

**Oracle9i**

Recovery Manager ユーザーズ・ガイド

リリース 2 (9.2)

2002 年 7 月

部品番号 : J06269-01

**ORACLE®**

---

Oracle9i Recovery Manager ユーザーズ・ガイド, リリース 2 (9.2)

部品番号 : J06269-01

原本名 : Oracle9i Recovery Manager User's Guide, Release 2 (9.2)

原本部品番号 : A96566-01

原本著者 : Lance Ashdown

原本協力者 : Beldalker Anand, Tammy Bednar, Don Beusee, Senad Dizdar, Muthu Olagappan, Francisco Sanchez, Steve Wertheimer, Valarie Moore

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

Printed in Japan

制限付権利の説明

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

当プログラムのリバース・エンジニアリング等は禁止されています。

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

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

危険な用途への使用について

オラクル社製品は、原子力、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションを用途として開発されておりません。オラクル社製品を上述のようなアプリケーションに使用することについての安全確保は、顧客各位の責任と費用により行ってください。万が一かかる用途での使用によりクレームや損害が発生いたしましても、日本オラクル株式会社と開発元である **Oracle Corporation**（米国オラクル）およびその関連会社は一切責任を負いかねます。当プログラムを米国国防総省の米国政府機関に提供する際には、『**Restricted Rights**』と共に提供してください。この場合次の **Notice** が適用されます。

**Restricted Rights Notice**

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

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

---

---

# 目次

はじめに .....	xix
------------	-----

Recovery Manager の新機能 .....	xxix
-----------------------------	------

## 第 I 部 Recovery Manager のスタート・ガイド

### 1 概要

Recovery Manager .....	1-2
Recovery Manager を使用する理由 .....	1-2
Recovery Manager 環境の概要 .....	1-6
Recovery Manager の実行可能ファイル .....	1-6
ターゲット・データベース .....	1-6
Recovery Manager リポジトリ .....	1-7
Recovery Manager メディア管理インタフェース .....	1-9

### 2 Recovery Manager を使用したデータベースへの接続

Recovery Manager の起動: 概要 .....	2-2
データベース接続のタイプ .....	2-2
データベース接続に必要な認証 .....	2-3
Recovery Manager 起動時のコマンドライン・オプション .....	2-4
データベースに接続しない Recovery Manager の起動 .....	2-4
リカバリ・カタログを使用しないターゲット・データベースへの接続 .....	2-5
カタログを使用しないコマンドラインからのターゲット・データベースへの接続 .....	2-5
カタログを使用しない Recovery Manager プロンプトからのターゲット・データベースへの接続 .....	2-6

ターゲット・データベースおよびリカバリ・カタログへの接続 .....	2-6
コマンドラインからのターゲット・データベースおよびリカバリ・カタログへの接続 .....	2-7
Recovery Manager プロンプトからのターゲット・データベースおよび リカバリ・カタログへの接続 .....	2-7
Oracle Real Application Clusters 内のターゲット・データベースへの接続 .....	2-8
補助データベースへの接続 .....	2-9
コマンドラインからの補助データベースへの接続 .....	2-10
データベースへの接続時にパスワードを非表示にする方法 .....	2-11
パイプを使用した Recovery Manager コマンドの実行 .....	2-12
Recovery Manager の終了 .....	2-13

### 3 クイック・スタート

Recovery Manager のサンプル・スクリプトの使用方法および使用例 .....	3-2
Recovery Manager 認証方式の選択 .....	3-2
基本的な Recovery Manager コマンドの使用方法 .....	3-3
ターゲット・データベースへの接続 .....	3-4
データベースの起動および停止 .....	3-5
ターゲット・データベースの現行スキーマのレポート .....	3-6
データベースのバックアップ .....	3-6
表領域のバックアップ .....	3-8
アーカイブ・ログのバックアップ .....	3-9
データ・ファイルのコピー .....	3-9
バックアップおよびコピーのリスト表示 .....	3-10
バックアップのリストアの妥当性チェック .....	3-11
データベースのリストアおよびリカバリ .....	3-12
表領域のリストアおよびリカバリ .....	3-13
Recovery Manager 構成の表示 .....	3-14
Recovery Manager でのリカバリ・カタログの使用・不使用の決定 .....	3-14
リカバリ・カタログを Recovery Manager リポジトリとして使用した場合の利点 .....	3-15
リカバリ・カタログを Recovery Manager リポジトリとして使用した場合のコスト .....	3-16

