御ファイルの 1 つのバックアップを説明します。このビューは、VSPROXY_DATAFILE に対応します。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	ターゲット・データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	数値	NOT NULL	ターゲット・データベースのインカーネーションの主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2(8)	NOT NULL	このレコードが属するデータベースのインカーネーションの DB_NAME。

列	データ型	NULL	説明
XDF_KEY	数値	NOT NULL	リカバリ・カタログのプロキシ・コピーの主キー。リカバリ・カタログに接続している間に list コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	数値	NOT NULL	VSPROXY_DATAFILE からのプロキシ・コピー・レコードの識別子。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
STAMP	数値	NOT NULL	V\$DATAFILE_COPY からのプロキシ・コピーのスタンプ。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
TAG	VARCHAR2(32)		プロキシ・コピーのタグ。
RESETLOGS_CHANGE#	数値	NOT NULL	レコードを作成したときの最新のリセットログ SCN。
RESETLOGS_TIME	日付	NOT NULL	レコードを作成したときの最新のリセットログ・タイムスタンプ。
CHECKPOINT_CHANGE#	数値	NOT NULL	コピーを作成したときのチェックポイント SCN。
CHECKPOINT_TIME	日付	NOT NULL	コピーを作成したときのチェックポイント時刻。
CREATION_TIME	日付	NOT NULL	制御ファイルの作成時刻。
BLOCK_SIZE	数値	NOT NULL	コピーのブロック・サイズ (バイト単位)。
MIN_OFFR_RECID	数値	NOT NULL	内部使用のみ。
OLDEST_OFFLINE_RANGE	数値	NOT NULL	内部使用のみ。
DEVICE_TYPE	VARCHAR2(255)	NOT NULL	順次メディア・デバイスのタイプ。
HANDLE	VARCHAR2(1024)	NOT NULL	プロキシ・コピーのファイル名。これは RMAN が OSD レイヤーに渡す値で、ファイルを識別します。
COMMENTS	VARCHAR2(255)		プロキシ・コピーについてのコメント。
MEDIA	VARCHAR2(80)		このバックアップを作成したメディア・マネージャについての詳細コメント。
MEDIA_POOL	数値		プロキシ・コピーを格納するメディア・プールの数。
START_TIME	日付	NOT NULL	コピーを開始したときの時刻。
COMPLETION_TIME	日付	NOT NULL	プロキシ・コピーが完了した時刻。
ELAPSED_SECONDS	数値		プロキシ・コピーの実行時間。
STATUS	VARCHAR2(1)	NOT NULL	バックアップ・セットの状態。A (使用可能) U (使用不能) X (期限切れ) または D (削除済み)。

RC_PROXY_DATAFILE

このビューでは、プロキシ・コピーで作成したデータ・ファイルのバックアップについて説明します。これは、VSPROXY_DATAFILE 動的性能ビューに対応します。プロキシ・コピーでは、メディア・マネージャがデータのバックアップと復元の操作を行います。各行で、1 データベース・ファイルの 1 バックアップを説明します。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	ターゲット・データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	数値	NOT NULL	ターゲット・データベースのインカーネーションの主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2(8)	NOT NULL	このレコードが属するデータベースのインカーネーションの DB_NAME。
XDF_KEY	数値	NOT NULL	リカバリ・カタログのプロキシ・コピーの主キー。リカバリ・カタログに接続している間に list コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	数値	NOT NULL	VSPROXY_DATAFILE からのプロキシ・コピー・レコードの識別子。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
STAMP	数値	NOT NULL	VSDATAFILE_COPY からのプロキシ・コピーのスタンプ。RECID と STAMP が、ターゲット・データベース制御ファイルでこのレコードを識別する一意の連結主キーとなります。
TAG	VARCHAR2(32)		プロキシ・コピーのタグ。
FILE#	数値	NOT NULL	プロキシ・コピーを作成したデータ・ファイルの絶対ファイル番号。
CREATION_CHANGE#	数値	NOT NULL	データ・ファイル作成時の SCN。
RESETLOGS_CHANGE#	数値	NOT NULL	データ・ファイル・ヘッダーの最新リセットログ SCN。
RESETLOGS_TIME	日付	NOT NULL	データ・ファイル・ヘッダーの最新リセットログ・タイムスタンプ。
INCREMENTAL_LEVEL	数値		このコピーが増分バックアップ計画の一部の場合は 0、その他の場合は NULL。
CHECKPOINT_CHANGE#	数値	NOT NULL	コピーを作成したときのチェックポイント SCN。
CHECKPOINT_TIME	日付	NOT NULL	コピーを作成したときのチェックポイント時刻。
ABSOLUTE_FUZZY_CHANGE#	数値		認識されている場合、ファイルのいずれかのブロックについている一番大きい値の SCN。

列	データ型	NULL	説明
RECOVERY_FUZZY_CHANGE#	数値		ファイルをファジーにしないために、回復操作の到達点とするべき SCN。NULL でない場合、このファイルでデータベースをオープンする前に、少なくとも指定された SCN までの回復が必要です。
RECOVERY_FUZZY_TIME	日付		ファイルをファジーにしないために、回復操作の到達点とするべきタイムスタンプ。NULL でない場合、このファイルでデータベースをオープンする前に、少なくとも指定された時刻までの回復が必要です。
ONLINE_FUZZY	VARCHAR2(3)		YES または NO。YES に設定されている場合、このコピーはクラッシュまたは IMMEDIATE モードによるオフラインの後に作成されたものです（あるいはデータベースがオープンされている間に不適切に作成されたコピーのコピーです）。ファイルに一貫性をもたせるには、次のクラッシュ回復マーカまでのすべての REDO を適用する回復が必要です。
BACKUP_FUZZY	VARCHAR2(3)		YES または NO。YES に設定されている場合、このコピーは BEGIN BACKUP/END BACKUP により作成されています。このコピーに一貫性をもたせるには、ALTER TABLESPACE END BACKUP コマンドを使用するときに REDO ストリームに置かれたマーカまで、すべての REDO を適用する回復が必要です。
BLOCKS	数値	NOT NULL	ブロック単位のデータ・ファイル・コピーのサイズ（およびコピーを作成したときのデータ・ファイルのサイズ）
BLOCK_SIZE	数値	NOT NULL	コピーのブロック・サイズ（バイト単位）
DEVICE_TYPE	VARCHAR2(255)	NOT NULL	順次メディア・デバイスのタイプ。
HANDLE	VARCHAR2(1024)	NOT NULL	プロキシ・コピーのファイル名。これは RMAN が OSD レイヤーに渡す値で、ファイルを識別します。
COMMENTS	VARCHAR2(255)		プロキシ・コピーについてのコメント。
MEDIA	VARCHAR2(80)		このバックアップを作成したメディア・マネージャについての詳細コメント。
MEDIA_POOL	数値		プロキシ・コピーを格納するメディア・プールの数。
START_TIME	日付	NOT NULL	コピーを開始したときの時刻。
COMPLETION_TIME	日付	NOT NULL	プロキシ・コピーが完了した時刻。
ELAPSED_SECONDS	数値		プロキシ・コピーの実行時間。
STATUS	VARCHAR2(1)	NOT NULL	バックアップ・セットの状態。A（使用可能） U（使用不能） X（期限切れ）または D（削除済み）

RC_REDO_LOG

このビューは、データベースのすべてのインカーネーションのオンライン REDO ログについての情報をリストします。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	ターゲット・データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	数値	NOT NULL	ターゲット・データベースのインカーネーションの主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2(8)	NOT NULL	このレコードが属するデータベースのインカーネーションの DB_NAME。
THREAD#	数値	NOT NULL	REDO スレッドの番号。
GROUP#	数値	NOT NULL	オンライン REDO ログ・グループの番号。
NAME	VARCHAR2(1024)	NOT NULL	オンライン REDO ログ・ファイルの番号。

RC_REDO_THREAD

このビューは、データベースのすべてのインカーネーションの REDO スレッドについての情報をリストします。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	ターゲット・データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	数値	NOT NULL	ターゲット・データベースのインカーネーションの主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2(8)	NOT NULL	このレコードが属するデータベースのインカーネーションの DB_NAME。
THREAD#	数値	NOT NULL	データベースのインカーネーションの REDO スレッド番号。
STATUS	VARCHAR2(1)	NOT NULL	REDO スレッドの状態。D（使用禁止） E（使用可能）または O（オープン）。
SEQUENCE#	数値	NOT NULL	最後に割り当てられたログ順序番号。
ENABLE_CHANGE#	数値		このスレッドが使用可能になったときの SCN。
ENABLE_TIME	日付		このスレッドが使用可能になったときの時刻。
DISABLE_CHANGE#	数値		スレッドが最後に使用禁止になった SCN。

列	データ型	NULL	説明
DISABLE_TIME	日付		スレッドが最後に使用禁止になった時刻。

RC_RESYNC

このビューは、リカバリ・カタログの再同期化についての情報をリストします。完全再同期化ではいつもターゲット・データベースの制御ファイルのスナップショットを作成し、このスナップショットからリカバリ・カタログを再同期化します。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	ターゲット・データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	数値	NOT NULL	ターゲット・データベースのインカーネーションの主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2(8)	NOT NULL	このレコードが属するデータベースのインカーネーションの DB_NAME。
RESYNC_KEY	数値	NOT NULL	再同期化の主キー。
CONTROLFILE_CHANGE#	数値	NOT NULL	カタログの再同期化における制御ファイル・チェックポイントの SCN。
CONTROLFILE_TIME	日付		カタログの再同期化における制御ファイル・チェックポイントのタイムスタンプ。
CONTROLFILE_SEQUENCE#	数値	NOT NULL	制御ファイルの順序番号。
CONTROLFILE_VERSION	日付	NOT NULL	カタログの再同期化の開始時点としての制御ファイルのバージョンの作成時刻。
RESYNC_TYPE	VARCHAR2(7)	NOT NULL	再同期化のタイプ。全体または部分。
DB_STATUS	VARCHAR2(7)		ターゲット・データベースの状態。オープンまたはマウント済み。
RESYNC_TIME	日付	NOT NULL	再同期化の時刻。

RC_STORED_SCRIPT

このビューは、リカバリ・カタログに格納されたスクリプトについての情報をリストします。このビューでは、1 ストアド・スクリプトの各行に 1 行の説明があります。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	このスクリプトを使用するデータベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DB_NAME	VARCHAR2(8)	NOT NULL	このレコードが属するデータベースのインカーネーションの DB_NAME。
SCRIPT_NAME	VARCHAR2(100)	NOT NULL	スクリプトの名前。

RC_STORED_SCRIPT_LINE

このビューは、リカバリ・カタログに格納されたスクリプトの行についての情報をリストします。このビューでは、1 ストアド・スクリプトの各行に 1 行の説明があります。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	このスクリプトを使用するデータベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
SCRIPT_NAME	VARCHAR2(100)	NOT NULL	ストアド・スクリプトの名前。
LINE	数値	NOT NULL	スクリプトの行番号。スクリプトの行は、SCRIPT_NAME および LINE で一意に識別します。
TEXT	VARCHAR2(1024)	NOT NULL	スクリプトの行のテキスト。

RC_TABLESPACE

このビューは、リカバリ・カタログに登録されたすべての表領域、すべての削除済み表領域、およびすべての旧データベースのインカーネーションに属する表領域についての情報をリストします。これは、V\$TABLESPACE 動的性能ビューに対応します。表領域属性の現在の値を表示します。

列	データ型	NULL	説明
DB_KEY	数値	NOT NULL	ターゲット・データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。

列	データ型	NULL	説明
DBINC_KEY	数値	NOT NULL	ターゲット・データベースのインカーネーションの主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2(8)	NOT NULL	このレコードが属するデータベースのインカーネーションの DB_NAME。
TS#	数値	NOT NULL	ターゲット・データベースでの表領域識別子。表領域を削除し、再作成した場合、同じインカーネーションに複数の TS# が存在します。
NAME	VARCHAR2(30)	NOT NULL	表領域名。表領域を削除し、再作成した場合、同じインカーネーションにその名前が複数存在する可能性があります。
CREATION_CHANGE#	数値	NOT NULL	作成時の SCN (最初のデータ・ファイルからの)。
CREATION_TIME	日付		表領域の作成時刻。制御ファイルを作成後にオフラインの表領域は NULL。
DROP_CHANGE#	数値		表領域を削除したときに記録された SCN。同じ TS# を持つ新規の表領域が発見された場合、DROP_CHANGE# はその表領域の CREATION_CHANGE# に設定されます。その他の場合は、値は RC_CHECKPOINT.CKP_SCN に設定されます。
DROP_TIME	日付		表領域を削除したときの時刻。

第 IV 部

オペレーティング・システム環境での
バックアップおよびリカバリの実行

オペレーティング・システム環境での バックアップの実行

Recovery Manager を使用しない場合には、オペレーティング・システムのユーティリティを使用してデータベースのバックアップを作成し、SQL*Plus を使用してデータ・ファイルを回復することができます。この章では、Oracle データベースをバックアップするための O/S コマンドの使用方法について説明します。ここでは次の項目を取り上げます。

- バックアップ実行前のデータベース・ファイルのリスト表示
- O/S のバックアップの実行
- オンライン表領域のバックアップ失敗からの回復
- 保護強化のための Export と Import の使用方法

バックアップ実行前のデータベース・ファイルのリスト表示

バックアップをとる前に、データベース内のすべてのファイルを確認します。次に、バックアップをとる必要があるファイルを確認します。

データ・ファイル、オンライン REDO ログおよび制御ファイルをリスト表示する方法

- 1. SQL*Plus を起動し、V\$DATAFILE に対して SELECT 文を発行し、データ・ファイルのリストを取得します。

```
SQL> SELECT name FROM v$datafile;
NAME
-----
/vobs/oracle/dbs/tbs_01.f
/vobs/oracle/dbs/tbs_02.f
/vobs/oracle/dbs/tbs_03.f
/vobs/oracle/dbs/tbs_11.f
/vobs/oracle/dbs/tbs_12.f
/vobs/oracle/dbs/tbs_21.f
/vobs/oracle/dbs/tbs_22.f
/vobs/oracle/dbs/tbs_23.f
/vobs/oracle/dbs/tbs_24.f
9 rows selected.
```

また、V\$TABLESPACE ビューと V\$DATAFILE ビューを結合すれば、データ・ファイルのリストに加えて関連する表領域も取得できます。

```
SQL> SELECT t.name "Tablespace", f.name "Datafile"
2> FROM v$tablespace t, v$datafile f
3> WHERE t.ts# = f.ts#
4> ORDER BY t.name;
```

Tablespace	Datafile
-----	-----
SYSTEM	/vobs/oracle/dbs/tbs_01.f
SYSTEM	/vobs/oracle/dbs/tbs_02.f
SYSTEM	/vobs/oracle/dbs/tbs_03.f
TBS_1	/vobs/oracle/dbs/tbs_11.f
TBS_1	/vobs/oracle/dbs/tbs_12.f
TBS_2	/vobs/oracle/dbs/tbs_21.f
TBS_2	/vobs/oracle/dbs/tbs_22.f
TBS_2	/vobs/oracle/dbs/tbs_23.f
TBS_2	/vobs/oracle/dbs/tbs_24.f

- 2. V\$LOGFILE ビューを使用し、オンライン REDO ログ・ファイルのリストを取得します。たとえば、次の問合せを発行します。

```
SELECT member FROM v$logfile;
MEMBER
```

```
-----
/vobs/oracle/dbs/tl_log1.f
/vobs/oracle/dbs/tl_log2.f
2 rows selected.
```

3. CONTROL_FILES パラメータを使用して、現行の制御ファイルの名前を取得します。たとえば、次の問合せを発行します。

```
SHOW PARAMETERS control_files;
NAME                                TYPE      VALUE
-----
control_files                      string    /vobs/oracle/dbs/cf1.f
```

4. BACKUP CONTROLFILE TO 'filename' オプションを指定し、ALTER DATABASE コマンドを実行して制御ファイルのバックアップをとるときは、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。

O/S のバックアップの実行

Oracle データベースのバックアップ作成のツールとしては Recovery Manager をお勧めしますが、また O/S ユーティリティを使用してバックアップをとることもできます。使用中のオペレーティング・システムによって、選択できるユーティリティは異なります。

ここでは、O/S のバックアップを作成する場合のさまざまな局面について説明します。項目は次のとおりです。

- 全体データベース・バックアップの実行
- 表領域およびデータ・ファイル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

全体データベース・バックアップの実行

データベースを NORMAL モード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全体データベース・バックアップをとります。データベースのオープン中、あるいはインスタンス・クラッシュまたは SHUTDOWN ABORT の後に作成した全体データベース・バックアップは一貫性がありません。このような場合、ファイルはチェックポイント SCN と比較して一貫性のないものになります。

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、全体データベース・バックアップをとることができます。しかし、データベースを NOARCHIVELOG モードで実行する場合には、バックアップに一貫性が必要です。つまり、データベースはバックアップ前に正しく停止しておく必要があります。

一貫性のある全体データベース・バックアップの結果として作成されたバックアップ・ファイル・セットは、すべてのファイルが同一の SCN に対応しているため、一貫性があります。このため回復の実行なしに、データベースを復元できます。データベースを ARCHIVELOG モードで運用している場合は、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するための追加の回復手順を実行できます。また、データベースが ARCHIVELOG モードであれば、一貫性のない全体データベース・バックアップをとることもできます。

データベース全体のバックアップによって作成されたバックアップ制御ファイルは、そのバックアップ中にとられたその他のファイルの復元にのみ使用してください。完全あるいは不完全なデータベース回復には使用しないでください。これは、Oracle が、ALTER DATABASE BACKUP CONTROLFILE 文を使用して作成されたバックアップ制御ファイルをバックアップ制御ファイルとして認識するためです。制御ファイルの O/S コピーは、Oracle からは現行の制御ファイルに見えます。全体データベース・バックアップを作成する場合以外は、制御ファイルのバックアップに必ず SQL 文を使用してください。

関連項目：制御ファイルのバックアップの詳細は、13-11 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。

一貫性のある全体データベース・バックアップの作成

データベースのデータ・ファイルの一貫性を確保するために、全体データベース・バックアップを作成する前に、NORMAL、IMMEDIATE または TRANSACTIONAL オプションを指定して、データベースを停止してください。データベースが ARCHIVELOG モードでない限り、インスタンスの障害後または SHUTDOWN ABORT 文を使用してデータベースを停止した後は、全体データベース・バックアップは決して実行しないでください。

一貫性のある全体データベース・バックアップを作成する方法

1. データベースがオープン状態の場合は、SQL*Plus を使用し、NORMAL、IMMEDIATE または TRANSACTIONAL オプションを指定してデータベースを停止します。

```
SHUTDOWN NORMAL;  
SHUTDOWN IMMEDIATE;  
SHUTDOWN TRANSACTIONAL;
```

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、全体データベース・バックアップを実行しないでください。この場合は、データベースを再度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. O/S のコマンドまたはバックアップ・ユーティリティを使用して、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。

たとえば、UNIX 上で /disk1/oracle/dbs の中にすべてのデータ・ファイル、制御ファイルおよびパラメータ・ファイルが含まれているときには、次のように、/disk2/backup にディレクトリをバックアップできます。

```
% cp -r /disk1/oracle/dbs /disk2/backup
```


3. データベースを再起動します。

```
STARTUP;
```

関連項目 : データベースの起動および停止の詳細は、『Oracle8i 管理者ガイド』を参照してください。

バックアップの検証

DBVERIFY 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。DBVERIFY は主に、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、データ破損の問題が発生したときの診断基準の 1 つとして使用してください。

DBVERIFY の名前と格納場所は、使用しているオペレーティング・システムによって異なります。たとえば、UNIX 上でデータ・ファイル `tbs_52.f` の整合性チェックを行うには、次のように、**dbv** コマンドを実行します。

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

```
DBVERIFY: Release 8.1.5.0.0
```

```
(c) Copyright 1998 Oracle Corporation. All rights reserved.
```

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

```
DBVERIFY - Verification complete
```

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

関連項目 : DBVERIFY の使用方法の詳細は、『Oracle8i ユーティリティ・ガイド』を参照してください。

表領域およびデータ・ファイルのバックアップの実行

表領域およびデータ・ファイルのバックアップは、ARCHIVELOG モードで運用している場合にのみ作成してください。NOARCHIVELOG モードで運用している場合には、データ・ファイルを同一の時点にまで回復するためのアーカイブ REDO ログがないため、個々のデータ・ファイルのバックアップを使用してデータベースを復元することはできません。

この項では次の項目を取り上げます。

- [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](#)
- [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](#)

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ときには、オンライン表領域のすべてのデータ・ファイルまたは指定のデータ・ファイルのバックアップをとることができます。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を更新しなくなります。

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。これにより、データベースは[ホット・バックアップ・モード \(hot backup mode\)](#) に入ります。データベースのオープン中に O/S バックアップを作成するためには、データベースをホット・バックアップ・モードにする必要があります。バックアップが完了すると、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が、これは、ユーザーが表領域のホット・バックアップ・モードを終了するために ALTER TABLESPACE END BACKUP 文を実行した後に行われます。

データ・ファイルを復元すると、ヘッダーの中には、オンライン表領域のバックアップ前に発生した、データ・ファイルの最新のチェックポイントの記録が含まれます。これには、バックアップ中に発生したものは含まれません。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ます。

オープン状態のデータベースでオンライン表領域をバックアップする方法

1. 表領域のバックアップを開始する前に、DBA_DATA_FILES データ・ディクショナリ・ビューを使用し、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 表領域をバックアップする場合。次のように入力します。

```
SELECT tablespace_name, file_name
FROM sys.dba_data_files
WHERE tablespace_name = 'USERS';
```

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

この例では、/oracle/dbs/tbs_21.f および /oracle/dbs/tbs_22.f は、USERS 表領域のデータ・ファイルに対応する完全修飾のファイル名です。

2. オンライン表領域のバックアップの開始を指示します。たとえば、表領域 USERS のオンライン・バックアップの開始を指示するには次の文を使用します。

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

警告： オンライン表領域のバックアップの開始を指示す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用できなくなります。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、結果的には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「fuzzy files」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. O/S コマンドを使用し、オンライン表領域のオンライン・データ・ファイルのバックアップをとります。たとえば、UNIX ユーザーは次のように入力できます。

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

4. オンライン表領域のデータ・ファイルをバックアップした後、END BACKUP オプションを指定して SQL コマンドの ALTER TABLESPACE を使用し、オンライン・バックアップの終了を指示します。たとえば、表領域 USERS のオンライン・バックアップの終了を指示す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 が発生すると、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 文を使用します。

関連項目：DBA_DATA_FILES データ・ディクショナリ・ビューの詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。STARTUP コマンドおよび SHUTDOWN コマンドの詳細は、『SQL*Plus ユーザーズ・ガイドおよびリファレンス』を参照してください。

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・ステータスをチェックするには、V\$BACKUP ビューを問い合わせます。このビューはすべてのオンライン・ファイルをリスト表示し、それらのファイルのバックアップ・ステータスを示します。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、表領域をホット・バックアップ・モードにしていたかどうかを判断するために使用します。

現在使用中の制御ファイルが、メディア障害が発生し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含ま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 列では、INACTIVE はファイルが現在バックアップ中ではないことを示し、ACTIVE は、ファイルが現在バックアップ中であることを示します。

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とる場合には、次の手順のいずれかを使用してください。

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方法

1. 必要な ALTER TABLESPACE 文をすべて一度に発行し、すべての表領域のバックアップをとる準備をします。たとえば、表領域 TS1、TS2 および TS3 をホット・バックアップ・モードにします。

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

2. 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ります。たとえば、UNIX ユーザーは次のように、tbs_1.f、tbs_2.f および tbs_3.f のバックアップをとることができます。

```
% cp /oracle/dbs/tbs_1.f /oracle/backup/tbs_1.bak  
% cp /oracle/dbs/tbs_2.f /oracle/backup/tbs_2.bak  
% cp /oracle/dbs/tbs_3.f /oracle/backup/tbs_3.bak
```

3. オンライン・バックアップが終了したことを指示します。

```
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シリアル方式で作成する方法

1. 表領域のオンライン・バックアップをとる準備をします。たとえば、表領域 TBS_1 をホット・バックアップ・モードにするには、次のように入力します。

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

2. オンライン表領域の中のデータ・ファイルのバックアップをとります。たとえば、次のように入力します。

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

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

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

4. 目的の表領域すべてのバックアップを作成するまで、表領域のそれぞれでこの手順を繰り返します。

ALTER TABLESPACE 間の時間を最小化できるシリアル・オプションをお勧めします。BEGIN/END BACKUP 文を使用したオンライン・バックアップの間には、その表領域について通常よりも多くの REDO 情報が生成されます。

サスペンド・モードでのミラーの分割 ディスクまたはファイルのミラー化が可能なシステムも考えられます。つまり、プライマリ・データの正確な複製を別の場所に保存し、ミラーを分割することができます。ミラーの分割は、各コピーを個別に使用できるように分離します。

RAID デバイスの中には、分割操作の発生中に書き込みを中断することによる利益を享受できるものがあります。使用中のシステムでこの機能を享受できるかどうかは、RAID のベンダーに確認してください。

SUSPEND および RESUME を使用するための基本的な手順は次のとおりです。

1. ALTER TABLESPACE BEGIN BACKUP 文を使用して、データベース表領域をホット・バックアップ・モードにします。たとえば、表領域 USERS をホット・バックアップ・モードにするには、次のように入力します。

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

2. 使用しているミラー化システムで、ディスク書き込み中のミラーの分割に問題がある場合には、次のコマンドを発行します。

```
ALTER SYSTEM SUSPEND;
```

- ミラーの分割は、O/S レベルまたはハードウェア・レベルで行ってください。
- 次のコマンドを発行します。

```
ALTER SYSTEM RESUME;
```

- 指定した表領域のホット・バックアップ・モードを終了します。たとえば、表領域 USERS のホット・バックアップ・モードを終了するには、次のように入力します。

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

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

警告： 表領域をホット・バックアップ・モードにするための代替策として SUSPEND コマンドを使用しないでください。

関連項目： SUSPEND/RESUME 機能の詳細は、『Oracle8i 管理者ガイド』を参照してください。RESUME および SUSPEND オプションを指定した ALTER SYSTEM コマンドの詳細は、『Oracle8i SQL リファレンス』を参照してください。

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。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。

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。

オフラインの表領域をバックアップするには

- 表領域のバックアップを開始する前に、DBA_DATA_FILES 表を使用し、その表領域のデータ・ファイルを確認します。たとえば、USERS 表領域をバックアップする場合、次のように入力します。

```
SELECT tablespace_name, file_name
FROM sys.dba_data_files
WHERE tablespace_name = 'USERS';
```

```
TABLESPACE_NAME
```

```
FILE_NAME
```

USERS

/oracle/dbs/users.f

この例では、/oracle/dbs/users.f は、USERS 表領域の中のデータ・ファイルに対応する、完全修飾のファイル名です。

2. 可能であれば、NORMAL オプションで表領域をオフラインにします。NORMAL オプションを使用すると、表領域回復を行わずに、オンラインに戻せるため、NORMAL オプションをお勧めします。たとえば、次の文は、USERS という名前の表領域を NORMAL オプションでオフラインにします。

```
ALTER TABLESPACE users OFFLINE NORMAL;
```

表領域を NORMAL オプションでオフラインにした場合、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。たとえば、UNIX ユーザーは次のように入力して、データ・ファイル users.f のバックアップをとることができます。

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

4. 表領域をオンラインにします。たとえば、次の文は、表領域 USERS をオンラインに戻します。

```
ALTER TABLESPACE users ONLINE;
```

注意： TEMPORARY または IMMEDIATE オプション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い限り、その表領域をオンラインにしないでください。

表領域をオンラインにすると、表領域はオープンされ、使用できるようになります。

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。

制御ファイルのバックアップをとるには 2 つの方法があります。

- 物理ファイルへの制御ファイルのバックアップの作成
- 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

物理ファイルへの制御ファイルのバックアップの作成

制御ファイルのバックアップをとる第一の方法は、SQL 文を使用してバイナリ・ファイルを生成する方法です。

構造的な変更を行った後に制御ファイルをバックアップする方法

1. データベースに必要な変更を加えます。たとえば、新しいデータ・ファイルを作成することが考えられます。

```
ALTER DATABASE CREATE DATAFILE '/oracle/dbs/tbs_20.f' AS '/oracle/dbs/tbs_4.f';
```

2. データベースの制御ファイルのバックアップをとります。次の SQL 文は、データベースの制御ファイルを /oracle/backup/cf.bak にバックアップします。

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

REUSE オプションを指定すると、既存のバックアップに新しい制御ファイルを上書きできます。

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 文の 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 オプションを指定すると、物理的バックアップは生成されず、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。トレース・ファイルの中に書き込まれた文は、データベースを起動し、制御ファイルを再作成し、データベースの回復とオープンを適切に行います。

トレース・ファイルの中の各 SQL 文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用してデータベースを回復できます。トレース・ファイルは、USER_DUMP_DEST 初期化パラメータで指定されている場所にあります。

制御ファイルのバックアップをトレース・ファイルに作成するには、データベースをマウントし、次の文を発行します。

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

トレース・ファイルの作成例 SALES データベース用の制御ファイルを再作成するためのスクリプトを生成すると仮定します。データベースの特性は次のとおりです。

- 使用可能なスレッドは 3 つあり、そのうちスレッド 2 がパブリック、スレッド 3 がプライベートである。
- REDO ログは、それぞれが 2 つのメンバーを持つ、3 つのグループに多重化されている。
- データベースには次のデータ・ファイルが入っている。
 - /diska/prod/sales/db/filea.dbf（オンライン表領域の中のアフライン・データ・ファイル）

- /diska/prod/sales/db/database1.dbf (オンライン)
- /diska/prod/sales/db/fileb.dbf (読取り専用表領域の中の唯一のファイル)

次の文を発行し、トレース・ファイルを作成します。

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

次に、トレース・ファイルを編集し、トレース・ファイルの生成時に現行のものであった制御ファイルに基づき、新しい制御ファイルを作成するスクリプトを作成します。NORMAL オプションによるオフライン表領域または読取り専用表領域の回復の実行を回避するには、これらの表領域を CREATE CONTROLFILE 文から除きます。ディクショナリのチェックではデータベースのオープン時に、これらのファイルに MISSING とマークを付けます。RENAME コマンドは、これらのファイルの名前をそれぞれのファイル名に戻します。

たとえば、次のようなスクリプトが考えられます。

```
# The following statements 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 datafiles. Use this only if the current version of all online logs
# are available.

STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE SALES NORESETLOGS ARCHIVELOG
    MAXLOGFILES 32
    MAXLOGMEMBERS 2
    MAXDATAFILES 32
    MAXINSTANCES 16
    MAXLOGHISTORY 1600
LOGFILE
    GROUP 1
        '/diska/prod/sales/db/log1t1.dbf',
        '/diskb/prod/sales/db/log1t2.dbf'
    ) SIZE 100K
    GROUP 2
        '/diska/prod/sales/db/log2t1.dbf',
        '/diskb/prod/sales/db/log2t2.dbf'
    ) SIZE 100K,
    GROUP 3
        '/diska/prod/sales/db/log3t1.dbf',
        '/diskb/prod/sales/db/log3t2.dbf'
    ) SIZE 100K
DATAFILE
    '/diska/prod/sales/db/database1.dbf',
    '/diskb/prod/sales/db/filea.dbf'
;

# This datafile is offline, but its tablespace is online. Take the datafile offline
# manually.
```

```
ALTER DATABASE DATAFILE '/diska/prod/sales/db/filea.dbf' OFFLINE;

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

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

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

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

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文が含まれた、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。
- インスタンスまたは 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が必要であるとみなします。

たとえば、次のように表示されることがあります。

```
SQL> startup
ORACLE instance started.
Total System Global Area                19839308 bytes
Fixed Size                               63820 bytes
Variable Size                           11042816 bytes
Database Buffers                        8192000 bytes
Redo Buffers                             540672 bytes
Database mounted.
ORA-01113: file 12 needs media recovery
```

```
ORA-01110: data file 12: '/oracle/dbs/tbs_41.f'
```

表領域のメディア回復の実行を回避する方法

1. V\$BACKUP ビューを使用し、データベースの再起動前にバックアップ中であった表領域のデータ・ファイルをリスト表示します。

```
SQL> SELECT * FROM v$backup WHERE status = 'ACTIVE';
FILE#          STATUS          CHANGE#    TIME
-----
          12 ACTIVE                      20863 25-NOV-98
          13 ACTIVE                      20863 25-NOV-98
          20 ACTIVE                      20863 25-NOV-98
3 rows selected.
```

2. ALTER DATABASE DATAFILE ... END BACKUP 文を発行し、ホット・バックアップを終了します。たとえば、データ・ファイル 12、13 および 20 のホット・バックアップ・モードを終了するには、次のように入力します。

```
ALTER DATABASE DATAFILE 12,13,20 END BACKUP;
```

警告： ALTER DATABASE DATAFILE ...END BACKUP は、影響を受けるファイルのいずれかをバックアップから復元した場合には使用しないでください。

3. データベースをオープンします。

```
ALTER DATABASE OPEN;
```

END BACKUP 文を使用せずにデータベースを回復する方法

1. データベースをマウントします。

```
STARTUP MOUNT;
```

2. データベースを回復します。

```
RECOVER DATABASE;
```

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

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

関連項目：データベースの回復の詳細は、[第 14 章の「オペレーティング・システム環境での回復の実行」](#)を参照してください。

保護強化のための Export と Import の使用方法

Export および Import は、Oracle データを Oracle データベースに入出力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・ファイルに特別なバイナリ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。

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。項目は次のとおりです。

- [Export の使用方法](#)
- [Import の使用方法](#)

関連項目：Export および Import ユーティリティの詳細は、『Oracle8i ユーティリティ・ガイド』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用すると、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で、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データを、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 を使用し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でください。可能であれば、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、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いてください。

[表 13-1](#) に、使用できるエクスポート・モードを示します。

表 13-1 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・データのみ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 の情報のみの場合、表 A のみ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・データのみ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 の情報のみの場合、表 A および表 B に対する変更内容のみ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用すると、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用してデータベースを回復するには、次のようにします。

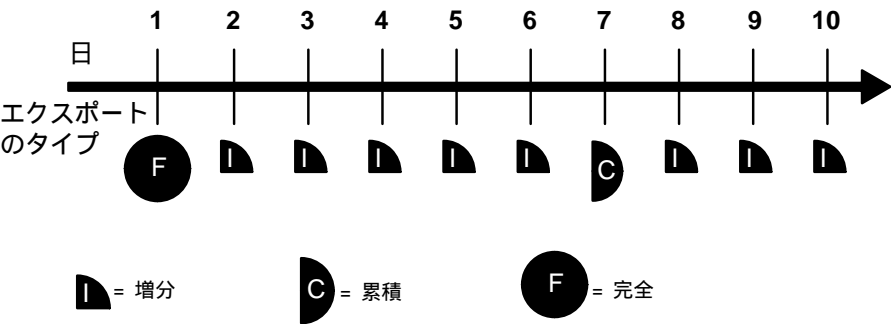
1. すべての表領域およびユーザーも含めて、データベース構造を再作成する。

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いけません。

2. 適切なエクスポート・ファイルをインポートし、データベースを可能な限り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。

例えば図 13-1 に示すスケジュールを使用して、Oracle データベースからデータをエクスポートするとします。

図 13-1 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、回復には次の手順を使用します。

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。

オペレーティング・システム環境での 回復の実行

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。項目は次のとおりです。

- [メディア回復](#)
- [回復するファイルの判別](#)
- [ファイルの復元](#)
- [基本的なメディア回復手順の理解](#)
- [完全メディア回復の実行](#)
- [不完全メディア回復の実行](#)
- [メディア回復後のデータベースのオープン](#)

メディア回復

メディア回復とは、特定の時点にデータベースを回復することです。これは、次のような基本的な段階に従って実行されます。

- [ファイルの復元](#)
- [データ・ファイルの回復](#)

ファイルの復元

メディア回復の最初の作業は、データベース・ファイルの復元です。復元作業では、1 つまたは複数のファイルのコピーを、バックアップから再構築します。たとえば、データ・ファイルを確認するために V\$DATAFILE を問い合わせるとします。

```
SQL> SELECT name FROM v$datafile;  
NAME
```

```
-----  
/oracle/dbs/tbs_01.f  
/oracle/dbs/tbs_02.f  
/oracle/dbs/tbs_03.f  
/oracle/dbs/tbs_11.f  
/oracle/dbs/tbs_12.f  
/oracle/dbs/tbs_21.f  
/oracle/dbs/tbs_22.f  
7 rows selected.
```

データ・ファイルを復元するには、O/S コマンドを使用し、データ・ファイルのバックアップを、デフォルトの位置にコピーしてそこに現在ある同じ名前のファイルを上書きするか、新しい位置にコピーします。たとえば、データ・ファイル tbs_12.f をバックアップ tbs_12.bak から復元する場合には、次のように入力します。

```
% cp /oracle/backup/tbs_12.bak /oracle/dbs/tbs_12.f
```

データ・ファイルの回復

2 番目の作業は、指定したデータ・ファイルのメディア回復の実行です。メディア回復とは、復元されたデータ・ファイルを現行の時刻または現行以外の時刻まで更新するために、オンラインおよびアーカイブ REDO ログをデータ・ファイルに適用することです。データベースを最新の SCN に回復するために使用可能なすべての REDO を使用したメディア回復は完全メディア回復です。これを行っていないメディア回復は不完全メディア回復です。

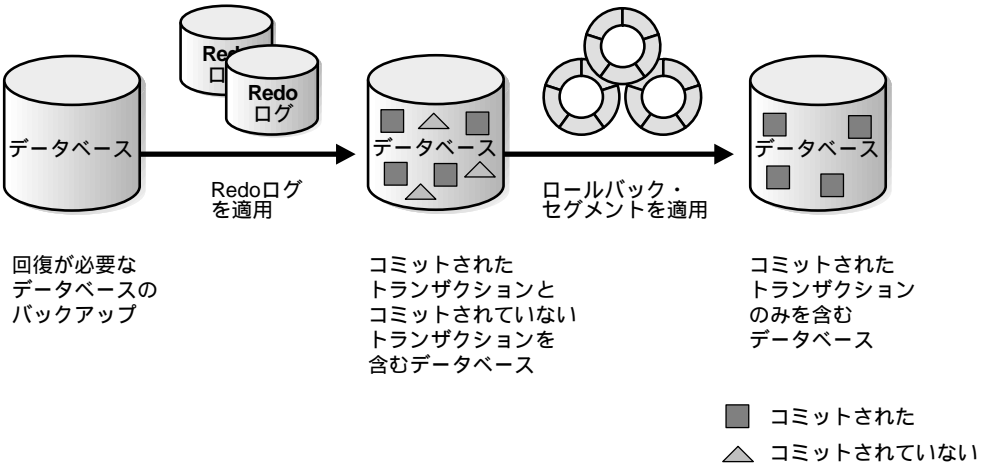
Oracle では回復は 2 つのフェーズで行われます。

- ロールフォワード
- ロールバック

回復のロールフォワード・フェーズでは、オンライン、アーカイブまたはその両方の REDO ログ・レコードの中に記録されている変更が、データ・ファイルに適用されます。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 REDO ログ・ファイルのロールフォワードが行われます。ロールフォワード後のデータ・ブロックには、コミットされた変更も、コミットされていない変更も含まれています。

ロールバック・フェーズでは、ロールフォワード・フェーズ中に適用された、コミットされていないトランザクションの影響を取り除くために、ロールバック・セグメントが使用されます。ロールバック・セグメントは、REDO ログに記録されているトランザクションの中で、コミットされていないものを特定するために使用されます。図 14-1 は、ロールフォワードとロールバックという 2 つの段階を示しています。

図 14-1 基本的な回復手順



回復するファイルの判別

表 VSRECOVER_FILE を使用することにより、回復するファイルを判別できます。このビューには、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ます。

次の問合せは、メディア回復が必要なデータ・ファイルの ID 番号と、回復が必要な理由 (わかっている場合)、そして回復を開始するべき SCN/ 時刻を表示します。

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

FILE#	ONLINE	ERROR	CHANGE#	TIME
14	ONLINE		0	

```
15 ONLINE FILE NOT FOUND 0
21 OFFLINE OFFLINE NORMAL 0
```

注意： 現在使用中の制御ファイルが、メディア障害の発生後に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、このビューは役に立ちません。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。

回復が必要なデータ・ファイルのファイル名および表領域名を取得するためには、V\$DATAFILE および V\$TABLESPACE を検索してください。たとえば、次のように入力します。

```
SQL> SELECT d.name, t.name
      2 FROM v$datafile d, v$tablespace t
      3 WHERE t.ts# = d.ts#
      4 AND d.file# in (14,15,21); # use values obtained from V$RECOVER_FILE query
```

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

回復が必要なファイルを決定すると共に、回復してはならないファイルも判別する必要があります。メディア回復では、次の概念は特殊な意味を持ちます。

- 回復不能な表および回復不能な索引
- 読取り専用表領域

回復不能な表および回復不能な索引

表および索引は、CREATE TABLE AS SELECT コマンドを使用して作成できます。また、それらの表および索引を回復不能 (UNRECOVERABLE) として作成するよう指定できます。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混在している場合は、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けられるので注意してください。回復不能オブジェクトにアク

セスしようとする、ORA-01578のエラー・メッセージが戻されます。回復不能なオブジェクトを削除し、必要であれば再作成してください。

表を回復不能として作成した後で、その表の索引を回復可能として作成することも考えられるため、メディア回復の実行後も、索引には、論理的に破損していることを示すマークは付きません。しかし、表は回復不能である（回復後に破損を示すマークが設定される）ため、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てください。

関連項目 : 回復不能操作がスタンバイ・データベースに与える影響の詳細は、16-28 ページの「[UNRECOVERABLE 操作後のバックアップの必要性の判別](#)」を参照してください。

読取り専用表領域

USING BACKUP CONTROLFILE オプションを使用したメディア回復では、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は回復できません。このエラーを回避するためには、バックアップ制御ファイルを使用して回復を実行する前に、読取り専用表領域のデータ・ファイルをオフラインにします。

回復には、制御ファイルの正しいバージョンを使用してください。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用する必要があります。同様に、回復の完了時に表領域を読み書き可能モードにするには、制御ファイルも読み書き可能モードにする必要があります。

適切な制御ファイルが使用できない場合には、15-12 ページの「[現行の制御ファイルのすべてのコピーの消失](#)」の説明に従って、CREATE CONTROLFILE 文を実行してください。読取り専用表領域を含むデータベースの制御ファイルを再作成する必要がある場合には、次のコマンドを入力して、必要な手順を取得してください。

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

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する必要がある点を除けば、NORMAL モードでオフラインにされた表領域についての手順と同様です。

関連項目 : 制御ファイルのトレース・バックアップの作成方法の詳細は、13-12 ページの「[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](#)」を参照してください。

ファイルの復元

メディア回復が必要であると判断した場合には、メディア回復を実行するために、必要なファイルを復元します。ここでは、次の作業の方法を示します。

- [バックアップ・データ・ファイルの復元](#)
- [バックアップが使用できない場合のデータ・ファイルの再作成](#)
- [必要なアーカイブ REDO ログ・ファイルの復元](#)

バックアップ・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・ファイルが永久的に破損した場合、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。破損したデータ・ファイルを元の位置に復元できない（たとえば、ディスクを交換する必要があるため、ファイルを別のディスクに復元する）場合には、対応するデータベースの制御ファイルに、これらのファイルの新しい位置を指定する必要があります。

バックアップ・データ・ファイルをデフォルトの位置に復元する方法

1. 14-3 ページの「[回復するファイルの判別](#)」で説明している方法を使用して、回復するデータ・ファイルを判別します。
2. O/S コマンドを使用し、破損したデータ・ファイルのバックアップをデフォルトの位置にコピーします。たとえば、UNIX 上で `tbs_14.f` を復元するには、次のコマンドを発行します。

```
% cp /disk2/backup/tbs_14.bak /disk1/oracle/dbs/tbs_14.f
```

バックアップが使用できない場合のデータ・ファイルの再作成

データ・ファイルが破損し、そのファイルに使用できるバックアップがないときでも、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アーカイブ・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）。

回復のためのデータ・ファイルの再作成

1. 対応するバックアップを持たない破損したデータ・ファイルを置換するために、新しく空のデータ・ファイルを作成します。たとえば、データ・ファイル `disk1:users1` 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';
```

この文は、失われたファイルに対応する空のファイルを作成します。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。古いデータ・ファイルは、新しいデータ・ファイルとして改名されます。

2. 空のデータ・ファイルにメディア回復を実行します。たとえば、次のように入力します。

```
RECOVER DATAFILE 'disk2:users1'
```

3. 回復時には、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 ログを復元し、失われたデータ・ファイルのかわりの空のファイルに再適用してください。

注意： SYSTEM 表領域の場合には、必要な REDO データを使用できないため、ALTER DATABASE 文の CREATE DATAFILE 句を使用してデータ・ファイルを再作成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。

必要なアーカイブ REDO ログの復元

1. どのアーカイブ REDO ログ・ファイルが必要かを判別するためには、V\$LOG_HISTORY および V\$RECOVERY_LOG を検索します。データ・ファイルのバックアップが使用できない場合には、データ・ファイルがデータベースに追加されてからの、すべての REDO 情報が必要です。

ビュー	説明
V\$LOG_HISTORY	すべてのアーカイブ・ログがリスト表示され、初期化パラメータ LOG_ARCHIVE_FORMAT により設定された、アーカイブ・ログ・ファイルの推定名称も示されます。
V\$RECOVERY_LOG	回復の実行に必要なアーカイブ REDO ログのみ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 に従って示されています。

2. 領域に余裕があれば、必要なアーカイブ REDO ログ・ファイルを、LOG_ARCHIVE_DEST_1 または LOG_ARCHIVE_DEST によって指定されている位置に復元してください。Oracle は、メディア回復中に必要に応じて正しいログを自動的に検出します。

たとえば、次のように入力します。

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

3. 宛先を指定する初期化パラメータで指定された位置に十分な領域がない場合には、必要なアーカイブ REDO ログ・ファイルのすべてまたは一部を、別の位置に復元してください。SQL*Plus の SET 文の LOGSOURCE パラメータを使用するか、SQL の ALTER DATABASE 文の RECOVER ... FROM パラメータを使用して、メディア回復前またはメディア回復中に位置を指定します。たとえば、次のように入力します。

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

4. アーカイブ・ログを適用し、バックアップとして各アーカイブ・ログ・グループのコピーがまだ存在していることを確認した後で、アーカイブ REDO ログ・ファイルの復元コピーを削除し、ディスク領域を解放します。たとえば、ログ・ディレクトリを作業ディレクトリにした後で、次のように入力します。

```
% rm *.arc
```

関連項目：データ・ディクショナリ・ビューの詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。

基本的なメディア回復手順の理解

回復を開始する前に、次の項目を理解する必要があります。

- [メディア回復文の使用法](#)
- [アーカイブ REDO ログの適用](#)
- [NOARCHIVELOG モードでのデータベースの回復](#)
- [ARCHIVELOG モードでのデータベースの回復](#)
- [パラレル方式でのメディア回復の実行](#)

メディア回復文の使用法

Oracle では次のメディア回復用の SQL*Plus 文が使用されます。これらの文では、回復すべきファイルのセットをシステムが判別する方法のみが異なります。

- RECOVER DATABASE
- RECOVER TABLESPACE
- RECOVER DATAFILE

各文では、ファイルが回復可能かを判断するために同じ基準が使用されます。Oracle は、2つの回復セッションが同じファイルを回復することを防止し、使用中のファイルのメディア回復を防止します。

SQL 文の ALTER DATABASE RECOVER を使用することもできますが、アーカイブ REDO ログの名前をプロンプトから要求される方式の、SQL*Plus の RECOVER 文を使用するように強くお勧めします。

関連項目：SQL*Plus の RECOVER 文の詳細は、『SQL*Plus ユーザーズ・ガイドおよびリファレンス』を参照してください。ALTER DATABASE RECOVER 文の詳細は、『Oracle8i SQL リファレンス』を参照してください。

RECOVER DATABASE 文

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。たとえば、データベース全体を回復するには、SQL プロンプトで次のコマンドを発行します。

```
RECOVER DATABASE
```

すべてのインスタンスを正しく停止し、バックアップを何も復元していない場合に、RECOVER DATABASE を発行すると、回復が必要ないことを示すエラーが戻されます。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。データベース全体に対してメディア回復を実行するには、そのデータベースを EXCLUSIVE オプションによりマウントし、クローズする必要があります。

RECOVER TABLESPACE 文

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。たとえば、表領域 TBS_1 を回復するには、SQL プロンプトで次のコマンドを発行します。

```
RECOVER TABLESPACE tbs_1
```

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。回復が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 文

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。たとえば、データ・ファイル /oracle/dbs/tbs_22.f を回復するには、SQL プロンプトで次のように入力します。

```
RECOVER DATAFILE '/oracle/dbs/tbs_22.f'
```

メディア回復ロックを取得できれば、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・ファイルのみです。

関連項目：メディア回復文の詳細は、『Oracle8i SQL リファレンス』を参照してください。

アーカイブ REDO ログの適用

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ため、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 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、SQL*Plus を使用している場合は、次のメッセージとプロンプトが表示されます。

```
ORA-00279: Change ##### generated at DD/MM/YY HH:MM:SS needed for thread#
ORA-00289: Suggestion : logfile
ORA-00280: Change ##### for thread # is in sequence #
Specify log: [<RET> for suggested | AUTO | FROM logsource | CANCEL ]
```

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

アーカイブ REDO ログのファイル名の提案

RECOVER コマンドでは初期化パラメータ LOG_ARCHIVE_DEST_1 または LOG_ARCHIVE_DEST と LOG_ARCHIVE_FORMAT の現行の設定値を連結し、さらに制御ファイルの情報を使用して、アーカイブ REDO ログ・ファイルの名前を推定して提案します。たとえば、アーカイブ・ログでは次の設定が考えられます。

```
LOG_ARCHIVE_DEST_1 = /oracle/arc_dest/arc
LOG_ARCHIVE_FORMAT = r_%t_%s.arc
```

```
SQL> SELECT name FROM v$archived_log;
```

```
NAME
```

```
-----
/oracle/arc_dest/arcr_1_467.arc
/oracle/arc_dest/arcr_1_468.arc
/oracle/arc_dest/arcr_1_469.arc
/oracle/arc_dest/arcr_1_470.arc
```

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_1 または LOG_ARCHIVE_DEST にマウントされ、LOG_ARCHIVE_FORMAT の値が変更されていない場合には、Oracle が必要なログ・ファイルを推定して適用するため、メディア回復を自動的に実行できます。

デフォルト位置以外の位置にアーカイブ REDO ログを復元する方法

1. パラメータの値を新しい位置に変更します。たとえば、次のように入力します。

```
LOG_ARCHIVE_DEST_1 = /oracle/new_location
```

2. ログ・ファイルを新しい位置に移動します。たとえば、次のように入力します。

```
% cp /oracle/arc_dest/* /oracle/new_location
```

3. 新しいインスタンスを起動し、データベースをマウントします。

```
STARTUP MOUNT
```

4. 通常の手順で、メディア回復を開始します。たとえば、次のように入力します。

```
RECOVER DATABASE
```

REDO ログ・ファイルの宛先パラメータの現行の設定値を、上書きする場合にこのようにします。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となきに、宛先パラメータによって指定された位置には必要な REDO ログ・ファイルをマウントするための領域が十分でない場合などです。

デフォルト位置以外の位置にあるログを使用して回復する方法

1. アーカイブ REDO ログをデフォルト以外の場所に復元します。たとえば、次のように入力します。

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

2. 回復操作の代替位置を指示します。SET 文の LOGSOURCE パラメータを使用するか、ALTER DATABASE 文の RECOVER ... FROM パラメータを使用します。たとえば、次のように入力します。

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

3. オフライン表領域を回復します。

```
RECOVER TABLESPACE offline_tbsp
```

注意： REDO ログのアーカイブ先を上書き指定しても、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 REDO 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場合には、宛先パラメータの現行の設定値を上書きすることを考えてください。この場合は、ログ・ファイルのソースを、複数の位置への検索パスとして機能する、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：このような機能はオペレーティング・システムに依存しています。詳細は、使用しているオペレーティング・システム固有の Oracle マニュアルを参照してください。

SQL*Plus RECOVER 文を使用してログを自動的に適用する方法

SQL*Plus を使用する場合には、回復時に必要なアーカイブ REDO ログのデフォルトのファイル名の適用を自動化するために、次のコマンドを使用してください。

```
SET AUTORECOVERY ON
```

必要なファイルが正しい名前で正しい位置にある限り、RECOVER 文の発行後に対話は必要ありません。

SET AUTORECOVERY ON を使用するときには使用されるファイル名は、初期化パラメータ LOG_ARCHIVE_DEST または LOG_ARCHIVE_DEST_1 と、LOG_ARCHIVE_FORMAT の値を組み合わせで導出されます。デフォルト・オプションである SET AUTORECOVERY OFF を実行する場合には、手動でファイル名を入力するか、提示されたデフォルトのファイル名を受け入れる必要があります。

アーカイブ REDO ログの適用を自動化する方法

1. オフライン・データ・ファイルのバックアップを復元します。この例は、データベース全体の一貫性バックアップを復元します。

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

2. データベースがマウントされていることを確認します。たとえば、データベースが停止している場合には、次のように入力します。

```
SQL> STARTUP MOUNT
```

3. AUTORECOVERY をオンにします。

```
SQL> SET AUTORECOVERY ON
Autorecovery                                ON
```

4. 必要なデータ・ファイルを回復します。この例ではデータベース全体を回復します。

```
SQL> RECOVER DATABASE
```

5. 必要なアーカイブ・ログは自動的に推定され、適用されます。

```
ORA-00279: change 53577 generated at 01/26/99 19:20:58 needed for thread 1
ORA-00289: suggestion : /oracle/work/arc_dest/arcr_1_802.arc
ORA-00280: change 53577 for thread 1 is in sequence #802
Log applied.
ORA-00279: change 53584 generated at 01/26/99 19:24:05 needed for thread 1
ORA-00289: suggestion : /oracle/work/arc_dest/arcr_1_803.arc
ORA-00280: change 53584 for thread 1 is in sequence #803
ORA-00278: log file '/oracle/work/arc_dest/arcr_1_802.arc' no longer needed for
```

```

this recovery
Log applied.
ORA-00279: change 53585 generated at 01/26/99 19:24:14 needed for thread 1
ORA-00289: suggestion : /oracle/work/arc_dest/arc_r1_804.arc
ORA-00280: change 53585 for thread 1 is in sequence #804
ORA-00278: log file '/oracle/work/arc_dest/arc_r1_803.arc' no longer needed for
this recovery
Log applied.
Media recovery complete.

```

OPS 構成を使用した場合、不完全回復を実行するか、バックアップ制御ファイルを使用すると、最初のスレッドからの最初のアーカイブ REDO ログ・ファイルの名前しか算出できません。他のスレッドからの最初のログ・ファイルを適用する必要がある場合。あるスレッド内の最初のログ・ファイルが指定されると、Oracle はそのスレッド内の後続のログ・ファイルの名前を推定できます。

関連項目 : ログ・ファイル適用の例は、使用しているオペレーティング・システム固有の Oracle マニュアルを参照してください。

ALTER DATABASE RECOVER を使用した、ログの個別の適用

SQL 文を使用してメディア回復を実行した場合、メディア回復の開始後に、ログ・ファイルのためのプロンプトは表示されません。かわりに、ALTER DATABASE RECOVER LOGFILE 文を使用して正しいログ・ファイルを指定する必要があります。たとえば、メッセージから log1.arc が提案された場合、次の文を使用して、このログ・ファイルを適用します。

```
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 の入力を太字で示し、変数情報をイタリックで示してあります）。

```

SQL> 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 #
SQL> 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 #
SQL> ALTER DATABASE RECOVER LOGFILE 'logfile2';
. . .
Repeat until all logs are applied.)
Statement processed.
SQL> ALTER TABLESPACE users ONLINE;
Statement processed.

```

ALTER DATABASE RECOVER を使用した、ログの自動適用

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となっています。SQL*Plus で使用した方法と同様に、回復前および回復中には、それぞれ次の文を使用して、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_filename;
```

最初の文の例を次に示します。

```
SQL> ALTER DATABASE RECOVER AUTOMATIC TABLESPACE users;  
Statement processed.  
SQL> 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となっています。

ALTER DATABASE RECOVER AUTOMATIC LOGFILE 文の例を次に示します。

```
SQL> 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 #  
SQL> ALTER DATABASE RECOVER AUTOMATIC LOGFILE 'logfile1';  
Statement processed.  
SQL> 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となっています。

注意： ALTER DATABASE RECOVER 文を発行すると、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せん。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： 回復関連のすべてのビューの内容の詳細は、『Oracle8i 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

SQL*Plus の回復オプション (SQL 文ではなく) を使用している場合には、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されます。

```
Log applied.
```

Oracle はログ順序の中の次のログを要求するプロンプトを表示するか、最後に適用されたログが必要な最後のログである場合には回復を終了します。

REDO ログの適用の失敗

推定されたファイルが正しくなかったか、指定したファイル名が間違っていた場合には、エラー・メッセージが戻されます。たとえば、次のようなメッセージが表示されます。

```
ORA-00308: cannot open archived log '/oracle/work/arc_dest/arcr_1_811.arc'  
ORA-27037: unable to obtain file status  
SVR4 Error: 2: No such file or directory  
Additional information: 3
```

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・メッセージが戻された場合には、次の状況が考えられます。

エラー	考えられる原因	解決策
ORA-27037: ファイルのステータスを取得できません	誤ったファイル名が入力された。 ログがありません。	正しいファイル名を入力し直してください。 バックアップのアーカイブ REDO ログを復元してください。
ORA-27047: ファイルのヘッダー・ブロックを読み取れません	ログが部分的に書き込まれているか、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、そのコピーを適用します。回復操作を再度実行する必要はありません。 ログのコピーが存在していない場合に、有効な REDO エントリが最後に実行された時刻がわかっているときには、不完全回復を実行してください。この場合には、バックアップの復元も含めて、回復を最初から再度開始します。

REDO ログの適用の中断

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、次のいずれかの方法を使用して、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する必要がある場合は、オペレーティング・システムの割込みシグナルを使用する。

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、その取消し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再開されます。

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用して再起動する場合や、同じバックアップを使用するが、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。

NOARCHIVELOG モードでのデータベースの回復

データベースが NOARCHIVELOG モードであり、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 を使用し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する必要があることです。一方、データベースが ARCHIVELOG モードの場合は、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。アーカイブ REDO ログを使用すれば、完全回復または不完全回復を使用してデータベースを再構築できるため、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、それを最新の一貫性全体バックアップ（この時点での唯一のオプション）から復元するには、次の手順に従ってください。

最新の全体データベース・バックアップをデフォルト位置に復元する方法

1. データベースがオープンしている場合は、インスタンスを終了します。

```
SHUTDOWN ABORT
```

2. バックアップのデータベース・ファイルを元の位置に復元できるように、メディアの問題を解決します。

3. O/S コマンドを使用し、最新の全体データベース・バックアップを復元します。破損したファイルのみでなく、全体データベース・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。次の例では、全体データベース・バックアップを復元します。

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

4. 次の文を発行し、不完全データベース回復を疑似実行します。

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

5. データベースをオープンし、現行の REDO ログ順序を 1 にリセットします。

```
ALTER DATABASE OPEN RESETLOGS;
```

RESETLOGS 操作は、オンライン・ログの中のすべての REDO ログを無効にします。全体データベース・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。

最新の全体データベース・バックアップを新しい位置に復元するには

1. データベースをオープンしている場合は、停止します。

```
SHUTDOWN NORMAL
```

2. ハードウェアの問題が解決されていないため、データベース・ファイルの一部またはすべてを別の位置に復元する必要がある場合には、最新の全体データベース・バックアップを新しい位置に復元します。破損したファイルのみでなく、全体データベース・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。たとえば、次のように入力します。

```
% cp /disk2/BACKUP/tbs* /disk3/oracle/dbs
% cp /disk2/BACKUP/cf.f /disk3/oracle/dbs
```

3. 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。

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

4. 復元し編集したパラメータ・ファイルを使用してインスタンスを起動し、データベースをマウントします。ただし、データベースはまだオープンしないでください。たとえば、次の文は初期化ファイル initPROD1.ora を使用してデータベースをマウントします。

```
STARTUP MOUNT pfile=initPROD1.ora
```

5. 復元されたデータ・ファイルのファイル名が変更されている場合には、制御ファイルの中の復元されたデータ・ファイルを改名します。たとえば、次のように入力します。

```
ALTER DATABASE RENAME FILE '/disk1/oracle/dbs/tbs1.f' TO  
'/disk3/oracle/dbs/tbs1.f';
```

6. 必要に応じて、オンライン REDO ログ・ファイルを改名します。たとえば、次のように入力します。

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

7. 次の文を発行し、不完全データベース回復を疑似実行します。

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

8. 次の文を発行し、不完全データベース回復を疑似実行します。

```
ALTER DATABASE RECOVER CANCEL;
```

9. データベースをオープンし、現行の REDO ログ順序を 1 にリセットします。

```
ALTER DATABASE OPEN RESETLOGS;
```

RESETLOGS 操作は、オンライン・ログの中のすべての REDO ログを無効にします。全体データベース・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の詳細は、『Oracle8i 管理者ガイド』を参照してください。

ARCHIVELOG モードでのデータベースの回復

ARCHIVELOG モードでデータベースを運用しているときにメディア回復操作を開始するには、次のオプションのいずれかを使用します。

- ALTER DATABASE RECOVER 文
- SQL*Plus RECOVER 文（推奨）

どの種類のメディア回復を開始する場合にでも、次の制限を守る必要があります。

- 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必要です。
- 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。

- 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。

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

パラレル方式でのメディア回復の実行

メディア回復のロールフォワード・フェーズを調整するには、ブロックのパラレル回復を使用します。ブロックのパラレル回復では、ロールフォワード時にデータ・ブロックごとに異なるプロセスを割り当てる「役割分担」方式を使用することにより、作業の効率を高めています。たとえば、REDO ログに多数のエントリが含まれる場合に、スレーブ 1 はログ・ファイルの一部分を担当し、スレーブ 2 は別の部分を担当し、スレーブ 3 はさらに別の部分を担当します。複数のディスク・ドライブに分散した多数のデータ・ファイルの破損、インスタンスおよびメディア回復ではブロックのパラレル回復が有効です。

RECOVERY_PARALLELISM 初期化パラメータは、インスタンスまたはメディア回復操作で使用する同時実行回復プロセスの数を指定するために使用します。クラッシュ回復はインスタンスの起動時に行われるため、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、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このパラメータ値が使用されます。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にする必要があります。パラレル回復の方がシリアル回復より効率がよいのは、最低 8 つ程度の回復プロセスがある場合です。

回復は、通常はデータ・ブロックに対する読取りという意味で I/O により制限されることに注意してください。このため、ブロック・レベルでパラレル化を行っても、非同期 I/O に関する O/S の制約を排除することにより I/O の総計が増加しない限り、回復のパフォーマンスは改善されません。非同期 I/O が十分効率的に行われているシステムでは、ブロックのパラレル回復を使用することによる改善はわずかなものです。

関連項目：パラレル回復の詳細は、『Oracle8i チューニング』を参照してください。

RECOVERY_PARALLELISM パラメータの詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。

完全メディア回復の実行

完全回復を実行するときには、データベース全体を一度に回復することも、個々の表領域またはデータ・ファイルを回復することもできます。完全回復の実行後には、不完全回復の実行後のように RESETLOGS オプションを使用してデータベースをオープンする必要はないため、同時にいくつかのデータ・ファイルを回復しておき、残りのデータ・ファイルを後で回復できます。

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。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回復

関連項目: 第3章の「バックアップおよび回復計画の立案」で説明されている、基本的な回復の概念と方法をお読みください。

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する場合があります。

この項では、次のような順序で、取消しベースのメディア回復の実行方法を説明します。

1. データベースを停止し、可能であればメディアの損傷を修復します。
2. 必要なファイルを復元します。
3. データベースを回復します。

クローズ状態のデータベースの回復の準備

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

```
SHUTDOWN ABORT
```

2. メディア障害からの回復の場合は、可能であればそのメディア障害を修復します。
3. 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あり、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、データベースを起動し、通常の操作を再開します。

```
STARTUP
```

必要なファイルの復元

1. 14-3 ページの「[回復するファイルの判別](#)」で説明している方法を使用して、回復するデータ・ファイルを判別します。
2. ファイルの破損が永久的な場合には、破損したファイルの最新のバックアップを指定します。メディア障害によって破損したデータ・ファイルのみを復元してください。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。

たとえば、破損したファイルが `/oracle/dbs/tbs_10.f` の場合には、記録を調べ、最新のバックアップが `/oracle/backup/tbs_10.backup` であることを確認できます。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、それを回復に使用できます。

3. O/S ユーティリティを使用し、ファイルをデフォルト位置または新しい位置に復元します。たとえば、UNIX ユーザーが `/oracle/dbs/tbs_10.f` をデフォルト位置に復元するときには、次のように入力します。

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

データ・ファイルのバックアップをどこに復元するかを判断するときには、以下のガイドラインを使用してください。

状況	方針
ハードウェアの問題が修復されたため、データ・ファイルをデフォルト位置に復元できる	データ・ファイルをデフォルト位置に復元し、メディア回復を開始する。
ハードウェアの問題が続いているため、データ・ファイルを元の位置に復元できない	データ・ファイルを代替の記憶デバイスに復元する。これらのファイルの新しい位置を制御ファイルに指示します。『Oracle8i 管理者ガイド』の「データ・ファイルの改名および再配置」に記載されている操作を必要に応じて使用してください。

復元したデータ・ファイルの回復

1. 管理者権限を使用して Oracle に接続し、データベースの新しいインスタンスを起動してマウントしますが、データベースはオープンしないでください。たとえば、次のように入力します。

```
STARTUP MOUNT
```

2. 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` ビューを問い合わせます。たとえば、次のように入力します。

```
SELECT name FROM v$datafile;
```

3. データベースのすべてのデータ・ファイルがオンラインになっていることを確認します。たとえば、`/oracle/dbs/tbs_10.f` という名前のデータ・ファイルがオンラインになっていることを確認するには、次のように入力します。

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

指定のデータ・ファイルがすでにオンラインになっている場合には、この文は無視されます。必要であれば、すべてのデータ・ファイルを一度にオンラインにする、次のようなスクリプトを作成します。

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

4. データベース、表領域またはデータ・ファイルを回復するコマンドを発行します。たとえば、次のように入力します。

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

どの文を実行するかを判断するために、次のガイドラインを使用してください。

目的	方針
破損したすべてのファイルを一度に回復する	RECOVER DATABASE（推奨）または ALTER DATABASE RECOVER DATABASE を実行する
個別の表領域を回復する	RECOVER TABLESPACE（推奨）または ALTER DATABASE RECOVER TABLESPACE を実行する
破損した個々のデータ・ファイルを回復する	RECOVER DATAFILE（推奨）または ALTER DATABASE RECOVER DATAFILE を実行する
データベース全体の回復または個別のデータ・ファイルの回復をパラレル化する	14-19 ページの「 パラレル方式でのメディア回復の実行 」の項を参照してください。

5. アーカイブ REDO ログの適用を自動化しない場合には、プロンプトから必要な REDO ログ・ファイルが提示されるたびに、ログ・ファイルを受け入れるか、拒否します。回復を自動化してある場合には、必要なログが自動的に適用されます。必要なアーカイブおよびオンライン 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。
6. Oracle はメディア回復の完了を通知します。完全メディア回復でアーカイブ REDO ログが必要ない場合には、必要なオンライン REDO ログ・ファイルがすべて適用された後、回復が終了します。
7. データベースをオープンします。

```
ALTER DATABASE OPEN;
```

関連項目 : REDO ログ・ファイルの適用の詳細は、14-19 ページの「[完全メディア回復の実行](#)」を参照してください。

オープン状態のデータベースの回復

データベースのオープン中にメディア障害が発生し、破損していないデータ・ファイルがオンラインのまま、使用可能な状態で残ることもあります。破損したデータ・ファイルは自動的にオフラインとなります。

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。

この項では、次のような順序で、取消しベースのメディア回復の実行方法を説明します。

1. データベースがオープン状態であることを確認し、影響を受ける表領域をオフラインにし、データベースの回復の準備をします。
2. 必要なファイルを復元します。
3. データベースを回復します。

関連項目 : SYSTEM 表領域の完全メディア回復を先に進めるには、14-20 ページの「[クローズ状態のデータベースの回復](#)」に記載された手順を使用します。

データベースの停止中におけるオープン状態のデータベースの回復の準備

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

```
STARTUP
```

2. 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。たとえば、表領域 TBS_1 に破損したデータ・ファイルが含まれている場合には、次のように入力します。

```
ALTER TABLESPACE tbs_1 OFFLINE TEMPORARY;
```

3. 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し、データベース回復の次のステップに進みます。

オープン状態のデータベースの回復の準備

1. 回復が必要であると判断した時点でデータベースがオープン状態であった場合は、破損したデータ・ファイルを含むすべての表領域をオフラインにします。たとえば、表領域 TBS_1 に破損したデータ・ファイルが含まれている場合は、次のように入力します。

```
ALTER TABLESPACE tbs_1 OFFLINE TEMPORARY;
```

可能であれば、回復の量を最小限にするために、TEMPORARY オプションを指定し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し、データベース回復の次のステップに進みます。

オープン状態のデータベースにデータ・ファイルを復元する方法

1. 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・ファイルのみを復元します。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決され、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修復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、ALTER DATABASE CREATE DATAFILE を使用して空の置換ファイルを作成し、それを回復に使用できます。

2. 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ときには、こ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定してください。指定するときに、必要に応じて『Oracle8i 管理者ガイド』の中の「データ・ファイルの改名および再配置」の項で説明されている手順を使用してください。

オープン状態のデータベースのオフライン表領域の回復

1. 管理者権限を使用してデータベースに接続します。たとえば、SYS としてデータベース PROD1 に接続します。

```
% sqlplus sys/sys_pwd@prod1
```

2. 1 回の作業で、1 つ以上のオフライン表領域の中の破損しているすべてのデータ・ファイルのオフライン表領域の回復を開始します。

```
RECOVER TABLESPACE tbs_1 # begins recovery on datafiles in tbs_1
```

注意： パフォーマンスを最大にするには、パラレル回復を使用してデータ・ファイルを回復します。14-19 ページの「[パラレル方式でのメディア回復の実行](#)」を参照してください。

3. 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。SET AUTORECOVERY ON を使用してファイルの適用が自動化されている場合以外は、必要な REDO ログ・ファイルごとに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。

4. 破損した表領域がメディア障害の発生時点まで回復されると、オフライン表領域をオンラインにします。たとえば、表領域 TBS_1 をオンラインにするには、次のコマンドを発行します。

```
ALTER TABLESPACE tbs_1 ONLINE;
```

関連項目 : REDO ログの適用の詳細は、14-19 ページの「[完全メディア回復の実行](#)」を参照してください。データ・ファイルの作成の詳細は、『Oracle8i 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。次の項目を取り上げます。

- [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](#)

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 回設定されているため、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、問題が発生する可能性があります。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、取消しベースのメディア回復の実行方法を説明します。

1. データベースをバックアップし、メディア障害を修復して回復の準備を行います。
2. バックアップ制御ファイル（必要であれば）とバックアップ・データ・ファイルを復元します。
3. RECOVER DATABASE 文を使用し、CANCEL で終了し、復元されたバックアップのメディア回復を実行します。

取消しベースの回復の準備

1. データベースが引き続きオープン状態であり、メディア回復が必要な場合は、インスタンスを終了します。

SHUTDOWN ABORT

2. 回復作業の途中でエラーが発生した場合に備えて、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、その障害の原因となったハードウェア障害を修復します。

取消しベースの回復に必要なファイルの復元

1. 現行の制御ファイルが、回復の目標時点のデータベースの物理構造と一致しない場合、つまり、回復の目標時点以降にデータ・ファイルが追加されている場合には、制御ファイルのバックアップを復元します。

復元された制御ファイルには、データベースの物理ファイル構造を反映する必要があります。つまり、制御ファイルには、不完全メディア回復の終了目標時点での、データ・ファイルおよびオンライン REDO ログ・ファイルの名前が含まれている必要があります。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。代替策は、新しい制御ファイルを作成することです。

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、この制御ファイルは、データベースに関連するパラメータ・ファイルの中の CONTROL_FILES パラメータ・リストから除いてください。

2. データベースのすべてのデータ・ファイルの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られたバックアップを復元します。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にしてください。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。
3. 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。

4. メディア障害の原因となっていたハードウェアの問題を解決したため、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、ファイルを復元してください。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用している場合は、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。

取消しベースの回復の実行

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

```
% sqlplus sys/change_on_install@prod1
```

2. 新しいインスタンスを起動し、データベースをマウントします。

```
STARTUP MOUNT
```

3. 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定してください。
4. 取消しベースの回復を開始します。

```
RECOVER DATABASE UNTIL CANCEL
```

この不完全回復でバックアップ制御ファイルを使用している場合には、RECOVER コマンドの中で USING BACKUP CONTROLFILE オプションを指定してください。

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

注意： RECOVER 文で UNTIL CANCEL 句を指定しないと、完全回復を実行するまでデータベースをオープンできなくなります。

5. 復元されたデータ・ファイルを再構築するために、必要な REDO ログ・ファイルが適用されます。ファイルの適用が自動化されている場合以外は、Oracle は、LOG_ARCHIVE_DEST_1 または 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が必要です。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。

注意： OPS 構成を使用した場合、不完全回復を実行するか、バックアップ制御ファイルを使用すると、最初のスレッドからの最初のアーカイブ REDO ログ・ファイルの名前しか算出できません。他のスレッドからの最初の REDO ログ・ファイルの名前は、ユーザーが指定する必要があります。あるスレッド内の最初のログ・ファイルが指定されると、Oracle はそのスレッド内の後続のログ・ファイルの名前を推定できます。

6. 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。
7. 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、回復を取り消します。

CANCEL

回復が成功したかどうかを示すメッセージが戻されます。回復が完了する前に回復を取り消した後で、データベースをオープンしようとすると、ファイルでさらに回復が必要な場合は、ORA-1113 エラーが発生します。

時間ベースの回復の実行

この項では、時間ベースのメディア回復手順の実行方法を説明します。

1. 作業の途中でエラーが発生した場合に備えてデータベースをバックアップし、メディア障害がある場合は障害を修復してください。
2. バックアップ制御ファイル（必要であれば）とバックアップ・データ・ファイルを復元します。
3. UNTIL TIME オプションを指定した RECOVER DATABASE 文を使用し、復元されたバックアップのメディア回復を実行します。

注意： 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オラクル社カスタマ・サポートに連絡してください。

時間ベースの回復の準備

14-28 ページの「[取消しベースの回復の実行](#)」で説明している準備手順を使用してください。

時間ベースの回復に必要なファイルを復元し、ファイルをオンラインにする方法

1. 現行の制御ファイルが、回復の予定時点のデータベースの物理構造と一致しない場合には、不完全メディア回復の終了時点での、データベースの物理ファイル構造を反映し

た、バックアップ制御ファイルを復元します。どのバックアップ制御ファイルを使用するかを判断するには、次のようにします。

- 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択してください。
- 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、欠落している制御ファイルと置換することもできます。

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、この制御ファイルは、データベースに関連するパラメータ・ファイルの中の CONTROL_FILES パラメータ・リストから除いてください。

2. データベースのすべてのデータ・ファイルのバックアップを復元します。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は、回復の目標時点以前にとられたものである必要があります。たとえば、1月2日の午後2時の時点にまで回復する場合には、この時間より前に完了していたバックアップを使用し、すべてのデータ・ファイルを復元します。次のガイドラインを使用してください。

状況	方針
データ・ファイルのバックアップがない	空の置換ファイルを作成し、このファイルを回復に使用する。
回復の目標時点以降にデータ・ファイルが追加されている	このファイルは、回復の完了後にデータベースで使用されることはないため、このファイルのバックアップは復元しない。
障害の原因となっていたハードウェアの問題が解決されたため、すべてのデータ・ファイルをデフォルト位置に復元できる	ファイルを復元し、この手順のステップ 5 を省略する。
ハードウェアの問題が解決されていない	破損したデータ・ファイルを代替の記憶デバイスに復元する。

注意： 制御ファイルのバックアップを使用している場合は、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。

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

```
% sqlplus sys/change_on_install@prod1
```

4. 新しいインスタンスを起動し、データベースをマウントします。

```
STARTUP MOUNT
```

5. 破損した 1 つ以上のデータ・ファイルをステップ 2 で代替位置に復元した場合は、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定してください。たとえば、次のように入力します。

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

6. 回復に必要なデータ・ファイルの名前を次の方法で確認します。

- 使用中の制御ファイルに通常含まれるデータ・ファイルのリストをチェックする。
- VSDATAFILE ビューを問い合わせる。

7. データベースのすべてのデータ・ファイルがオンラインになっていることを確認します。オフライン表領域が NORMAL オプション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。たとえば、user1 という名前（完全修飾ファイル名）のデータ・ファイルを実際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用している場合（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用するダイアログ・ボックスまたはコマンドの中でこのことを指示します。指定のデータ・ファイルがすでにオンラインになっている場合は、この文は無視されます。

時間ベースの回復の実行

1. 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 'YYYY-MM-DD:HH24:MI:SS' という書式を使用し、引用符で区切って指定します。次の文を実行すると、制御ファイルのバックアップが使用され、データベースは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
```

2. 復元されたデータ・ファイルを再構築するために、必要な REDO ログ・ファイルを適用します。ファイルの適用が自動化されている場合以外は、Oracle は、LOG_ARCHIVE_DEST_1 または 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が必要です。制御ファイルがバックアップの場合には、オンライン・ログの名前を指定する必要があります。

3. 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・ファイルを適用していきます。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。

変更ベースの回復の実行

この項では、指定された SCN までの回復の実行方法を説明します。

1. 作業の途中でエラーが発生した場合に備えてデータベースをバックアップし、メディア障害がある場合は障害を修復してください。
2. バックアップ制御ファイル（必要であれば）とバックアップ・データ・ファイルを復元します。
3. UNTIL CHANGE オプションを指定した RECOVER DATABASE 文を使用し、復元されたバックアップのメディア回復を実行します。

変更ベースの回復の準備

14-28 ページの「[取消しベースの回復の実行](#)」で説明している準備手順を使用してください。

変更ベースの回復に必要なファイルを復元する方法

14-28 ページの「[時間ベースの回復の実行](#)」で説明している復元手順を使用してください。

変更ベースの回復の実行

1. 回復を終了する時点の SCN を指定し、変更ベースの回復を開始します。SCN には、引用符なしの 10 進数を指定します。たとえば、SCN 100 まで回復するには、次のコマンドを発行します。

```
RECOVER DATABASE UNTIL CHANGE 100;
```

2. 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、Oracle は、LOG_ARCHIVE_DEST_1 または 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が必要です。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。
3. 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。

メディア回復後のデータベースのオープン

回復後にデータベースをオープンするときに、ログ順序番号を維持するには、次の文を実行します。

```
ALTER DATABASE OPEN NORESETLOGS;
```

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、次の文を実行します。

```
ALTER DATABASE OPEN RESETLOGS;
```

RESETLOGS オプションは、不完全メディア回復後には必須です。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再度初期化する。
- 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 のいずれを指定するか決定

RESETLOGS または NORESETLOGS のいずれを指定するかを決定するには、次のルールに従います。

- 不完全メディア回復の後には必ず RESETLOGS オプションを指定してください。たとえば、現在より先の時刻や SCN でなく、以前の時刻や SCN を指定した場合などです。
- 完全メディア回復を実行した後は、NORESETLOGS オプションを指定してください（バックアップ制御ファイルを使用していない限り）。このルールが当てはまるのは、意図的に完全回復を実行した場合か、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。
- 完全回復を実行した場合でも、不完全回復を実行した場合でも、回復のために制御ファイルのバックアップを使用した場合には、必ず RESETLOGS オプションを指定してください。

- 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用する場合は、RESETLOGS オプションを指定しないでください。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する必要があります。

ログ順序番号をリセットした場合には、完全回復が実行されたか、不完全回復が実行されたかにより、戻されるメッセージは異なります。完全回復の場合は、alert.log ファイルに次のメッセージが出力されます。

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

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。

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

リセットしてはならない場合にログをリセットしようとしたり、リセットする必要がある場合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。

RESETLOGS モードでオープンする場合のガイドライン

RESETLOGS モードでデータベースをオープンした後、次のアクションを実行してください。

- 全体データベース・バックアップの作成
- アラート・ログのチェック

全体データベース・バックアップの作成 すぐにデータベースを NORMAL オプションで停止し、全データベースのバックアップを作成します。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。全バックアップをとるまでは、前述の手順（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。）

アラート・ログのチェック RESETLOGS オプションを使用してデータベースをオープンした後、alert.log をチェックし、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 NORMAL オプションでオフラインにされたデータ・ファイルのみ指すようにします。一方、MISSINGnnnn が読取り専用でも NORMAL オプションによる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できません。これは、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になるためです。この場合には、データ・ファイルを含む表領域を削除する必要があります。

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.log 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、14-19 ページの「[完全メディア回復の実行](#)」を参照してください。

RESETLOGS 以前のバックアップからの回復

Oracle8i より前のリリースでは、直ちに復元が実行できるように計画していた場合には、RESETLOGS オプションを使用したデータベースのオープンを避けるために、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の例は、ディスクのメンテナンス（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）を行う場合です。DBA は、RESETLOGS モードで再起動を行わない場合には、データベースの復元の直後に全体データベース・バックアップを実行する必要はありません。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、全体データベース・バックアップが必要でした。

Oracle8 以降では、ロールフォワードに RESETLOGS 前のバックアップを使用できる状況が 1 つだけあります。データベースの一貫性バックアップがあり、このバックアップが RESETLOGS オプションを使用してデータベースをオープンする直前にとられたものであり、さらに、RESETLOGS を使用してデータベースをオープンした後で有効な制御ファイルがある場合のみに限られます。このため、オンライン REDO ログをバックアップまたは復元する必要はありません。

次の例は、RESETLOGS 前のバックアップを使用できる状況を説明しています。ハードウェア・ストライプ化の再構成を実行する場合を考えてみます。これには、データベース・ファイルのバックアップと削除、ハードウェアの再構成、そしてデータベースの復元が必要です。

金曜日の夜に、次の作業を行います。

1. データベースを正しく停止します。

```
SHUTDOWN IMMEDIATE
```

2. 全体データベース・バックアップ（制御ファイルおよびデータ・ファイル）を実行します。たとえば、次のように入力します。

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

注意： この時点でデータベースを再オープンしないでください。

3. O/S メンテナンスを実行します。

4. データ・ファイルと制御ファイルを復元します。たとえば、次のように入力します。

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

5. データベースをマウントします。

```
STARTUP MOUNT
```

6. 取消しベースの回復を開始します。

```
RECOVER DATABASE UNTIL CANCEL
```

7. RESETLOGS オプションを使用してデータベースをオープンします。

```
ALTER DATABASE OPEN RESETLOGS;
```

土曜日の朝には、スケジュールされたバッチ・ジョブが実行され、アーカイブ REDO ログが生成されます。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、土曜日に作成されたログを使用してロールフォワードできます。

土曜日の夜には次の作業を行います。

1. インスタンスを終了します（まだ存在している場合）。

```
SHUTDOWN ABORT
```

2. すべての破損ファイルを、金曜日の夜に作成したバックアップから復元します。

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

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。復元した場合は、RESETLOGS オプションを指定してデータベースをオープンした後に有効だった制御ファイルも復元する必要があります。

3. 土曜日に作成されたすべてのアーカイブ・ログを適用し、完全回復を開始します。ログの適用を自動化するには SET AUTORECOVERY ON を使用します。

```
SET AUTORECOVERY ON  
RECOVER DATABASE
```

4. データベースをオープンします。

STARTUP

上記の例では、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の制御ファイルがない場合には、金曜日の夜のバックアップを使用してロールフォワードできなくなります。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。