## 第 II 部 アーキテクチャおよび概念

### 4 Recovery Manager のアーキテクチャ

Recovery Manager アーキテクチャの概要 .....	4-2
Recovery Manager 環境 .....	4-2
Recovery Manager セッション・アーキテクチャ .....	4-5
Recovery Manager メタデータの格納 .....	4-5
Recovery Manager コマンド・インタフェース .....	4-5
Recovery Manager と PL/SQL パッケージ .....	4-6
Recovery Manager コマンドのコンパイルおよび実行方法 .....	4-6
Recovery Manager コマンドのタイプ .....	4-7
Recovery Manager コマンドの実行 .....	4-9
Recovery Manager パイプ・インタフェース .....	4-11
Oracle Enterprise Manager での Recovery Manager のジョブ・スクリプト .....	4-11
Recovery Manager リポジトリ .....	4-12
Recovery Manager リポジトリのリカバリ・カタログへの格納 .....	4-12
Recovery Manager リポジトリの制御ファイルへの格納 .....	4-16
メディア管理 .....	4-18
メディア・マネージャを使用したバックアップおよびリストア操作 .....	4-18
プロキシ・コピー .....	4-19
メディア・マネージャのテスト .....	4-19
Backup Solutions Program .....	4-19

### 5 概念 I: チャンネル、バックアップおよびコピー

Recovery Manager の自動チャンネルおよび手動チャンネルの割当て .....	5-2
Recovery Manager のチャンネル .....	5-2
自動チャンネルおよび手動チャンネルの割当て .....	5-4
自動チャンネルのデバイスの構成 .....	5-5
自動チャンネルのデフォルト・デバイス・タイプ .....	5-6
自動チャンネルのネーミング規則 .....	5-7
自動チャンネルの一般的な構成 .....	5-8
特定の自動チャンネルの構成 .....	5-9
自動チャンネル設定の消去 .....	5-10
手動によるチャンネル割当ての平行化 .....	5-10

手動および自動チャネル用のチャネル制御オプション .....	5-11
メディア・マネージャによるハードウェアの多重化 .....	5-13
<b>バックアップ・セット</b> .....	5-13
バックアップ・セット .....	5-14
プロキシ・コピー .....	5-15
バックアップ・セットの格納 .....	5-16
バックアップ・セットの圧縮 .....	5-16
アーカイブ・ログのバックアップ .....	5-17
多重バックアップ・セット .....	5-20
二重バックアップ・セット .....	5-23
バックアップの平行化 .....	5-25
バックアップ・セットのバックアップ .....	5-28
<b>バックアップ・オプション: 名前、サイズおよびスピード</b> .....	5-31
バックアップ・ピースのファイル名 .....	5-31
バックアップ・ピースのサイズ .....	5-32
バックアップ・セットの数とサイズ: 基本アルゴリズム .....	5-33
バックアップ・セットの数とサイズ: 拡張アルゴリズム .....	5-39
バックアップの I/O 読取りレート .....	5-40
<b>バックアップ・タイプ</b> .....	5-41
バックアップ・タイプ .....	5-41
全体バックアップ .....	5-42
増分バックアップ .....	5-43
Recovery Manager がバックアップ可能なファイルのタイプ .....	5-50
<b>バックアップ・エラー</b> .....	5-51
<b>制御ファイルとサーバー・パラメータ・ファイルの自動バックアップ</b> .....	5-52
<b>バックアップの保存方針</b> .....	5-55
バックアップの保存方針について .....	5-55
リカバリ・ウィンドウ .....	5-56
バックアップ冗長性 .....	5-59
不要なバックアップとコピーのバッチ削除 .....	5-60
バックアップの保存方針からの除外 .....	5-61
<b>バックアップの最適化</b> .....	5-62
バックアップの最適化のアルゴリズム .....	5-62
バックアップの最適化を有効化および無効化するための要件 .....	5-65
保存方針がバックアップの最適化に与える影響 .....	5-66
<b>再起動可能なバックアップ</b> .....	5-68

イメージ・コピー .....	5-69
Recovery Manager のイメージ・コピー .....	5-69
ユーザー管理のイメージ・コピー .....	5-70
バックアップおよびイメージ・コピーのタグ .....	5-71
長期間の Recovery Manager コピー .....	5-72
バックアップのテストおよび整合性チェック .....	5-72
テストおよび整合性チェック .....	5-73
物理的なブロック破損の検出 .....	5-73
論理的なブロック破損の検出 .....	5-74
オープン・バックアップ時の分裂ブロックの検出 .....	5-74
Recovery Manager を使用したバックアップのテスト .....	5-75

## 6 概念 II: リストア、リカバリおよび複製

Recovery Manager を使用したファイルのリストア .....	6-2
RESTORE コマンド .....	6-2
リストア操作の仕組み .....	6-3
リストア処理時のファイル選択 .....	6-4
リストアの最適化 .....	6-5
Real Application Clusters でリストアする場合の自動的なバックアップ位置検索 .....	6-5
Recovery Manager を使用したデータ・ファイルのメディア・リカバリ .....	6-6
Recovery Manager メディア・リカバリ：基本手順 .....	6-6
リカバリの仕組み：増分バックアップおよび REDO ログ .....	6-8
不完全リカバリ .....	6-10
表領域の Point-in-Time リカバリ .....	6-10
制御ファイルの自動バックアップを使用した障害時リカバリ .....	6-11
Recovery Manager を使用したブロック・メディア・リカバリ .....	6-11
ブロック・メディア・リカバリ .....	6-11
ブロック・メディア・リカバリを使用する場合 .....	6-12
REDO が存在しない場合のブロック・メディア・リカバリ .....	6-13
Recovery Manager を使用したデータベースの複製 .....	6-14
Recovery Manager を使用したスタンバイ・データベースの作成 .....	6-16

## 7 概念 III: メンテナンス

Recovery Manager のレポート機能 .....	7-2
LIST コマンドの出力 .....	7-2
REPORT コマンドによる出力 .....	7-3
SHOW コマンドによる出力 .....	7-7
PRINT SCRIPT コマンドによる出力 .....	7-8
Recovery Manager のバックアップおよびコピーのクロスチェック .....	7-8
Recovery Manager のバックアップおよびコピーの削除 .....	7-10
Recovery Manager による削除 .....	7-11
Recovery Manager による削除方法のまとめ .....	7-12
DELETE コマンドによるバックアップおよびコピーの削除 .....	7-13
リポジトリとメディアが対応していない場合の DELETE コマンドの動作 .....	7-15
BACKUP ... DELETE INPUT コマンドによるバックアップおよびコピーの削除 .....	7-15
Recovery Manager のメンテナンス・コマンドへの複数チャネルの割当て .....	7-17
Recovery Manager のバックアップおよびコピーの可用性の変更 .....	7-18
Recovery Manager のバックアップおよびコピーの保存ステータスの変更 .....	7-19

## 第 III 部 Recovery Manager を使用したバックアップおよびリカバリの実行

### 8 Recovery Manager 環境の構成

メディア・マネージャにバックアップを作成する場合の Recovery Manager の構成 .....	8-2
Recovery Manager でメディア・マネージャを使用する場合の前提条件 .....	8-2
Oracle とメディア・マネージャの通信方法 .....	8-3
Recovery Manager とメディア・マネージャの統合: 基本手順 .....	8-4
メディア・マネージャ・ライブラリが正常に統合されているかどうかのテスト .....	8-6
自動チャネルをメディア・マネージャで使用する場合の構成 .....	8-9
自動チャネルの構成 .....	8-11
自動チャネルの構成: 概要 .....	8-11
デフォルト・デバイス・タイプの変更 .....	8-12
自動チャネルの平行化の構成 .....	8-12
デバイス・タイプ用の一般的な自動チャネルの構成 .....	8-13
デバイス・タイプ用の特定のチャネルの構成 .....	8-15
チャネルおよびデバイス設定の消去 .....	8-17



制御ファイルとサーバー・パラメータ・ファイルの自動バックアップの構成 .....	8-18
制御ファイルの自動バックアップの有効化と無効化 .....	8-19
制御ファイルの自動バックアップ書式の構成 .....	8-20
バックアップの保存方針の構成 .....	8-21
リカバリ・ウィンドウのための保存方針の構成 .....	8-21
冗長性のための保存方針の構成 .....	8-22
保存方針の無効化 .....	8-22
保存方針をデフォルト設定に戻す方法 .....	8-23
バックアップ・セットの最大サイズの設定 .....	8-23
バックアップの最適化の構成 .....	8-24
バックアップ・コピー数の設定 .....	8-25
データベース全体のバックアップから除外する表領域の構成 .....	8-26
Recovery Manager のグローバリゼーション・サポートの環境変数の設定 .....	8-28
NLS_DATE_FORMAT および NLS_LANG の設定 .....	8-28
データベース・キャラクタ・セットの指定 .....	8-28
スナップショット制御ファイルの場所の設定 .....	8-29
スナップショット制御ファイルの場所をデフォルト値に設定する方法 .....	8-29
スナップショット制御ファイルの場所をデフォルト以外の値に設定する方法 .....	8-30
Recovery Manager を共有サーバーで使用する場合は設定 .....	8-31
リカバリ・カタログの設定 .....	8-32