オペレーティング・システム環境での 回復シナリオ

この章では、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。項目は次のとおりです。

- [メディア障害の種類を理解](#)
- [データ・ファイルの消失後の回復](#)
- [ADD DATAFILE 操作を使用した回復方法](#)
- [移動された表領域の回復](#)
- [オンライン REDO ログ・ファイルの消失後の回復](#)
- [アーカイブ REDO ログ・ファイルの消失後の回復](#)
- [制御ファイルの消失後の回復](#)
- [ユーザー・エラーからの回復](#)
- [分散環境でのメディア回復の実行](#)

メディア障害の種類の理解

メディア障害には、一般的に 2 つの種類があります。永久的なものと一時的なものです。障害の種類によって、異なる回復方法を使用します。

永久メディア障害は、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問題です。失われたデータを回復するには、障害が発生し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可能になるようなハードウェア問題ですが、このために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れば、ディスク上のデータにアクセスできます。
- 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・ファイルの消失後の回復

データ・ファイルがメディア障害の影響を受けた場合は、回復手順は次の状況によって異なります。

- データベースのアーカイブ・モード ARCHIVELOG または NOARCHIVELOG.
- メディア障害の種類
- メディア障害の影響を受けたファイル

次の項では、データベース・モードの適切な回復の方針について説明します。

- [NOARCHIVELOG モードでのデータ・ファイルの消失](#)
- [ARCHIVELOG モードでのデータ・ファイルの消失](#)

NOARCHIVELOG モードでのデータ・ファイルの消失

NOARCHIVELOG モードで運用されているデータベースのデータ・ファイルのどれかが永久または一時メディア障害の影響を受けた場合には、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 つの回復方法のいずれかを使用できます。

- メディア障害が一時的な場合には、ハードウェア問題を修復した後で、データベースを再起動します。通常はインスタンス回復が可能であり、オンライン REDO ログを使用してすべてのコミットされたトランザクションを回復できます。
- 永久メディア障害の場合には、14-16 ページの「[NOARCHIVELOG モードで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。

ARCHIVELOG モードでのデータ・ファイルの消失

ARCHIVELOG モードで運用されているデータベースのデータ・ファイルが永久または一時メディア障害の影響を受ける状況として、次の例が想定されます。

破損したデータ・ファイル	データベース・ステータス	解決策
SYSTEM 表領域の中のデータ・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。	Oracle は停止します。	ハードウェア問題が一時的な場合は、修復後にデータベースを再起動してください。通常の場合は、インスタンス回復によりトランザクションを回復できます。 ハードウェア問題が永久的な場合には、14-20 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。
SYSTEM データ・ファイル以外のデータ・ファイルか、アクティブ・ロールバック・セグメントを含まないデータ・ファイル。	Oracle は影響を受けたデータ・ファイルをオフラインにしますが、データベースはオープンしたままです。	データベースのうち影響を受けなかった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。一時オプションを使用して破損したデータ・ファイルを含む表領域をオフラインにした後、14-20 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。

ADD DATAFILE 操作を使用した回復方法

ADD DATAFILE 操作によってデータベース回復がロールフォワードされた場合には、Oracle は ADD DATAFILE REDO データの適用時に回復を停止し、ユーザーはファイルの位置を確認できます。

たとえば、/db/db2.f および /db/db3.f という 2 つのデータ・ファイルを含む新しい表領域を作成するとします。その後 CREATE TABLESPACE 操作を使用してメディア回復を実行した場合、Oracle は CREATE TABLESPACE REDO データの適用時に次のエラーを発行する可能性があります。

```
ORA-00283: recovery session canceled due to errors
ORA-01244: unnamed datafile(s) added to controlfile by media recovery
ORA-01110: data file 3: '/db/db2.f'
ORA-01110: data file 2: '/db/db3.f'
```

ADD DATAFILE 操作による回復

1. V\$DATAFILE から選択し、追加されたファイルを表示します。

```
SELECT file#, name FROM v$datafile;

FILE#          NAME
-----
1              /db/db1.f
```

```
2          /db/UNNAMED00002
3          /db/UNNAMED00003
```

2. 名前のないファイルが複数存在している場合は、次の方法の1つを使用し、名前のない各ファイルがどのデータ・ファイルに対応しているかを判別します。

- 名前のない各ファイルの元のファイル位置に関するメッセージを含む alert.log をオープンする。
- 名前のない各ファイルの元のファイル位置をエラー・メッセージと V\$DATAFILE から導出する。名前のないファイルは、エラー・メッセージの中の、同じファイル番号を持つファイルに対応しています。

3. ALTER DATABASE RENAME FILE コマンドを発行し、データ・ファイルを改名します。たとえば、次のように入力します。

```
ALTER DATABASE RENAME FILE '/db/UNNAMED00002' TO '/db/db3.f';
ALTER DATABASE RENAME FILE '/db/UNNAMED00003' TO '/db/db2.f';
```

4. 前述の回復コマンドを使用して回復を続行します。たとえば、次のように入力します。

```
RECOVER DATABASE
```

移動された表領域の回復

Oracle のトランスポートابل表領域機能を使用すると、1つのデータベースから別のデータベースに表領域のセットを移送できます。表領域をデータベースに移送する、つまり "組み込む" ことと、あらかじめロードされたデータを使用して表領域を作成することは類似しています。この機能は、次の理由から便利です。

- この方法で必要になる作業は、データ・ファイルのコピーと、メタデータの統合のみであるため、インポート / エクスポートまたはロード / アンロードよりも速い。
- この方法を使用して索引データを移動すると、索引を再作成する必要がなくなる。

移動された表領域は通常の表領域と同様に、回復可能です。通常の表領域はバックアップなしに回復できるのに対して、移動された表領域を回復するには、移動されたデータ・ファイルのバックアップが必要です。

移動された表領域を回復するには、移送されたデータ・ファイルのバックアップを復元し、通常の回復コマンドを発行します。このバックアップは、移送されたデータ・ファイルの最初のバージョンでも、表領域の移送後にとられた任意のバックアップでもかまいません。表領域の組み込み操作を通じて回復するときには、CREATE TABLESPACE 操作を使用して回復する場合と同じように、ORA-01244 が発行される可能性があります。この場合には、15-3 ページの「[ADD DATAFILE 操作を使用した回復方法](#)」で説明している手順を使用し、正しい位置へ名前のないファイルを改名してください。

関連項目 : 移動された表領域の使用方法的詳細は、『Oracle8i 管理者ガイド』を参照してください。

オンライン REDO ログ・ファイルの消失後の回復

データベースのオンライン REDO ログがメディア障害の影響を受けた場合、正しい回復手順は次の状況によって異なります。

- オンライン REDO ログの構成。ミラー化の有無。
- メディア障害の種類。一時か、永久か。
- メディア障害の影響を受けた、オンライン REDO ログ・ファイルの種類。現行、アクティブ、アーカイブされていない、またはアクティブでないか。

次の項で、これらの状況に適した回復方針について説明します。

- [多重化されたオンライン REDO ログ・グループのメンバーの消失後の回復](#)
- [オンライン REDO ログ・グループの全メンバーの消失後の回復](#)

多重化されたオンライン REDO ログ・グループのメンバーの消失後の回復

データベースのオンライン REDO ログを多重化している場合に、各オンライン REDO ログ・グループの少なくとも 1 つのメンバーがメディア障害の影響を受けていないときには、データベースは正常に機能し続けることができます。Oracle は、LGWR トレース・ファイルと、データベースの alert.log にエラー・メッセージを書き込みます。

次のアクションのいずれかを実行し、問題を解決してください。

- ハードウェア問題が一時的な場合は、修復する。この後 LGWR は、問題が存在しなかったかのように、以前は使用不可能であったオンライン REDO ログ・ファイルにアクセスします。
- ハードウェアの問題が永久的な場合には、次の手順を使用し、破損したメンバーを削除し、新しいメンバーを追加する。

注意： ログ・グループが再使用されるまでは、新しく追加されたメンバーによって冗長性が生じることはありません。

REDO ログ・グループの破損したメンバーの置換

1. VSLOGFILE の中で破損したメンバーのファイル名を検索します。ファイルにアクセス不可能な場合は、ステータスは INVALID になります。

```
SELECT group#, status, member FROM v$logfile;
```

GROUP#	STATUS	MEMBER
0001	-----	-----
0001		/oracle/dbs/log1a.f
0001		/oracle/dbs/log1b.f

```
0002          /oracle/dbs/log2a.f
0002      INVALID  /oracle/dbs/log2b.f
0003          /oracle/dbs/log3a.f
0003          /oracle/dbs/log3b.f
```

2. 破損したメンバーを削除します。たとえば、グループ 2 からメンバー log2b.f を削除するには、次のコマンドを発行します。

```
ALTER DATABASE DROP LOGFILE MEMBER '/oracle/dbs/log2b.f';
```

3. グループに新しいメンバーを追加します。たとえば、グループ 2 に log2c.f を追加するには、次のコマンドを発行します。

```
ALTER DATABASE ADD LOGFILE MEMBER '/oracle/dbs/log2c.f' TO GROUP 2;
```

追加するファイルがすでに存在している場合には、ファイルのサイズは他のグループ・メンバーのサイズと同じにし、REUSE を指定する必要があります。

```
ALTER DATABASE ADD LOGFILE MEMBER '/oracle/dbs/log2b.f' REUSE TO GROUP 2;
```

オンライン REDO ログ・グループの全メンバーの消失後の回復

メディア障害によりオンライン REDO ログ・グループの全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、種々のシナリオがあります。

破損したログ・グループがアクティブでない場合には、インスタンス回復は必要ありません。ログ・グループがアクティブな場合には、インスタンス回復が必要です。最初の作業は、破損したグループがアクティブか、アクティブでないかを判別することです。

破損したグループがアクティブかどうかの判別方法

1. 消失した REDO ログのファイル名を V\$LOGFILE の中で確認し、これに対応するグループ番号を探します。たとえば、次のように入力します。

```
SELECT group#, status, member FROM v$logfile;
```

GROUP#	STATUS	MEMBER
0001		/oracle/dbs/log1a.f
0001		/oracle/dbs/log1b.f
0002	INVALID	/oracle/dbs/log2a.f
0002	INVALID	/oracle/dbs/log2b.f
0003		/oracle/dbs/log3a.f
0003		/oracle/dbs/log3b.f

2. どのグループがアクティブかを判別します。たとえば、次のように入力します。

```
SELECT group#, members, status, archived FROM v$log;
```


GROUP#	MEMBERS	STATUS	ARCHIVED
-----	-----	-----	-----
0001	2	INACTIVE	YES
0002	2	ACTIVE	NO
0003	2	CURRENT	NO

3. 影響を受けたグループがアクティブでない場合には、15-7 ページの「[アクティブでない、オンライン REDO ログ・グループの消失](#)」で説明している手順に従ってください。影響を受けたグループがアクティブな場合（前述の例のように）には、15-9 ページの「[アクティブなオンライン REDO ログ・グループの消失](#)」で説明している手順に従ってください。

アクティブでない、オンライン REDO ログ・グループの消失

INACTIVE ステータスのオンライン REDO ログ・グループの全メンバーが破損している場合には、アクティブでない REDO ログ・グループに損傷を与えたメディア問題を修正できるかどうかで、手順が異なります。

メディア障害の種類	手順
一時	問題を解決します。LGWR は必要なときに REDO ログ・グループを再使用できます。
永久	破損したアクティブでないオンライン REDO ログ・グループにより、結果的にデータベースの通常の運用が停止します。次に説明するように、ALTER DATABASE CLEAR LOGFILE を使用し、破損したグループを手動で消去、つまり初期化し直してください。

データベースをオープンしているときも、クローズしているときも、アクティブな REDO ログ・グループを消去できます。手順は、破損したグループがアーカイブ済みかどうかによって異なります。

アーカイブ済みの、アクティブでないオンライン REDO ログ・グループの消去

1. データベースを停止している場合には、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ますが、データベースはオープンしないでください。

```
STARTUP MOUNT
```

2. 破損したログ・グループを再初期化します。たとえば、REDO ログ・グループ 2 を再初期化するには、次のコマンドを発行します。

```
ALTER DATABASE CLEAR LOGFILE GROUP 2;
```

アーカイブされていない、アクティブでないオンライン REDO ログ・グループの消去

アーカイブされていないログを消去した場合には、このログをアーカイブせずに、再使用できます。このアクションにより、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き、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。このため、消去したログ・ファイルがバックアップの回復のために必要である場合には、そのバックアップを回復することはできません。

1. データベースを停止している場合には、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ますが、データベースはオープンしないでください。

```
STARTUP MOUNT
```

2. UNARCHIVED キーワードを使用してログを初期化します。たとえば、ログ・グループ 2 を初期化するには、次のコマンドを発行します。

```
ALTER DATABASE CLEAR LOGFILE UNARCHIVED GROUP 2;
```

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 を指定する必要があります。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する必要があります。なぜなら、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、そのコピーもないからです。たとえば、次のように入力します。

```
ALTER DATABASE CLEAR LOGFILE UNARCHIVED GROUP 2 UNRECOVERABLE DATAFILE;
```

3. O/S ユーティリティを使用し、データベースを即時にバックアップします。これで、消去したログ・グループに頼ることなく、このバックアップを完全回復に使用できます。たとえば、次のように入力します。

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

4. データベースの制御ファイルのバックアップをとります。

```
ALTER DATABASE BACKUP CONTROLFILE TO 'filename';
```

CLEAR LOGFILE 操作の失敗 ALTER DATABASE CLEAR LOGFILE コマンドは、次の 2 つの作業が不可能な場合には、メディア障害による I/O エラーのために失敗する可能性があります。

- 現在構成されている REDO ログ・ファイル名を使用して再作成することにより、REDO ログ・ファイルを別のメディアに再配置する。
- REDO ログ・ファイルの名前自体が無効または使用不可能であるため（たとえばメディア障害により）、このファイルを再作成するために現在構成されているログ・ファイル名を再使用する。

これらの場合には、CLEAR LOGFILE コマンドは、ログが消去されているためアーカイブが不要ないことを（I/O エラーの受信前に）、制御ファイルに正常に伝達しているはずで

この I/O エラーは、CLEAR LOGFILE で新しい REDO 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。

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 コマンドを発行します。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、15-9 ページの「[アクティブなオンライン REDO ログ・グループの消失](#)」の手順を実行できます。このコマンドが失敗した場合や、データベースが停止した場合には、どのようなアーカイブ・モードかによって、次の手順のいずれかを実行してください。

NOARCHIVELOG モードで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方法

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. O/S ユーティリティを使用し、全体データベース・バックアップからデータベースを復元します。たとえば、次のように入力します。

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

3. データベースをマウントします。

```
STARTUP MOUNT
```

4. RESETLOGS オプションを使用してデータベースをオープンします。

```
ALTER DATABASE OPEN RESETLOGS;
```

5. データベースを通常の方法で停止します。

```
SHUTDOWN IMMEDIATE
```

6. 全体データベース・バックアップを作成します。たとえば、次のように入力します。

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

ARCHIVELOG モードで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方法

1. メディア障害が一時的なものであれば修復し、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. 不完全メディア回復を実行します。14-25 ページの「[不完全メディア回復の実行](#)」で説明した手順を使用し、破損したログの直前のログまでを回復します

3. 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、破損したオンライン REDO ログ・グループのメンバーの名前を新しい位置に改名します。たとえば、次のように入力します。

```
ALTER DATABASE RENAME FILE '/oracle/dbs/log_1.rdo' TO '/temp/log_1.rdo';
ALTER DATABASE RENAME FILE '/oracle/dbs/log_2.rdo' TO '/temp/log_2.rdo';
```

4. RESETLOGS オプションを使用してデータベースをオープンします。

```
ALTER DATABASE OPEN RESETLOGS;
```

注意： 不完全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を消失した場合には、回復が最も困難なログに対する回復方法を使用してください。困難の度合いを、最も困難なものから順番にリストすると、次のようになります。

- 1. 現行のオンライン REDO ログ
- 2. アクティブなオンライン REDO ログ
- 3. アーカイブしていない 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・ファイルの消失後の回復

データベースを ARCHIVELOG モードで運用しているときに、アーカイブ REDO ログ・ファイルの唯一のコピーが破損した場合でも、データベースの現在の運用に影響はありません。しかし、REDO ログがいつ作成されたか、そしてデータ・ファイルがいつバックアップされたかによって、次の状況が発生する可能性があります。

何をバックアップしたか	方針
満杯になったオンライン REDO ログ・グループ（現在アーカイブしているもの）の書込み後のすべてのデータ・ファイル。	完全メディア回復操作には、満杯になったオンライン REDO ログ・グループのアーカイブされたバージョンは必要ありません。
満杯になったオンライン REDO ログ・グループの書込み前の特定のデータ・ファイル。	対応するデータ・ファイルが永久メディア障害によって破損した場合には、破損したデータ・ファイルの最新のバックアップを使用し、破損したログまでの不完全回復を実行します。

警告： アーカイブ REDO ログ・グループが破損していることが判明したときは、即時にすべてのデータ・ファイルのバックアップを作成し、破損したアーカイブ REDO ログを必要としない、全体データベース・バックアップを作成しておいてください。

制御ファイルの消失後の回復

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（制御ファイルが多重化されているかどうかに関係なく）その後最初に Oracle のバックグラウンド・プロセスで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発生した時点で、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し、データベースの自動停止を防止してください。しかし、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には、問題を修復し、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、制御ファイルを多重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。

- [多重化された制御ファイルのメンバーの消失](#)
- [現行の制御ファイルのすべてのコピーの消失](#)

多重化された制御ファイルのメンバーの消失

次の条件のすべてが当てはまるときには、この後の手順を使用してデータベースを回復します。

- 永久メディア障害により、データベースの 1 つ以上の制御ファイルが破損した。
- メディア障害による破損がない制御ファイルが少なくとも 1 つある。

注意： 制御ファイルを多重化した構成ですべての制御ファイルが破損した場合には、15-12 ページの「[現行の制御ファイルのすべてのコピーの消失](#)」で説明した手順に従ってください。

制御ファイルをデフォルト位置に復元する方法

1. インスタンスをまだ実行している場合は、これを終了します。

```
SHUTDOWN ABORT
```

2. メディア障害の原因になったハードウェア問題を修復します。ハードウェア問題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. データベースの現行の制御ファイルの多重化された無傷のコピーを使用し、破損した制御ファイルに上書きします。たとえば、`bad_cf.f` を `good_cf.f` で置換する場合には、次のように入力します。

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

4. インスタンスを起動し、データベースをオープンします。

```
STARTUP
```

制御ファイルのデフォルト以外の位置への復元

1. インスタンスをまだ実行している場合は、これを終了します。

```
SHUTDOWN ABORT
```

2. メディア障害の原因となったハードウェア問題を解決できない場合には、無傷の制御ファイルを代替位置にコピーします。たとえば、`good_cf.f` を `new_cf.f` に改名するには、次のコマンドを発行します。

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

3. `CONTROL_FILES` パラメータにすべての制御ファイルの現在位置が反映され、復元されていない制御ファイルはすべてここから除外されるようにするために、データベースのパラメータ・ファイルを編集します。たとえば、`new_cf.f` を追加するには、次のように入力します。

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

4. 新しいインスタンスを起動し、データベースをオープンします。

```
STARTUP
```

現行の制御ファイルのすべてのコピーの消失

永久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、新しい制御ファイルを作成することによりデータベースを回復できます。

新しい制御ファイルの作成

1. NORESETLOGS オプションを指定した CREATE CONTROLFILE 文を使用し、制御ファイルを作成します。表 15-1 でオプションを参照してください。
2. 通常の方法でデータベースを回復します。

```
RECOVER DATABASE
```

制御ファイルのバックアップの存在と作成の時期に応じて、CREATE CONTROLFILE コマンドの作成時に次のオプションが使用できます。

表 15-1 制御ファイル作成のオプション

状況	方針
データベースの最後の構造変更後に ALTER DATABASE BACKUP CONTROLFILE TO TRACE NORESETLOGS を実行し、SQL コマンド出力を保存してある。	出力からの CREATE CONTROLFILE 文をそのまま使用する。
データベースの構造変更を行う前に、ALTER DATABASE BACKUP CONTROLFILE TO TRACE の最新の実行を行っている。	その変更を反映するように、ALTER DATABASE BACKUP CONTROLFILE TO TRACE の出力を編集する。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 文の DATAFILE 句に追加します。
TO TRACE オプションを使用して制御ファイルをバックアップせず、ALTER DATABASE BACKUP CONTROLFILE の TO <i>filename</i> オプションを使用している。	制御ファイルのコピーを使用して SQL コマンドの出力を取得する。バックアップ制御ファイルをコピーし、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してください。
TO TRACE 書式でも TO <i>filename</i> 書式でも制御ファイルのバックアップがない。	手動で CREATE CONTROLFILE 文を生成する。

注意： キャラクタ・セットがデフォルトの US7ASCII でない場合には、CREATE CONTROLFILE 文の引数としてキャラクタ・セットを指定する必要があります。

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： 厳選した適切なユーザーのみに重要な権限（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。

次のシナリオでは、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、オンラインのまま通常で使える状態にしておくことができます。データベースは、オープンしたままでも、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発生した場合に備えて、既存のデータベースのすべてのデータ・ファイルのバックアップをとります。
2. 時間ベースの回復を使用し、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。永久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代替位置に復元し（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代替位置に作成します。また、データベースの永久コピーに影響を与えないように、すべてのデータ・ファイルを代替位置に復元します（ステップ5）。
3. Oracle の Export ユーティリティを使用し、データベースを一時的に復元したものから、失われたデータをエクスポートします。この場合、誤って削除された表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用し、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久コピーにインポートします。
5. ディスク領域節約のため、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 および Export ユーティリティの詳細は、『Oracle8i ユーティリティ・ガイド』を参照してください。

分散環境でのメディア回復の実行

メディア回復の実行方法は、データベースが分散データベース・システムの一部であるかどうかによって異なります。Oracle 分散データベース・アーキテクチャは、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種類によっては、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する必要がある場合があります。

表 15-2 では、各種の回復操作を示し、分散データベース・システムのノード間の調整が必要かどうかを簡単に説明しています。

表 15-2 分散データベース環境の回復操作

状況	方針
リモート・ノードからアクセスされたことのないデータベース用の全体バックアップの復元	独立したデータベース回復を使用する。
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、同じ時点に実行した全体バックアップを使用し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の実行	独立したデータベース回復を使用する。
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の実行	独立したデータベース回復を使用する。
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の実行	分散データベース内のすべてのデータベースについて、同じグローバルな時点の状態への、不完全メディア回復を使用する。

Time-Based および Change-Based での分散データベース・リカバリの調整

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する場合があります。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する必要がよくあります。この操作を、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、次の作業を実行する必要があります。

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い、最初に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. データベースの回復が終了した後、RESETLOGS オプションを使用し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.log から RESETLOGS メッセージを探します。

メッセージが "RESETLOGS after complete recovery through change xxx," 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。回復すると、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。

メッセージが "RESETLOGS after incomplete recovery UNTIL CHANGE xxx," であった場合には、不完全回復が正常に完了したことになります。メッセージの変更番号を記録し、次のステップに進んでください。

3. 変更ベースの回復を使用し、ステップ2で記録した変更番号（SCN）を指定し、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・データベースを個別に、過去の時点まで回復した（つまり、調整式の、時間ベースの分散データベース回復を実行していない）場合には、消失時間中にリフレッシュされた、すべての依存リモート・スナップショットは、マスター表との整合性がなくなります。この場合には、マスター・データベースの管理者は、整合性の失われたすべてのスナップショットについて、完全リフレッシュを実行するようにリモート側の管理者に指示する必要があります。

スタンバイ・データベースの管理

この章では、スタンバイ・データベースの管理方法について説明します。ここでは次の項目を取り上げます。

- スタンバイ・データベースの計画
- スタンバイ・データベースの作成
- スタンバイ・データベース・モードの選択
- スタンバイ・データベースの回復モードのメンテナンス
- スタンバイ・データベースの読取り専用モードでのオープン
- スタンバイ・データベースのアクティブ化
- プライマリ・データベースの物理構造の変更
- OPS 構成でのスタンバイ・データベースの使用方法

スタンバイ・データベースの計画

バックアップおよび回復方針を作成するときには、[スタンバイ・データベース \(standby database\)](#) を持つかどうかを検討してください。本番データベースの重複コピーをメンテナンスしたい場合には、スタンバイ・データベースを使用できます。

本番サイトのすべてのメディアが破壊された場合には、破壊されたまたは破損したデータベースをスタンバイ・データベースで置換できます。災害に対する備えを万全にするために、スタンバイ・データベースのデータ・ファイル、REDO ログ・ファイルおよび制御ファイルは、プライマリ・データベースとは別の地域にある、別個の物理メディアに入れておいてください。

また、本番データベースとスタンバイ・データベースを同一のデータ・センターに配置することも、同一のマシン上に配置することも可能です。スタンバイ・データベースをレポート生成のために読み取り専用モードで使用する場合には、この構成が便利です。

この項では以下の項目を取り上げます。

- [スタンバイ・データベースの利点](#)
- [スタンバイ・データベースの要件](#)

スタンバイ・データベースの利点

スタンバイ・データベースは、災害の防止と、レポート生成のための強力なツールとして使用できます。次の作業を行うことができます。

- アーカイブ REDO ログを本番データベースからスタンバイ・データベースに自動的に転送することにより、スタンバイ・データベースを最新の状態に保つ。
- 本番データベースから地域的に離れた場所にスタンバイ・データベースを配置する、あるいは複数の地域的に離れた場所に複数のスタンバイ・データベースを配置する。
- 本番データベースが完全に破壊された場合に、時間とデータの損失を最小限に抑え、スタンバイ・データベースを新しい本番データベースにする。
- 破損したデータを含むアーカイブ・ログが伝播する前にスタンバイ・データベースをアクティブにすることにより、本番データベースによるバッチ・ジョブのエラーやアプリケーションの破壊を防止する。

スタンバイ・データベースの要件

スタンバイ・データベースをメンテナンスする場合は、次の要件に注意してください。

- スタンバイ・データベースは、Oracle リリース 7.3 以降でしか使用できない。
- スタンバイ・データベースに適用する REDO ログは、アーカイブ REDO ログまたは現行でない REDO ログのいずれかにする必要があります。
- 本番サイトとスタンバイ・サイトでは同一バージョン、リリース、およびパッチのオペレーティング・システムを使用する必要がある。しかし、スタンバイ・サイトでは異なるディスク構成を使用できます。
- フェイルオーバーが影響されることのないように、本番データベースとスタンバイ・データベースでは Oracle RDBMS の同一バージョン、リリース、およびパッチを使用する。
- プライマリ・データベースの制御ファイルをスタンバイ・データベースでは使用できない。
- 本番データベースとスタンバイ・データベースを同一ホスト上に配置した場合、オペレーティング・システムによっては、同一マシン上に同一のデータベース名を持つ 2 つのインスタンスを同時にマウントできないことがある。
- スタンバイ・データベースをアクティブ化した後で、スタンバイ回復モードに戻すことはできない。アクティブ化されたスタンバイ・データベースは通常の本番データベースとして機能するため、スタンバイ・データベースとして再作成する必要がある。

スタンバイ・データベースの作成

スタンバイ・データベースは、本番データベースと同じホスト上に作成することも、リモート・ホスト上に作成することもできます。同一ホスト上にスタンバイ・データベースを作成する場合には、重要なファイルを上書きしないように、慎重に作業を進めてください。

スタンバイ・データベースの作成

1. 本番用の `init.ora` ファイルをコピーしてスタンバイ用の `init.ora` ファイルを作成します。16-25 ページの「[初期化パラメータの設定](#)」の説明を参考に、スタンバイ初期化パラメータを作成します。
2. プライマリ・データベースで SQL*Plus セッションを開始し、`V$DATAFILE` に対して `SELECT` を発行してデータ・ファイルのリストを取得します。たとえば、次のように入力します。

```
SELECT name FROM v$datafile;
NAME
-----
/oracle/dbs/tbs_01.f
/oracle/dbs/tbs_02.f
```

```
/oracle/dbs/tbs_03.f
/oracle/dbs/tbs_11.f
/oracle/dbs/tbs_12.f
/oracle/dbs/tbs_21.f
/oracle/dbs/tbs_22.f
7 rows selected.
```

3. プライマリ・データベースを正しく停止します。

```
SHUTDOWN IMMEDIATE
```

4. O/S ユーティリティを使用し、プライマリ・データベースからデータ・ファイルの一貫性バックアップを作成します。
5. データベースをオープンします。

```
STARTUP
```

6. 本番データベースに接続し、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。たとえば、次のように入力します（この *filename* は完全修飾パス名です）。

```
ALTER DATABASE CREATE STANDBY CONTROLFILE AS 'filename';
```

7. プライマリ・データベースの現行のオンライン REDO ログをアーカイブします。

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

この操作は、データ・ファイル、制御ファイル、および REDO ログ・ファイルの一貫性を確保するためのものです。

8. 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用し、スタンバイ・データベースの制御ファイルとアーカイブ・ログ・ファイル、そしてバックアップ・データ・ファイルを、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用してください（16-6 ページの「[スタンバイ・データベースの回復モードのメンテナンス](#)」を参照してください）。たとえば、次のように入力します。

```
% cp /oracle/dbs/*.f /standby/oracle/dbs/*.f
```

9. プライマリ・データベースと同じファイル・システムを持つリモート・ホスト上にスタンバイ・データベースがある場合には、スタンバイ・データベースでもプライマリ・データベースと同じファイル名を使用できます。同じファイル名ではなく、区別をつけたい場合には、16-15 ページの「[データ・ファイルおよびアーカイブ REDO ログのファイル名の変換](#)」で説明しているファイル名変換初期化パラメータを使用してください。

関連項目： ALTER DATABASE と ALTER SYSTEM の構文については、『Oracle8i SQL リファレンス』を参照してください。

スタンバイ・データベース・モードの選択

スタンバイ・データベースは、3つの運用モードのいずれか1つで実行できますが、これらのモードを同時に実行することはできません。スタンバイ・データベースをマウントした後、次の運用モードを実行できます。

- データベースを手動または管理回復モードでメンテナンスする（16-6 ページの「[スタンバイ・データベースの回復モードのメンテナンス](#)」を参照）。
- 問合せに使用するために、読取り専用でデータベースをオープンする（16-19 ページの「[スタンバイ・データベースの読取り専用モードでのオープン](#)」を参照）。その後、データベースを回復モードに戻すことができます。
- アクティブにする。この時点で、このデータベースはスタンバイ・データベースではなくなります（16-21 ページの「[スタンバイ・データベースのアクティブ化](#)」を参照）。データベースは読み書き両用または読取り専用でオープンできます。

災害に備えるために、スタンバイ・データベースは回復モードにしておいてください。つまり、スタンバイ・データベースは、災害時回復を開始する目的以外で、問い合わせたり、オープンすることはできません。データベースでは、手動回復モードまたは管理回復モードを使用できます。手動回復モードではスタンバイ・データベースに絶えずアーカイブ REDO ログを転送および適用する必要がありますが、管理回復モードではこの手順は自動化されています。

スタンバイ・データベースは読取り専用モードでオープンすることもできます。このオプションを使用すると、データベースを問い合わせることができ、データ・ファイルまたは REDO ログに影響を与えることなく、一時表領域にデータを格納することもできます。回復モードに戻す場合は、いつでも戻すことができます。

注意： スタンバイ・データベースを読取り専用モードでオープンする場合には、COMPATIBLE パラメータを 8.1 またはこれ以上の値に設定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すると、スタンバイ・データベースはスタンバイ状態ではなくなり、本番データベースとして機能するようになります。別のスタンバイ・データベースを再作成しない限り、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本番データベースのログには互換性がありません。

スタンバイ・データベースの回復モードのメンテナンス

この項では、スタンバイ・データベースの回復モードをメンテナンスする方法を説明します。これにより、災害発生時に、データベースを復元および回復できるようになります。アーカイブ・ログを手動で、または自動的に適用するのかわによって、手動回復モードか、管理回復モードを選択できます。また、古くなったオンライン・ログ・ファイルの管理方法およびファイル名の変換方法も制御できます。

この項では次の項目を取り上げます。

- [スタンバイ・データベースを手動回復モードにする](#)
- [スタンバイ・データベースを管理回復モードにする](#)
- [スタンバイ・データベースへのアーカイブ REDO ログの転送](#)
- [スタンバイ・データベースの回復モードのメンテナンス](#)

スタンバイ・データベースを手動回復モードにする

データベースを起動およびマウントした後、データベースを手動回復モードにすることができます。スタンバイ・データベースを最新の状態に保つには、ターゲット・データベースからスタンバイ・データベースに手動でアーカイブ REDO ログを適用する必要があります。データベースを現行日付以外の日付に回復するなど、各種のメディア回復オプションの詳細は、14-25 ページの「[不完全メディア回復の実行](#)」を参照してください。

スタンバイ・データベースを手動回復モードにする方法

1. スタンバイ・サイトの初期化パラメータを構成します。16-6 ページの「[スタンバイ・データベースの回復モードのメンテナンス](#)」および 16-25 ページの「[初期化パラメータの設定](#)」を参照してください。
2. SQL*Plus を使用し、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。たとえば、次のように入力します。

```
STARTUP NOMOUNT
```
3. スタンバイ・データベースをマウントします。

```
ALTER DATABASE MOUNT STANDBY DATABASE;
```
4. スタンバイ・ホスト上の目的の場所に、アーカイブ REDO ログを転送します。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用してください。
5. スタンバイ・データベースを回復モードにします。ここではオプションとして、FROM 'location' オプションを指定できます。ユーザーがこのパラメータを省略した場合、アーカイブ REDO ログ・ファイル・グループは、初期化パラメータ LOG_ARCHIVE_DEST_*n* (この *n* は 1 から 5 までの整数) または LOG_ARCHIVE_DEST 初期化パラメータに指定された場所にあると想定されます。たとえば、次のように入力します。

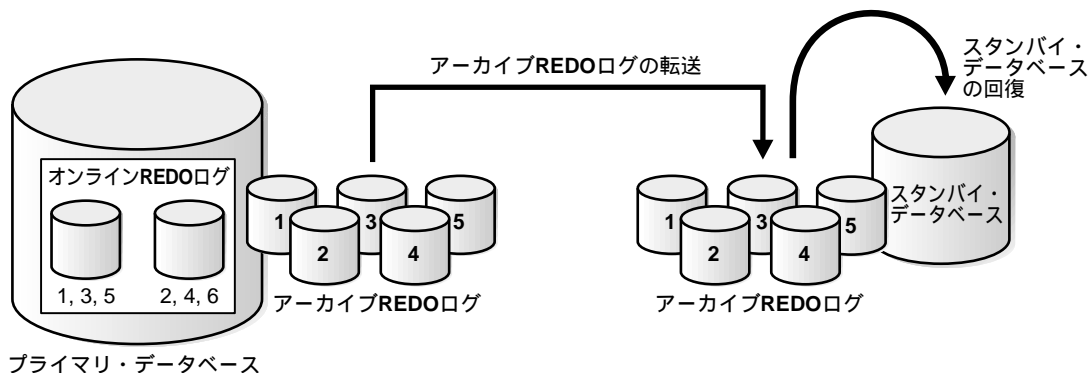

```
RECOVER STANDBY DATABASE # uses archiving location for logs specified in
init.ora
RECOVER FROM '/logs' STANDBY DATABASE # specifies non-default location
```

注意： Oracle はスタンバイ・データベースを最新の状態に保つために、アーカイブ REDO ログを生成しながら、スタンバイ・データベースにログを絶えず転送し、適用していきます。

スタンバイ・データベースを管理回復モードにする

本番データベースを**管理回復モード (managed recovery mode)** で運用するときには、ローカル・ホストまたはリモート・ホストへのアーカイブ処理を自動化できます。Oracle は、プライマリ・データベースからのアーカイブ・ログの転送を待ち、これをスタンバイ・データベースに自動的に適用することにより、スタンバイ・データベースをプライマリ・データベースと同期させます。この機能により、ユーザーがアーカイブ・ログのファイル名を用いて対話形式で回復処理を行う必要がなくなります。

図 16-1 スタンバイ・データベースへのアーカイブ REDO ログの転送と適用



データベースを管理回復モードにする場合には、RECOVER 文の TIMEOUT オプションを使用し、オプションのタイムアウト間隔を指定します。これを指定した管理回復操作では、Oracle は指定された分数の間待機してから、スタンバイ制御ファイルのディレクトリに、必要なアーカイブ・ログ・エントリを書き込みます。

スタンバイ制御ファイルの中で必要な次のログ・エントリを検出できなかったために Oracle がタイムアウトした場合には、システムは適切なメッセージを発行し、管理回復モードを終了します。デフォルト設定では、管理回復モードは必要なアーカイブ REDO ログを無期限で待ち続けます。管理回復モードは、ユーザーが介入した場合、停止またはクラッシュの場合にのみ終了します。

IMMEDIATE オプションを指定するか、指定せずに RECOVER MANAGED STANDBY DATABASE CANCEL を発行し、いつでも操作を取り消すことができます。RECOVER MANAGED STANDBY DATABASE CANCEL は管理回復操作が現行の REDO ログ・ファイルの処理を終了するのを待ち、回復操作を終了させます。

IMMEDIATE オプションを指定して CANCEL 文を使用すると、管理回復操作は、REDO ログ・ファイルからの次のブロックの読取りか、次の REDO ログ・ファイルのオープンの、いずれか先に発生した事象の前に停止されます。

次の REDO ログのオープン前に回復が終了した場合には、CANCEL IMMEDIATE は CANCEL と同じです。ログの処理中に回復が終了した場合には、CANCEL IMMEDIATE によってデータベースは一貫性のない状態になります。一貫性のない状態のデータベースのオープンは実行できませんので注意してください。

スタンバイ・データベースを管理回復モードにする方法

1. SQL*Plus を使用し、スタンバイ・データベースをマウントせずに起動します。必要であれば、パラメータ・ファイルを指定します。

```
STARTUP NOMOUNT pfile=initSTANDBY.ora
```

2. データベースをマウントします。

```
ALTER DATABASE MOUNT STANDBY DATABASE;
```

3. スタンバイ・データベースを管理回復モードにします。

```
RECOVER MANAGED STANDBY DATABASE
```

オプションのタイムアウト・オプションを使用する場合には、次の例のように、コマンド構文に TIMEOUT 整数を追加してください。

```
RECOVER MANAGED STANDBY DATABASE TIMEOUT 60
```

注意： FROM と MANAGED オプションは一緒に使用しないでください。

スタンバイ・データベースへのアーカイブ REDO ログの転送

バックグラウンド・アーカイバ・プロセスがスタンバイの位置までアーカイブするためには、次の条件を満たす必要があります。

- LOG_ARCHIVE_DEST_*n* パラメータ (この *n* は 1 から 5 までの整数) は、SERVICE 属性を使用して正しく定義する必要がある。ローカル位置とリモート位置の両方で SERVICE 属性を使用する必要があります。
- プライマリ上の tnsnames.ora と、スタンバイ上の listener.ora には、対応する正しいエントリがある。
- リモート・ファイル・サーバー (RFS) を起動し、アーカイブ処理を行うプライマリ・プロセスからネットワーク要求を受け取り始めるようにするためには、スタンバイ・インスタンスをスタンバイ・サイトで起動する必要がある。スタンバイをアーカイブ先にするためには、RFS プロセスが必要です。

各 ARC*n* プロセスは、スタンバイ宛先ごとに、対応する RFS を作成します。たとえば、3 つの ARC*n* プロセスで 2 つのスタンバイ・データベースにアーカイブするときには、6 つの RFS 接続が確立されます。

スタンバイ init.ora ファイルの中の STANDBY_ARCHIVE_DEST および LOG_ARCHIVE_FORMAT パラメータは、スタンバイ・サイトでのアーカイブ REDO ログのファイル名を決定します。これらのファイル名はスタンバイ・データベースの制御ファイルに格納されます。これらのファイル名にアクセスするには、V\$ARCHIVED_LOG 動的パフォーマンス・ビューを使用します。

この項では以下の項目を取り上げます。

- [プライマリのパラメータ・ファイルの中でアーカイブ先を指定する](#)
- [必須の宛先とオプションの宛先の指定](#)
- [アーカイブ先を使用可能にする方法](#)
- [失敗した宛先への再アーカイブ](#)
- [スタンバイ・データベースのアーカイブ・ログのファイル名の指定](#)
- [アーカイブ先指定の例](#)

関連項目: アーカイブ REDO ログの管理方法の詳細は、『Oracle8i 管理者ガイド』の中のアーカイブ REDO ログ関連の章を参照してください。Oracle のネットワーク・オプションの詳細は、『Net8 管理者ガイド』を参照してください。ARC*n* アーカイバ・プロセスの概要は、『Oracle8i 概要』を参照してください。

プライマリのパラメータ・ファイルの中でアーカイブ先を指定する

次の初期化パラメータを設定し、プライマリ・データベースのアーカイブ・ログの場所の数を指定します。

パラメータ	ホスト	例
LOG_ARCHIVE_DEST_n (この n は 1 から 5 までの整数)	リモート または ローカル	LOG_ARCHIVE_DEST_1 = 'LOCATION = /disk1/arc/' LOG_ARCHIVE_DEST_2 = 'SERVICE = standby1'
LOG_ARCHIVE_DEST および LOG_ARCHIVE_DUPLEX_DEST	ローカルの み	LOG_ARCHIVE_DEST = /oracle/arc LOG_ARCHIVE_DUPLEX_DEST = /bak

スタンバイ・データベースをメンテナンスする場合には、LOG_ARCHIVE_DEST_n パラメータを使用し、1 から 5 までの数の、異なるアーカイブの宛先を指定します。LOG_ARCHIVE_DEST_1、LOG_ARCHIVE_DEST_2 など、数値のサフィックスの付いた各パラメータは、一意に識別される別個の宛先を指定します。

LOG_ARCHIVE_DEST と、オプションである LOG_ARCHIVE_DUPLEX_DEST を一緒に使用することにより最大 2 つの位置を指定することもできます。LOG_ARCHIVE_DEST は LOG_ARCHIVE_DEST_n と一緒に使用することはできませんので注意してください。

次のキーワードを使用して位置を指定します。

キーワード	指定対象	例
LOCATION	ローカル・ファイル・システムの位置	LOG_ARCHIVE_DEST_1 = 'LOCATION=/arc/'
SERVICE	Net8 サービス名を通じたりモート・アーカイブ	LOG_ARCHIVE_DEST_2 = 'SERVICE=standby1'

LOCATION キーワードを使用する場合は、オペレーティング・システムの有効なパス名を指定してください。SERVICE が指定されると、Oracle は tnsnames.ora ファイルのネット・サービス名を接続記述子に変換します。この記述子には、リモート・データベースに接続するために必要な情報が含まれています。スタンバイ・データベースの制御ファイルのログ履歴を正しく更新できるようにするために、サービス名には関連するデータベース SID が必要になります。

関連項目 : LOG_ARCHIVE_DEST_n およびアーカイブ・プロセスの詳細は、『Oracle8i 管理者ガイド』の中のアーカイブ REDO ログに関する章を参照してください。STANDBY_ARCHIVE_DEST、SERVICE_NAME または関連するその他の初期化パラメータの情報は、『Oracle8i リファレンス・マニュアル』を参照してください。tnsnames.ora ファイルまたはネットワーク構成パラメータの情報は、『Net8 管理者ガイド』を参照してください。

必須の宛先とオプションの宛先の指定

LOG_ARCHIVE_DEST_*n* パラメータを使用すると、宛先の属性を OPTIONAL (デフォルト) または MANDATORY に指定できます。たとえば、次のようなパラメータを設定できます。

```
LOG_ARCHIVE_DEST_3 = 'SERVICE=standby1 MANDATORY'
```

LOG_ARCHIVE_MIN_SUCCEED_DEST=*n* パラメータは、すべての MANDATORY 宛先と、OPTIONAL の非スタンバイ宛先のいくつかを使用して、LGWR がオンライン REDO ログに書きできるかどうかを判別します。

関連項目 : OPTIONAL および MANDATORY キーワードの詳細は、『Oracle8i 管理者ガイド』の中のアーカイブ REDO ログに関する章を参照してください。

アーカイブ先を使用可能にする方法

LOG_ARCHIVE_DEST_STATE_*n* (この *n* は 1 から 5 までの整数) パラメータは、指定された宛先のステータスを識別します。宛先パラメータには 2 つの値を使用できます。ENABLE および DEFER。ENABLE は Oracle が宛先を使用できることを表示し、DEFER は宛先が使用できないことを表示します。

たとえば、次のようなパラメータを設定できます。

```
LOG_ARCHIVE_DEST_STATE_2 = ENABLE
```

関連項目 : アーカイブ先の状態の詳細は、『Oracle8i 管理者ガイド』の中のアーカイブ REDO ログに関する章を参照してください。LOG_ARCHIVE_DEST_STATE_*n* パラメータの詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。

失敗した宛先への再アーカイブ

エラー後に ARC*n* が失敗した宛先に対して再アーカイブを行うか、そしていつ行うかを決定するには、LOG_ARCHIVE_DEST_*n* パラメータの REOPEN 属性を使用します。REOPEN は OPEN のエラーのみでなく、すべてのエラーで使用できます。

REOPEN=*n* は、ARC*n* が失敗した宛先の再オープンを試行するまで待つ最小秒数を設定します。REOPEN を指定すると、デフォルト値の 300 秒が設定されます。REOPEN を指定しない場合、値は 0 になり、このオプションを使用しない場合と同じになります。REOPEN キーワードを指定しない場合、ARC*n* はエラー後に宛先の再オープンを行いません。

アーカイブ・ログの再接続と転送を試行する回数の制限を指定するために REOPEN を使用することはできません。REOPEN は成功するか失敗するかのいずれかであり、この場合、REOPEN 情報はリセットされます。

たとえば、再オープン時間を 60 秒に指定するには、パラメータを次のように設定します。

```
LOG_ARCHIVE_DEST_2 = 'SERVICE=standby2 OPTIONAL REOPEN=60'
```


データベースを管理回復モードで実行するには、次の状況が発生した場合、RECOVER STANDBY DATABASE を発行する必要があります。

- スタンバイ・ホストにアーカイブ REDO ログがない。つまり、RFS プロセスはスタンバイの制御ファイルにその名前を記録していない。
- 管理回復操作が失敗した。

このような状況で RECOVER STANDBY DATABASE を発行する場合には、必要なアーカイブ REDO ログを探すために LOG_ARCHIVE_DEST パラメータを使用する必要があります。スタンバイ・データベースを管理回復モードで使用する場合には、STANDBY_ARCHIVE_DEST および LOG_ARCHIVE_DEST を同じ値に設定することをお勧めします。これにより、各種の回復操作で、アーカイブ REDO ログの同じセットにアクセスできるようになります。

アーカイブ先指定の例

この例は次の状況を想定しています。

- プライマリ・データベース PROD1 はホスト LOCAL 上にある。
- 2 つのスタンバイ・データベース、STANDBY1 と STANDBY2 がある。
- STANDBY1 はローカル・ホスト LOCAL 上にあり、STANDBY2 はリモート・ホスト REMOTE2 上にある。

PROD1 の INIT.ORA の設定 プライマリ・データベース PROD1 の init.ora ファイルの中の LOG_ARCHIVE_DEST_1 および LOG_ARCHIVE_DEST_2 の設定の例を紹介します。

```
# This example specifies net service name "standby1", makes archiving mandatory, and
# enables the destination.

# A REOPEN value of 5 indicates that if the LOG_ARCHIVE_DEST_1 location
# encounters an error during archival of a redo log file, Oracle will remain inactive
# until the archival of a redo file is about to begin and 5 seconds has elapsed. At that
# time, Oracle re-attempts the archival to LOG_ARCHIVE_DEST_1.

# If Oracle encounters an error when archiving to a destination, that destination
# is inactive for the duration of the archival of the current redo log file.
# The destination may be reactivated (based on the REOPEN attribute) at the start
# of the archival of another redo log.

LOG_ARCHIVE_DEST_1 = 'SERVICE=standby1 MANDATORY REOPEN=5'
LOG_ARCHIVE_DEST_STATE_1 = ENABLE

# Specifies net service name "standby2", makes archiving optional, and specifies that
# Oracle should re-try archiving after 5 seconds should an error occur. The destination
# is enabled.

LOG_ARCHIVE_DEST_2 = 'SERVICE=standby2 OPTIONAL REOPEN=5'
LOG_ARCHIVE_DEST_STATE_2 = ENABLE
```

TNSNAMES.ORA の設定 前述の例のスタンバイ・データベース、STANDBY1 および STANDBY2 の tnsnames.ora ファイルの中の設定値です。

```
# The standby1 standby database is on the same node as the primary.
standby1 = (DESCRIPTION=
  (ADDRESS=
    (PROTOCOL=ipc)
    (KEY=stby1))
  (CONNECT_DATA=
    (SID=stby1)
    (SERVER=DEDICATED)))

# The standby2 standby database is on a different node from the primary.
standby2 = (DESCRIPTION=
  (ADDRESS=
    (PROTOCOL=tcp)
    (HOST=remote2)
    (PORT=1512)
  )
  (CONNECT_DATA=
    (SID=stby2)
    (GLOBAL_NAME=standby2)
    (SERVER=DEDICATED)))
```

LISTENER.ORA の設定 スタンバイ・データベース、STANDBY1 および STANDBY2 の listener.ora ファイルの中の設定値です。

```
# The listener settings for standby1 on host local
LISTENER = (ADDRESS_LIST=
  (ADDRESS=
    (PROTOCOL=ipc)
    (KEY=stby1)))

SID_LIST_LISTENER = (SID_LIST=
  (SID_DESC=(SID_NAME=stby1)(ORACLE_HOME=/oracle))

# The listener settings for standby2 on the remote host remote2
LISTENER = (ADDRESS_LIST=
  (ADDRESS=
    (PROTOCOL=tcp)
    (KEY=stby2)
    (HOST=remote2)
    (PORT=1512)))

SID_LIST_LISTENER = (SID_LIST=
  (SID_DESC=(SID_NAME=stby2)(ORACLE_HOME=/oracle))
```


STANDBY1 および STANDBY2 の INIT.ORA の設定 前述の例のスタンバイ・データベース、STANDBY1 および STANDBY2 の init.ora ファイルの中の設定値です。これらの設定値は、スタンバイ・データベース上のアーカイブ REDO ログのファイル名を決定します。

```
# The init.ora values for the standby1 database, which is on the same host as the primary
STANDBY_ARCHIVE_DEST = /oracle/standby/arc
LOG_ARCHIVE_DEST = /oracle/standby/arc
LOG_ARCHIVE_FORMAT = log%s.arc

# The init.ora values for the standby2 database, which is on host remote2
STANDBY_ARCHIVE_DEST = /oracle/standby/arc
LOG_ARCHIVE_DEST = /oracle/standby/arc
LOG_ARCHIVE_FORMAT = log%s.arc
```

スタンバイ・データベースの回復モードのメンテナンス

スタンバイ・データベースを実行するときには、さまざまなメンテナンスの問題が発生する可能性のあることに注意してください。可能であれば、スタンバイ・データベースを回復モードにする前に、考えられる問題の解決策を調べておいてください。

この項では次の項目を取り上げます。

- [データ・ファイルおよびアーカイブ REDO ログのファイル名の変換](#)
- [最後に適用されたアーカイブ REDO ログの判別](#)
- [オンライン REDO ログの消去](#)
- [スタンバイ・データベースのバックアップの作成](#)

データ・ファイルおよびアーカイブ REDO ログのファイル名の変換

スタンバイ・データベースが、プライマリ・データベースの制御ファイルからファイルを変換できるようにするために、次の初期化パラメータを設定します。プライマリ・データベースとスタンバイ・データベースが同じノードを占有している場合には、これらのパラメータを使用することにより、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を区別できます。同一ノードのシステムでは、OPS を使用して LOCK_NAME_SPACE パラメータを別々の値に設定しておかないと、ORA-1102 エラーが発生するので注意してください。

表 16-1 ファイル名の変換

パラメータ	機能
DB_FILE_NAME_CONVERT	プライマリ・データベースのデータ・ファイルのファイル名をスタンバイ・データ・ファイルのファイル名に変換する。たとえば、tbs_* から standbytbs_* に変換します。

表 16-1 ファイル名の変換

パラメータ	機能
LOG_FILE_NAME_CONVERT	プライマリ・データベースの REDO ログのファイル名をスタンバイ REDO ログのファイル名に変換する。たとえば、log_* から standbylog_* に変換します。
LOCK_NAME_SPACE	分散ロック・マネージャ（DLM）がロックの名前生成に使用する名前空間を指定する。同一クラスタ上に、同じデータベース名を持つスタンバイ・データベースがあるときには、OPS 構成の中でこの値を設定する必要があります。

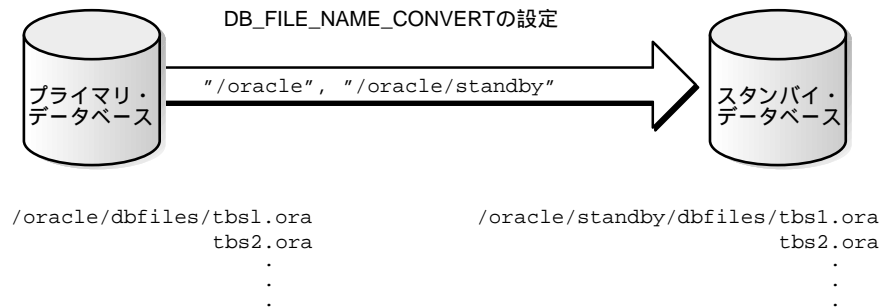
プライマリ・データベース上の新規のデータ・ファイルのファイル名をスタンバイ・データベース上のファイル名に変換するには DB_FILE_NAME_CONVERT を使用します。プライマリ・データベース上の新規の REDO ログのファイル名をスタンバイ・データベース上のファイル名に変換するには、LOG_FILE_NAME_CONVERT を使用します。プライマリ・データベースにデータ・ファイルまたはログを追加すると、対応するファイルをスタンバイ・データベースに追加する必要が生じます。

スタンバイ・データベースが更新されると、このパラメータを使用して、プライマリ・データベース上のデータ・ファイル名がスタンバイ・データベース上のデータ・ファイル名に変換されます。スタンバイ・データベース上にこのファイルが存在し、これが書き込み可能になっていないと、回復プロセスでエラーが発生し、プロセスが停止します。

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。その文字列が一致すると、これが 2 番目の文字列で置換され、スタンバイ・データベースのファイル名が作成されます。

図 16-2 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 16-2 ファイル名変換パラメータの設定



次の文を実行すると、影響を受けたファイルには変換パラメータは適用されなくなります。

- ALTER TABLESPACE RENAME DATAFILE
- ALTER DATABASE RENAME FILE
- ALTER DATABASE CREATE DATAFILE ... AS

関連項目：スタンバイ・データベースにデータ・ファイルを追加する方法の詳細、16-23 ページの「[データ・ファイルの追加](#)」を参照してください。

最後に適用されたアーカイブ REDO ログの判別

スタンバイ・データベースに最後にアーカイブ REDO ログが適用された時間を判別するには、次の方法のいずれかを使用してください。

- 適用された最新のログ順序番号を記録している V\$LOG_HISTORY ビューを問い合わせる。
- alert.log 中のトレース・データを表示する。トレース・データは、アーカイブ REDO ログが適用されるたびに更新されます。

注意： V\$LOG は回復時には更新されません。

オンライン REDO ログの消去

パフォーマンスを最適化するために、ALTER DATABASE CLEAR LOGFILE GROUP 整数コマンドを使用して、スタンバイ・データベースのオンライン REDO ログを消去できます。メンテナンス時にこの操作を実行しなかった場合、アクティブ化されるときにオンライン・ログは自動的に消去されます。

スタンバイ・データベースのバックアップの作成

必要であれば、スタンバイ・データベースをバックアップすることができますが、データベースが手動回復モードまたは管理回復モードの間は、バックアップを作成できません。スタンバイ・データベースの管理回復モードを終了し、バックアップをとった後、管理回復を再開します。データベースが停止しているときや、読取り専用モードになっているときには、バックアップを作成できます。

データベースのサイズによっては、この作業に時間がかかります。これは、スタンバイ・データベースのダウンにより、プライマリ・データベースでのログのアーカイブを待機する必要がある場合があることを意味します。これを解決するには、次の手順のいずれかを使用してください。

スタンバイ・データベースの停止中に表領域をバックアップする方法

1. スタンバイ・データベースで SQL*Plus セッションを起動し、データベースの管理回復モードを終了させます。

```
RECOVER MANAGED STANDBY DATABASE CANCEL
```

2. データベースを停止します。

```
SHUTDOWN IMMEDIATE
```

3. O/S ユーティリティを使用し、いくつかの表領域のコールド・バックアップをとります。データベースのダウン時間を最小化します。たとえば、UNIX 上で表領域 TBS_1 中のデータ・ファイル tbs11.f、tbs12.f および tbs13.f をバックアップするには、次のように入力します。

```
% cp /disk1/oracle/dbs/tbs11.f /disk2/backup/tbs11.bk
% cp /disk1/oracle/dbs/tbs12.f /disk2/backup/tbs12.bk
% cp /disk1/oracle/dbs/tbs12.f /disk2/backup/tbs13.bk
```

4. SQL*Plus を使用し、マウントせずに、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。必要であれば、パラメータ・ファイルを指定します。

```
STARTUP NOMOUNT pfile = initSTANDBY.ora
```

5. データベースをマウントします。

```
ALTER DATABASE MOUNT STANDBY DATABASE;
```

6. データベースを回復します。

```
RECOVER MANAGED STANDBY DATABASE
```

オプションのタイムアウト・オプションを使用する場合には、次の例のように、コマンド構文に TIMEOUT 整数を追加してください。

```
RECOVER MANAGED STANDBY DATABASE TIMEOUT 60
```

7. データベース内の各表領域のバックアップが終わるまでこの手順を繰り返します。

読取り専用モードのスタンバイ・データベースで表領域をバックアップする方法

スタンバイ・データベースの制御ファイルではなく、プライマリ・データベースの制御ファイルをバックアップする必要があります。

1. スタンバイ・データベースで SQL*Plus セッションを起動し、データベースの管理回復モードを終了させます。

```
RECOVER MANAGED STANDBY DATABASE CANCEL
```

2. 読取り専用モードでデータベースをオープンします。

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

3. O/S ユーティリティを使用し、いくつかの表領域のバックアップをとります。スタンバイの制御ファイルをバックアップしないでください。

データベースのダウン時間を最小化します。たとえば、UNIX 上で表領域 TBS_1 の中のデータ・ファイル tbs11.f、tbs12.f および tbs13.f をバックアップするには、次のように入力します。

```
% cp /disk1/oracle/dbs/tbs11.f /disk2/backup/tbs11.bk
% cp /disk1/oracle/dbs/tbs12.f /disk2/backup/tbs12.bk
% cp /disk1/oracle/dbs/tbs12.f /disk2/backup/tbs13.bk
```

4. スタンバイ・データベースのアクティブなユーザー・セッションをすべて終了させます。
5. 次の文を発行します。

```
RECOVER MANAGED STANDBY DATABASE # you can also set the TIMEOUT option
```

6. O/S ユーティリティを使用し、プライマリ・データベース上の制御ファイルをバックアップします。
7. データベース内の各表領域のバックアップが終わるまでこの手順を繰り返します。

スタンバイ・データベースの読取り専用モードでのオープン

読取り専用モードではユーザーはオープン状態のデータベースを問い合わせることができます。このモードにより、オンライン・データが変更される可能性を排除できます。この機能を使用すると、ユーザーはスタンバイ・データベースを一時的なレポーティング・データベースとして使用できます。一時表領域を使用すると、問合せを行うために、読取り専用モードで一時ファイル・エントリを追加できます。一時ファイルの追加および変更によって REDO エントリが生成されることはありません。

主に災害回復対策としてスタンバイ・データベースを使用する場合には、情報源としてスタンバイ・データベースにあまり依存しないでください。災害発生時には、スタンバイ・データベースを迅速にアクティブ化する必要があるため、すべてのユーザー活動を即時に停止する必要があります。また、スタンバイ・データベースで問合せを行う場合には、データベースで管理回復を行うことはできません。スタンバイ・データベースをプライマリ・データベースと同期させるために、ある時点でスタンバイに対して回復操作を実行する必要があります。この作業により、災害時回復データベースとしてのスタンバイの役割が制約を受けます。

災害対策が必要であると同時に、スタンバイで問合せも行いたい場合には、複数のスタンバイ・データベースをメンテナンスし、あるスタンバイ・データベースを読取り専用にし、別のスタンバイ・データベースを管理回復モードにすることができます。読取り専用データベースでは再同期化を行う必要がありますが、回復モードのデータベースで、災害対策を行うことができます。

関連項目：一時ファイルおよび一時表領域の使用の詳細は、『Oracle8i 管理者ガイド』を参照してください。

スタンバイ・データベースの停止中にそれを読取り専用モードでオープンする方法

1. SQL*Plus を使用し、マウントすることなく、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。

```
STARTUP NOMOUNT pfile=initSTANDBY.ora
```

2. スタンバイ・データベースをマウントします。

```
ALTER DATABASE MOUNT STANDBY DATABASE;
```

3. 読取り専用モードでデータベースをオープンします。

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

手動回復モード中に読取り専用モードでスタンバイ・データベースをオープンする方法

1. 次のように入力し、回復を取り消します（アーカイブ REDO ログのフローを終了させ、プロンプトを表示します）。

```
RECOVER CANCEL
```

2. 読取り専用モードでデータベースをオープンします。

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

管理回復モード中にスタンバイ・データベースを読取り専用モードでオープンする方法

1. SQL*Plus セッションを起動し、次のコマンドを実行します。

```
RECOVER MANAGED STANDBY DATABASE CANCEL
```

2. 読取り専用モードでデータベースをオープンします。

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

スタンバイ・データベースを読取り専用モードから管理回復モードに戻す方法

1. スタンバイ・データベースのアクティブなユーザー・セッションをすべて終了させます。
2. 次の文を発行します。

```
RECOVER MANAGED STANDBY DATABASE # you can also set the TIMEOUT option
```

スタンバイ・データベースを読取り専用モードから手動回復モードに戻す方法

1. スタンバイ・データベースのアクティブなユーザー・セッションをすべて終了させます。
2. 次の文を発行します。

```
RECOVER STANDBY DATABASE # you can also set the TIMEOUT option
```

スタンバイ・データベースのアクティブ化

緊急時以外は、スタンバイ・データベースをアクティブにしないでください。スタンバイ・データベースをアクティブにすると、このデータベースは通常の本番データベースとなるため、スタンバイ・ステータスを失います。

災害の性質によっては、プライマリ・データベースのファイルにアクセスできないことがあります。ファイルにアクセスできる場合には、可能であれば、次の作業を行ってください。

1. プライマリ・データベースのオンライン REDO ログをアーカイブします。

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

2. これらのログをスタンバイ・サイトに転送します。
3. スタンバイ・データベースをアクティブにする前に、これらのログを適用します。

この手順を実行すると、スタンバイ・データベースは、プライマリ・データベースの障害の直前の時点にまでロールフォワードされます。スタンバイ・データベースには、現行の REDO ログ以外のすべての REDO ログを適用できます。現行のものでないオンライン REDO ログを消失した場合、これらのログがアーカイブされていない場合には、プライマリ・データベースの非アーカイブ REDO 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します。

スタンバイ・データベースをアクティブにした後、オンライン REDO ログをリセットします。これにより、スタンバイ・データベースからの REDO ログとプライマリ・データベースからの REDO ログの間の互換性が失われることに注意してください。また、スタンバイ・データベースはアクティブ化のときにマウントされていません。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。

スタンバイ・データベースをアクティブにする方法

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. スタンバイ・データベースをアクティブにします。

```
ALTER DATABASE ACTIVATE STANDBY DATABASE;
```

3. スタンバイ・インスタンスを停止します。

```
SHUTDOWN IMMEDIATE
```

4. 新しい本番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本番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は、アクティブ化した後の変更内容を回復できないため、安全策として行うことをお勧めします。
5. 新しい本番用インスタンスを読み書き両用モードまたは読取り専用モードで起動します。

```
STARTUP MOUNT
```

```
ALTER DATABASE READ ONLY; # opens the database in read-only mode
```

```
ALTER DATABASE READ WRITE; # opens the database in read-write mode
```

注意： スタンバイ・データベースをアクティブにすると、元の本番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

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。

この項では次の項目を取り上げます。

- [データ・ファイルの追加](#)
- [データ・ファイルの改名](#)
- [REDO ログの変更](#)
- [制御ファイルの変更](#)

- 初期化パラメータの設定
- 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
- ダイレクト・パス操作の実行
- 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

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 データが生成されます。この REDO データがスタンバイ・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配置されれば、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・ファイルを配置できない場合には、回復は終了します。

回復プロセスが停止した場合には、次の手順を実行してください。スタンバイ・データベースに新しいデータ・ファイルを入れない場合には、次の構文を使用して、これをオフラインにすることができます。

```
ALTER DATABASE DATAFILE 'filename' OFFLINE DROP;
```

プライマリ・データベースに表領域またはデータ・ファイルを追加し、これをスタンバイに転送する方法

1. 通常の方法でプライマリ・データベース上に表領域を作成します。たとえば、表領域 TBS_2 の中に新しいデータ・ファイル t_db2.f を作成するには、次コマンドを発行します。

```
CREATE TABLESPACE tbs_2 DATAFILE 't_db2.f' SIZE 2M;
```

2. 新しく作成した空のデータ・ファイルをスタンバイ・サイトにコピーします。たとえば、両方のデータベースが同一ホスト上にある場合には、次のように入力します。

```
% cp t_db2.f /private1/stby/t_db2.f
```

3. スタンバイ・インスタンスをマウントせずに起動します。たとえば、次のように入力します。

```
STARTUP NOMOUNT pfile=/private1/stby/initSTANDBY.ora
```

4. スタンバイ・データベースをマウントし、管理回復モードにします。

```
ALTER DATABASE MOUNT STANDBY DATABASE;  
RECOVER MANAGED STANDBY DATABASE
```

5. プライマリ上の REDO ログを切り替え、スタンバイ・データベースへの REDO のアーカイブを開始します。

```
ALTER SYSTEM SWITCH LOGFILE;
```

- すべてのアーカイブ REDO ログが適用されたら、管理回復を取り消します。

```
RECOVER MANAGED STANDBY DATABASE CANCEL
```

CREATE TABLESPACE REDO を適用すると、スタンバイ制御ファイルに新しいファイル名が追加されます。次の alert.log エントリが生成されます。

```
WARNING! Recovering data file 2 from a fuzzy file. If not the current file it
might
be an online backup taken without entering the begin backup command.
Successfully added datafile 2 to media recovery
Datafile #2: '/privatel/stby/t_db2.f'
```

- スタンバイ・データベース上にデータ・ファイルを作成します。たとえば、次のコマンドを発行します。

```
ALTER DATABASE CREATE DATAFILE '/privatel/stby/t_db2.f' AS '/privatel/stby/t_
db2.f';
```

- スタンバイを管理回復モードにします。

```
RECOVER MANAGED STANDBY DATABASE
```

プライマリ・データベースの通常の処理を続行します。これで、プライマリ・データベースとスタンバイ・データベースは同期化されました。

関連項目：オフラインのデータ・ファイルの変更の詳細は、16-27 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。

データ・ファイルの改名

プライマリ・データベース上で行われたデータ・ファイルの改名は、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・ファイルの同期を保つには、スタンバイ・データベースでも同様の操作を実行してください。

REDO ログの変更

スタンバイ・データベースに影響を与えずに、REDO 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。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 REDO ログの構成を、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することができます。その場合は、プライマリ・データベースで ALTER DATABASE ENABLE THREAD 文を使用してログ・ファイル・スレッドを使用可能にするとき、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。16-29 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](#)」で手順を参照してください。

ALTER DATABASE CLEAR UNARCHIVED LOGFILE 文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 オプションを使用してプライマリ・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 コマンドを使用して次のいずれかの操作を実行した場合、スタンバイ・データベースの制御ファイルが無効になる場合があります。

- 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には、16-29 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](#)」で説明している手順を使用して再作成してください。

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ます。しかし、CONTROL_FILES および DB_FILE_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。

表 16-2 スタンバイの初期化パラメータの設定

パラメータ	ガイドライン
COMPATIBLE	このパラメータは、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する必要があります。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。読取り専用モードでスタンバイ・データベースをオープンしたい場合には、COMPATIBLE パラメータは 8.1 以上に設定する必要があります。
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にする必要があります。
CONTROL_FILES	このパラメータは、プライマリ・データベースとスタンバイ・データベースで異なる設定にする必要があります。スタンバイ・データベース用の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必要があります。
DB_FILE_NAME_CONVERT	スタンバイのデータ・ファイルのファイル名を、プライマリ・データベースのファイル名と区別できるようにする場合に設定します。詳細は、16-15 ページの「 データ・ファイルおよびアーカイブ REDO ログのファイル名の変換 」を参照してください。
LOG_FILE_NAME_CONVERT	スタンバイのログのファイル名を、プライマリ・データベースのログのファイル名と区別できるようにする場合に設定します。このパラメータの詳細は、16-15 ページの「 データ・ファイルおよびアーカイブ REDO ログのファイル名の変換 」を参照してください。
STANDBY_ARCHIVE_DEST	<p>このパラメータは、スタンバイ RFS プロセスがアーカイブ・ログを入れるディレクトリを判別する目的でのみ使用されます。Oracle は、この値と LOG_ARCHIVE_FORMAT を使用し、スタンバイ・サイトのログ・ファイル名を生成します。完全修飾のファイル名がスタンバイの制御ファイルに格納されます（このデータが必要な場合は V\$ARCHIVED_LOG を問い合わせます）。管理回復では、この情報を使用して回復操作が実行されます。</p> <p>RECOVER STANDBY DATABASE コマンド（MANAGED オプションを除く）は、アーカイブ・ファイルの位置を、LOG_ARCHIVE_DEST が、ユーザーが入力したファイル名のいずれかより判別します。スタンバイ・サイトでログが紛失している場合、つまり RFS がスタンバイ制御ファイルにその名前を記録してなく、管理回復操作が失敗した場合には、RECOVER STANDBY DATABASE を発行する必要があります。この文を使用する場合は、アーカイブ・ログを見つけるために LOG_ARCHIVE_DEST パラメータを使用する必要があります。</p> <p>管理状態のスタンバイ・データベースでは、STANDBY_ARCHIVE_DEST パラメータと LOG_ARCHIVE_DEST パラメータを同じ値に設定してください。手動の回復が必要な場合は、紛失しているアーカイブ・ログを、他のすべてのアーカイブ・ログと同じ位置にコピーし、手動のスタンバイ回復操作を実行し、スタンバイを管理回復モードに戻します。</p>

関連項目：初期化パラメータの詳細は、『Oracle8i 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。

スタンバイ・データベースで次の文を使用し、データ・ファイルをオフラインにします。

```
ALTER DATABASE DATAFILE 'filename' OFFLINE DROP;
```

この文を実行する場合、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

次のいずれかの方法で始まるダイレクト・ロードを実行した場合、パフォーマンスの向上は、プライマリ・データベースにのみ適用されます（これにより、スタンバイ・データベースで回復プロセスのパフォーマンスが向上することはありません）。

- ダイレクト・パス・ロード
- 副問合せによる CREATE TABLE
- プライマリ・データベースでの CREATE INDEX

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 オプションを使用したプライマリ・データベースのプロセスは、これらのプロセスがアーカイブ REDO ログ中に示されないため、スタンバイ・データベースには反映されません。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、次の作業のうちの、1 つを実行してください。

- 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。（16-27 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）。
- 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。（16-3 ページの「[スタンバイ・データベースの作成](#)」を参照してください）。
- 影響を受けた表領域をバックアップし、プライマリ・データベースの中の現行のログをアーカイブし、データ・ファイルをスタンバイ・データベースに転送してから、スタンバイの回復を再開します。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 操作を実行した後、スタンバイ・データベースで回復を試みた場合には、回復中にエラー・メッセージが表示されなくなります。この場合、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記録されます。スタンバイ・データベースの警告ログを定期的に調べてください。

UNRECOVERABLE オプションを使用してロードされたブロックをスタンバイ・サイトで読み取ろうとすると、次のエラー・メッセージが表示されます。

```
26040, 00000, "Data block was loaded using the NOLOGGING option\n"
/** Cause: Trying to access data in block that was loaded without
/**      redo generation using the NOLOGGING/UNRECOVERABLE option
/** Action: Drop the object containing the block.
```

関連項目 : 詳細は、16-27 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。

注意 : UNRECOVERABLE オプションを使用してロードされたブロックには、スタンバイ・サイトの回復時に、論理的に破損しているというマークが付けられます。これらのデータ・ブロックを問い合わせた場合にはエラー・メッセージが発行されます。

UNRECOVERABLE 操作後のバックアップの必要性の判別

プライマリ・データベースで UNRECOVERABLE 操作を実行した場合には、V\$DATAFILE ビューを使用し、最新の無効化 REDO データが生成された SCN または時刻を判別してください。

別のバックアップを実行する必要があるかどうかを判断するために、次の SQL コマンドを発行します。

```
SELECT unrecoverable_change#, to_char(unrecoverable_time, 'mm-dd-yyyy hh:mi:ss')
FROM v$datafile;
```

データ・ファイルが最後にバックアップされて以降にデータ・ファイルに回復不能時間があることが問合せにより報告されたら、このデータ・ファイルのバックアップをもう 1 つ作成してください。

関連項目 : V\$DATAFILE ビューの詳細は、『Oracle8i 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。ファイルの追加や削除など、プライマリ・データベースの大幅な構造変更を行った後は、制御ファイルをリフレッシュしてください。

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

1. スタンバイ・インスタンス上で SQL*Plus セッションを起動し、スタンバイ・データベース上で CANCEL コマンドを発行し、回復プロセスを停止します。

```
RECOVER CANCEL # for manual recovery mode
RECOVER MANAGED STANDBY DATABASE CANCEL # for managed recovery mode
```

2. スタンバイ・インスタンスを停止します。

```
SHUTDOWN IMMEDIATE
```

3. 本番用インスタンス上で SQL*Plus セッションを起動し、スタンバイ・データベースの制御ファイルを作成します。

```
ALTER DATABASE CREATE STANDBY CONTROLFILE AS 'filename';
```

4. プライマリ・データベースの現行のオンライン REDO ログをアーカイブします。

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

5. バイナリ・ファイルに適した O/S ユーティリティを使用し、スタンバイ制御ファイルとアーカイブ・ログ・ファイルをスタンバイ・サイトに転送します。

6. スタンバイ・インスタンスに接続し、スタンバイ・データベースをマウントします（オープンはしません）。

```
ALTER DATABASE MOUNT STANDBY DATABASE;
```

7. スタンバイ・データベースで回復プロセスを再起動します。

```
RECOVER STANDBY DATABASE # recovers using location for logs in init.ora
RECOVER FROM 'location' STANDBY DATABASE # recovers using specified location
```

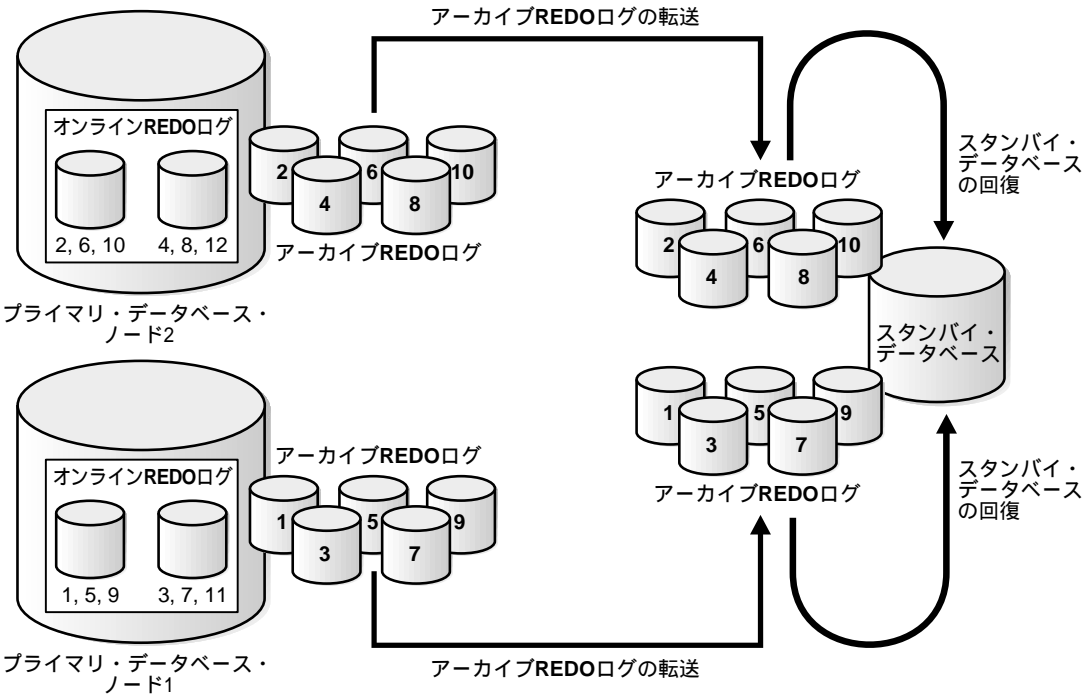
OPS 構成でのスタンバイ・データベースの使用法

スタンバイ・データベースは、Oracle マルチスレッド・サーバー・オプションを指定して使用できます。次の表は、プライマリ・データベースとスタンバイ・データベースでの有効および無効なノードの組合せを示しています。

	シングル・インスタンスのスタンバイ	複数インスタンスのスタンバイ
シングル・インスタンスのプライマリ	有効	無効
複数インスタンスのプライマリ	有効	有効

それぞれの状況では、プライマリ・データベースの各ノードは、自分のアーカイブ REDO ログのスレッドをスタンバイ・データベースに転送します。たとえば、[図 16-3](#) は、シングル・インスタンスのスタンバイに REDO ログを転送する、2 つのノードを持つ OPS データベースを示しています。

図 16-3 複数インスタンスのプライマリ・データベースからのアーカイブ・ログの転送



この場合、プライマリ・データベースのノード 1 は、ログ 1、3、5、7、9 を転送するのに対して、ノード 2 は、ログ 2、4、6、8、10 を転送します。スタンバイ・データベースが管理回復モードになっている場合には、アーカイブ REDO ログを適用する正しい順番が自動的に判断されます。

プライマリ・データベースとスタンバイ・データベースの両方が OPS 構成であり、スタンバイ・データベースが管理回復モードになっている場合には、スタンバイ・データベースの 1 つのノードが、プライマリ・ノードから転送されたすべてのログ・セットを適用します。この場合、管理回復の進行中は、REDO を適用していないスタンバイ・ノードを読取り専用モードにすることはできません。非回復ノードはマウントすることもできますが、通常の場合は、これを停止してください。

関連項目：データベースの OPS 構成の詳細は、『Oracle8i Parallel Server セットアップおよび構成ガイド』を参照してください。

Recovery Manager による表領域の Point-in-Time 回復の実行

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。

- [RMAN の TSPITR の概要](#)
- [TSPITR の計画](#)
- [TSPITR のための補助インスタンスの準備](#)
- [TSPITR の実行](#)
- [TSPITR 後に使用するためのターゲット・データベースの準備](#)
- [失敗した TSPITR への対応](#)
- [TSPITR のパフォーマンスのチューニング](#)


RMAN の TSPITR の概要

Recovery Manager（RMAN）の自動化された表領域の Point-in-Time 回復（TSPITR）機能を使用すると、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迅速に回復させることができます。

RMAN の TSPITR は次のような回復の場合最も有用です。

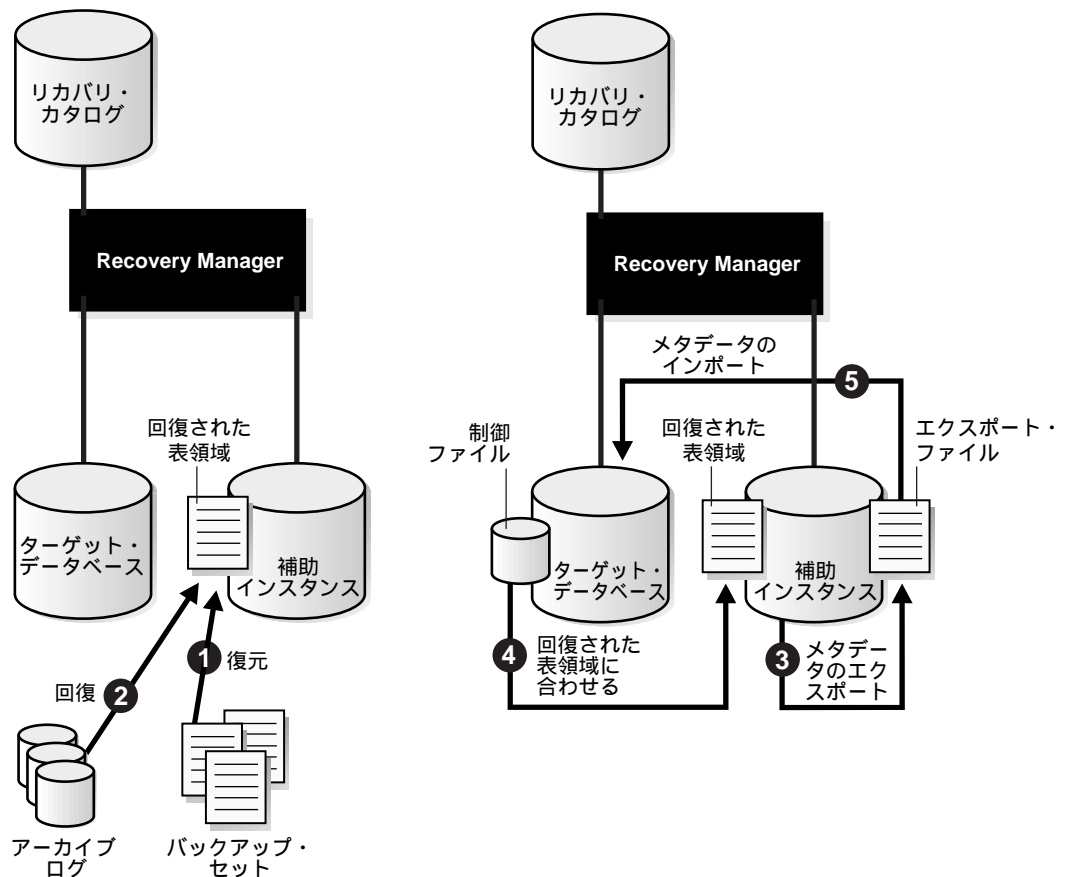
- DROP TABLE または TRUNCATE TABLE 操作のエラーから回復する場合。
- 論理的に壊れた表を回復させる場合。
- データベースのサブセットのみに影響を与えた不正確なバッチ・ジョブ、またはその他の DML 文から回復する場合。
- 1 つの物理データベースの別々の表領域の中に複数の論理データベースが存在する場合に、物理データベースの残りの部分とは異なる時点になっている論理データベースを回復させる場合。

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。

 [A-1](#) で示すように、Recovery Manager は次の作業を行います。

1. 指定された表領域のバックアップを、一時補助インスタンス内に復元する。
2. 表領域を回復する。
3. 補助インスタンスからメタデータをエクスポートする。
4. ターゲット・データベースの制御ファイルが、新しく回復されたデータ・ファイルを指すようにする。
5. ターゲット・データベースにメタデータをインポートする。

図 A-1 RMAN の TSPITR



TSPITR の用語

次の用語と略称はこの章で頻繁に使用されるので理解しておいてください。

TSPITR

表領域の Point-in-Time 回復

補助インスタンス

バックアップ表領域の回復に使用される補助インスタンス TSPITR によって作成されたデータベースは、独立した存在ではありません。これは、中間的な作業領域にすぎません。

回復セット

TSPITR を実行する必要がある表領域。

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
- SYSTEM 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）エクスポートのソート操作のために小さい領域が必要になります（ソートの領域の問題の詳細は、A-12 ページの「[失敗した TSPITR への対応](#)」を参照してください）。