## 9 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・ジョブおよびコピー・ジョブで使用するチャンネルの構成および割当て .....	9-2
Recovery Manager を使用したデータベース・ファイルおよびアーカイブ・ログのバックアップ .....	9-3
Recovery Manager のバックアップ .....	9-4
Recovery Manager を使用した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 .....	9-6
Recovery Manager を使用したデータベース全体のバックアップ .....	9-7
Recovery Manager を使用した表領域のバックアップ .....	9-8
Recovery Manager を使用したデータ・ファイルおよびデータ・ファイルのコピーのバックアップ .....	9-8
Recovery Manager を使用した制御ファイルのバックアップ .....	9-9
Recovery Manager を使用したサーバー・パラメータ・ファイルのバックアップ .....	9-13
Recovery Manager を使用したアーカイブ REDO ログのバックアップ .....	9-13
二重化バックアップ .....	9-18
CONFIGURE BACKUP COPIES を使用した多重化バックアップ .....	9-18
BACKUP COPIES を使用した多重化バックアップ .....	9-19

<b>Recovery Manager を使用した増分バックアップの作成</b> .....	9-20
<b>Recovery Manager を使用したミラーの分割によるバックアップの作成</b> .....	9-21
<b>Recovery Manager を使用したスタンバイ・データベース・サイト上のファイルのバックアップ</b> .....	9-23
Recovery Manager によるスタンバイ・データベース・データ・ファイルおよび アーカイブ・ログのバックアップ .....	9-23
Recovery Manager スタンバイ・データベース・バックアップ作成時の制限事項および 使用方法 .....	9-24
RC_ARCHIVED_LOG ビューの解析 .....	9-25
スタンバイ・データベースのアーカイブ REDO ログをバックアップする時期の判断 .....	9-26
Recovery Manager を使用したスタンバイ・データベースのバックアップ .....	9-28
<b>Recovery Manager を使用したバックアップ・セットのバックアップ</b> .....	9-30
<b>Recovery Manager によるバックアップの再起動および最適化</b> .....	9-31
バックアップの最適化を使用したファイルのバックアップ .....	9-31
部分的に完了したバックアップの再起動 .....	9-32
<b>Recovery Manager を使用したバックアップの妥当性チェックの実行</b> .....	9-32
<b>Recovery Manager を使用したファイルのコピー</b> .....	9-34
データベース全体のコピー .....	9-34
任意のデータ・ファイルのコピーおよびアーカイブ REDO ログのコピー .....	9-36
<b>制御ファイルの自動バックアップ書式の上書き</b> .....	9-37
<b>Recovery Manager のバックアップおよびコピーの例</b> .....	9-38
BACKUP コマンドでのデバイス・タイプの指定 : 例 .....	9-39
データベースのバックアップ時の表領域のスキップ : 例 .....	9-39
バックアップの再起動 : 例 .....	9-40
複数ディスク・ドライブへのバックアップの分散 : 例 .....	9-41
複数ファイル・システムへの大規模データベースのバックアップ : 例 .....	9-42
バックアップ・セットのサイズ指定 : 例 .....	9-43
バックアップ・ピースのサイズ制限 : 例 .....	9-45
バックアップにおけるデータ・ファイルの多重化 : 例 .....	9-45
フェイルオーバーでのアーカイブ REDO ログのバックアップの使用例 : 例 .....	9-47
オンライン・バックアップのリカバリに必要なアーカイブ・ログのバックアップ : 例 .....	9-47
アーカイブ REDO ログの複数コピーのバックアップと削除 : 例 .....	9-48
差分増分バックアップの実行 : 例 .....	9-49
累積増分バックアップの実行 : 例 .....	9-49
チャネルによるバックアップ作業負荷分散の決定 : 例 .....	9-50
NOARCHIVELOG モードでのバックアップ : 例 .....	9-51
Oracle Real Application Clusters 環境でのバックアップ : 例 .....	9-52

オペレーティング・システム・コピーのカatalog化 : 例 .....	9-53
長期バックアップの保持 : 例 .....	9-53
バックアップの最適化 : 例 .....	9-54
バックアップおよびコピー時のエラー処理 : 例 .....	9-57