TSPITR の計画

Recovery Manager の TSPITR では慎重な計画が必要です。次に進む前に、この章全体を読んでください。

この項では次の項目を取り上げます。

- [リカバリ・カタログを使用しない TSPITR の実行](#)
- [一般的な制約事項の理解](#)
- [非一貫性の調査と解決](#)
- [データの関連の管理](#)

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、[付録 B の「オペレーティング・システム環境での表領域の Point-in-Time 回復の実行」](#)にも記載されていますが、異なる点もあります。それらの違いについては、この章で説明します。

リカバリ・カタログを使用しない TSPITR の実行

RMAN の TSPITR は、リカバリ・カタログを使用しても、使用しなくても実行できます。リカバリ・カタログを使用しない場合には、次の制約事項に注意してください。

- RMAN は TSPITR の中にロールバック・セグメントの履歴データを持っていないため、現行のロールバック・セグメントが、回復の実行時に存在していたものと同じセグメントであると想定します。
- RMAN がかなり以前の時刻まで回復した場合には、コピーおよびバックアップの記録が Oracle によって再使用されているため、TSPITR を実行できないこともあります。

一般的な制約事項の理解

RMAN の TSPITR を実行するときには、次の作業は実行できません。

- ターゲット・データベースと補助データベースを別々のノード上で実行する。しかし、ターゲット・データベースと補助データベースは、共有ディスクを使用して、1 つの同じクラスタ構成に入れることができます。
- 削除された表領域を回復する。
- 削除された後、同じ名前を使用して再作成された表領域を回復する。
- 表領域に追加されたデータ・ファイルを削除する。そのファイルが、RMAN の TSPITR の回復対象となる時点より後に追加された場合、RMAN の TSPITR の完了後も、そのファイルは空のまま表領域の一部として存在します。
- 補助データベースで DML 文を実行する。補助データベースは回復専用です。
- TSPITR の完了後に、回復セットのデータ・ファイルの既存のバックアップを使用して回復を行う。このかわりに、回復されたファイルの新しいバックアップをとってください。TSPITR の実行前にとられたバックアップを使用して回復を試みると、回復は失敗します。
- 統計情報が算出されているオブジェクトのオブティマイザ統計情報を回復する。TSPITR の実行後に統計情報を再度算出してください。
- 次のオブジェクトのいずれかを回復セットに入れる。
 - レプリケートされたマスター表
 - VARRAY 列のある表
 - ネストした表が入っている表
 - 外部ファイルを伴う表
 - 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）

警告： RMAN の TSPITR を初めて実行する場合は、本番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場合も実行しないでください。

非一貫性の調査と解決

RMAN の TSPITR に関する最大の問題は、参照依存性が明示的ではなく暗黙的であるため、回復済みの表領域と未回復の表領域の中の表の間で、アプリケーション・レベルの一貫性が失われる可能性があることです。次の問題に注意し、この先に進む前に、非一貫性が生じた場合の解決方法を準備しておいてください。

- RMAN はすべての表が入っている回復セットのみをサポートする
- ロールバック・セグメントが入っている表領域をサポートしない
- TS_PITR_CHECK は SYS が所有するオブジェクトをチェックしない

RMAN はすべての表が入っている回復セットのみをサポートする

注意： これは、RMAN の TSPITR に固有の制限事項です。

RMAN の TSPITR は、すべての表が入っている回復セットのみをサポートします。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領域に分かれていると、RMAN はエクスポート・フェーズ時にエラー・メッセージを戻します。回復セットに制約に必要な索引を伴わない表が入っている場合、または制約に必要な表がなく制約を含む表のみが入っている場合もエラーになります。

ロールバック・セグメントが入っている表領域をサポートしない

注意： これは、RMAN の TSPITR に固有の制限事項です。

O/S の 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。これにより、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。RMAN の TSPITR 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、B-9 ページの「[ステップ 3: 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。しかし、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。

ロールバック・セグメントも含めて、SYS によって所有されるオブジェクトが回復セットに含まれている場合には、これらのオブジェクトを正常に回復できる保証はありません。TSPITR は Export および Import ユーティリティを使用しますが、これらのユーティリティ

は、SYS が所有するオブジェクトをサポートしていません。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 ('tablespace_name_1', 'tablespace_name_2',
                          'tablespace_name_n')
AND owner = 'SYS';
```

関連項目 : TS_PITR_CHECK ビューの詳細は、B-7 ページの「[ステップ 2: プライマリ・データベースに対する依存性を調査および解決する](#)」を参照してください。

データの関連の管理

TSPITR は、回復セット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、TSPITR は正常に実行されません。

関連項目 : 詳細は、B-7 ページの「[ステップ 2: プライマリ・データベースに対する依存性を調査および解決する](#)」を参照してください。

TSPITR のための補助インスタンスの準備

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。

- [補助インスタンスの Oracle パスワード・ファイルの作成](#)
- [補助インスタンスのパラメータ・ファイルの作成](#)
- [補助インスタンスの起動](#)
- [補助インスタンスへの Net8 接続性の確保](#)
- [Recovery Manager のコマンド行インタフェースの起動](#)

補助インスタンスの Oracle パスワード・ファイルの作成

Oracle パスワード・ファイルの作成とメンテナンスの詳細は、『Oracle8i 管理者ガイド』を参照してください。

補助インスタンスのパラメータ・ファイルの作成

補助インスタンス用に `init.ora` ファイルを作成し、次の必須パラメータを設定してください。

パラメータ	指定対象
DB_NAME	ターゲット・データベースと同じ名前
LOCK_NAME_SPACE	同じ <code>\$ORACLE_HOME</code> 内のどのデータベースとも異なる値。単純にするために、 <code>_DBNAME</code> と指定してください。
DB_FILE_NAME_CONVERT	補助データベースのデータ・ファイル用にファイル名を変換するためのパターン。このパラメータを使用し、 <code>set auxname.</code> を使用して命名しなかったファイルのファイル名を生成できます。
LOG_FILE_NAME_CONVERT	補助データベースのオンライン REDO ログ用にファイル名を変換するためのパターン。
CONTROL_FILES	ターゲットのパラメータ・ファイルの中の <code>CONTROL_FILES</code> パラメータとは異なる値。

Net8 を通じて SYSDBA として接続できるようにするためのパラメータも含めて、必要に応じてその他のパラメータを設定します。

補助インスタンスの `init.ora` パラメータの設定の例を紹介します。

```
DB_NAME=prod1
LOCK_NAME_SPACE=_prod1
CONTROL_FILES=/oracle/aux/cf/aux_prod_cf.f
DB_FILE_NAME_CONVERT=("/oracle/prod/datafile","/oracle/aux/datafile")
LOG_FILE_NAME_CONVERT=("/oracle/prod/redo_log","/oracle/aux/redo_log")
```

関連項目：DB_FILE_NAME_CONVERT の詳細は、A-13 ページの「[TSPITR のパフォーマンスのチューニング](#)」を参照してください。Net8 の詳細は、『Net8 管理者ガイド』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。

補助インスタンスの起動

RMAN の TSPITR を開始する前に、SQL*Plus を使用して補助インスタンスに接続し、NOMOUNT モードで起動してください（必要であればパラメータ・ファイルを指定します）。

```
SQL> connect sys/aux_pwd@aux_str;
SQL> startup nomount pfile='/oracle/aux/dbs/initAUX.ora';
```

補助インスタンスにはまだ制御ファイルがないため、このインスタンスは NOMOUNT モードでしか起動できません。制御ファイルの作成や、TSPITR の補助インスタンスのマウントやオープンを試みないでください。

補助インスタンスへの Net8 接続性の確保

補助インスタンスには有効なネット・サービス名が必要です。先に進む前に、SQL*Plus を使用し、補助インスタンスへの接続を確立できることを確認します。

Recovery Manager のコマンド行インタフェースの起動

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。

- [O/S コマンド行から接続する](#)
- [RMAN プ롬プトからの接続](#)

O/S コマンド行から接続する 補助インスタンスおよびターゲット・インスタンス、そしてオプションとしてリカバリ・カタログに接続するには、Recovery Manager の起動時に次の情報を渡します。

```
% rman target sys/target_pwd@target_str catalog rman/cat_pwd@cat_str auxiliary \
> sys/aux_pwd@aux_str
```

このとき、

<i>sys</i>	SYSDBA 権限を持つユーザー
<i>rman</i>	リカバリ・カタログの所有者
<i>target_pwd</i>	ターゲット・データベースの <code>orapwd</code> ファイルの中に指定された、SYSDBA として接続するためのパスワード
<i>target_str</i>	ターゲット・データベースのネット・サービス名
<i>cat_pwd</i>	リカバリ・カタログの <code>orapwd</code> ファイルの中に指定された、ユーザーの RMAN のパスワード
<i>cat_str</i>	リカバリ・カタログ・データベースのネット・サービス名
<i>aux_pwd</i>	補助データベースの <code>orapwd</code> ファイルの中に指定された、SYSDBA として接続するためのパスワード

aux_str 補助データベースのネット・サービス名

RMAN プロンプトからの接続 補助インスタンスに接続せずに RMAN コマンド行インタフェースを起動してから、RMAN プロンプトで **connect** コマンドを使用できます。

```
% rman
RMAN> connect auxiliary sys/aux_pwd@aux_str
RMAN> connect target target sys/target_pwd@target_str
RMAN> connect catalog rman/cat_pwd@cat_str
```

TSPITR の実行

計画に必要な条件をすべて満たした後、RMAN の TSPITR を実行します。**run** で次のコマンドを実行します。この *tablespace_list* は回復セットの中の表領域名のリストであり、*recovery_end_time* は回復の目標となる時点です。

```
allocate auxiliary channel . . .
recover tablespace tablespace_list until recovery_end_time;
```

allocate auxiliary channel コマンドによって、少なくとも 1 つの補助チャンネルを割り当てる必要があります。

注意： 表領域の回復セットに、SYSTEM 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。

次の例文は、1999 年 1 月 10 日の午後 8 時を指定して、表領域 TBS_2 および TBS_3 に対して RMAN の TSPITR を実行します。

```
run {
    allocate auxiliary channel dev1 type 'sbt_tape';
    recover tablespace tbs_2, tbs_3 until time 'Jan 10 1999 20:00:00';
}
```

Recovery Manager は、TSPITR 中に自動的に次の作業を実行します。

1. 補助インスタンスにデータ・ファイルを復元する。
2. 復元されたデータ・ファイルを指定された時間まで回復する。
3. RESETLOGS オプションを使用して補助データベースをオープンする。
4. 回復された表領域の中のオブジェクトに関するディクショナリ・メタデータ（オブジェクトを作成するための DDL および回復済みデータ・ファイルの中のオブジェクトの物理位置へのポインタ）をターゲット・データベースにエクスポートする。
5. 補助データベースをクローズする。

6. ターゲットの制御ファイルが、補助データベースで回復されたばかりの、回復セットの中のデータ・ファイルを示すようにするために、**switch** コマンドを発行する。
7. 補助データベースからエクスポートされたディクショナリ・メタデータをインポートし、回復されたオブジェクトをアクセス可能にする。

注意： RMAN は回復中のデータ・ファイルを復元するかわりに、データ・ファイルのコピーを探します。ファイルのコピーが見つからない場合、RMAN は復元操作を実行し、switch は実行しません。set auxname コマンドを使用してデータ・ファイルの名前を構成しており、これらの auxname 位置に適切なデータ・ファイルのコピーが存在している場合には、RMAN は最適化のために復元を取り止め、auxname データ・ファイルのコピーへの切替えを実行します。

TSPITR 後に使用するためのターゲット・データベースの準備

回復セット内の表領域は、RMAN の TSPITR が正常に完了するまでオフラインのままです。

TSPITR 後の再使用のためのターゲット・データベースの準備

1. 回復セット内の表領域をオンラインにする前に、これらの表領域のバックアップを作成しておいてください。回復セット内の以前のバックアップはすべて無効になっているので注意してください。たとえば、次のコマンドは表領域 TBS_4 をバックアップします。

```
run {
  allocate channel chl type disk;
  backup tablespace tbs_4;
}
```

2. 回復済みの表領域をオンラインにします。たとえば、次のように入力します。

```
sql "ALTER TABLESPACE TBS_4 ONLINE";
```

3. RMAN の TSPITR が正常に実行された後、補助データベースは使用できなくなるため、データベースを停止し、メモリーを解放してください。

```
shutdown abort;
```

4. 次のものを削除します。
 - RMAN の TSPITR の実行時に一時的な格納場所に復元された補助セット・データ・ファイル
 - 補助データベースの制御ファイル
 - 補助データベースの REDO ログ・ファイル

失敗した TSPITR への対応

TSPITR はさまざまな問題によってエラーとなる可能性があります。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には、補助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、**startup nomount** を発行して、RMAN の TSPITR を再実行する必要があります。

また、Export コーティリティのためのソート領域が足りないために失敗することもあります。この場合には、`recover.txt` ファイル（UNIX の場合は `$ORACLE_HOME/admin` に入っています）を編集する必要があります。このファイルの内容は次のとおりです。

```
#
# tsiptr_7: do the incomplete recovery and resetlogs. This member is used once.
#
define tsiptr_7
<<<
# make the controlfile point at the restored datafiles, then recover them
recover clone database tablespace &l&;
sql clone "alter database open resetlogs";
# PLUG HERE the creation of a temporary tablespace if export fails due to lack of
# temporary space.
# For example in Unix these two lines would do that:
#sql clone "create tablespace aux_tsiptr_tmp
#          datafile '"/tmp/aux_tsiptr_tmp.dbf"' size 500K";
#
}>>>
```

コメントの最後の 2 行から '#' 記号を削除し、文を変更して一時表領域を作成します。エクスポート操作が成功するまで、表領域のサイズを拡大しながら、TSPITR 操作を再試行します。

TSPITR が失敗した場合には、次の手順に従ってください。

失敗した TSPITR への対応

1. RMAN の TSPITR が失敗した場合は、補助インスタンスを停止します。

```
shutdown abort;
```

2. エラーを識別し、訂正します。

3. 補助インスタンスをマウントせずに起動します。たとえば、次のように入力します。

```
startup nomount pfile=initAUX.ora;
```

4. A-10 ページの「[TSPITR の実行](#)」で説明しているように、TSPITR を再実行します。

TSPITR のパフォーマンスのチューニング

この項では、RMAN の TSPITR のパフォーマンスをチューニングするための手順を説明します。

- 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 による TSPITR 用の補助名の設定とデータ・ファイル・コピーの使用
- 補助制御ファイル内での変換されたファイル名の使用
- まとめ: データ・ファイルの命名方法

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、補助インスタンスにある回復セットおよび補助セット内の表領域に所属するすべてのデータ・ファイルを復元および回復します。補助セットには、SYSTEM 表領域と、ロールバック・セグメントを持つすべての表領域が含まれることに注意してください。

Recovery Manager コマンドの **set newname** を使用し、補助セットの表領域内のデータ・ファイルに新しい名前を指定します。RMAN 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。この新しい名前は、`init.ora` ファイルの中の `DB_FILE_NAME_CONVERT` パラメータの設定値も上書きします。たとえば、`datafile 2` を `new_df_name.f` に改名するには、次のように入力します。

```
set newname for datafile 2 to '/oracle/dbs/new_df_name.f';
```

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。データ・ファイルの新しい名前を指定すると、ターゲット制御ファイルの中の元のデータ・ファイルがこのデータ・ファイルで置換されるため、既存のファイル名が新しいファイル名で置換されます。

新しいファイル名を設定する場合、RMAN は補助データベースとターゲット・データベースにあるデータ・ファイルの名前が競合しないかどうかを確認しません。競合があった場合、TSPITR 時に RMAN エラーが発生します。

Recovery Manager による TSPITR 用の補助名の設定とデータ・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データ・ファイルの適切なコピーを使用してください。

次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・ファイルの補助名としてリカバリ・カタログに登録されている（*filename* はデータ・ファイルの名前または番号、*auxiliary_datafile_name* はデータ・ファイルの補助名）。

```
set auxname for datafile filename to auxiliary_datafile_name;
```

2. データ・ファイル・コピーは、次の RMAN コマンドを使用して、until 句で指定された時刻よりも前に作成されている（*'filename'* はデータ・ファイルのファイル名）。

```
run {  
    copy datafile 'filename' to auxname;  
    ...  
}
```

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。

```
set auxname for datafile '/oracle/prod/datafile_1_1.dbf'  
to '/oracle/prod_copy/datafile_1_1.dbf';  
  
run {  
    allocate channel chl type disk;  
    copy datafile '/oracle/prod/datafile_1_1.dbf'  
    to auxname;  
}
```

同じデータ・ファイルに対して **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 が正常に完了した場合、リカバリ・カタログ内の *auxiliary_datafile_name* には、**deleted** のマークが設定されます。RMAN の TSPITR の完了後、ターゲットの元のデータ・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。

補助制御ファイル内での変換されたファイル名の使用

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補助名も設定されていない場合、Recovery Manager は、補助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用して、復元と回復を実行できます。Recovery Manager は、補助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競合がないか確認します。競合があると、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補助名も設定されていない場合、あるいは補助名のファイルが使用不可能な場合、Recovery Manager データ・ファイルの元の位置を使用します。

まとめ：データ・ファイルの命名方法

次のコマンドおよびパラメータは、TSPITR 時に補助セットおよび回復セット内のデータ・ファイルに名前を付けるために使用されるものです。表内では、優先順位を上から下の順番で示しています。つまり、**set newname** は **set auxname** および **DB_FILE_NAME_CONVERT** よりも優先されます。

	コマンド / パラメータ	補助セット	回復セット
1	set newname	X	X
2	set auxname	X	X
3	DB_FILE_NAME_CONVERT	X	

補助セット内でファイル名が変換されていない場合、RMAN は TSPITR 時にエラーを発行します。

オペレーティング・システム環境での表領域 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オラクル社カスタマ・サポートに連絡することをお勧めします。

この章では、O/S 環境での表領域の Point-in-Time 回復（TSPITR）の実行方法について説明します。項目は次のとおりです。

- [O/S 環境での表領域の Point-in-Time 回復の概要](#)
- [表領域の Point-in-Time 回復の計画](#)
- [データベースでの TSPITR の準備](#)
- [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](#)
- [トランスポータブル表領域を使用した TSPITR の実行](#)

O/S 環境での表領域の Point-in-Time 回復の概要

表領域の Point-in-Time 回復（TSPITR）機能を使用すると、1 つ以上の非 SYSTEM 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迅速に回復させることができます。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。

TSPITR は次のような状況で回復を行うのに最も有用です。

- DROP TABLE または TRUNCATE TABLE 操作のエラーから回復する場合
- 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のみ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の中に複数の論理データベースが存在する場合に、物理データベースの残りの部分とは異なる時点になっている論理データベースを回復させる場合。
- VLDB（非常に大規模なデータベース）の中の表領域を回復させる場合、これはバックアップからデータベース全体を復元してロールフォワードするよりも TSPITR が効率的です（決定する前に B-4 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）。

この項では次の項目を取り上げます。

- [TSPITR の利点](#)
- [TSPITR の方法](#)
- [TSPITR の用語](#)

TSPITR の利点

Oracle8 より前のバージョンでは、Point-in-Time 回復は、データベースのサブセットに対して次の手順でを使用することしかできませんでした。

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. 本番データベースから関連するオブジェクトを削除する。
5. オブジェクトを本番データベースにインポートする。

大規模なオブジェクトのエクスポートおよびインポートではパフォーマンスのオーバーヘッドが発生していたので、新しい方法を開発する必要性がありました。TSPITR では次の作業を行えます。

1. クローン・データベースと呼ばれる、データベースの一時コピーを作成する。
2. クローン・データベースのサブセットを回復する。
3. O/S ユーティリティを使用し、関連するデータ・ファイルを、回復したデータベースから本番データベースにコピーする。
4. データ・ファイルの内容に関するデータ・ディクショナリ・メタデータ（たとえば、ファイル内の回復されたセグメント）を、クローン・データベースから本番データベースにエクスポートする。ファイルのコピーも、特殊なインポート処理によって本番データベースに追加されます。

TSPITR の方法

O/S 環境での TSPITR は 2 つの異なる方法で実行できます。

方法	結論
従来の O/S TSPITR	クロンの <code>init.ora</code> ファイルの作成、クローン・データベースのマウントなどを行うために、特殊な手順を使用する必要があります。この手順は、クローン・データベースの回復中に同一のコンピュータ上にあるプライマリ・データベースが破損することを防止するために、エラー・チェックを提供します。
トランスポートابل表領域機能を使用した TSPITR	この方法と標準の O/S の TSPITR との主な相違点は、この方法では、TSPITR の最後のステップを実行するためにトランスポートابل表領域が使用される点にあります。この方法を使用するには、COMPATIBLE 初期化パラメータを 8.1 以上に設定する必要があります。

これら 2 つの方法の主な相違点は、トランスポートابل表領域を使用して TSPITR を実行することにより、O/S の TSPITR の特殊な手順のいくつかが緩和される点にあります。バックアップをプライマリ・データベースとは別のホスト上に復元する場合には、クローン・データベースの MOUNT コマンドではなく、通常データベースの MOUNT コマンドを使用して、クローン・データベースをプライマリ・データベースであるかのように起動することができます。

関連項目：トランスポートابل表領域機能の詳細は、『Oracle8i 管理者ガイド』を参照してください。

TSPITR の用語

次の用語と略称はこの章で頻繁に使用されるので、理解しておいてください。

TSPITR

表領域の Point-in-Time 回復

クローン・データベース

TSPITR での回復に使用されるデータベースのコピーです。このデータベースは、通常のデータベースとは実質的な種々の違いがあります。

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）

エクスポート時のソート操作のため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、SYSTEM 表領域ファイルに対して AUTOEXTEND を ON に設定することにより、ソート領域を作成してください。

トランSPORTABLE表領域

あるデータベースから表領域を取り出して別のデータベースに組み込むための機能です。詳細は、15-4 ページの「[移動された表領域の回復](#)」を参照してください。詳細は、『Oracle8i 管理者ガイド』を参照してください。

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。次に進む前に、この章全体を読んでください。

警告： TSPITR を初めて実行する場合は、本番用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場合も実行しないでください。

TSPITR の制限事項

考慮すべき最大の問題は、参照依存性が明示的ではなく暗黙的であるため、回復済みの表領域と未回復の表領域の中の表の間で、アプリケーション・レベルの非一貫性が生じる可能性があることです。これらの依存性を理解し、次に進む前に、すべての非一貫性を解決するための手段を用意しておく必要があります。

この項では次の項目を取り上げます。

- [一般的な制約事項](#)
- [データ整合性と TSPITR](#)

一般的な制約事項

TSPITR にはいくつかの制約事項があります。次の作業で TSPITR は使用できません。

- SYSTEM 表領域を回復する。
- 削除された表領域を回復する。
- 削除された後、同じ名前を使用して再作成された表領域を回復する。
- 誤った表領域に追加されたデータ・ファイルを削除する。表領域の回復された時点より後にそのファイルが追加された場合、TSPITR の完了後も、そのファイルは（空のまま）表領域の一部として存在します。
- 補助データベースで DML 文を使用する。補助データベースは回復専用です。
- TSPITR の実行後に、回復セット・データ・ファイルの既存のバックアップを使用して回復を行う。このかわりに、回復されたファイルの新しいバックアップをとってください。TSPITR の実行前にとられたバックアップを使用して回復を試行すると、回復は失敗します。
- 統計情報が計算されているオブジェクトのオブティマイザ統計情報を回復する。統計情報は TSPITR の実行後に再計算してください。
- 次のオブジェクト・タイプのいずれかを TSPITR の回復セットに含める。
 - レプリケートされたマスター表
 - スナップショット・ログ
 - スナップショット表

これらのオブジェクトのいずれかが含まれている場合には、TSPITR の前に削除してください。

データ整合性と TSPITR

TSPITR は、回復中の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、TSPITR は正常に実行されません。

関連項目：詳細は、B-7 ページの「[ステップ 2: プライマリ・データベースに対する依存性を調査および解決する](#)」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しているすべてのファイルが、クローン・データベースの回復セットに入っていることを確認する。さもないと、TSPITR のエクスポート・フェーズは失敗します。
- 次の SQL 文を使用し、補助セットに制御ファイルのバックアップを作成する。

```
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ある必要があります。そうしないと、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。

- クローン・データベースに必要な十分なディスク領域を割り当てる。
- クローン・インスタンスを起動するのに十分な実メモリーを提供する。

関連項目：詳細は、B-10 ページの「[ステップ 4: クローン・パラメータ・ファイルを準備する](#)」を参照してください。

データベースでの TSPITR の準備

ここでは、クローン・データベースで TSPITR を実行する準備方法を次の順序で説明します。

- [ステップ 1: オブジェクトが消失するかどうかを確認する](#)
- [ステップ 2: プライマリ・データベースに対する依存性を調査および解決する](#)
- [ステップ 3: 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・パラメータ・ファイルを準備する](#)
- [ステップ 5: クローン・データベースを準備する](#)

ステップ 1: オブジェクトが消失するかどうかを確認する

表領域に対して TSPITR を実行すると、回復の時点以降に作成されたオブジェクトはすべて消失します。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 1-3 に示します。

表 1-3 TS_PITR_OBJECTS_TO_BE_DROPPED ビュー

列名	NULL?	型
OWNER	NOT NULL	VARCHAR2(30)
NAME	NOT NULL	VARCHAR2(30)
CREATION_TIME	NOT NULL	DATE
TABLESPACE_NAME		VARCHAR2(30)

このビューを問い合わせるときには、データ・フィールドのすべての要素を指定します。指定しないとデフォルトの設定値が使用されます。また、TO_CHAR および TO_DATE ファンクション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。

```
SELECT owner, name, tablespace_name, to_char(creation_time, 'YYYY-MM-DD:HH24:MI:SS'),
FROM ts_pitr_objects_to_be_dropped
WHERE tablespace_name IN ('TS1','TS2')
AND creation_time > to_date('97-JUN-02:07:03:11','YY-MON- DD:HH24:MI:SS')
ORDER BY tablespace_name, creation_time;
```

関連項目 : TS_PITR_OBJECTS_TO_BE_DROPPED ビューの詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。

ステップ 2: プライマリ・データベースに対する依存性を調査および解決する

TS_PITR_CHECK ビューを使用し、回復セットの境界が重複しているオブジェクト間の関係を明らかにします。このビューが問合せに対して行を戻した場合には、問題を調べて修正します。TS_PITR_CHECK ビューが行を戻さなかった場合にのみ、TSPITR を実行してください。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 の完了後に、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。

TS_PITR_CHECK ビューは、次の場合以外は行を戻します。

- 回復セットの中にパーティション表のすべてのパーティションおよびサブパーティションを含める。あるパーティション表の一部のパーティションのみを回復する場合は、それらのパーティションを独立した表に変換します（B-15 ページの「[パーティション表の部分的な TSPITR の実行](#)」を参照）。

- 回復セットに次のものをすべて含める。
 - 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)
- 回復セットから次のオブジェクト・タイプを除外する。
 - レプリケートされたマスター表
 - 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 つの条件を指定します。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。

```
SELECT * FROM sys.ts_pitr_check
WHERE (ts1_name IN ('TS1','TS2') AND ts2_name NOT IN ('TS1','TS2'))
OR (ts1_name NOT IN ('TS1','TS2') AND ts2_name IN ('TS1','TS2'));
```

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 "own1"
column OBJ1_OWNER format a4
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 "own2"
column OBJ2_OWNER format a4
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
```

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（適切な形式を仮定）。

```

own1   name1 subname1 obj1type ts1_name name2 subname2 obj2type own2 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

```

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。TPIND を削除するか、回復セットに ID1 および ID2 を含めてください。

関連項目：TS_PITR_CHECK ビューの詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。

ステップ 3: プライマリ・データベースを準備する

次の作業を実行します。

1. 現行のオンライン REDO ログをアーカイブします。

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

2. 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. プライマリ・データベースの回復セット表領域を NORMAL オプションでオフラインにします。I/O エラーのためにファイルに書き込みできない場合や、ファイルが使用不能な場合には、OFFLINE FOR RECOVER オプションを使用します。たとえば多数のデータ・ファイルがあり、いずれにしても過去の時点に回復されるのでファイル・ヘッダーを更新してもかまわないときには、パフォーマンス上の理由から OFFLINE FOR RECOVER を使用できます。

```
ALTER TABLESPACE tablespace_name OFFLINE FOR RECOVER;
```

この文により、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、問合せを行う物理的にも論理的にも破損していないデータのサブセットがある場合は、プライマリ・データベースの回復セット表領域を、クローンの回復中のみ READ ONLY に変更します。回復セット表領域を、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します（B-13 ページの「[ステップ 5: 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）。

関連項目： ALTER SYSTEM 文および ALTER ROLLBACK SEGMENT 文の詳細は、『Oracle8i SQL リファレンス』を参照してください。

ステップ 4: クローン・パラメータ・ファイルを準備する

本番データベースの init.ora ファイルを使用するかわりに、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。DB_BLOCK_BUFFERS、SHARED_POOL_SIZE または LARGE_POOL_SIZE などのパラメータには、低い値を設定し、メモリーを節約してください。しかし、クローン・データベースに本番用パラメータ・ファイルが使用されている場合には、他のパラメータ値（共有ブール内からメモリーを割り当てる ENQUEUE_RESOURCES パラメータなど）が高い値に設定されている場合でも、これらのパラメータ値を低く設定することで、クローン・データベースが起動するよう設定することができます。

クローンの init.ora ファイルに次のパラメータを設定します。

パラメータ	目的
CONTROL_FILES	クローン制御ファイルを指定します。クローン制御ファイルの名前と位置に設定してください。
LOCK_NAME_SPACE	クローン・データベースの名前がプライマリ・データベースの名前と同じであっても、クローン・データベースを起動できるようにします。=CLONE など、一意の値に設定します。 注意： DB_NAME パラメータは変更しないでください。
DB_FILE_NAME_CONVERT	データ・ファイルのファイル名を変換します。必要であれば新しい値に設定します。

パラメータ	目的
LOG_FILE_NAME_CONVERT	<p>REDO ログ・ファイルを改名します。たとえば、プライマリ・データベースのデータ・ファイルが <code>/ora/primary</code> ディレクトリにあり、クローンが <code>/ora/clone</code> 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。</p> <p>注意: ALTER DATABASE RENAME FILE 文を使用して REDO ログを改名することもできます。B-11 ページの「ステップ 5: クローン・データベースを準備する」を参照してください。</p>

ステップ 5: 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、次の作業を実行します。

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することもできます。詳細は、B-15 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、環境を構成します。たとえば UNIX の場合には、ORACLE_SID をクローンの名前に設定します。
3. 必要であればパラメータ・ファイルを指定し、クローン・データベースをマウントせずに起動します。

```
STARTUP NOMOUNT PFILE=/path/initCLONE.ora;
```

4. クローン・データベースをマウントします。

```
ALTER DATABASE MOUNT CLONE DATABASE;
```

データベースはクローンであるため、この時点で自動的に ARCHIVELOG モードではなくなります。すべてのファイルはオフラインです。

5. DB_FILE_NAME_CONVERT および LOG_FILE_NAME_CONVERT を設定していない場合には、新しい位置を反映するように、ファイルを改名します。

```
ALTER DATABASE RENAME FILE 'name_of_file_in_primary_location'
TO 'name_of_corresponding_file_in_clone_location';
```

DB_FILE_NAME_CONVERT および LOG_FILE_NAME_CONVERT を設定していても、異なる位置に復元されたファイルがある場合には、これらのファイルを改名します。

6. DB_FILE_NAME_CONVERT および LOG_FILE_NAME_CONVERT を設定した場合でも、クローン・データベースのすべてのファイルが、指定された位置にあるとは想定しないでください。ディスク領域の不足により、別の位置に復元されたクローン・ファイルがある可能性もあります。次の SQL 文を発行して、回復セットと補助セットのすべてのファイル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。

TSPITR 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: クローン・データベースを回復する
- ステップ 2: クローン・データベースをオープンする
- ステップ 3: クローン・データベースでエクスポートの準備を行う
- ステップ 4: メタデータをエクスポートする
- ステップ 5: 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 6: プライマリ・データベースにメタデータをインポートする
- ステップ 7: プライマリ・データベースを使用する準備をする
- ステップ 8: プライマリ・データベース内の回復した表領域をバックアップする

ステップ 1: 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・データベースを必要な時点まで回復します。次のように、不完全回復のいずれかの形式を使用します。

```
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になっていないと、エラー・メッセージが発行されます。

ステップ 2: クローン・データベースをオープンする

次の文を使用し、RESETLOGS オプションを指定してクローン・データベースをオープンします。

```
ALTER DATABASE OPEN RESETLOGS;
```

データベースはクローンであるため、この時点では SYSTEM ロールバック・セグメントのみがオンラインになります。このため、どのユーザー表領域に対しても 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する試みは失敗し、エラー・メッセージが発行されます。

ステップ 3: クローン・データベースでエクスポートの準備を行う

プライマリ・データベースの場合と同様に TS_PITR_CHECK ビューを使用して依存性を解消し、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 (B-7 ページの「[ステップ 2: プライマリ・データベースに対する依存性を調査および解決する](#)」を参照)。TS_PITR_CHECK が行を戻さない場合にのみ、TSPITR のエクスポート・フェーズを完了できます。

ステップ 4: メタデータをエクスポートする

次の文を使用し、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=tablespace_1,tablespace_2,tablespace_n
```

エクスポート・フェーズが失敗し、エラー・メッセージが生成された場合には、TS_PITR_CHECK を再度問い合せ、問題を解決してから、エクスポートを再実行します。ユーザー SYS として TSPITR のエクスポート・フェーズを実行します。そうしないと、エクスポートは失敗します。

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。

```
SHUTDOWN IMMEDIATE;
```

ステップ 5: 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表領域が読取り専用になっている場合には、それらをオフラインにする必要があります。O/S ユーティリティを使用し、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に上書きしないように注意してください。

ステップ 6: プライマリ・データベースにメタデータをインポートする

次のコマンドを使用し、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。

注意： プライマリ・データベースの中に同じ名前のオブジェクトがすでに存在している場合には、オブジェクト名の競合が発生します。これらの競合は明示的に解決してください。

関連項目： Export の詳細は、『Oracle8i チューニング』を参照してください。

ステップ 7: プライマリ・データベースを使用する準備をする

プライマリ・データベースを使用する準備は、次の手順です。

1. 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。
2. 回復セットの表領域を読み書き可能に変更します（読取り専用に変更されていた場合は、B-9 ページの「[ステップ 3: 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）。
3. 依存性を解消するために行ったすべての作業を元に戻します。たとえば、索引を再作成したり、制約を再度有効にします（B-7 ページの「[ステップ 2: プライマリ・データベースに対する依存性を調査および解決する](#)」を参照）。
4. 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、独立した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、B-15 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 8: プライマリ・データベース内の回復した表領域をバックアップする

表領域に対する TSPITR が完了した後で、O/S ユーティリティを使用して表領域のバックアップをとります。

警告： 表領域が消失する可能性があるため、必ずバックアップを作成してください。たとえば、メディア障害が発生した場合に、データベースの最新のバックアップからのアーカイブ REDO ログが、回復された表領域に論理的にリンクしていないこともあります。TSPITR の実行前にとら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。:

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、次の順序で説明します。

- 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: パーティションとスタンドアロン表を入れ替える
- 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: パーティションとスタンドアロン表を入れ替える
- 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: 回復セット表領域をオンラインにする
- ステップ 12: パーティションとスタンドアロン表を入れ替える
- ステップ 13: プライマリ・データベース内の回復した表領域をバックアップする

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する必要がある場合がよくあります。B-18 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

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る必要があります。次のように表を作成します。

```
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用されます (B-16 ページの「[ステップ 3: パーティションとスタンドアロン表を入れ替える](#)」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。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 (B-17 ページの「[ステップ 6: 回復するパーティションの索引を削除する](#)」を参照)。TSPITR が完了したら、索引を再作成します。

ステップ 3: パーティションとスタンドアロン表を入れ替える

次のコマンドを発行し、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、RESETLOGS オプションを指定してオープンした後、回復中のパーティション表と同じ列名と列データ型を持つ表を作成します。回復したいパーティションごとに表を作成します。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（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: パーティションとスタンドアロン表を入れ替える

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=y
recovery_tablespace=tablespace_1,tablespace_2,tablespace_n
```

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 を再度問い合せて問題を解決し、エクスポートを再実行してください。ユーザー SYS として TSPITR のエクスポート・フェーズを実行します。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: ユーザーは SYS としてログインしていません」が発行されます）。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 になっている場合には、OFFLINE に変更します。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に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

次のコマンドを使用し、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: 回復セット表領域をオンラインにする

プライマリ・データベースの各回復セット表領域をオンラインにします。

```
ALTER TABLESPACE tablespace_name ONLINE;
```

ステップ 12: パーティションとスタンドアロン表を入れ替える

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、対応するスタンドアロン表を入れ替えます。

```
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には、索引を再作成します。

ステップ 13: プライマリ・データベース内の回復した表領域をバックアップする

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： 表領域が消失する可能性があるため、必ずバックアップを作成してください。たとえば、メディア障害が発生した場合に、データベースの最新のバックアップからのアーカイブ REDO ログが、回復された表領域に論理的にリンクしていないこともあります。TSPITR の実行前にとら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う失敗します。

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: パーティションとスタンドアロン表を入れ替える

- 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: 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・データベース内の回復した表領域をバックアップする

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・データベースの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

RESETLOGS オプションを指定してクローンをオープンした後、回復中のパーティション表と同じ列名と列データ型を持つ表を作成します。回復するパーティションごとに表を作成します。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用されます。

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: パーティションとスタンドアロン表を入れ替える

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=y
recovery_tablespace=tablespace_1,tablespace_2,tablespace_n
```

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 を再度問い合わせ問題を確認し、エクスポートを再実行してください。ユーザー SYS として TSPITR のエクスポート・フェーズを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: ユーザーは SYS としてログインしていません」が発行されます）。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

次のコマンドを使用し、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する必要があります。そのため最初に、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ため、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、ALTER TABLE ADD PARTITION 文を使用して追加します。

ステップ 13: プライマリ・データベース内の回復した表領域をバックアップする

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消失します。

警告： 表領域をバックアップしておかないと、表領域が消失する可能性があるため、バックアップを作成してください。たとえば、メディア障害が発生した場合に、データベースの最新のバックアップからのアーカイブ REDO ログが、回復された表領域に論理的にリンクしていないこともあります。TSPITR の実行前にとら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とう失敗します。

注意： B-5 ページの「TSPITR 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 を使用して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

ALTER TABLE SPLIT PARTITION 文または ALTER TABLE ADD PARTITION 文を使用し、プライマリ・データベースに表をインポートし、パーティション表に挿入します。

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 2-13: パーティション表領域の部分的 TSPITR と同じ手順を使用する](#)

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

範囲が分割されているパーティションを回復する場合には、そのパーティションごとに、2 つのパーティションの下位のものを削除し、上位のものが下方向に拡張されるようにします。つまり、上位のパーティションは分割前と同じ範囲になります。たとえば、P1 が P1A と P1B という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 と同じ範囲になります。

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、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 2-13: パーティション表領域の部分的 TSPITR と同じ手順を使用する

B-15 ページの「[パーティション表の部分的な TSPITR の実行](#)」のステップ 2-13 の手順を使用します。

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、回復セットのファイルを " 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。お薦めするのは、ファイルを別の位置に復元してから、TSPITR のインポート・フェーズが終了する前に、コピーする方法です (B-14 ページの「[ステップ 6: プライマリ・データベースにメタデータをインポートする](#)」を参照)。

別の場所へ回復することの長所と短所

別の場所に回復する長所は次のとおりです。

- より高い可用性と柔軟性。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。
- クローンで回復が行われている間に、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る。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 (「[ステップ 3: プライマリ・データベースを準備する](#)」B-9 ページの を参照)。この場合は、回復セットの表領域をプライマリ・データベースで READ ONLY に変更します。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、このオプションは使用できません。

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。

短所は、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には（B-14 ページの「[ステップ 6: プライマリ・データベースにメタデータをインポートする](#)」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ることです。この間データは使用できません。回復中は、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 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。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・データベースが RESETLOGS オプションでオープンされてしまいます。クローンを回復する SQL*Plus コマンドは次のようになります。

```
RECOVER DATABASE USING BACKUP CONTROLFILE UNTIL CANCEL;  
CANCEL;  
OPEN DATABASE RESETLOGS;
```

この時点では REDO 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： 一貫性のある正しい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た場合には、通常データベースで RESETLOGS オプションは正しく機能します。そうしておかないと、現存のバックアップ・スクリプトとの動作の互換性がなくなります。

トランスポータブル表領域を使用した TSPITR の実行

トランスポータブル表領域機能は、表領域の Point-in-Time 回復を実行するために使用できます。この方法は、これ以前の項で説明した O/S TSPITR と似ていますが、回復された表領域をクローン・データベースからプライマリ・データベースに移動するためにトランスポータブル表領域機能を使用する点で異なります。データベース間で表領域を移送する方法を知りたい場合は、『Oracle8i 管理者ガイド』を参照してください。

O/S の TSPITR と、トランスポータブル表領域を使用した TSPITR の主な相違点は、前者では、クローンの `init.ora` ファイルの作成、クローン・データベースのマウントなどに、特殊な手順を使用する必要があることです。O/S の TSPITR では、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コンピュータ上に置かれることを想定しています。特殊なクローン・データベース・コマンドは、クローン・データベースを回復するとともに、プライマリ・データベースの破損を防止するためのエラー・チェックを提供します。

トランスポータブル表領域を使用して TSPITR を実行することにより、この要件を緩和できます。バックアップをプライマリ・データベースとは別のコンピュータ上に復元する場合には、クローン・データベースの `MOUNT` コマンドではなく、通常データベースの `MOUNT` コマンドを使用して、クローン・データベースをプライマリ・データベースであるかのよう起動することができます。しかし、プライマリ・データベースと同じコンピュータ上にバックアップを復元する場合には、O/S の TSPITR で説明した、クローン・データベースを作成する特殊な手順を使用してください。この手順は、同一コンピュータ上にクローン・データベースを回復するときに、プライマリ・データベースの偶発的な破損を防止します。

トランスポータブル表領域を使用した TSPITR は、基本的には O/S の TSPITR と同じ機能を提供しますが、次の理由により、より柔軟なものになっています。

- 削除した表領域を回復できる。O/S の TSPITR を使用した場合は不可能です。
- プライマリ・データベースに永久的変更を行う前に手順をテストするために、回復した表領域をクローン・データベースから別のデータベースに移送できる。

トランスポータブル表領域機能を使用した TSPITR の実行

1. バックアップを復元し、クローン・データベースを組み立てます。プライマリ・データベースと同一のコンピュータまたは別のコンピュータにクローン・データベースを作成します。
2. クローン・データベースの特殊な手順を使用してクローン・データベースを作成する場合には、回復セットおよび補助セットのすべてのファイルをオンラインにします。

```
ALTER DATABASE DATAFILE 'datafile_name' ONLINE;
```

通常データベースとしてクローン・データベースを作成する場合（プライマリ・データベースとは別のコンピュータ上にクローン・データベースを作成する場合）は、回復セットおよび補助セットにないすべてのデータ・ファイルをオフラインにします。

```
ALTER DATABASE DATAFILE 'datafile_name' OFFLINE;
```

3. 指定された時点までクローン・データベースを回復します。
4. RESETLOGS オプションを使用してクローン・データベースをオープンします。
5. ALTER TABLESPACE READ ONLY コマンドを発行し、回復セット内の表領域を読取り専用にします。
6. EXPORT を実行し、トランスポータブル・セットを生成します。回復セット内のすべての表領域を含めます。
7. プライマリ・データベースで、DROP TABLESPACE コマンドを使用して回復セット内の表領域を削除します。
8. IMPORT を実行し、トランスポータブル・セットをプライマリ・データベースにプラグインします。
9. 必要であれば、ALTER TABLESPACE READ WRITE コマンドを発行し、回復した表領域を読み書き両用にします。

用語集

ARCHIVELOG モード (ARCHIVELOG mode)

Oracle によって満杯になったオンライン REDO ログがディスクにコピーされる、データベース・モード。このモードは、データベース作成時に指定するか、または ALTER DATABASE コマンドを使用して指定する。動的に ALTER SYSTEM コマンドを使用するか、または初期化パラメータ LOG_ARCHIVE_START を TRUE に設定すると、自動アーカイブを使用可能にできる。

データベースを ARCHIVELOG モードで実行した場合、NOARCHIVELOG モードのときと比較していくつかの利点がある。次のことが可能。

- ユーザーがデータベースを開いてアクセスしているときもデータベースのバックアップが可能になる。
- データベースを任意の時点の状態にまで回復できる。

ARCHIVELOG モードのデータベースを障害から保護するために、アーカイブ・ログのバックアップを作成すること。

[アーカイブ REDO ログ \(archived redo log \)](#)、[NOARCHIVELOG モード \(NOARCHIVELOG mode \)](#) も参照。

ATL (自動テープ・ライブラリ) (automated tape library)

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、テープ・サイロとも呼ばれ、オペレータの介入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る。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る。たとえば、ロボット・アームでは、バーコード・リーダーが各テープのバーコードを走査して識別できる。

[メディア・マネージャ \(media manager \)](#) も参照。

DB 識別子 (DB identifier)

一意となるよう生成された内部番号で、データベースの識別に使用される。データベースを作成すると、Oracle が自動的にこの番号を作成する。

LogMiner

オンライン REDO ログまたはアーカイブ REDO ログに含まれた情報を、さまざまな選択基準に基づいて読み取ることができるユーティリティ。たとえば、V\$LOGMINER_CONTENTS のビューの情報を選択すると、次のことができる。

- 特定の表に加えられた変更を追跡する。
- 特定のユーザーが加えた変更を追跡する。
- データ・アクセスのパターンをマップする。
- データベースに対して加えられた特定の変更を取り消す、あるいは再実行するための SQL 構文を表示する。
- アーカイブ・データを使用して、チューニングを実行し、容量計画を立てる。

[アーカイブ REDO ログ \(archived redo log \)](#) も参照。

NOARCHIVELOG モード (NOARCHIVELOG mode)

満杯になったオンライン REDO ログを、ディスクにアーカイブすることを Oracle が要求しないデータベースのモード。このモードはデータベース作成時に指定する。このモードを変更する場合は ALTER DATABASE コマンドを使用する。NOARCHIVELOG モードでは、失われたデータが回復される可能性がごく小さくなるため、オラクルでは同モードによる実行を推奨しない。

[アーカイブ REDO ログ \(archived redo log \)](#)、[ARCHIVELOG モード \(ARCHIVELOG mode \)](#) も参照。

recover

(1)Recovery Manager のコマンド。最初に増分バックアップ (存在する場合)、次にアーカイブまたはオンライン REDO ログを適用し、復元されたデータ・ファイルを更新する。

(2)SQL*Plus のコマンド。アーカイブまたはオンライン REDO ログを適用し、復元されたファイルを更新する。

[回復 \(recovery \)](#) も参照。

Recovery Manager

Oracle データベースのバックアップ、復元および回復を実行するユーティリティ。これは、リカバリ・カタログと呼ばれる情報リポジトリとあわせて使用することも、またリカバリ・カタログなしで使用することもできる。リカバリ・カタログを使わない場合、RMAN はデータベースの制御ファイルを使用し、バックアップおよび回復処理に必要な情報を格納する。RMAN とメディア・マネージャをあわせて使うと、ファイルをテープなどの 3 次記憶装置にバックアップできるようになる。

[バックアップ・ピース \(backup piece \)](#)、[バックアップ・セット \(backup set \)](#)、[コピー \(copy \)](#)、[メディア・マネージャ \(media manager \)](#)、[リカバリ・カタログ \(recovery catalog \)](#) も参照。

REDO レコード (redo record)

データベースに対する単一かつ最小単位の変更が記述されている変更ベクトルのグループ。Oracle は、すべてのデータ・ブロック変更について、REDO レコードを作成し、これをディスク上の現行オンライン REDO ログに保存する。データが消失しても、REDO データを使用してデータベース・ブロックの変更による再構築が可能である。

[REDO ログ \(redo log \)](#) も参照。

REDO ログ (redo log)

REDO レコードを含むファイル。REDO ログには、次の 2 つのタイプがある。オンライン REDO ログおよびアーカイブ REDO ログ

オンライン REDO ログとは、Oracle データ・ファイルおよび制御ファイルへのすべての変更が記録される、2 つ以上のファイルのセットである。LGWR プロセスは、REDO レコードをこのログに記録する。現行のオンライン REDO ログとは、LGWR によって現在書込みが行われているオンライン REDO ログのことである。

オフライン REDO ログとも呼ばれるアーカイブ REDO ログは、保存用の宛先にコピーされたオンライン REDO ログのコピーである。データベースが ARCHIVELOG モードになっている場合、満杯になったオンライン REDO ログは、1 つ以上の ARC_n プロセスで、1 つ以上のアーカイブ・ログの宛先にコピーされる。

[アーカイブ REDO ログ \(archived redo log \)](#) [オンライン REDO ログ \(online redo log \)](#)
[REDO レコード \(redo record \)](#) も参照。

REDO ログ・グループ (redo log groups)

各オンライン REDO ログは、1 つのグループに属している。グループには、1 つ以上の同一のメンバーが含まれている。多重 REDO ログとは、REDO グループが複数のメンバーを有している REDO ログのことを指す。

REDO ログ・バッファ (redo log buffer)

Oracle が REDO レコードを記録する、システム・グローバル領域 (System Global Area: SGA) のメモリー・バッファ。このバッファは、バックグラウンドの LGWR プロセスによって現行のオンライン REDO ログにフラッシュされる。

[REDO レコード \(redo record \)](#) も参照。

RESETLOGS オプション (RESETLOGS option)

データベースをオープンする方法の一つ。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新しく作成される。さらに、ログ順序番号が 1 にリセットされ、オンライン REDO ログが再フォーマットまたは再作成される。次の操作の後には、RESETLOGS キーワードを指定し、データベースをオープンする必要がある。

- 不完全回復
- バックアップ制御ファイルを使用した回復

resync

再同期化 (resynchronization) を参照。

RMAN

Recovery Manager を参照。

SBT

System Backup to Tape インタフェース。

メディア管理インタフェース (media management interface) も参照。

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 文の ALTER DATABASE RENAME FILE '*original_name*' TO '*new_name*' と同じ機能で、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する。

TSPITR

表領域の Point-in-Time 回復 (tablespace point-in-time recovery: TSPITR) を参照。

アーカイブ (archiving)

ARCn バックグラウンド・プロセスが、満杯になったオンライン REDO ログを保存用の宛先にコピーする操作。REDO ログをアーカイブするには、データベースを ARCHIVELOG モードで実行する必要がある。

アーカイブ REDO ログ (archived redo log)

オンライン REDO ログ・グループに属する、満杯になったメンバーのコピーで、データベースが ARCHIVELOG モードになっているときに作成される。LGWR プロセスの際、各オンライン REDO ログが REDO レコードで満杯になると、Oracle はこれらのログを 1 つ以上のオフライン・アーカイブ・ログの宛先にコピーする。このコピーをアーカイブ REDO ログという。オフライン REDO ログと呼ばれることもある。

アクセス不可能なデータ・ファイル (inaccessible datafile)

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ある。

データ・ファイル (datafile) メディア障害 (media failure) も参照。

圧縮 (compression)

使用されたデータ・ブロックのみを RMAN バックアップ・セットにコピーする処理。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る。RMAN によって

作成されたバックアップ・セットには、使用されたことのあるブロックのみが含まれる。つまり、RMAN は、未使用ブロックをバックアップ・セットに書き出さない。

一貫性バックアップ (consistent backup)

メディア回復を実行することなく、RESETLOGS オプションを指定してオープンできる全体データベース・バックアップ。つまり、このバックアップのデータ・ファイルに REDO を適用しなくても一貫性が保たれる。一貫性バックアップのデータ・ファイルは、次のようになっているなければならない。

- ヘッダー内に同一のチェックポイント SCN が設定されていること。ただし、データ・ファイルが表領域に存在し、かつ読取り専用または NORMAL オプションを指定したオフラインである場合を除く（この場合、そのデータ・ファイルには、チェックポイント SCN より前の正しい SCN が設定される）。
- チェックポイント SCN 以降の変更が含まれていないこと。つまりファジーでないこと。
- 制御ファイルに格納されたデータ・ファイル・チェックポイント情報が一致していること。

一貫性バックアップは、データベースが正しく停止した後でのみ取ることができる。バックアップが完了するまで、データベースをオープンしないこと。

[正しい停止 \(clean shutdown\)](#)、[ファジー・ファイル \(fuzzy file\)](#)、[非一貫性バックアップ \(inconsistent backup\)](#)、[システム変更番号 \(system change number: SCN\)](#)、[全体データベース・バックアップ \(whole database backup\)](#) も参照。

イメージ・コピー (image copy)

単独のデータ・ファイル、アーカイブ REDO ログまたは制御ファイルのコピーで、かつ次のようなものを指す。

- そのまま使用して、回復を実行できるもの（RMAN 固有の形式のバックアップ・セットとは異なる）。
- RMAN の **copy** コマンド、または UNIX の **dd** など、O/S のコマンドを使用して生成されたもの。

[コピー \(copy\)](#) も参照。

インカーネーション (incarnation)

物理データベースのバージョン。RESETLOGS オプションを指定して、データベースをオープンすると、データベースのインカーネーションが変更される。RESETLOGS オプションを指定してオープンした後、NORMAL モードを指定したオフラインでも読取り専用でもないすべてのファイルについて、全体データベース・バックアップを行う。RMAN を使用している場合は、RESETLOGS モードでオープンした後、**reset database** コマンドを発行すること。

インスタンス (instance)

SGA、Oracle 実行プログラムおよびバックグラウンド・プロセス。インスタンスを作成するには、次のコマンドのいずれかを発行する。

- STARTUP NOMOUNT— インスタンスが開始するが、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ない。
- 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る。データベースはオープンされない。
- 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る。

インスタンスを停止するには、SHUTDOWN 文を発行する。

[システム・グローバル領域 \(System Global Area: SGA\)](#) も参照。

インスタンス回復 (instance recovery)

OPS 構成で、インスタンスを使用して REDO データをオープン・データベースに適用すること。これが行われるのは、このインスタンスが、もう一方のインスタンスのクラッシュを検出したときである。クラッシュしていないインスタンスが、自動的に REDO ログを使用して、そのインスタンスのバッファ・キャッシュに存在したデータを回復する。インスタンスがクラッシュすると、Oracle は、そのインスタンスで処理中であったあらゆるコミットされていないトランザクションを取り消し、回復の完了後そのインスタンスによってかけられていたすべてのロックを解除する。

[回復 \(recovery\)](#)、[REDO レコード \(redo record\)](#) も参照。

エクスポート (export)

Export ユーティリティを使用して、データベースから論理データ（物理ファイルではなく）を抽出すること。データをデータベースにインポートするには、Import ユーティリティを使用する。

[全体エクスポート \(full export\)](#) も参照。

オープン状態のデータベース (open database)

ユーザーが問合せおよび更新できるデータベース。データベースのオープンは、STARTUP 文で自動的に行うか、または ALTER DATABASE OPEN 文で明示的に行う。

オープン・バックアップ (open backup)

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。データベースをオープンしたまま、O/S バックアップを作成する場合は、ALTER TABLESPACE BEGIN BACKUP コマンドを発行して、表領域をホット・バックアップ・モードにする必要がある。データベースがオープンされているときに、Recovery Manager を使ってバックアップを作成する場合は、表領域をホット・バックアップ・モードにする必要はない。

[ホット・バックアップ・モード \(hot backup mode \)](#) も参照。

オフライン・データ・ファイル (offline datafile)

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。例外的な状況では、必要に応じて自動的にデータ・ファイルがオフラインになる。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ある。

データ・ファイルをオフラインにするには、次のいずれかの方法を実行する。

- ALTER TABLESPACE OFFLINE を実行する。
- コマンド ALTER DATABASE DATAFILE *filename* OFFLINE を発行する。データ・ファイルをオンライン状態に戻す前に、回復を実行する必要がある。この文は、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る。

[オフライン表領域 \(offline tablespace \)](#) も参照。

オフライン表領域 (offline tablespace)

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。表領域をオフラインにできるのは、データベースがオープンしている場合のみ。表領域がオフラインになっている場合は、その表領域に含まれたオンライン・データ・ファイルもすべてオフラインになる。

表領域をオフラインにするには、ALTER TABLESPACE OFFLINE 文で次の 3 つのオプションを指定する。

- NORMAL
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、オフラインになる。このオプションはオフライン・クリーンと呼ばれることがある。表領域のどのデータ・ファイルも使用できない場合、その表領域は NORMAL オプションを指定したオフラインにはならない。表領域内で正常にオフラインになっているデータ・ファイルには表領域をオンライン状態に戻す前に、回復する必要はない。
- TEMPORARY
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る。OFFLINE TEMPORARY コマンドでチェックポイントしたファイルについては、回復の必要はない。OFFLINE IMMEDIATE コマンドの使用時にアクセスできなかったためにチェックポイントされなかったデータ・ファイルを表領域をオンライン状態に戻す前に、回復する必要がある。
- IMMEDIATE
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とることを試行せずに、オフラインになる。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、回復する必要がある。

[オフライン・データ・ファイル \(offline datafile \)](#) も参照。

オンライン REDO ログ (online redo log)

オンライン REDO ログとは、Oracle データ・ファイルおよび制御ファイルへのすべての変更が記録される、2 つ以上のファイルのセットである。データベースに変更が加えられるたびに、Oracle は REDO レコードを REDO バッファに生成する。REDO レコードの内容は、LGWR プロセスによってオンライン REDO ログにフラッシュされる。

現行のオンライン REDO ログとは、LGWR によって現在書込みが行われているオンライン REDO ログのことを指す。LGWR は、ファイルの最後に到達した時点で、ログ・スイッチを実行し、新しいログ・ファイルへの書込みを開始する。ARCHIVELOG モードでデータベースを使用している場合は、1 つまたは複数の ARC*n* プロセスによって、REDO データがアーカイブ REDO ログにコピーされる。

[アーカイブ REDO ログ \(archived redo log \)](#) も参照。

オンライン・データ・ファイル (online datafile)

ユーザーによるアクセスが可能なデータ・ファイル。コマンド ALTER DATABASE DATAFILE *filename* ONLINE により、データベースはオープン、またはマウントされる。データベースがオープンしている場合に、データ・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る。データベースがマウントされている場合、データ・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る。ただし、データベースをオープンする前に回復を実行する必要がある。

[オンライン表領域 \(online tablespace \)](#) も参照。

オンライン表領域 (online tablespace)

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 *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る。表領域をオンラインに変えるには、その前に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る。

[オンライン・データ・ファイル \(online datafile \)](#) も参照。

回復 (recovery)

REDO データまたは増分バックアップをデータベース・ファイルに適用して、消失した変更を再構築すること。回復のタイプには、インスタンス回復、クラッシュ回復およびメディア回復の 3 つがある。Oracle では、オンライン REDO ログを使用して、最初の 2 つのタイプの回復を自動的に実行します。メディア回復を実行する場合は、バックアップを復元してコマンドを発行する必要がある。増分バックアップを適用してデータ・ファイルを回復する方法は、Recovery Manager のみである。

[完全回復 \(complete recovery \)](#)、[不完全回復 \(incomplete recovery \)](#)、[メディア回復 \(media recovery \)](#) も参照。

回復セット (recovery set)

TSPITR の実行時、以前の時点までの回復が行われる 1 つ以上の表領域。

[補助セット \(auxiliary set \)](#) [表領域の Point-in-Time 回復 \(tablespace point-in-time recovery: TSPITR \)](#) も参照。

完全回復 (complete recovery)

復元バックアップ以降に生成された、オンライン REDO およびアーカイブ REDO をすべて適用して、データベースを回復すること。通常、完全メディア回復は、1 つ以上のデータ・ファイルまたは制御ファイルが、メディア障害によって破損した場合に実行する。破損したファイルを完全に回復するには、復元バックアップ以降に生成した REDO をすべて使用する。RMAN を使用している場合は、完全回復時に増分バックアップの適用も可能。

[不完全回復 \(incomplete recovery \)](#) [メディア回復 \(media recovery \)](#) も参照。

完全再同期化 (full resynchronization)

Recovery Manager が行う処理の一つで、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・カタログを更新する。カタログの完全再同期化を実行するには、RMAN コマンドの **resync catalog** を発行する。Recovery Manager は、特定のコマンドの実行時に、必要に応じて再同期化処理を実行する。

[制御ファイル \(control file \)](#) [リカバリ・カタログ \(recovery catalog \)](#) [再同期化 \(resynchronization \)](#) も参照。

管理回復モード (managed recovery mode)

スタンバイ・データベースのモード。スタンバイ・データベースは、ターゲット・データベースからのアーカイブ・ログ・ファイルを待機し、これらのファイルが使用可能になると、自動的に REDO ログを適用する。この機能を使用すると、対話形式でアーカイブ REDO ログのファイル名を回復プロセスに指示する必要がなくなる。

[スタンバイ・データベース \(standby database \)](#) も参照。

クラッシュ回復 (crash recovery)

単一インスタンス・データベースのクラッシュか、または OPS データベースの全インスタンスのクラッシュのどちらかが発生した場合に、オンライン REDO レコードをデータベースに自動的に適用すること。クラッシュ回復に必須なのはオンライン・ログの REDO のみ。アーカイブ REDO ログは必須ではない。

クラッシュ回復では、インスタンスはデータベースをオープン前に自動的に回復する。通常、クラッシュ後または SHUTDOWN ABORT 後にデータベースをオープンした最初のインスタンスが、クラッシュ回復を自動的に実行する。

[回復 \(recovery \)](#) [REDO レコード \(redo record \)](#) も参照。

クローズ状態のデータベース (closed database)

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る場合でも、インスタンスの起動は可能であり、さらにそのデータベースをマウントすることもできる。

[オープン状態のデータベース \(open database\)](#) も参照。

クローズ状態のバックアップ (closed backup)

データベースがクローズしている間に作成される、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、全体データベース・バックアップである。データベースが正常にクローズされていれば、バックアップに含まれているファイルはすべて一貫性が保たれた状態になっている。SHUTDOWN ABORT を使用してデータベースを停止した場合、またはインスタンスが異常終了した場合、バックアップの一貫性は保たれない。

[正しい停止 \(clean shutdown\)](#)、[一貫性バックアップ \(consistent backup\)](#) も参照。

クロスチェック (crosscheck)

ディスクまたはメディア管理カタログのファイルが、リカバリ・カタログ (使用している場合) および制御ファイルの情報に対応しているかを調べるためのチェック。テープには、メディア・マネージャによって期限切れまたは使用不可のマークが付けられることがあり、ファイルはディスクから削除されたり破損することがあるため、リカバリ・カタログおよび制御ファイルには、バックアップとイメージ・コピーに関する古い情報が含まれる場合がある。

チェックするバックアップ・セットまたはバックアップ・ピースのリストが必要な場合は、**change ... crosscheck** を使用する。クロスチェックを特定のデバイス・タイプ、オブジェクト・タイプあるいはデータ範囲に限定し、RMAN でバックアップ・セットまたはバックアップ・ピースのリストを生成する場合は **crosscheck backupset** を使用する。ファイルの復元が可能かどうかを判断するには、**validate** または **restore ... validate** を使用する。

[メディア・マネージャ \(media manager\)](#)、[リカバリ・カタログ \(recovery catalog\)](#)、[妥当性チェック \(validation\)](#) も参照。

現行のオンライン REDO ログ (current online redo log)

LGWR バックグラウンド・プロセスが、REDO レコードの記録中になっているオンライン REDO ログ・ファイル。LGWR による書込みが行われていないファイルは非アクティブ・ファイルと呼ばれる。

どのデータベースにも、最低 2 つのオンライン REDO ログ・ファイルが含まれている必要がある。オンライン REDO ログを多重化すると、LGWR によって同一の REDO データが同時に複数のファイルに書き込まれる。この場合の個々のファイルは、オンライン REDO ログ・グループのメンバーと呼ばれる。

[オンライン REDO ログ \(online redo log\)](#)、[REDO ログ \(redo log\)](#)、[REDO ログ・バッファ \(redo log buffer\)](#)、[REDO ログ・グループ \(redo log groups\)](#) も参照。

現行の制御ファイル (current control file)

ディスクにある制御ファイル。データベースの現行のインカーネーションに使用される制御ファイルのうち、最後に変更されたものを指す。制御ファイルが、回復時に現行のものと認められるには、バックアップから復元されるべきではない。

制御ファイル (control file) も参照。

現行のデータ・ファイル (current datafile)

RMAN では、制御ファイルが示しているターゲット・データベースのデータ・ファイルを指す。データ・ファイルのバックアップを再度 " 現行 " にするには、**switch** コマンドを実行する。

コールド・バックアップ (cold backup)

クローズ状態のバックアップ (closed backup) を参照。

コピー (copy)

(1) データをレプリケートすること。Oracle データ・ファイル、制御ファイルおよびアーカイブ REDO ログは、次の 2 つの方法でコピーできる。

- O/S ユーティリティを使用する (UNIX の **cp** や **dd** など)。
- Recovery Manager の **copy** コマンドを使用する。

(2) Recovery Manager のコマンドの一つ。これを使用すると、データベースのデータ・ファイル、制御ファイル、またはアーカイブ REDO ログの複製が作られる。この複製は、Oracle サーバー・プロセスで作成され、Recovery Manager チャンネルに割り当てられる。そこで Oracle ファイルの読み込み、ディスクへの複製の書き込みが行われる。Recovery Manager を使うと、表領域をホット・バックアップ・モードにすることなしに、オープン・データベースのファイルをコピーできる。

バックアップ (backup) \ ホット・バックアップ・モード (hot backup mode) も参照。

コマンド・ファイル (command file)

コマンド行から実行できる、一連の RMAN コマンドが記述されているファイル。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る。

孤立したバックアップ (orphaned backups)

データベースのインカーネーションには属しているものの、このインカーネーションが現行のインカーネーションの直接の親元でないため、使用できないバックアップおよびコピー。4-22 ページの「[孤立したバックアップのレポート作成](#)」で、孤立したバックアップを図で説明している。

再同期化 (resynchronization)

ターゲット・データベースの制御ファイルの現行の情報を使って、リカバリ・カタログを更新する処理。カタログの完全再同期化を実行するには、**resync catalog** コマンドを発行する。

部分再同期化とは、アーカイブ REDO ログ、バックアップ・セット、およびデータ・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ある。部分再同期化では、次のような情報は転送されない。

- 新規データ・ファイル
- 新規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

完全再同期化または部分再同期化が必要であると Recovery Manager が判断した場合、その処理が、**backup**、**restore**、**copy**、**recover** などのコマンドの前で自動的に実行される。

差分バックアップ (differential backup)

増分バックアップの 1 つのタイプで、レベル n 以下の最後のバックアップ以降に変更されたブロックがすべてバックアップされる。たとえば、差分レベル 2 のバックアップでは、レベル 2、レベル 1、レベル 0 のバックアップのうち、どれが最も新しいかを RMAN が判断し、それ以降に変更された全ブロックをバックアップする。差分バックアップは、非累積増分バックアップとも呼ばれ、増分バックアップのデフォルトのタイプとなっている。

累積バックアップ (cumulative backup)、増分バックアップ (incremental backup)、マルチレベル増分バックアップ (multi-level incremental backups) も参照。

時間ベースの回復 (time-based recovery)

データベース・ファイルを現行以外の時刻まで不完全回復すること。時間ベースの回復は、Point-in-Time 回復とも呼ばれる。これには 2 つのタイプがある。

- データベースの Point-in-Time 回復 (DBPITR)。全データ・ファイルおよび制御ファイルを、最近の時点より前に不完全回復する。
- 表領域の Point-in-Time 回復 (TSPITR)。補助データベースの 1 つ以上の表領域に含まれた全データ・ファイルを、最近の時点より前の指定した時点に不完全回復する。その後、表領域は元のデータベースに再統合される。

不完全回復 (incomplete recovery)、メディア回復 (media recovery)、回復 (recovery)、TSPITR も参照。

システム・グローバル領域 (System Global Area: SGA)

1 つの Oracle データベース・インスタンスのデータおよび制御情報が含まれている共有メモリー構造のグループ。Oracle インスタンスは、SGA プロセスおよび Oracle プロセスで構成される。インスタンスを起動するたびに、Oracle は自動的に SGA にメモリーを割り当てます。インスタンスをシャットダウンすると、オペレーティング・システムはこのメモリーを再生する。各インスタンスはただ 1 つの SGA を持つ。

システム変更番号 (system change number: SCN)

コミットされたバージョンのデータベースを、ある時点で定義するスタンプ。Oracle は、コミットされたすべてのトランザクションに一意的 SCN を割り当てる。

ジョブ・コマンド (job commands)

backup、**copy** および **recover** などの RMAN コマンドで、**run** コマンドのかっこ内で実行する必要があるもの。

[スタンドアロン・コマンド \(stand-alone commands \)](#) も参照。

シリアル回復 (serial recovery)

回復方式の一つで、1 つのプロセスが REDO ログ・ファイルにある変更を順次適用すること。

[パラレル回復 \(parallel recovery \)](#) も参照。

スタンドアロン・コマンド (stand-alone commands)

RMAN のコマンド。このコマンドは、**run** コマンドのかっこ内で実行する必要がある。

[ジョブ・コマンド \(job commands \)](#) も参照。

スタンバイ・データベース (standby database)

本番データベースと同じ内容のコピーで、災害の保護に使用するもの。スタンバイ・データベースを現行にするには、本番データベースのアーカイブ REDO ログを使用して更新する。本番データベースが災害によって損傷を受けても、スタンバイ・データベースをアクティブにし、新しい本番データベースとすることができる。

スタンバイ転送 (standby transmission)

ネットワークを介して、アーカイブ REDO ログ・ファイルをローカルまたはリモート・スタンバイ・データベースに転送すること。

[スタンバイ・データベース \(standby database \)](#) も参照。

ステージング (staging)

アーカイブ・ログを、テープなどの 3 次記憶領域からディスクに復元して回復を実行する処理。**recover** コマンドが実行されると、RMAN はこのログをディスクに移動する。この機能を使用するには、メディア・マネージャを構成する必要がある。

[メディア・マネージャ \(media manager \)](#) も参照。

ストアド・スクリプト (stored script)

リカバリ・カタログに格納された一連の RMAN コマンド。

[リカバリ・カタログ \(recovery catalog \)](#) も参照。

スナップショット制御ファイル (snapshot control file)

Recovery Manager が作成した、データベースの制御ファイルのコピー。RMAN は、リカバリ・カタログの再同期化または制御ファイルのバックアップを実行する場合、スナップショット制御ファイルを使用して制御ファイルの一貫性のあるバージョンを読み取る。ス

ナッシュショット制御ファイルは、次に示すように、バックアップ制御ファイルを作成するのと同じ Oracle コードを使って Recovery Manager が作成する。ALTER DATABASE BACKUP CONTROL FILE TO 'location'.

スレッド (thread)

各 Oracle 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る。このようなグループを、オンライン REDO のスレッドと呼ぶ。OPS 以外の環境では、1 つのデータベースには 1 つのスレッドしか存在せず、さらにこのスレッドが、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ている。OPS 環境では、インスタンスごとに別々のスレッドがある。つまり、各インスタンスは、独自のオンライン REDO ログを持っている。各スレッドには、独自の現行ログ・メンバーがある。

制御ファイル (control file)

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バイナリ・ファイル。Oracle では、データベースの使用、制御ファイルが絶えず更新される。したがって、データベースがマウントまたはオープンされている間は、制御ファイルは常に書き込み可能になっている必要がある。

[バックアップ制御ファイル \(backup control file \)](#)、[現行の制御ファイル \(current control file \)](#) も参照。

全体エクスポート (full export)

データベース全体のエクスポート。

[エクスポート \(export \)](#) も参照。

全体データベース・バックアップ (whole database backup)

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・ファイルをバックアップすること。

[バックアップ \(backup \)](#) も参照。

全体バックアップ (full backup)

RMAN による非増分バックアップ。この場合の " 全体 " とは、データベースのバックアップ割合を示しているのではなく、バックアップが増分バックアップではないことを示している。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は可能である。

全体バックアップとレベル 0 の増分バックアップの唯一の違いは、全体バックアップは、それ以後に行われる増分バックアップのブロックの数に影響しないことである。

[増分バックアップ \(incremental backup \)](#) も参照。

増分バックアップ (incremental backup)

修正されたブロックのみがバックアップされる RMAN によるバックアップ。増分バックアップをレベルによって分類する。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、同じ機能である。ただし、全体

バックアップでは、それ以後に行われる増分バックアップでも、バックアップされるブロックに影響はないが、増分バックアップでは、それ以後に行われる増分バックアップで、バックアップされるブロックに影響が及ぶという違いがある。

レベル 1 以上の増分バックアップの場合、前回の増分バックアップ以降に変更されたブロックのみバックアップされる。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はバックアップされない。

増分バックアップは、次のように 2 つのタイプに分類される。差分および累積。差分では、レベル n 以下の最新のバックアップ以後に変更されたブロックがすべてバックアップされる。たとえば、レベル 2 の差分バックアップでは、レベル 2、レベル 1、レベル 0 のバックアップのうち、最も新しいバックアップ以後に修正された全ブロックをバックアップする。累積バックアップは、最新のレベル $n-1$ 以下のバックアップ移行に使用された全ブロックをバックアップする。たとえば、レベル 2 の累積バックアップでは、レベル 1、レベル 0 のバックアップのうち、最も新しいバックアップ以後に修正された全ブロックをバックアップする。

累積バックアップ (cumulative backup) 増分バックアップ (incremental backup) も参照。

ターゲット・データベース (target database)

RMAN でバックアップまたは復元するデータベース。

タグ (tag)

ユーザーが指定する文字列で、バックアップ・セットまたはイメージ・コピーの記号名となるもの。restore コマンドまたは change コマンドを実行するときタグを指定できる。タグの最大長は 30 文字。

多重化 (multiplexing)

■ オンライン REDO ログ

同じ内容の複数のオンライン REDO ログのコピーを、自動的にメンテナンスすること。オンライン・ログを多重化するには、各 REDO ログ・グループ内に複数のメンバーを作成する。多重化の程度は、各グループのメンバー数に直接関係する。

■ 制御ファイル

同じ内容の複数のデータベース制御ファイルのコピーを、自動でメンテナンスすること。制御ファイルを多重化するには、CONTROL_FILES の初期化パラメータに複数のエントリを作成する。

■ バックアップ・セット

同じ RMAN バックアップ・セット内のデータ・ファイルのブロックは多重化 (つまり混合) される。バックアップ・セット内の全データ・ファイルのブロックは、セット内の他のデータ・ファイルのブロックとともに分散される。

■ アーカイブ REDO ログ

Oracle プロセス ARC_n では、REDO ログの複数のコピーをアーカイブできる。アーカイブ REDO ログを多重化するには、LOG_ARCHIVE_DEST__n (_n には 1 ~ 5 の整数で、最大 4 つの追加コピーが作成可能) か、LOG_ARCHIVE_DUPLEX_DEST (追加コピーを 1 つ作成可能) を、INIT.ORA ファイルに設定する。

[ミラー化 \(mirroring\)](#) も参照。

正しい停止 (clean shutdown)

SHUTDOWN コマンドで、IMMEDIATE、TRANSACTIONAL または NORMAL オプションを指定して停止したデータベース。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、回復する必要はない。

妥当性チェック (validation)

バックアップ・セットまたはバックアップ・コピーが、復元可能かどうかをチェックするテスト。RMAN は、指定したバックアップ・セットのコピーまたはバックアップ・ピースをすべて走査し、チェックサムを参照してこれらの内容が正しく復元したかどうかを検証する。

バックアップ・セット内の、1 つ以上のコピーまたはバックアップ・ピースが、消失あるいは破損したと考えられるときは、**restore ... validate** または **validate backupset** コマンドを使用する。**restore ... validate** および **validate backupset** は、ファイルが復元可能かどうかを実際にテストしますが、**change ... crosscheck** および **crosscheck** の方は、ファイル・ヘッダーを調べるだけであることに注意すること。

[クロスチェック \(crosscheck\)](#) [メディア・マネージャ \(media manager\)](#) [リカバリ・カタログ \(recovery catalog\)](#) も参照。

チェックサム (checksum)

Oracle データ・ブロックの内容から数学的に導かれた数値。Oracle ではチェックサムを使用してブロックの一貫性を検証する。

[データ・ブロック \(data block\)](#) も参照。

チェックポイント (checkpoint)

REDO レコードで指定された SCN より前の変更が、DBW_n によってデータ・ファイルにすべて書き込まれたことを示すポイント。REDO ログ内の各 REDO レコードには、データベース・ブロックに加えた変更または最小変更単位のセットが記述されています。REDO エントリのチェックポイントによって、前の REDO エントリに記述された変更が、メモリー・バッファのみでなくディスクにも確実に書き込まれる。バックグラウンド・プロセス CKPT は、チェックポイントを 3 秒ごとに自動的に制御ファイルに記録する。

[制御ファイル \(control file\)](#) [REDO レコード \(redo record\)](#) も参照。

チェックポイントの繰上げ (advancing the checkpoint)

チェックポイントをマークする REDO ログ・エントリが変更されたときに実行される動作。たとえば、CKPT プロセスで REDO レコード 356 がチェックポイントとして記録され、その 3 秒後に REDO レコード 358 がチェックポイントとして記録される場合。

[チェックポイント \(checkpoint\)](#) [REDO レコード \(redo record\)](#) も参照。

チェンジ・ベクター (change vector)

単一データ・ブロックに対する単一の変更。チェンジ・ベクターは、REDO ログに記録される変更の最小単位。

[REDO レコード \(redo record\)](#) も参照。

チャネル (channel)

Recovery Manager とターゲット・データベースの接続。割り当てられた各チャネルは、新しい Oracle サーバー・セッションを開始する。このセッションは、バックアップ、復元および回復作業を行う。チャネルのタイプによって、Oracle サーバー・プロセスが、ディスクに対して読取りと書込みを行うか、またはサード・パーティのメディア・マネージャを介して行うかが決まる。チャネルのタイプは次のとおり。

- **disk** 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または書込みを行う。
- **'sbt_tape'** の場合、サーバー・プロセスは、サード・パーティのメディア・マネージャに対してバックアップの読取りまたは書込みを行います。

チャネルは、タイプに関係なく、ディスクに対する、読取りと書込みを常時実行できます。

[メディア・マネージャ \(media manager\)](#) [ターゲット・データベース \(target database\)](#) も参照。

データ・ファイル (datafile)

データ・ファイルは、Oracle で作成されたディスク上の物理 O/S ファイルで、表や索引などのデータ構造を含んでいる。データ・ファイルは、1 つのデータベースにのみ属することができる。

[アクセス不可能なデータ・ファイル \(inaccessible datafile\)](#) も参照。

データ・ファイルのコピー (datafile copy)

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
- O/S ユーティリティ

[バックアップ \(backup\)](#) [コピー \(copy\)](#) も参照。

データ・ファイル・ヘッダー (datafile header)

[ファイル・ヘッダー \(file header\)](#) を参照。

データ・ブロック (data block)

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る。

[破損ブロック \(corrupt block\)](#) も参照。

データベースの Point-in-Time 回復 (DataBase Point-In-Time Recovery: DBPITR)

データベースを、指定した現行以外の時刻、SCN またはログ順序番号の状態に回復すること。

[不完全回復 \(incomplete recovery\)](#)、[表領域の Point-in-Time 回復 \(tablespace point-in-time recovery: TSPITR\)](#) も参照。

テープ・サイロ (tape silo)

[ATL \(自動テープ・ライブラリ\) \(automated tape library\)](#) を参照。

テープ・ドライブ (tape drive)

磁気テープからの読み込み、および磁気テープへの書き込みを行うハードウェア。

テープのストリーミング (tape streaming)

テープが常にビジーである位、高速でテープ・ドライブに出力を書き込むこと。

テープ・ボリューム (tape volume)

物理的な 1 巻のテープ・メディア。

登録 (registration)

RMAN で、**register database** コマンドを実行し、ターゲット・データベースの存在をリカバリ・カタログに記録すること。

トランスポータブル表領域 (transportable tablespace)

表領域のセットをあるデータベースから別のデータベースに移送する機能。表領域をデータベースに移送する、つまり "組み込む" ことと、あらかじめロードされたデータで表領域を作成することは類似している。この機能の利点は次のとおり。

- インポート / エクスポート、またはアンロード / ロードよりも速い。必要となる操作が、データ・ファイルのコピーおよびメタデータの統合のみだからである。
- 索引データを移動できる。それにより索引を再作成する必要がなくなる。

取消しベースの回復 (cancel-based recovery)

不完全メディア回復の一種。UNTIL CANCEL 句を指定して RECOVER コマンドを使用する。CANCEL コマンドを発行するまで回復処理が継続する。

[不完全回復 \(incomplete recovery\)](#) [メディア回復 \(media recovery\)](#) も参照。

パスワード・ファイル (password files)

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 SYSDBA として接続する場合、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る。詳細は Oracle8i 管理者ガイドを参照。

破損データ・ファイル (corrupt datafile)

1 つ以上の破損ブロックを含むデータ・ファイル。

[破損ブロック \(corrupt block\)](#) も参照。

破損ブロック (corrupt block)

Oracle 形式で認識されない、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する。

- 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：形式が正しくないブロック。このようなブロックは、次のような状態になっていることが考えられる。
 - 無効な形式

メディアが破損したブロックは、次の方法でしか修復できない。

- ブロックを置き換えて、回復を実行する。ブロックの置換えは、データ・ファイルを回復するか、または増分バックアップを適用することによって行われる。
- ブロックを再生する。ブロックを再生するには、破損したブロックを含む表（またはその他のデータベース・オブジェクト）を削除します。再生されたブロックは他のオブジェクトで再使用される。

メディア破損の原因がハードウェア故障の場合、前述のどちらの解決策も、そのハードウェアの故障を直さなければ効果がない。

[データ・ブロック \(data block\)](#) [分裂ブロック \(fractured block\)](#) も参照。

バックアップ (backup)

(1) データベース、表領域、表、データ・ファイル、制御ファイルまたはアーカイブ REDO ログなどのデータのコピー。バックアップは次の方法で作成できる。

- Export ユーティリティを使用して 1 つ以上の表をコピーする。
- Recovery Manager を使用して、1 つ以上のデータ・ファイル、制御ファイルまたはアーカイブ REDO ログのバックアップを作成する。

- 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コピーを作成する。

(2) バックアップ・セットを作成する RMAN コマンド。バックアップ・コマンドの出力が使用できるのは RMAN のみです。RMAN **copy** コマンドの出力は、追加処理なしで使用できます。

[コピー（copy）](#) [バックアップ・セット（backup set）](#) [多重化（multiplexing）](#) RMAN も参照。

バックアップ、クローズ状態（**backup、closed**）

[クローズ状態のバックアップ（closed backup）](#) を参照。

バックアップ、データベース全体（**backup、whole database**）

[全体データベース・バックアップ（whole database backup）](#) を参照。

バックアップ制御ファイル（**backup control file**）

制御ファイルのバックアップ。このバックアップは次の方法で作成する。

- (1) Recovery Manager の **backup** コマンドまたは **copy** コマンドを使用する。バックアップ制御ファイルの作成には O/S コマンドを使用しない。
- SQL のコマンドである ALTER DATABASE BACKUP CONTROLFILE TO '*filename*' を使用する。

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するが、ある種の Point-in-Time 回復の実行前に復元することもある。

[制御ファイル（control file）](#) も参照。

バックアップ・セット（**backup set**）

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の論理グループで、RMAN に固有のもの。RMAN の **backup** コマンドの出力がバックアップ・セットである。バックアップ・セットからファイルを抽出するには RMAN の **restore** コマンドを使用する。ファイルを多重化してバックアップ・セットにする、つまりいくつかの入力ファイルのブロックを混合して 1 つのバックアップ・セットにすることが可能。

バックアップ・セットには、次の 2 つのタイプがある。

- データ・ファイル・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指す。このタイプのバックアップ・セットは圧縮される。つまり、使用されたことのあるデータ・ファイル・ブロックのみが含まれ、未使用のブロックは省かれる。
- アーカイブ・ログ・バックアップ・セット。アーカイブ REDO ログのバックアップを指す。

バックアップ・ピース (backup piece) も参照。圧縮 (compression) \ 多重化 (multiplexing) \ RMAN

バックアップ・ピース (backup piece)

バックアップ・ピースとは、RMAN 固有のフォーマットで作成される物理ファイルの一種で、1 つのバックアップ・セットのみに属す。通常、バックアップ・セットは、1 つのバックアップ・ピースのみで構成される。RMAN が複数のバックアップ・ピースを作成するのは、**set limit kbytes** コマンドでピース・サイズを制限した場合のみ。これらのコマンドは、バックアップの書き込み先の記憶領域またはメディア・マネージャが、ある一定のサイズ以上のファイルの書き込みをサポートできない場合に使用する。

バックアップ (backup) も参照。バックアップ・セット (backup set) \ RMAN

バッファ・キャッシュ (buffer cache)

データ・ファイルから読み込まれたデータ・ブロックのコピーが保管されている SGA 部分。インスタンスに同時接続されたユーザー・プロセスはすべて、データベース・バッファ・キャッシュへのアクセスを共有する。

このキャッシュのバッファは、次の 2 つのリストで編成される。使用済みリストおよび最低使用頻度 (least recently used: LRU) リスト。使用済みリストには使用済みバッファが保管されている。使用済みバッファには、変更はされたがディスクへの書き込みがまだ済んでいないデータが含まれる。最低使用頻度 (least recently used: LRU) リストには、空きバッファ (未修正で使用可能) \ 使用中バッファ (現在アクセス中) \ そして使用済みリストにまだ移動されていない使用済みバッファが保管されている。

システム・グローバル領域 (System Global Area: SGA) も参照。

パラレル回復 (parallel recovery)

回復方式の一つで、複数のプロセスが REDO ログ・ファイルからの変更を同時に適用すること。SQL/SQL*Plus RECOVER コマンドで初期化パラメータまたはオプションを指定すると、インスタンスおよびメディア回復のパラレル化を自動的に行うことができる。Oracle は 1 つのプロセスを使用して、ログ・ファイルを順次読み込み、REDO 情報を複数の回復プロセスにディスパッチする。これにより回復プロセスは、読み込んだログ・ファイルからの変更を、データ・ファイルに適用する。

シリアル回復 (serial recovery) も参照。

パラレル化 (parallelization)

複数のチャンネルを、Recovery Manager のバックアップ処理および回復処理に割り当てること。パラレル化が可能な処理は次のとおり。

- バックアップ・セットの作成。複数のチャンネルを割り当て、**backup** コマンドを発行する。
- ファイル・コピーの作成。複数のチャンネルを割り当て、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。

- 復元操作。並列度は、割り当てられたチャネルの数、および復元操作時に読み込む必要のあるバックアップ・セットまたはファイル・コピーの数によって異なる。
- 増分バックアップの適用時の復元操作。並列度は、割り当てられたチャネルの数、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る。

非アクティブ REDO ログ (inactive redo log)

REDO レコードに含まれた変更が、データベースに適用済みとなっているため、インスタンス回復では必要とされない REDO ログ・ファイル。現行の REDO ログ・ファイルが非アクティブとなることはない。ARCHIVELOG モードでデータベースを使用している場合は、ARCn プロセスによって、非アクティブ REDO ログ・ファイルはアーカイブされる。

現行のオンライン REDO ログ (current online redo log)、オンライン REDO ログ (online redo log)、REDO ログ (redo log)、REDO ログ・バッファ (redo log buffer)、REDO ログ・グループ (redo log groups) も参照。

非一貫性バックアップ (inconsistent backup)

バックアップの一部のファイルに、それらのファイルのチェックポイント以後の変更が含まれるようなバックアップ。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、回復処理が必要である。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。それは、ファイルのバックアップ中、そのデータベースがオープンされていた場合である。非一貫性バックアップは、データベースがクローズ状態でも、次のような時に作成される。

- Oracle インスタンス (または Oracle Parallel Server クラスタ内のすべてのインスタンス) がクラッシュした直後
- SHUTDOWN ABORT によってデータベースを停止した後

非一貫性バックアップが有効なのは、データベースが ARCHIVELOG モードの場合のみであることに注意すること。

一貫性バックアップ (consistent backup)、オープン・バックアップ (open backup)、システム変更番号 (system change number: SCN)、全体データベース・バックアップ (whole database backup) も参照。

標準アーカイブ送信 (normal archiving transmission)

アーカイブ REDO ログをローカル・ディスクに送信すること。

スタンバイ転送 (standby transmission) も参照。

表領域 (tablespace)

データベースは、表領域と呼ばれる 1 つ以上の論理記憶領域単位に分割される。各表領域には、その表領域にのみ対応付けられたデータ・ファイルが 1 つ以上格納される。

データ・ファイル (datafile) も参照。

表領域の Point-in-Time 回復 (tablespace point-in-time recovery: TSPITR)

1 つ以上の非 SYSTEM 表領域を、データベースとは異なる時点の状態まで回復すること。TSPITR は、RMAN と O/S の方式のいずれを使用しても実行できる。

ファイル・ヘッダー (file header)

Oracle データ・ファイルの最初のブロック。ファイル・ヘッダーには、チェックポイント SCN など、このファイルに関連する情報が入っている。データ・ファイル・ヘッダーのチェックポイント SCN が、制御ファイルに格納されているファイル・ヘッダー情報と一致していない場合は、Oracle でメディア回復を実行する必要がある。

[チェックポイント \(checkpoint \)](#) も参照。

ファジー・ファイル (fuzzy file)

チェックポイント SCN よりも新しい SCN が付されたブロックを、ブロック・ヘッダーに少なくとも 1 つ含んでいるデータ・ファイル。たとえば、ホット・バックアップ・モードになっているデータ・ファイルが、Oracle によって更新されたとき、このような状況が発生する。復元されたファジー・ファイルは、必ず回復を実行する必要がある。

[チェックポイント \(checkpoint \)](#)、[ホット・バックアップ・モード \(hot backup mode \)](#) も参照。

不完全回復 (incomplete recovery)

復元バックアップの作成以後に生成された変更のうち、一部を適用しないままデータベースを回復すること。

不完全回復は、通常、次のような場合に行う。

- 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・ログまで回復する。
- ユーザーのエラーが原因で、エラー発生直前の状態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ある。たとえば、給与支払いのトランザクションを給与支払機関に送る前に、ユーザーが誤って削除してしまったとする。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の状態まで不完全回復を実行する必要がある。

- 回復に必要なアーカイブ REDO 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 REDO ログの内容が破損している場合。この場合、ログを紛失した時点まで回復するしか方法はない。

いずれの場合も、メディア回復を実行した後、RESETLOGS オプションを指定してデータベースをオープンする。リカバリ・カタログで RMAN を使用する場合は、データベースのリセットも必要である。

完全回復 (complete recovery) 、メディア回復 (media recovery) 、回復 (recovery) 、REDO レコード (redo record) も参照。

復元 (restore)

消失したファイルまたは破損したファイルを、バックアップと置き換えること。ファイルを復元するには、UNIX の `cp` などの O/S コマンド、または RMAN の `restore` コマンドを使用する。

`recover` も参照。

複数 ARC n 処理 (multiple ARC n processing)

複数 ARC n 処理を使用して、オンライン REDO ログを 1 つ以上の場所にアーカイブすること。LGWR によるオンライン REDO ログへの書込みが、単独のアーカイブ・プロセスによるアーカイブ宛先への書込みよりも速い場合、発生するボトルネックを、複数 ARC n 処理を行うことにより防ぐことができる。初期化パラメータ `LOG_ARCHIVE_MAX_PROCESS = n` (n は 1 ~ 10 の整数) を設定すると、起動時または実行時にこの機能が使用可能になる。

複製データベース (duplicate database)

RMAN の `duplicate` コマンドを使用して、ターゲット・データベースのバックアップから作成したデータベース。

補助データベース (auxiliary database) も参照。

物理スキーマ (physical schema)

ある時点でデータベースに存在しているデータ・ファイル、表領域、REDO スレッドおよび REDO ログ。表領域およびデータ・ファイルのリストを取得するには、RMAN の `report schema` コマンドを発行する。

リカバリ・カタログの完全再同期化では、物理スキーマ情報も含め、変更された RMAN のメタデータがすべて更新される。ターゲット・データベースがオープンしている場合は、RMAN はロールバック・セグメントの情報収集も行う。リカバリ・カタログを部分再同期化した場合は、物理スキーマもロールバックも更新される。

再同期化 (resynchronization) も参照。

物理バックアップ (physical backups)

ある場所から別の場所にコピーされた物理データベース・ファイル。このファイルは、データ・ファイル、アーカイブ REDO ログまたは制御ファイルのいずれでも構わない。物理バックアップを作成するには、Recovery Manager を使用する、または UNIX `dd` などの O/S コマンドを指定する。

部分再同期化 (partial resync)

再同期化 (resynchronization) を参照。

プラグ可能表領域 (pluggable tablespace)

トランスポータブル表領域 (transportable tablespace) を参照。

プロキシ・コピー (proxy copy)

RMAN によるバックアップ操作および復元操作時の、メディア記憶デバイスとディスク間のデータ転送を、メディア・マネージャが実行できる機能。

メディア・マネージャ (media manager) も参照。

分割ブロック (split block)

分裂ブロック (fractured block) を参照。

分裂ブロック (fractured block)

メディア破損の 1 タイプで、バックアップするブロックを O/S ユーティリティが読み込んでいる最中に、DBWn によってブロックの書き込みが行われると発生する場合がある。O/S によって読み込まれたブロックが、分割されている可能性がある。たとえば、ブロックの先頭が書き込まれる時点と、ブロックの最後が書き込まれる時点が異なる場合である。分裂ブロックが含まれたファイルを復元して、Oracle がそのブロックを読み込んだ場合、同ブロックは破損していると見なされる。

分裂ブロックが発生する可能性があるため、O/S オンライン・バックアップを行う前に、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る。データベースをホット・バックアップ・モードにすると、変更しようとする Oracle データ・ブロック全体が REDO ログに書き込まれる。その結果、バックアップ時にブロックが分割されても、REDO を使用して修復できる。Recovery Manager ではこの問題は発生しない。バックアップまたはコピーを行うサーバーは、各ブロックを読み込んでそれが分割されているかどうかを判断し、一貫性のあるバージョンが得られるまでブロックの再読み込みを行うためである。

破損ブロック (corrupt block) 破損データ・ファイル (corrupt datafile) ホット・バックアップ・モード (hot backup mode) も参照。

変更ベースの回復 (change-based recovery)

不完全メディア回復の一種で、指定した SCN までを回復する。CANCEL コマンドを発行するまで回復を行う取消しベースの回復、および指定した時間まで回復を行う時間ベースの回復を実行することも可能。

取消しベースの回復 (cancel-based recovery) 不完全回復 (incomplete recovery) メディア回復 (media recovery) システム変更番号 (system change number: SCN) 時間ベースの回復 (time-based recovery) も参照。

補助セット (auxiliary set)

TSPITR の場合に、回復セットにはないファイルの集合。ただし、TSPITR セットが正しく作動するためには、これらのファイルをクローン・データベースで復元する必要がある。補助セットのファイルには、次のようなものがある。

- バックアップ制御ファイル

- SYSTEM 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）

補助データベース（auxiliary database）、回復セット（recovery set）、TSPITR を参照。

補助データベース（auxiliary database）

(1) RMAN **duplicate** コマンドを使用して、ターゲット・データベースのバックアップから作成したデータベース。

(2) 新しい場所に復元され、表領域の Point-in-Time 回復（TSPITR）時に新しいインスタンス名で起動される一時データベース。TSPITR 補助データベースは、回復セットと補助セットで構成される。

回復セット（recovery set）、補助セット（auxiliary set）、TSPITR も参照。

ホット・バックアップ（hot backup）

オープン・バックアップ（open backup）を参照。

ホット・バックアップ・モード（hot backup mode）

オープン・バックアップを行う前に、ALTER TABLESPACE *tablespace_name* BEGIN BACKUP コマンドを発行すると実行されるデータベース・モード。ALTER TABLESPACE *tablespace_name* END BACKUP コマンドを発行すると、表領域のホット・バックアップ・モードがオフになる。

オンライン表領域の、1 つ以上のデータ・ファイルの O/S バックアップを作成する場合は、このコマンドを使用する必要がある。Recovery Manager を使用している場合は、データベースをホット・バックアップ・モードにする必要はない。ホット・バックアップ・モードで表領域を更新すると、通常よりも多くの REDO が作成される。変更が加えられるたびに、Oracle は、変更されたデータのみでなくブロック全体を最初に REDO ログに書き込むためである。

破損ブロック（corrupt block）、分裂ブロック（fractured block）、オープン・バックアップ（open backup）も参照。

マウントされたデータベース（mounted database）

起動済みのインスタンスで、オープン・データベースに関連付けられた制御ファイルを持つもの。データベースは、オープンしなくてもマウントできる。通常、メンテナンスまたは復元処理、回復処理を実行する場合に、データベースをこの状態にする。

インスタンス（instance）も参照。

マルチレベル増分バックアップ（multi-level incremental backups）

RMAN によって生成される増分バックアップ。これを使用し、どのブロックをいつバックアップするか計画を立てることによって領域の節約が可能になる。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る。このバックアップでは、データが含まれているブロックがすべてコピーされる。レベル n の増分バックアップを生成するときに、 n が 0 より大である場合は、次のいずれかのバックアップが作成される。

- レベル n 以下の最新のバックアップ以後に変更された全ブロックこれは増分バックアップのデフォルトのタイプで、差分バックアップと呼ばれる。
- レベル $n-1$ 以下の最新のバックアップ以後に使用された全ブロックこのタイプのバックアップは、累積バックアップと呼ばれる。

日ごとに異なるレベルのバックアップ方針を立て、バックアップするデータの量を制御できる。

[累積バックアップ \(cumulative backup\)](#) [差分バックアップ \(differential backup\)](#) [増分バックアップ \(incremental backup\)](#) も参照。

ミラーの復元 (resilvering a mirror)

破壊されたミラーを、更新済みの一方のミラーからリフレッシュし、ミラーの両側をメンテナンスするよう、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指示すること。

[ミラー化の解除 \(breaking a mirror\)](#) [ミラー化 \(mirroring\)](#) も参照。

ミラー化 (mirroring)

O/S を使用して、Oracle データと同一内容のコピーをメンテナンスすること。一般的にミラー化は、二重化されたハードディスクにおいてオペレーティング・システム・レベルで実行する。したがって、いずれかのディスクが使用不可能になっても、中断することなくもう一方のディスクで要求の処理を継続できる。たとえば、データ・ファイルをミラー化すれば、Oracle は 2 つのディスク・ドライブに同じ情報を書き込むことになる。バックアップを作成するためにミラー化を解除し、後でミラーを復元できる。

ファイルをミラー化すると、オペレーティング・システムが複数のディスクに書込みを行う間に、Oracle は書込みを 1 回行う。ファイルを多重化すると、Oracle は複数のファイルに同じデータを書き込む。

[ミラー化の解除 \(breaking a mirror\)](#) [ミラーの復元 \(resilvering a mirror\)](#) も参照。

ミラー化の解除 (breaking a mirror)

ディスクのミラー化手順が終了した結果、ミラー・イメージが最新の状態に保たれなくなる。オペレーティング・システム・データベースのバックアップを作成するには、そのデータベースにホット・バックアップ・モードで表領域を設定し、ミラー化を解除する。ホット・バックアップ・モードをオフにして表領域を取り出した後、ミラー化が解除された方をテープにバックアップする。バックアップが完了すると、ミラーの復元が可能になる。

[ホット・バックアップ・モード \(hot backup mode\)](#) [ミラー化 \(mirroring\)](#) [ミラーの復元 \(resilvering a mirror\)](#) も参照。

メディア回復 (media recovery)

オンラインのアプリケーションまたはアーカイブ REDO ログ・レコードに復元バックアップを適用して、指定した時点の状態を現行のものとする。メディア回復を実行して回復できるものは次のとおり。

- データベース全体
- 表領域
- データ・ファイル

REDO データをすべて使用した場合は完全回復となる。REDO データの一部のみを使用した場合は不完全回復となる。通常、メディア回復は、メディア障害の発生後に実行する。

ARCHIVELOG モードでは、完全回復か不完全回復かを選択できる。NOARCHIVELOG モードでは、通常、最新のバックアップから REDO データを適用せずに復元することのみが可能である。

データベースが ARCHIVELOG モードでなくても、例外的にデータベースまたはデータ・ファイルを回復できる。ただし、バックアップ後に上書きされたオンライン・ログがないことが条件である。

[完全回復 \(complete recovery \)](#) [不完全回復 \(incomplete recovery \)](#) [メディア障害 \(media failure \)](#) [回復 \(recovery \)](#) [REDO レコード \(redo record \)](#) も参照。

メディア管理インタフェース (media management interface)

Oracle が公開した API。メディア管理ソフトウェアのベンダーが、このインタフェースに準拠した互換性のあるソフトウェア・ライブラリを作成している。このソフトウェアは Oracle に統合される。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・ファイルを書き込んだり、順次記憶装置からファイルを読み込むことができる。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、ラベル付けおよびアンロードに必要なアクションを処理する。

メディア管理インタフェースは、メディア管理レイヤー、メディア管理ライブラリ (Media Management Library: MML)、SBT インタフェースとも呼ばれる。

[メディア・マネージャ \(media manager \)](#) も参照。

メディア障害 (media failure)

データベースの運用に必要なファイルの書込みまたは読取りに、Oracle が失敗したとき発生する物理的な問題。よくある例は、ディスク・ヘッドのクラッシュによりディスク・ドライブ上のすべてのデータが失われる場合である。ディスク障害により、データ・ファイル、REDO ログ・ファイルおよび制御ファイルなどさまざまな種類のファイルに影響があることがある。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、SGA 領域のバッファ・キャッシュ内のデータをデータ・ファイルに書き込めなくなる。

[バッファ・キャッシュ \(buffer cache \)](#) [メディア回復 \(media recovery \)](#) も参照。

メディア・マネージャ (media manager)

サード・パーティ・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、順次メディアのロード、ラベル付け、およびアンロードを行う機能がある。メディア・マネージャは、メディアの時間切れおよびリサイクルを構成する機能もある。自動テープ・ライブラリ (Automated Tape Libraries: ATLs) の制御機能を持っているメディア・マネージャもある。

[ATL \(自動テープ・ライブラリ \) \(automated tape library \)](#) も参照。

読取り専用データベース (read-only database)

ALTER DATABASE OPEN READ ONLY コマンドでオープンされるデータベース。その名称が示すとおり、読取り専用データベースは問合せ専用で、修正はできない。Oracle では、スタンバイ・データベースを、読出し専用モードで実行できる。つまりプライマリ・データベースの最新かつ緊急の代用として機能していても、問合せを行うことができる。

読取り専用表領域 (read-only tablespace)

更新禁止の状態に変更された表領域。表領域は、SQL 文の ALTER TABLESPACE< 表領域 >READ ONLY を実行することで読取り専用モードになる。バックアップの回数を減らす場合は、通常、表領域を読取り専用モードにする。これにより、表領域のバックアップ回数をたとえば毎晩 1 回ではなく、毎月 1 回にできる。

注意： 表領域のバックアップをとる間隔が長くなると、(バックアップの回数が減るため) バックアップ・メディアを占有する時間も長くなり、さらにバックアップ・メディアが破損するリスクも大きくなる。

リカバリ・カタログ (recovery catalog)

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。Recovery Manager はこのデータを使用して、Oracle データベースのバックアップ、復元および回復を管理する。リカバリ・カタログを使わない場合、RMAN はターゲット・データベースの制御ファイルを使用する。

[リカバリ・カタログ・データベース \(recovery catalog database \)](#) も参照。

リカバリ・カタログ・データベース (recovery catalog database)

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。リカバリ・カタログは、ターゲット・データベースに格納しないこと。

累積バックアップ (cumulative backup)

増分バックアップの一つで、 $n-1$ またはそれ以下のレベルの最新のバックアップ以降に変更された全ブロックをバックアップすること。たとえば、累積レベル 2 のバックアップでは、レベル 1 とレベル 0 のバックアップのどちらが新しいかを RMAN が判断し、新しい方のバックアップ以降に変更された全ブロックをバックアップする。

データ・ブロック (data block)、差分バックアップ (differential backup)、増分バックアップ (incremental backup)、マルチレベル増分バックアップ (multi-level incremental backups) も参照。

ロールバック (rolling back)

回復のロールフォワード段階でデータベースに適用される、コミットされていないトランザクションを、ロールバック・セグメントを使用して取り消すこと。

回復 (recovery)、ロールフォワード (rolling forward) も参照。

ロールフォワード (rolling forward)

REDO レコードまたは増分バックアップを、データ・ファイルおよび制御ファイルに適用して、これらのファイルに加えた変更を回復すること。

回復 (recovery)、ロールバック (rolling back) も参照。

ログ順序回復 (log sequence recovery)

RMAN で実行される不完全回復の一種で、指定したログ順序番号まで回復する。

不完全回復 (incomplete recovery) も参照。

ログ順序番号 (log sequence number)

REDO ログ・ファイルの REDO レコードのセットを一意的に識別するための番号。あるオンライン REDO ログ・ファイルが満杯になって別のオンライン REDO ログ・ファイルに切り替わると、Oracle は新しいファイルにログ順序番号を自動的に割り当てる。たとえば、2 つのオンライン・ログ・ファイルを持つデータベースを作成したとすると、最初のファイルにはログ順序番号 1 が割り当てられる。最初のファイルが満杯になると、Oracle は 2 番目のファイルに切り替え、ログ順序番号 2 を割り当てる。再び最初のファイルに切り替わると、Oracle はログ順序番号 3 を割り当てるといように続く。

ログ・スイッチ (log switch)、REDO ログ (redo log) も参照。

ログ・スイッチ (log switch)

LGWR がアクティブな REDO ログ・ファイルへの書き込みを停止し、使用可能な次の REDO ログ・ファイルに切り替える時点。LGWR によって切り替えが行われるのは、アクティブなログ・ファイルが、REDO レコードで満杯になった場合、手動で切り替えを強制的に実行した場合。

ARCHIVELOG モードでデータベースを使用している場合、Oracle は、非アクティブなログ・ファイルの REDO データを、アーカイブ REDO ログにアーカイブする。ログ・スイッチが発生し、LGWR が古い REDO データに上書きを開始しても、アーカイブ REDO ログにはその古いデータが含まれるため、データが失われることはない。NOARCHIVELOG モードで実行している場合、ログ・スイッチの時点で、Oracle は古い REDO データに対してアーカイブせずに上書きする。したがって、古い REDO データはすべて失われる。

REDO ログ (redo log) も参照。

ログの最後尾 (tail of the log)

REDO ログ・ファイルの中で最も新しい REDO レコード。ユーザーがデータベースに変更を加えると、最後尾は前に移動していく。最新のチェックポイントは、必ず一時的にログの最後尾の後に来るので、このことをログの最後尾の後送りと呼ぶ。チェックポイントによってログの最後尾の遅れが大きい場合は、回復の時間が延びる。

論理バックアップ (logical backups)

Export ユーティリティが SQL を使用してデータベースのデータを読み込み、このデータを O/S レベルのバイナリ・ファイルにエクスポートするバックアップ方法。このデータを再びデータベースにインポートするには、Import ユーティリティを使用する。

Export ユーティリティによるバックアップは、RMAN バックアップとは次のような点で異なる。

- データベースの論理オブジェクトのエクスポートは、論理オブジェクトを含んだファイルとは別個に行われる。
- 論理バックアップは、別のデータベース、さらに別のプラットフォームへもインポートできる。RMAN バックアップは、データベース間、プラットフォーム間で、移動は不可能。

物理バックアップ (physical backups) も参照。

A

- ABORT オプション
 - SHUTDOWN コマンド, 3-3, 14-16, 14-17, 14-26, 15-12
- ALERT ファイル, 15-15
 - RESETLOGS 後のチェック, 14-33
- allocate channel コマンド (RMAN), 11-9
 - 削除オプション, 8-26, 11-13
 - メンテナンス・オプション, 11-13
- ALTER DATABASE コマンド
 - BACKUP CONTROLFILE TO TRACE オプション, 13-12
 - BACKUP CONTROLFILE オプション, 3-6
 - CLEAR LOGFILE GROUP オプション, 16-17
 - CLEAR UNARCHIVED LOGFILE オプション, 2-14, 16-25
 - CREATE DATAFILE オプション, 14-5
 - DATAFILE OFFLINE DROP 句, 16-27
 - DATAFILE ONLINE オプション, 14-30
 - ENABLE THREAD オプション, 16-25
 - MOUNT STANDBY DATABASE オプション, 16-6, 16-29
 - NORESETLOGS オプション, 13-14, 14-32
 - OPEN READ ONLY オプション, 16-19, 16-20, 16-21
 - OPEN RESETLOGS オプション, 6-7
 - RECOVER AUTOMATIC LOGFILE オプション, 14-14
 - RECOVER ...FROM オプション, 14-7, 14-11
 - RECOVER LOGFILE オプション, 14-13
 - RENAME DATABASE 句, 9-9
 - RESETLOGS オプション, 13-14, 14-17, 14-18, 14-32
- alter database コマンド (RMAN), 11-15
- ALTER SYSTEM コマンド
 - ARCHIVE ALL オプション
 - ~を使用したオンライン REDO ログのアーカイブ, 3-5
 - ARCHIVE LOG CURRENT オプション, 16-4, 16-21
 - RESUME オプション, 13-10
 - SUSPEND オプション, 13-9
- ALTER TABLESPACE コマンド
 - BEGIN BACKUP オプション, 13-6, 13-9
 - BEGIN/END BACKUP オプション, 6-28
 - END BACKUP オプション, 13-9
- ARCHIVE LOG CURRENT オプション
 - ALTER SYSTEM コマンド, 16-4, 16-21
- archivelogRecoverSpecifier 句 (RMAN), 11-17
- ARCHIVELOG モード, 2-16
 - アーカイブ, 2-15
 - 概要, 1-10
 - データ・ファイルの消失, 15-3
 - ~での運用, 2-16
 - ~でのバックアップ計画, 3-17
 - ~の定義, 2-16
 - バックアップ・オプション, 3-21
 - 分散データベース, 3-22
 - 利点, 2-16
 - ~を使用したバックアップ計画, 3-17
- AS SELECT オプション
 - CREATE TABLE コマンド, 14-4
- AUTORECOVERY オプション
 - SET コマンド, 14-12
- available オプション RMAN
 - change コマンド, 6-8

B

BACKUP CONTROLFILE TO TRACE オプション
 ALTER DATABASE コマンド, 13-3, 13-12
BACKUP CONTROLFILE オプション
 ALTER DATABASE コマンド, 3-6, 13-3
backup files
 ユーザー作成、カタログ化, 6-28
Backup Solutions Program (BSP), 4-19
 Legato Storage Manager (LSM), 4-19
BACKUP_TAPE_IO_SLAVES 初期化パラメータ, 8-23
backup コマンド (RMAN), 11-21, 4-28, 8-2, 8-27
 format パラメータ, 4-31
 proxy only オプション, 4-18
 skip offline オプション, 8-16
 プロキシ・オプション, 4-18
BEGIN BACKUP オプション
 ALTER TABLESPACE コマンド, 13-6
BEGIN/END オプション
 ALTER TABLESPACE コマンド, 6-28
BSP
 Backup Solutions Program を参照

C

CASE1.RCV サンプル・スクリプト
 バックアップ・ピースのサイズ制限の設定, 5-20
CATALOG.SQL スクリプト, 6-2
catalog コマンド (RMAN), 6-5, 6-28, 11-32
CATPROC.SQL スクリプト, 6-2
CATRMAN.SQL スクリプト, 6-5
change コマンド (RMAN), 6-10, 11-35
 available オプション, 6-8
 delete オプション, 6-29
 unavailable オプション, 6-8
CLEAR LOGFILE GROUP オプション
 ALTER DATABASE コマンド, 16-17
CLEAR UNARCHIVED LOGFILE オプション
 ALTER DATABASE コマンド, 16-25
COMPATIBLE パラメータ, 16-26
 スタンバイ・データベース, 16-5
connect コマンド (RMAN), 11-44, 11-46
CONTROL_FILES 初期化パラメータ, 2-9, 9-11, 15-12, 16-25
copy コマンド (RMAN), 11-48
create catalog コマンド (RMAN), 11-52
CREATE CONTROLFILE コマンド

 および回復, 9-16
 スタンバイ・データベースへの影響, 16-25
CREATE DATAFILE オプション
 ALTER DATABASE コマンド, 14-6
create script コマンド (RMAN), 11-54
CREATE TABLESPACE コマンド, 15-3
CREATE TABLE コマンド
 AS SELECT オプション, 14-4
crosscheck コマンド (RMAN), 11-57

D

DATAFILE ONLINE オプション
 ALTER DATABASE コマンド, 14-30
DB_FILE_NAME_CONVERT 初期化パラメータ, 4-53
RMAN 複製コマンドで使う, 10-7
DB_FILE_STANDBY_NAME_CONVERT 初期化パラメータ, 16-26
 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, 16-25
DB_FILES 初期化パラメータ, 16-26
DBA_DATA_FILES ビュー, 13-6, 13-10
DBVERIFY ユーティリティ, 13-5
DB 識別子, 6-5
debug コマンド (RMAN), 11-62
delete expired backupset コマンド (RMAN), 11-63
delete script コマンド (RMAN), 11-65, 11-66
drop catalog コマンド (RMAN), 11-68
duplex パラメータ (RMAN)
 set コマンド, 8-23
duplicate コマンド (RMAN), 11-69, 4-52

E

ENABLE THREAD オプション
 ALTER DATABASE コマンド, 16-25
Enterprise Backup Utility, 3-11
Export ユーティリティ, 3-11, 3-15
 エクスポートの種類, 13-17
 バックアップ, 3-15, 3-23, 13-16
 バックアップと~, 3-23
 読取り一貫性, 13-16

F

format パラメータ (RMAN)
 backup コマンド, 4-31

H

host コマンド (RMAN), 11-74

I

Import コーティリティ, 13-16

 使用手順, 13-18

 データベース回復, 13-18

incarnation of database オプション RMAN

 list コマンド, 6-7

incarnation オプション RMAN

 reset database コマンド, 6-7

INIT.ORA ファイル, 4-49, 9-15

I/O エラー

 バックアップへの影響, 4-36

L

listObjList, 11-85

listObjList 句 (RMAN), 11-85

list コマンド (RMAN), 4-20, 11-76

 incarnation of database オプション, 6-7

LOG_ARCHIVE_DEST_*n* 初期化パラメータ, 2-19, 14-7, 14-10, 16-6, 16-10

 REOPEN オプション, 16-11

LOG_ARCHIVE_DEST 初期化パラメータ, 14-7, 14-10, 14-27, 14-30

 指定する、～を使用して宛先を, 16-10

 ～を使用した宛先の指定, 2-19

LOG_ARCHIVE_DEST パラメータ, 16-6

LOG_ARCHIVE_DUPLEX_DEST 初期化パラメータ

 指定する、～を使用して宛先を, 16-10

 ～を使用した宛先の指定, 2-19

LOG_ARCHIVE_FORMAT 初期化パラメータ, 14-7, 14-10

LOGSOURCE 変数

 SET コマンド, 14-7, 14-11

M

MOUNT STANDBY DATABASE オプション

 ALTER DATABASE コマンド, 16-6, 16-29

MOUNT オプション

 STARTUP コマンド, 14-27, 14-30

N

newname for datafile オプション (RMAN)

 set コマンド, 9-9

NLS_DATE_FORMAT 環境変数, 5-2, 8-20, 9-23

NLS_LANG 環境変数, 5-2, 8-20, 9-23

NOARCHIVELOG モード

 アーカイブ, 2-15

 回復, 14-16

 概要, 1-10

 定義, 2-15

 データ・ファイルの消失, 15-2

 ～での運用, 2-15

 ～でのクローズ状態の非一貫性バックアップ, 3-5

 ～でのバックアップ計画, 3-16

 ～のバックアップ, 8-24

 バックアップ・オプション, 3-21

 不利な点, 14-16

 分散データベースのバックアップ, 3-22

NORESETLOGS オプション

 ALTER DATABASE コマンド

 制御ファイルのバックアップ, 13-14

O

OPEN READ ONLY 句

 ALTER DATABASE コマンド, 16-19, 16-20, 16-21

OPEN RESETLOGS オプション

 ALTER DATABASE コマンド, 6-7

ORA-01578 のエラー・メッセージ, 14-5

Oracle Enterprise Manager, 3-10

Oracle8i コーティリティ

 Recovery Manager, 4-2

O/S コピーのカタログ化, 8-25

O/S のバックアップ, 13-3 ~ 13-18

O/S コーティリティ

 ～によるファイルのコピー, 8-25

P

PARALLEL_MAX_SERVERS 初期化パラメータ, 14-19

PARALLEL 句

 RECOVER コマンド, 14-19

Point-in-Time 回復

 表領域, A-2 ~ A-14, B-1 ~ B-26

print script コマンド (RMAN), 11-87

proxy only オプション (RMAN)

backup コマンド , 4-18

R

RECOVER AUTOMATIC LOGFILE オプション

ALTER DATABASE コマンド , 14-14

RECOVER DATAFILE コマンド , 14-9

RECOVER ...FROM オプション

ALTER DATABASE コマンド , 14-7 , 14-11

RECOVER TABLESPACE コマンド , 14-9

Recovery Manager

PL/SQL ジョブ・ステップ , 4-6

RMAN コマンドの使用方法 , 4-5

sbttest プログラム , 4-19 , 5-20

イメージ・コピー・バックアップ , 4-45

エラー , 4-12

回復 , 9-15

完全 , 9-17

データベースをオープン , 9-21

バックアップ制御ファイル , 9-19

不完全 , 9-23

メディア全体障害の後の , 9-35

概要 , 4-2 , 4-4

構文規則 , 11-2

コマンド

allocate channel , 11-9

allocate channel for delete , 8-26

alter database , 11-15

backup , 4-18 , 4-28 , 8-2 , 8-27

catalog , 6-5 , 6-28

change , 6-10 , 6-29

connect , 11-44

copy , 11-48

create catalog , 11-52

create script , 11-54

crosscheck , 11-57

debug , 11-62

delete expired backup , 11-63

delete script , 11-65

drop catalog , 11-68

duplicate , 11-69

execute script , 6-21

host , 11-74

list , 6-7

print script , 11-87

recover , 4-49 , 9-15

register , 6-6

release channel , 11-96

replace script , 11-98

replicate controlfile , 9-11

report , 7-5

reset database , 6-7

restore , 9-16

resync catalog , 6-34

rman , 11-122

run , 11-125

send , 5-20

set , 8-23 , 9-9

shutdown , 11-138

sql , 11-141

startup , 11-143

switch , 11-145

upgrade catalog , 11-149

validate , 11-151

概要 , 4-5

コマンド・ファイルの使用 , 4-10

ジョブ , 4-9

ジョブ・コマンド , 4-8

スタンドアロン , 4-9

スタンドアロン・コマンド , 4-8

対話形式の使用方法 , 4-9

コマンド行引数 , 4-9

コマンドのコンパイルおよび実行 , 4-6

コマンドの対話的な使用方法 , 4-9

コマンドの日付 , 5-2

サンプル・スクリプト , 5-21

時間を指定するパラメータ , 5-2

ストアド・スクリプト , 4-10

整合性チェック , 4-53

制約

バックアップ , 4-44

復元 , 4-48

接続

パスワード・ファイルを使用した , 5-9 , 5-11

パスワード・ファイルを使用しない , 5-10

複製データベース , 5-12

リカバリ・カタログなしでの ~ , 5-9 , 5-10

リカバリ・カタログを使用した , 5-10 , 5-11

接続オプション , 5-8

切断、~からの , 5-12

増分バックアップ

差分 , 4-40

累積 , 4-42

レベル 0 , 4-39

- 多重化
 - データ・ファイル, 4-32
- チャンネル制御
 - 概要, 4-24
- データベース・キャラクタ・セット, 5-3
- データベースの登録, 6-6
- データベース情報のリセット, 6-7
- ～での分裂ブロックの検出, 4-55
- 名前の変換, 4-6
- ～によるバックアップのためのユーザー・タグ, 4-46
- パスワード・ファイル, 5-2
- 破損データ・ファイル・ブロック, 4-54
 - I/O エラーと～の処理, 4-36
- バックアップ, 8-2
 - アーカイブ REDO ログ, 8-8
 - イメージ・コピー, 4-45
 - 制御ファイル, 8-6, 8-7
 - 増分, 8-9
 - タグの使用法, 4-46
 - データ・ファイル, 8-4, 8-5, 8-6
 - データベース全体, 8-3
 - 表領域, 8-4, 8-5, 8-6
 - 分裂ブロックの検出, 4-55
- バックアップ・セット, 4-31
- バックアップのタイプ, 4-28, 4-45
 - 二重バックアップ・セット, 4-33
- バックアップの平行化, 4-34
- バックアップ・ピース, 4-31
- バックアップ方法, 3-9
- 表領域の Point-in-Time 回復, 4-51
- 不完全回復
 - リカバリ・カタログなしで, 9-27
 - リカバリ・カタログを使って, 9-23
- 復元, 9-2
 - アーカイブ REDO ログ, 9-13
 - 制御ファイルを新しい位置に, 9-11, 9-12
 - デフォルトの場所にデータベースを, 9-3
- 複製データベースの作成, 10-2
- メタデータ, 4-12, 6-1
 - 制御ファイルへの格納, 4-15
- メディア管理
 - Backup Solutions Program (BSP), 4-19
 - 一意的なファイル名, 5-19
 - クロスチェック, 4-17
 - 最大ファイル・サイズ, 5-20
 - テスト, 4-19
 - ファイルのバックアップ, 4-17
 - ファイルの復元, 4-17
 - プロキシ・コピー, 4-18
 - メディア・マネージャ、とのリンク, 5-19
 - 要件, 5-19
- メンテナンスのためにチャンネルを割り当てる, 11-13
- リカバリ・カタログ, 4-13
 - アップグレード, 6-34
 - クロスチェック, 6-9
 - 再同期化, 6-23
 - 消失または破損した～の回復, 6-34
 - 使用・不使用の決定, 5-4
 - スキーマ, 6-2
 - スキーマの変更後の更新, 6-26
 - スナップショット制御ファイル, 4-14
 - データベースの登録, 6-5
 - 問合せ, 4-20, 7-2
 - ～なしでの運用, 4-15
 - ～のサイズの管理, 6-27
 - ～のバックアップ, 6-31
 - バックアップおよびコピーの可用性の変更, 6-8
 - 別のデータベースの作成, 5-5
 - レコードの削除, 6-17
 - ～を使用した運用, 4-13
 - ～を使用していない場合の制御ファイルの消失, 4-15
- リカバリ・カタログのクロスチェック, 6-9
- リスト, 7-2
- レポート, 7-5
 - データベース・スキーマ, 7-9
 - バックアップが必要なオブジェクト, 7-5
 - 不要なバックアップ, 7-6, 7-8
 - ～を使用したデータ・ファイルの復元, 4-47
 - ～を使用する利点, 4-4
 - スナップショット制御ファイルの位置, 5-3
- Recovery Manager のスタンドアロン・コマンド, 4-8
- RECOVERY_PARALLELISM 初期化パラメータ, 14-19
- RECOVER コマンド, 14-9
 - FROM 'location' オプション, 16-29
 - MANAGED STANDBY DATABASE オプション, 16-8, 16-18
 - PARALLEL 句, 14-19
 - UNTIL TIME オプション, 14-30
 - 回復不能オブジェクトとスタンバイ・データベース, 14-5
- recover コマンド (RMAN), 4-49, 9-15, 11-89

REDO ログ・ファイル, 1-8
アーカイブ済み, 2-15
内容, 2-15
利点, 2-14
概要, 1-8
グループ, 2-13
メンバー, 2-13
消去, 2-14
多重
一部のメンバーにアクセスできない場合, 2-14
グループ, 2-13
すべてにアクセスできない場合, 2-14
ダイアグラム, 2-13
目的, 1-9
バックアップのためのファイルのリスト表示, 13-2
バックアップの「ファジー」データと~, 3-17
ミラー化した~
ログ・スイッチと~, 2-14
命名, 14-10
メンバー, 2-13
制御ファイルに指定されているファイル, 2-3
多重, 2-12
REDO ログ・ファイルの消去, 2-14
register コマンド (RMAN), 6-5, 6-6, 11-94
release channel コマンド (RMAN), 11-96
メンテナンス・チャネルの解放, 11-97
RENAME DATABASE 句
ALTER DATABASE コマンド, 9-9
REOPEN オプション
LOG_ARCHIVE_DEST_n 初期化パラメータ, 16-11
replace script コマンド (RMAN), 11-98
replicate コマンド (RMAN), 9-11, 11-101
report コマンド (RMAN), 4-22, 11-103
need backup オプション, 7-5
reset, 11-111
reset database コマンド (RMAN), 6-7, 11-111
incarnation オプション, 6-7
RESETLOGS オプション
ALTER DATABASE コマンド, 14-17, 14-18, 14-32
REDO ログの複数の時間の流れ, 3-24
~ 使用後のデータベースのバックアップ, 3-20
制御ファイルのバックアップ, 13-14
~ 使用後のデータベースの回復, 14-34
restore コマンド (RMAN), 9-16, 11-113
RESUME オプション
ALTER SYSTEM コマンド, 13-10

resync catalog コマンド (RMAN), 4-14, 6-23, 11-119
from controlfilecopy オプション, 6-34
RFS
「リモート・ファイル・サーバー」を参照
RMAN
Recovery Manager を参照
rman コマンド (RMAN), 11-122
RMAN コマンドのコンパイルおよび実行, 4-6
RMAN コマンドの名前の変換, 4-6
RMAN のメタデータの管理, 6-1
RMAN を使用したリストの生成, 4-20
run コマンド (RMAN), 11-122, 11-125

S

sbttest プログラム, 4-19, 5-20
SCN
システム変更番号を参照
SCN(システム変更番号)
定義, 1-9
send コマンド (RMAN), 5-20, 11-128
SET コマンド
AUTORECOVERY オプション, 14-12
LOGSOURCE 変数, 14-7, 14-11
set コマンド (RMAN), 11-130
duplex パラメータ, 8-23
maxcorrupt for datafile オプション, 8-27
newname パラメータ, 4-53, 9-9
run の中で実行, 11-134
SHUTDOWN コマンド
ABORT オプション, 13-7, 14-16, 14-17, 14-26, 15-12
一貫性のある全体データベース・バックアップ, 3-3
shutdown コマンド (RMAN), 11-138
skip offline オプション (RMAN)
backup コマンド, 8-16
skip readonly オプション (RMAN)
backup コマンド, 8-16
SQL*Plus
ログ・ファイルの適用, 14-12
SQL コマンド
表領域の回復用, 14-13
ログ・ファイルの適用, 14-13
sql コマンド (RMAN), 11-141
STANDBY_ARCHIVE_DEST 初期化パラメータ, 16-26

STARTUP コマンド

MOUNT オプション, 14-27, 14-30

startup コマンド (RMAN), 11-143

SUSPEND オプション

ALTER SYSTEM コマンド, 13-9

switch コマンド (RMAN), 11-145

SYSTEM 表領域

回復, 14-23

SYS が所有するオブジェクト

および TSPITR、RMAN の使用方法, A-6

T

TNSNAMES.ORA ファイル, 16-10

TS_PITR_CHECK ビュー, A-6

TSPITR。「表領域の Point-in-Time 回復」を参照。

U

unavailable オプション (RMAN)

change コマンド, 6-8

UNRECOVERABLE 操作

~後のバックアップ, 16-28

~後のバックアップ実行, 3-20

UNTIL TIME オプション

RECOVER コマンド, 14-30

until 句 (RMAN), 11-42, 11-147

upgrade catalog コマンド (RMAN), 11-149

USING BACKUP CONTROLFILE オプション

RECOVER コマンド, 14-27

V

VSARCHIVE ビュー, 2-4, 2-12, 2-18

V\$BACKUP_CORRUPTION ビュー, 4-36

V\$BACKUP ビュー, 13-8

V\$DATABASE ビュー, 2-4, 2-18

V\$DATAFILE ビュー, 14-30, 16-28

バックアップのためのファイルのリスト表示, 13-2

V\$LOG_HISTORY ビュー, 14-7

V\$LOGFILE ビュー, 15-5, 15-6

バックアップのためのファイルのリスト表示, 13-2

V\$LOG ビュー, 2-4, 2-12, 2-18

アーカイブ状態の表示, 2-4, 2-12, 2-18

V\$LONGOPS ビュー, 4-36

V\$PROXY_ARCHIVEDLOG ビュー, 4-18

V\$PROXY_DATAFILE ビュー, 4-18

V\$RECOVER_FILE ビュー, 1-24, 14-3

回復するファイルの選択, 3-28

V\$RECOVERY_LOG ビュー, 14-7

validate コマンド (RMAN), 11-151

あ

アーカイブ

ALTER SYSTEM ARCHIVE ALL, 3-5

宛先の状態, 16-11

オープン・バックアップ後の~, 3-5

クローズ状態の非一貫性バックアップ後の~, 3-5

失敗した宛先への, 16-11

~の情報の表示, 2-4, 2-18

不利な点, 2-15

利点, 2-15

アーカイブ REDO ログ, 2-14, 2-17, 2-21

ALTER SYSTEM ARCHIVE ALL コマンド, 3-5

RMAN を使った復元, 9-13

宛先

再アーカイブ、失敗した位置への, 16-11

宛先の状態, 16-11

回復後の削除, 14-8

回復時の位置, 14-10

回復中のエラー, 14-15

コピー、リスト, 7-2

消失, 15-10

状態情報, 2-4, 2-18

多重化, 2-19, 16-10

ディスクへの復元, 14-7

転送, 2-21, 16-7

登録, 6-5

~の宛先の指定, 2-19

~のスタンバイ転送, 2-21

~のバックアップ, 8-19

~の標準転送, 2-21

メディア回復時の適用, 14-10, 14-13

アーカイブ REDO ログの宛先の状態, 16-11

アーカイブ REDO ログの転送, 2-21, 16-7

標準転送モードでの~, 2-21

アーカイブ・ログのバックアップ

RMAN を使用した, 8-8

アクティブ・オンライン REDO ログ

グループの消失, 15-9

アクティブでないオンライン REDO ログ

消失, 15-7

宛先の指定

アーカイブ REDO ログの～, 2-19
アプリケーション・エラー
定義, 1-11

い

一貫性のある全体データベース・バックアップ, 3-3, 13-4
一貫性のない全体データベース・バックアップ
定義, 3-4
一貫性バックアップ
定義, 1-16
イメージ・コピー, 4-45
クロスチェック, 6-12
～の復元のテスト, 6-18
インスタンス
回復, 1-18
障害, 1-11, 1-12

え

エクスポート
完全, 13-17
増分, 13-17
モード, 13-16
累積, 13-17
エラー
Recovery Manager, 4-12
RMAN バックアップ時の～, 8-26

お

オープン状態のデータベースのバックアップ
～中の分裂ブロックの検出, 4-55
オープン、データベース回復
RMAN を使って, 9-21
オフライン
スタンバイ・データベースのデータ・ファイルの, 16-27
バックアップ, 3-18
オンライン～
バックアップ, 3-18
オンライン REDO ログ, 1-8, 15-7
アーカイブ済みグループ, 15-5, 15-6
アクティブ・グループ, 15-5, 15-6
アクティブでないグループ, 15-5, 15-6
概要, 1-8

現行のグループ, 15-5, 15-6
消失
アクティブ・グループ, 15-9
回復, 15-5 ～ 15-10
グループ, 15-6
全メンバー, 15-6
ミラー化メンバー, 15-5
制御ファイルに記録された～, 2-3
多重, 1-9
～の意図しない復元, 3-23
～のバックアップ, 2-11, 3-23
バックアップのためのログ・ファイルのリスト表示, 13-2
複数のグループの消失, 15-10
メディア回復時の適用, 14-10
メンバーのステータス, 15-5, 15-6

か

ガイドライン
バックアップ
Export ユーティリティ, 3-23
OPEN RESETLOGS 後の全体データベース・バックアップ, 3-20
回復不能操作, 3-20
計画のテスト, 3-24
構造変更, 3-19
使用回数の多い表領域, 3-19
頻度, 3-18
ファイルの多重化, 3-18
古いバックアップの保管, 3-21
分散データベースの制約, 3-22

回復

Import ユーティリティ, 13-18
PARALLEL_MAX_SERVERS 初期化パラメータ, 14-19
RMAN を使い、データベースをオープンする, 9-21
RMAN を使って, 9-15
RMAN を使い完全な, 9-17
RMAN を使って障害時に, 9-35
RMAN を使って不完全な, 9-23
インスタンス, 1-18
クラッシュ, 1-18
コマンド, 14-2 ～ 14-9
時間ベースの～, 3-29, 14-28 ～ 14-31
消失または破損したリカバリ・カタログの～, 6-34

使用するプロセス数の設定, 14-19
取消しベースの~, 14-20, 14-23, 14-25, 14-28
~のためのパラレル処理, 14-19
バックアップ制御ファイルを使っての (RMAN),
9-19
分散データベース
スナップショットと, 15-16
変更ベースの~, 14-31 ~ 14-34
方針
回復が必要なものの判断, 3-28
メディア回復が必要となるタイミングの判断,
3-31
マルチスレッド, 14-13
メディア, 14-1, 15-1
完全, 3-28
不完全, 3-29
メディア回復も参照。
読取り専用表領域, 14-5
回復計画
概要, 1-17 ~ 1-24
回復した表
~のバックアップ, B-14
回復セット
RMAN TSPITR の, A-4
すべての表が入っている, A-6
プライマリ・データベースへのインポート, B-14
プライマリ・データベースへのコピー, B-13
回復不能オブジェクト
および RECOVER 操作, 14-5
回復不能操作後のバックアップ実行, 3-20
環境変数
NLS_DATE_FORMAT, 5-2
NLS_LANG, 5-2
完全エクスポート, 13-17
完全回復
RMAN を使って, 9-17
定義, 1-19, 3-28
手順, 14-19

き

キーワード
構文図の, 11-2
危険なバックアップ手段の回避, 3-23
キャラクタ・セット
RMAN で使用するための設定, 5-3

く

クラッシュ回復
定義, 1-18
グループ
アーカイブ REDO ログ, 15-5, 15-6
オンライン REDO ログ, 15-5, 15-6
クローズ状態のバックアップ, 3-4
クローン・データベース
TSPITR 用に準備する, B-11
エクスポート, B-13
オープン, B-13
回復, B-12
パラメータ・ファイルの準備, B-10
クロスチェック
定義, 4-17
リカバリ・カタログとメディア・マネージャの~,
6-9

け

警告
一貫性と Export バックアップ, 13-16
最初のバックアップのアーカイブ・モード, 3-17

こ

構文規則
Recovery Manager, 11-2
構文図
キーワード, 11-2
パラメータ, 11-3
の説明, 11-2
コールド・バックアップ, 3-4
全体データベース・バックアップ, 13-3
コマンド、Recovery Manager の
allocate, 11-9
allocate channel for delete, 8-26
alter database, 11-15
archivelogRecoverSpecifier 句 (RMAN), 11-17
backup, 8-2, 8-27, 11-21
proxy only オプション, 4-18
skip offline オプション, 8-16
プロキシ・オプション, 4-18
catalog, 6-5, 6-28, 11-32
change, 6-10, 11-35
delete オプション, 6-29

connect , 11-44 , 11-46
copy , 11-48
create catalog , 11-52
create script , 11-54
crosscheck , 11-57
debug , 11-62
delete expired backupset , 11-63
delete script , 11-65 , 11-66
drop catalog , 6-35 , 11-68
duplicate , 4-52 , 11-69
execute script , 6-21
host , 11-74
list , 4-20 , 11-76
 incarnation of database オプション , 6-7
listObjList 句 , 11-85
print script , 11-87
recover , 4-49 , 9-15 , 11-89
resync catalog
 from controlfilecopy オプション , 6-34
register , 6-6 , 11-94
release channel , 11-96
release channel (メンテナンス・タイプの) , 11-97
replace script , 11-98
replicate , 9-11 , 11-101
replicate controlfile , 9-11
report , 4-22 , 11-103
 need backup オプション , 7-5
reset database , 6-7 , 11-111
 incarnation オプション , 6-7
restore , 11-113
復元 , 9-16
resync catalog , 4-14 , 6-23 , 11-119
rman , 11-122
run , 11-125
send , 5-20 , 11-128
set , 11-130
 maxcorrupt for datafile オプション , 8-27
 newname for datafile オプション (RMAN) , 9-9
set (run コマンド内) , 11-134
shutdown コマンド , 11-138
sql , 11-141
startup , 11-143
switch , 11-145
until , 11-42 , 11-147
upgrade catalog , 6-34 , 11-149
validate , 11-151
概要 , 4-5 , 11-5

 を割り当てる , 11-13
コマンド、SQL
 ALTER DATABASE , 14-7 , 14-11 , 14-13 , 16-17 ,
 16-25
 CREATE CONTROLFILE , 16-25
コマンド、SQL*Plus
 RECOVER , 14-9
 UNTIL TIME オプション , 14-30
 RECOVER DATAFILE , 14-9
 RECOVER TABLESPACE , 14-9
 SET , 14-7 , 14-11 , 14-12
コマンド行
 RMAN の引数 , 4-9
コマンド・ファイル
 Recovery Manager , 4-10
コマンド例
 説明 , 11-3
コマンド、Recovery Manager の
 backup , 4-28
 set
 duplex パラメータ , 8-23
コマンド行パラメータ (RMAN) , 11-39

さ

災害時回復
 ～ のためのスタンバイ・データベースの使用方法 ,
 16-1 ~ 16-29
削除された表
 回復 , 15-14
作成
 スタンバイ・データベース , 16-3
サスペンド / 再開モード , 13-9
差分増分バックアップ , 4-40
サンプル・スクリプト
 RMAN , 5-21

し

時間ベースの回復 , 14-28 ~ 14-31
 定義 , 3-29
 分散データベース内の調整式の , 15-15
時間ベースの調整式回復
 分散データベース , 15-15
時間を指定するパラメータ
 Recovery Manager で使用するための設定 , 5-2
時刻書式

- RECOVER DATABASE UNTIL TIME 文, 14-30
- システム時間、変更
 - 回復への影響, 14-25
- システム障害, 1-11
- システム変更番号 (SCN)
 - 定義, 1-9
 - 分散回復での使用方法, 15-15
- 持続回復モード, 16-5
- 手動回復モード, 16-5
- 障害
 - インスタンス, 1-12
 - システム, 1-11
 - タイプ, 1-11
 - ディスク, 1-12
 - ハードウェア, 1-11
 - プロセス, 1-11
 - 文, 1-11
 - メディア, 1-12
 - 多重オンライン REDO ログ・ファイル, 2-12
 - ユーザー・エラー, 1-12
- 消失
 - アクティブでないログ・グループ, 15-7
- 使用例、Recovery Manager の
 - NOARCHIVELOG バックアップ, 8-24
 - OPS バックアップ, 8-25
 - O/S コピーのカタログ化, 8-25
 - RESETLOG 以前のバックアップの回復, 9-37, 9-38
 - アーカイブ REDO ログのバックアップ, 8-19
 - 回復不能データ・ファイルのレポート, 7-11
 - 増分バックアップ, 8-21
 - データベースが名前を共有する場合の復元, 9-30
 - データベース・スキーマのレポート, 7-13
 - 廃止バックアップおよびコピーの削除, 7-12
 - 廃止バックアップおよびコピーのリスト表示, 7-10
 - バックアップ・エラーの処理, 8-26
 - バックアップおよびコピーのメンテナンス, 8-26
 - バックアップおよびコピーのリスト表示, 7-10
 - バックアップ・セットのサイズ設定, 8-17
 - バックアップ・セットの二重化, 8-22
 - バックアップの平行化, 8-23
 - バックアップの必要なデータ・ファイルのレポート, 7-11, 8-15
 - 不要なバックアップのレポート, 7-11
 - メディア全体障害の後の回復, 9-35
 - 累積増分バックアップ, 8-22
- 初期化パラメータ
 - BACKUP_TAPE_IO_SLAVES, 8-23

- COMPATIBLE, 16-26
- CONTROL_FILES, 9-11, 15-12, 16-25
 - 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, 16-25
- DB_FILE_STANDBY_NAME_CONVERT, 16-26
 - 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, 16-25
- DB_FILES, 16-26
- LOG_ARCHIVE_DEST, 2-19, 14-7, 14-10, 14-27, 14-30, 16-6
- LOG_ARCHIVE_DEST_n, 2-19, 14-7, 14-10, 16-6, 16-10
- LOG_ARCHIVE_DEST_STATE_n, 16-11
- LOG_ARCHIVE_DUPLEX_DEST, 2-19
- LOG_ARCHIVE_FORMAT, 14-7, 14-10
- PARALLEL_MAX_SERVERS, 14-19
- RECOVERY_PARALLELISM, 14-19
- STANDBY_ARCHIVE_DEST, 16-26
 - 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, 16-25
- ジョブ, 4-8

す

- スキーマ変更
 - リカバリ・カタログの更新, 6-26
- スクリプト
 - CATRMAN.SQL, 6-5
- スタンバイ・データベース, 16-2 ~ 16-29
 - REDO ログ・ファイル
 - 変更, 16-24
 - アーカイブ REDO ログ
 - 転送, 16-7
 - アーカイブ REDO ログの転送, 16-7
 - メンテナンス, 16-6
 - 作成, 16-2
 - 手順, 16-3
 - 持続回復モード, 16-5
 - 手動回復モード
 - スタンバイ・データベース, 16-5
 - 手順, 16-6
 - 初期化パラメータ, 16-25
 - 制御ファイル
 - リフレッシュ, 16-29
 - ダイレクト・パス操作, 16-27
 - 定義, 16-2
 - データ・ファイル

- オフライン化, 16-27
- 改名, 16-24
- プライマリ・データベースの変更の影響, 16-22
- 変更、制御ファイルの, 16-25
- モード, 16-5
- 読取り専用モード, 16-5, 16-19
- 利点, 16-2
- ストアド・スクリプト
 - Recovery Manager, 4-10
 - 置換え, 6-22
 - 管理, 6-21
 - 削除, 6-22
 - 作成, 6-21
 - 実行, 6-22
 - 出力, 6-23
 - 全～の一覧表示, 6-23
- スナップショット
 - 分散データベースの回復, 15-16
 - メディア回復と～, 15-16
- スナップショット制御ファイル, 4-14
 - 位置の指定, 5-3

せ

- 制御ファイル, 2-3 ~ 2-8
 - CONTROL_FILES 初期化パラメータ
 - 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, 16-25
 - CONTROL_FILES の初期化パラメータの設定, 2-9
 - RMAN を使ったの復元, 9-11, 9-12
 - 概要, 1-7
 - 更新, 2-5
 - 時間ベースの回復, 14-29
 - 消失, 15-11, 15-12
 - スタンバイ・データベースのリフレッシュ, 16-29
 - スタンバイ・データベースへの影響, 16-25
 - スナップショット
 - ～の位置の指定, 5-3
 - 多重, 2-7, 2-8, 2-9
 - 定義, 1-7
 - 内容, 1-7, 2-3
 - ～の数, 2-9
 - ～のバックアップ, 13-3, 13-11
 - バックアップおよび回復, 9-19
 - バックアップの保持, 2-7, 2-9
 - ファイル名の検索, 13-3
 - 不完全回復, 14-26

- ミラー化した～, 2-9
 - 消失, 15-11
 - 目的, 1-7
 - リカバリ・カタログの代替として使用, 4-15
- 制御ファイルのバックアップ
 - RMAN を使用した, 8-6, 8-7
 - 定義, 3-8
- 整合性チェック, 4-53
- 制約
 - 復元, 4-48
- 接続オプション
 - Recovery Manager, 5-8
- 切断
 - Recovery Manager からの, 5-12
- 全体データベース・バックアップ, 13-3 ~ 13-5
 - ARCHIVELOG モード, 13-3
 - NOARCHIVELOG モード, 13-3
 - RMAN を使用した, 8-3
 - 一貫性のある～, 3-3
 - SHUTDOWN ABORT コマンドの使用, 3-3
 - バックアップ制御ファイルと～, 3-3
 - 一貫性のない～, 3-4
 - 回復、～からの, 14-16
 - 作成の準備, 13-4
 - 定義, 3-2
- 全体バックアップ, 4-38

そ

- 増分エクスポート, 13-17
- 増分バックアップ, 8-21
 - RMAN を使用した, 8-9
- 差分, 4-40

た

- ダイレクト・パス操作
 - スタンバイ・データベース, 16-27
- 多重化
 - Recovery Manager を使用したデータ・ファイルの～, 4-32
 - REDO ログ・ファイル, 2-12
 - グループ, 2-13
 - アーカイブ REDO ログ, 2-19, 16-10
 - 制御ファイル, 2-7, 2-8, 2-9
 - ミラー化との比較, 2-7

ち

チャネル

～のバラレル化, 4-26

割当て, 4-24

チャネル制御

RMAN の概要, 4-24

中断後の回復の再開, 14-16

て

ディスク障害, 1-12

データ・ディクショナリ

ビュー, 13-6, 13-10

データ・ファイル, 1-6

RMAN を使ったの復元, 9-2

回復

RMAN を使って, 9-15

ガイドライン, 4-49, 9-16

作成, 14-5

バックアップがない場合, 14-6

改名

スタンバイ・データベースへの影響, 16-24

概要, 1-6

コピー、リスト, 7-2

消失, 15-2

使用方法, 1-6

制御ファイルに指定されている～, 2-3

登録, 6-5

～のバックアップ, 3-8

Recovery Manager の使用方法, 4-28

オフライン, 13-10

バックアップが必要な～、リスト表示, 7-11, 7-13, 8-15

バックアップ・ステータスのチェック, 13-8

バックアップ、リスト, 7-2

表示

回復が必要なファイル, 14-3

バックアップ・ステータス, 13-8

復元, 4-47

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, 16-23

リスト

回復不能, 7-5

バックアップ用, 13-2

データ・ファイル・バックアップ

RMAN を使用した, 8-4, 8-5, 8-6

データベース

DB 識別子, 6-5

OPEN RESETLOGS オプション後の回復, 14-34

RMAN による複製の作成, 10-2

回復

制御ファイルの破損後, 15-12

災害時回復の計画, 16-1 ~ 16-29

サスペンド, 13-9

スタンバイ, 16-2 ~ 16-29

REDO ログ・ファイル、変更, 16-24

アーカイブ REDO ログの転送, 16-7

作成, 16-2

作成、～の手順, 16-3

持続回復モード, 16-5

手動回復モード, 16-5, 16-6

初期化パラメータ, 16-25

制御ファイル, 16-25, 16-29

ダイレクト・パス操作, 16-27

定義, 16-2

データ・ファイル、オフライン化, 16-27

データ・ファイルの改名, 16-24

メンテナンス, 16-6

モード, 16-5

読取り専用モード, 16-5, 16-19

利点, 16-2

制御ファイルに格納された名前, 2-3

全体データベース・バックアップ, 13-3, 13-5

テスト用作成, 10-2

～のバックアップ

Recovery Manager の使用方法, 4-28

バックアップのためのリスト表示, 13-2

複製の作成

リモート・ホスト上に, 10-10

物理構造, 1-6 ~ 1-10

物理構造の変更

スタンバイ・データベースへの影響, 16-22

メディア回復のシナリオ, 15-1 ~ 15-14

メディア回復の手順, 14-1 ~ 14-31

リカバリ・カタログからの～の登録解除, 6-6

リカバリ・カタログへの～の登録, 6-4, 6-5

データベース・スキーマ

レポートの生成, 7-9

データベースの構造

REDO ログ・ファイル, 1-8

オンライン REDO ログ, 1-8

概要, 1-6 ~ 1-11

データ・ファイル, 1-6

- 物理, 1-6 ~ 1-10
- ロールバック・セグメント, 1-8
- データベースのサスペンド, 13-9
- データベースの複製, 4-52
- テスト
 - バックアップ計画, 3-24
 - メディア・マネージャ, 4-19
- テスト・データベース、作成, 4-52
- テスト・データベースの作成, 4-52

と

- 問合せ
 - リカバリ・カタログ, 7-2
- 登録
 - アーカイブ REDO ログ, 6-5
 - データ・ファイル, 6-5
- トランザクション
 - コミットされていない, 1-8
 - ロールバック, 1-8
- トランSPORTABLE表領域
 - TSPITR と, B-3
 - 回復, 15-4
- 取消しベースのメディア回復
 - 定義, 3-29
 - 手順, 14-20, 14-23, 14-25, 14-28
- 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, 13-12
 - ログ・ライター・プロセスと～, 2-14

は

- パーティション表
 - 削除されたパーティション, B-18
 - 部分的な TSPITR の実行, B-15
 - 分割パーティション, B-22
- ハードウェア障害, 1-11
- パスワード・ファイル
 - ～と Recovery Manager, 5-2
 - ～を使用して Recovery Manager に接続する場合, 5-9, 5-11
 - ～を使用せずに Recovery Manager に接続する場合, 5-10
- 破損データ・ファイル・ブロック
 - RMAN と～, 4-36
 - 制御ファイルのレコード, 4-36
 - バックアップの最大値の設定, 8-27

- 破損の検出, 4-54
- バックアップ
 - ARCHIVELOG モードでの～, 3-17
 - backup コマンド (RMAN), 8-2
 - DBVERIFY ユーティリティ, 13-5
 - Enterprise Backup Utility, 3-11
 - Export ユーティリティ, 3-11, 3-15, 3-23
 - NOARCHIVELOG モードでの～, 3-16, 8-24
 - OPEN RESETLOGS オプション後の～, 3-20
 - Recovery Manager による～のタイプ, 4-28
 - Recovery Manager による～のためのユーザー・タ
グ, 4-46
 - RMAN エラーの処理, 8-26
 - UNRECOVERABLE 操作後, 16-28
 - アーカイブ REDO ログ, 8-19
 - RMAN を使用した, 8-8
 - アーカイブの効果, 2-16
 - 一貫性のある～, 1-16
 - 一貫性のあるデータベース全体, 3-3
 - 一貫性のない～, 1-16
 - NOARCHIVELOG モードでの～, 3-4
 - クローズ状態のデータベース, 3-4
 - データベース全体, 3-4
 - イメージ・コピー, 4-28, 4-45
 - オプション
 - ARCHIVELOG モード, 3-21
 - NOARCHIVELOG モード, 3-21
 - オフライン, 3-18
 - オンライン～, 3-18
 - オンライン REDO ログ, 3-23
 - ガイドライン, 3-15 ~ 3-24
 - Export ユーティリティ, 3-23
 - OPEN RESETLOGS 後の全体データベース・
バックアップ, 3-20
 - 回復不能操作, 3-20
 - 計画のテスト, 3-24
 - 構造変更, 3-19
 - 使用回数の多い表領域, 3-19
 - 頻度, 3-18
 - ファイルの多重化, 3-18
 - 古いバックアップの保管, 3-21
 - 分散データベースの制約, 3-22
 - 計画の選択, 3-15 ~ 3-24
 - 計画のテスト, 3-24
 - 避けるべき手段, 3-23
 - 重要性, 1-11
 - ステータス

- バックアップのチェック, 13-8
- ストアド・スクリプト, 6-21
- 制御ファイル, 2-7, 2-9, 3-8, 13-11
 - RMAN を使用した, 8-6, 8-7
- 全体~, 4-38
- 全体データベース・バックアップの復元, 14-16
- 増分, 8-21
 - RMAN を使用した, 8-9
 - 差分, 4-40
- タイプ, 3-2, 13-3 ~ 13-17
- 定義, 1-2
- ディスクへの~, 3-10
- データ・ファイル, 3-8
 - RMAN を使用した, 8-4, 8-5, 8-6
- データベース作成前の計画, 3-15
- データベース全体, 3-2, 13-3 ~ 13-5
 - RMAN を使用した, 8-3
 - 作成の準備, 13-4
 - バックアップ制御ファイルと~, 3-3
- データベースの構造変更時の~, 3-19
- 手順, 13-3 ~ 13-17
 - オフラインのデータ・ファイル, 13-10
 - オフライン表領域, 13-10
- 二重化, 8-22
- バックアップが必要なオブジェクトのレポート,
7-5
- バックアップ・セット, 4-28
- パラレル化, 4-34, 8-23
- パラレル・サーバー環境, 8-25
- 必要なファイルのリスト, 13-2
- 表領域, 3-19, 13-8
 - RMAN を使用した, 8-4, 8-5, 8-6
- 非累積増分, 4-40
- 頻度, 3-18
- 物理、定義, 1-2
- 分散データベース, 3-22
 - ARCHIVELOG モード, 3-22
 - NOARCHIVELOG モード, 3-22
- 保管, 3-21
- 保存, 8-26
- リカバリ・カタログ, 4-14
- 累積増分~, 4-42, 4-50, 4-53, 8-22, 16-7, 16-30
- レポートの生成, 7-2
- 論理、定義, 1-2
- バックアップ方法
 - Enterprise Backup Utility, 3-11
- バックアップ、RMAN を使用した, 8-2

- バックアップ・オプション
 - ARCHIVELOG モード, 3-21
 - NOARCHIVELOG モード, 3-21
- バックアップおよびコピーのリスト表示, 7-3
- バックアップ計画
 - 概要, 1-11 ~ 1-17
- バックアップ形式, 3-12 ~ 3-15
 - バックアップ・セット, 3-13
 - 論理バックアップ, 3-14
- バックアップ制御ファイル
 - ~を使った回復, 9-19
- バックアップ・セット
 - backup コマンドを使用した~の作成, 4-31
 - クロスチェック, 6-10
 - サイズ指定, 4-31
 - 二重化, 8-22
 - ~の復元のテスト, 6-18
 - バックアップ形式のタイプ, 3-13
 - 編成, 4-28
 - 命名, 4-30
 - リスト, 7-2
- バックアップ・セットのサイズ、設定, 4-31
- バックアップ・セットの二重化, 4-33, 8-22
- バックアップ・セットの命名, 4-30
- バックアップ・ピース
 - 命名, 4-31
- バックアップ方法, 3-8 ~ 3-11
 - Export ユーティリティ, 3-11
 - Recovery Manager, 3-9
 - オペレーティング・システム, 3-10
 - ~の比較, 3-11
- パラメータ
 - 構文図の, 11-3
- パラレル化
 - RMAN バックアップ, 4-34, 8-23
 - RMAN を使用したバックアップの~, 4-34, 8-23
 - チャンネル, 4-26
 - ~の度合いを左右する要素, 4-27
- パラレル回復, 14-19
- パラレル・サーバー
 - バックアップと~, 8-25

ひ

- 非一貫性バックアップ
 - 定義, 1-16
- 日付

- RMAN コマンドでの指定, 5-2
- ビュー
 - DBA_DATA_FILES, 13-6, 13-10
 - TS_PITR_CHECK, A-6
 - VSARCHIVE, 2-4, 2-12, 2-18
 - VSBACKUP, 13-8
 - VSBACKUP_CORRUPTION, 4-36
 - VSDATABASE, 2-4, 2-18
 - VSDATAFILE, 13-2, 14-30
 - VSLOG, 2-4, 2-12, 2-18
 - VSLOG_HISTORY, 14-7
 - VSLOGFILE, 13-2, 15-5, 15-6
 - VSLONGOPS, 4-36
 - VSPROXY_ARCHIVEDLOG, 4-18
 - VSPROXY_DATAFILE, 4-18
 - VSRECOVER_FILE, 3-28
 - VSRECOVERFILE, 14-3
 - VSRECOVERY_LOG, 14-7
- データ・ディクショナリ, 13-6, 13-10
- リカバリ・カタログ, 12-1
- 表領域
 - オープン状態のデータベースでのオフラインの回復, 14-23
 - ～のバックアップ, 3-19
 - オフライン, 13-10
 - オンライン～, 13-8
 - 頻度, 3-19
 - 読取り専用
 - 回復への影響, 14-5
 - ～のバックアップ, 8-16
- 表領域の Point-in-Time 回復, B-25
 - O/S 環境で, B-3
 - RMAN の使用方法, 4-51
 - 概要, A-2
 - 計画, A-4
 - 実行, A-10
 - すべての表が入っている回復セット, A-6
 - 制約事項, A-4
 - チューニングに関する考慮事項, A-13
 - 補助インスタンスの準備, A-7
 - 概要, B-2
 - クローン・データベース, B-4
 - 計画, B-4
 - 異なる方法, B-25
 - 実行, B-1 ~ B-26
 - 準備, B-6
 - 制限事項, B-5

- 制約事項, B-5
- チューニングに関する考慮事項, B-23
 - 回復セットの位置, B-23
 - バックアップ制御ファイル, B-24
- トランスポータブル表領域機能を使用する手順, B-25
- トランスポータブル表領域の方法, B-3, B-25
- 方法, B-3
- 要件, B-6
- 用語, B-4
- 利点, B-2
- 表領域バックアップ
 - RMAN を使用した, 8-4, 8-5, 8-6
- 非累積増分バックアップ, 4-40

ふ

- ファイル名
 - バックアップのためのリスト表示, 13-2
- 不完全回復
 - change-based (RMAN), 9-25
 - log sequence-based (RMAN), 9-26
 - RMAN を使って, 9-23
 - time-based (RMAN), 9-23
 - リカバリ・カタログなしで, 9-27
 - リカバリ・カタログを使って, 9-23
- 不完全メディア回復
 - OPS 構成内, 14-13
 - 時間ベースの～, 14-28 ~ 14-31
 - 定義, 1-19, 3-29
 - 手順, 14-25 ~ 14-34
 - バックアップ制御ファイルを使用, 14-13
 - 変更ベースの～, 14-31 ~ 14-34
- 復元
 - RMAN を使い、データベースを, 9-3
 - RMAN を使ってアーカイブ REDO ログを, 9-13
 - RMAN を使ったのデータ・ファイルの, 9-2
 - オンライン REDO ログのバックアップ, 3-23
 - 全体データベース・バックアップ, 14-16
 - データ・ファイル, 4-47
 - テスト, 6-18
 - バックアップ・ファイルの選択, 4-48
- 復元の制約, 4-48
- 復元の妥当性チェック, 6-18
- 複製データベース
 - DB_FILE_NAME_CONVERT パラメータ, 10-5
 - nofilenamecheck オプション, 10-6

- set auxname コマンド (RMAN), 10-5
- set newname コマンド (RMAN), 10-5
- 異なるファイル・システムを持つリモート・ホスト上に作成, 10-12
- 作成, 4-52, 10-2
- 制限, 10-3
- 非現行の作成, 10-19
- 表領域のスキップ, 10-5
- ファイル名の生成, 10-3
- 複製の準備, 10-7
- リモート・ホスト上に同じファイル・システムで作成, 10-10
- ローカル・ホスト上に作成, 10-16
- 複製データベースの作成, 10-2
- リモート・ホスト上に, 10-10
- 物理データベースの構造, 1-6 ~ 1-10
- プライマリ・データベース
 - 使用のための準備, B-14
- プロキシ・オプション (RMAN)
 - backup コマンド, 4-18
- プロキシ・コピー
 - 概要, 4-18
- プロセス障害, 1-11
 - 定義, 1-11
- 分散データベース
 - 回復, 15-14
 - 時間ベースの調整式回復, 15-15
 - バックアップ, 3-22
 - 変更ベースの回復, 15-15
 - メディア回復とスナップショット, 15-16
- 文障害, 1-11
 - 定義, 1-11
- 分裂ブロックの検出, 4-55

へ

- 変更ベースの回復, 14-31 ~ 14-34
 - 定義, 3-29
 - 分散データベース内の調整式の, 15-15

ほ

- 方針
 - 回復
 - 回復が必要なものの判断, 3-28
 - メディア回復が必要となるタイミングの判断, 3-31

- バックアップ, 3-15 ~ 3-24
 - ARCHIVELOG モード, 3-17
 - NOARCHIVELOG モード, 3-16
- 補助セット
 - RMAN TSPITR の, A-4
 - 表領域内のデータ・ファイルの命名, A-13
- 補助データベース
 - RMAN の TSPITR の
 - データ・ファイル・コピーの使用, A-13
 - 変換されたファイル名, A-14
- ホット・バックアップ
 - 一貫性のない全体データベース・バックアップ, 3-4
- ホット・バックアップ・モード
 - オンライン O/S バックアップに必要な, 13-7

ま

- マルチスレッド回復, 14-13

み

- ミラー化
 - 多重化との比較, 2-7
- ミラー化したファイル
 - オンライン REDO ログ, 2-13
 - 制御ファイル, 2-9
- ミラー化ディスク保管, 2-9
- ミラー化ファイル
 - オンライン REDO ログ
 - 消失, 15-5
 - メンバーの消失, 15-5
 - 制御ファイル
 - 消失, 15-11
 - ～の分割, 13-9
 - サスペンド / 再開モード, 13-9
- ミラーの分割
 - サスペンド / 再開モード, 13-9

め

- メタデータ
 - Recovery Manager, 4-12
 - 管理, 6-1
 - 制御ファイルへの格納, 4-15
- メディア回復, 14-1 ~ 14-35
 - ADD DATAFILE 操作, 15-3

Import ユーティリティの使用法, 13-18
NOARCHIVELOG モード, 14-16
OPEN RESETLOGS 操作後, 14-34
Recovery Manager の使用法, 4-49, 9-15
REDO ログ・ファイルのエラー, 14-15
SYSTEM 表領域, 14-23
アーカイブ REDO ログの適用, 14-10
アーカイブの効果, 2-16
エラー・メッセージ, 14-15
オープン状態のデータベースのオフラインの表領域, 14-23
オープン状態のデータベースのオフライン表領域の~, 14-23
オンライン REDO ログ・ファイル, 15-5
回復が必要なファイルの決定, 14-3
回復も参照。
概要, 1-19 ~ 1-23
完全, 1-19, 3-28, 14-19 ~ 14-25
完了, 14-22, 14-25
~ 後のデータベースのオープン, 14-32
コマンド, 14-2 ~ 14-9
再起動, 14-16
削除された表, 15-14
時間ベースの~, 14-25
~ 時の REDO ログの適用, 14-10
消失した制御ファイル, 15-11
消失したファイル
消失したアーカイブ REDO ログ・ファイル, 15-10
消失した制御ファイル, 15-11
消失したデータ・ファイル, 15-2
消失したミラー化制御ファイル, 15-11
スナップショット, 15-16
制御ファイルの破損後, 15-12
正常に適用された REDO ログ, 14-15
タイプ
分散データベース, 15-14
中断, 14-16
中断後の再開, 14-16
定義, 14-2
データ・ファイル
ガイドライン, 4-49, 9-16
バックアップがない場合, 14-6
トランSPORTABLE 表領域, 15-4
取消しベースの~, 3-29, 14-20, 14-23, 14-25, 14-28
破損していない表領域のオンライン~, 14-23

不完全, 3-29, 14-25 ~ 14-34
不完全、定義, 1-19
復元
アーカイブ REDO ログ・ファイル, 14-7
全体データベース・バックアップ, 14-16
破損したファイル, 14-5
分散データベース, 15-14
時間ベースの調整式, 15-15
変更ベースの~, 3-29, 14-25, 14-31 ~ 14-34
メソッド
選択, 3-31
ロールバック, 14-2
ロールフォワード, 14-2
ロールフォワード・フェーズ, 14-10
ログ順序回復, 3-29
メディア回復の中断, 14-16
メディア管理
Backup Solutions Program, 4-19
Legato Storage Manager (LSM), 4-19
sbttest プログラム, 4-19
一意的なファイル名、~ の生成, 5-19
クロスチェック, 4-17
最大ファイル・サイズ, 5-20
ソフトウェアへのリンク, 5-19
デバイス固有の文字列の送信, 5-20
トラブルシューティング, 5-20
ファイルのバックアップ, 4-17
ファイルの復元, 4-17
プロキシ・コピー, 4-18
要件, 5-19
メディア管理ライブラリ (MML), 4-17
メディア障害
NOARCHIVELOG モード, 14-16
アーカイブ REDO ログ・ファイルの消失, 15-10
永久, 15-2
オンライン REDO ログ・グループの消失, 15-6
オンライン REDO ログの消失, 15-5
回復, 14-19 ~ 14-31
分散データベース, 15-14
回復手順
例, 15-2 ~ 15-14
完全回復, 14-19 ~ 14-25
種類の理解, 15-2
制御ファイルの消失, 15-11, 15-12
説明, 15-2
定義, 1-12
ディスク・コントローラ, 15-2

データ・ファイルの消失, 15-2
メディア・マネージャのトラブルシューティング,
5-20

も

モード

ARCHIVELOG, 1-10
NOARCHIVELOG, 1-10
障害からの回復, 14-16

ゆ

ユーザー・エラー

回復、～から, 15-13
定義, 1-11
データベース障害, 1-12

ユーザー作成のバックアップ・ファイル

カタログ化, 6-28

ユーザー・タグ, 4-46

ユーティリティ

OS、コピー作成のための～の使用法, 8-25
Recovery Manager, 4-2

よ

読取り一貫性

Export ユーティリティ, 13-16

読取り専用表領域

回復, 14-5
回復への影響, 14-5
～のバックアップ, 8-16

読取り専用モード

スタンバイ・データベース, 16-5, 16-19

り

リカバリ・カタログ

作成, 6-2

リカバリ・カタログ, 4-13, 5-4 ~ 6-34

アップグレード, 6-34
および不完全回復, 9-23
概要, 4-13
カタログ化
O/S バックアップ, 6-28
クロスチェック, 6-9
更新

スキーマの変更後, 6-26

再同期化, 4-13, 6-23

削除, 6-35

作成

別のデータベースでの, 5-5

消失または破損した～の回復, 6-34

スキーマ, 5-5

設定, 6-2

スナップショット制御ファイル, 4-14

データベースの登録, 6-4, 6-5

データベースの登録解除, 6-6

問合せ, 4-20, 7-2

～なしでの運用, 4-15

～のサイズの管理, 6-27

～のバックアップ, 4-14

バックアップの可用性の変更, 6-8

バックアップの状態の deleted への変更, 6-13

ビュー, 12-1

リフレッシュ, 6-23

レコードの削除, 6-17

ログ・スイッチのレコード, 6-28

～を使用した運用, 4-13

～を使用したことにより得られる結果, 5-4

～を使用して Recovery Manager に接続する場合,
5-10, 5-11

～を使用せずに Recovery Manager に接続する場
合, 5-9, 5-10

リカバリ・カタログからのデータベースの登録解除,
6-6

リカバリ・カタログからのレコードの削除, 6-17

リカバリ・カタログのアップグレード, 6-34

リカバリ・カタログの再同期化, 4-13, 6-23

リカバリ・カタログの削除, 6-35

リカバリ・カタログの作成, 6-2

リカバリ・カタログのレコードの更新, 6-13

リスト、生成, 7-2 ~ 7-5

使用例, 7-10

バックアップおよびコピーのリスト表示, 7-3

リモート・ファイル・サーバー (RFS) プロセス, 16-9

る

累積エクスポート, 13-17

累積増分バックアップ, 4-42, 8-22

れ

- レベル 0 の増分バックアップ , 4-39
- レポート、生成 , 7-2 , 7-5
 - 回復不能なバックアップ , 7-6 , 7-8
 - 使用例 , 7-11 , 7-13
 - データベース・スキーマ , 7-9
 - バックアップが必要なオブジェクト , 7-5
 - 不要なバックアップ , 7-8 , 7-6
- レポートの生成 , 4-20 , 7-5

ろ

- ロールバック , 14-2
- ロールバック・セグメント , 1-8
 - および RMAN の TSPITR , A-6
- ロールフォワード , 14-2
- ログ順序回復 , 3-29
- ログ順序番号
 - 回復時に要求された , 14-10
- ログ・スイッチ
 - アーカイブ完了待ちの ~ , 2-14
 - 多重 REDO ログ・ファイルと ~ , 2-14
 - リカバリ・カタログ・レコード , 6-28
- ログ・ファイルの適用
 - SQL*Plus , 14-12
- ログ・ライター・プロセス (LGWR)
 - 多重 REDO ログ・ファイルと ~ , 2-13
 - トレース・ファイルと ~ , 2-14
- 論理バックアップ
 - 定義 , 3-14