## 10 Recovery Manager によるリストアとリカバリ

<b>Recovery Manager を使用したリストアおよびリカバリ : 概要</b> .....	10-2
メディア・リカバリの一般的な手順 .....	10-2
リストアとリカバリの使用例の相違 .....	10-3
<b>基本的な Recovery Manager メディア・リカバリの実行</b> .....	10-5
基本的なメディア・リカバリ .....	10-6
メディア・リカバリの準備 .....	10-7
完全リストアおよびリカバリの実行 .....	10-9
不完全リストアおよびリカバリの実行 .....	10-11
<b>サーバー・パラメータ・ファイルのリストア</b> .....	10-13
<b>バックアップ制御ファイルを使用したリカバリの実行</b> .....	10-15
バックアップ制御ファイルおよびリカバリ・カatalogを使用したリカバリの実行 .....	10-16
バックアップ制御ファイルを使用してリカバリ・カatalogを使用しないリカバリの実行 .....	10-18
<b>新しい場所へのファイルのリストア</b> .....	10-21
新しい場所へのデータ・ファイルのリストア .....	10-21
新しい場所への制御ファイルのリストア .....	10-22
アーカイブ REDO ログの新しい場所へのリストア .....	10-23
<b>新しいホストへのデータベースのリストア</b> .....	10-25
新しいホストへのデータベースのリストア .....	10-25
新しいホストにリストアする場合のファイル名の指定 .....	10-26
<b>Oracle Real Application Clusters データベースのリカバリ</b> .....	10-27
<b>Recovery Manager を使用した RESETLOGS 操作によるリカバリ</b> .....	10-28
Recovery Manager を使用した RESETLOGS 操作によるリカバリ .....	10-29
RESETLOGS 操作によるメディア・リカバリの要件 .....	10-29
RESETLOGS 操作によるリカバリ : 基本手順 .....	10-30
RESETLOGS 操作によるリカバリ : 例 .....	10-32
<b>障害時リカバリの実行</b> .....	10-37

<b>Recovery Manager を使用したデータ・ブロックのリカバリ</b> .....	10-40
使用可能なすべてのバックアップによるデータ・ブロックのリカバリ .....	10-41
任意のバックアップを使用したデータ・ブロックのリカバリ .....	10-41
V\$DATABASE_BLOCK_CORRUPTION にリストされるブロックのリカバリ .....	10-42
<b>バックアップとコピーのリストアの妥当性チェック</b> .....	10-43
<b>Recovery Manager のリストアおよびリカバリ例</b> .....	10-45
新しいホストへのデータ・ファイルのコピーのリストア : 例 .....	10-45
カタログ内の複数データベースが同じ名前を共有する場合のリストア : 例 .....	10-46
RESETLOGS 操作以前に作成されたバックアップのリカバリ : 例 .....	10-48
NOARCHIVELOG モードでのデータベースのリカバリ : 例 .....	10-49
バックアップがないデータ・ファイルを消失した場合のリカバリ : 例 .....	10-50
Recovery Manager を使用した異なるデータベースへの表領域バックアップの移動 : 例 .....	10-51

## 11 Recovery Manager による表領域の Point-in-Time リカバリの実行

<b>TSPITR の概要</b> .....	11-2
TSPITR の用語 .....	11-4
<b>TSPITR の計画</b> .....	11-4
リカバリ・カタログを使用しない TSPITR の実行 .....	11-5
一般的な制限事項の理解 .....	11-5
データの関連の管理 .....	11-6
<b>補助セットとリカバリ・セットに含まれるファイルのネーミング・メソッドの選択</b> .....	11-9
Recovery Manager TSPITR でのデータ・ファイルのネーミングの概要 .....	11-9
SET NEWNAME を使用した補助セット内とリカバリ・セット内のファイルの名前付け .....	11-10
補助セットとリカバリ・セットでのデータ・ファイルのコピーの使用 .....	11-11
初期化パラメータを使用した補助セットのファイルの名前付け .....	11-12
<b>TSPITR の実行に必要な補助インスタンスの準備</b> .....	11-13
タスク 1: 補助インスタンスの Oracle パスワード・ファイルの作成 .....	11-14
タスク 2: 補助インスタンス用パラメータ・ファイルの作成 .....	11-14
タスク 3: 補助インスタンスへの Oracle Net 接続性の確認 .....	11-16
タスク 4: 補助インスタンスの起動 .....	11-16
タスク 5: Recovery Manager コマンドライン・インタフェースの起動 .....	11-16
<b>TSPITR の実行</b> .....	11-18
<b>TSPITR の実行後に使用するターゲット・データベースの準備</b> .....	11-19
<b>TSPITR が失敗した場合の対応</b> .....	11-20

## 12 Recovery Manager を使用したデータベースの複製

複製データベースの作成 : 概要 .....	12-2
Recovery Manager によるデータベースの複製方法 .....	12-2
データベース複製オプション .....	12-4
前提条件および制限事項 .....	12-5
複製データベース用のファイルの作成 .....	12-5
複製制御ファイルの作成 .....	12-5
複製オンライン REDO ログの作成 .....	12-6
複製データ・ファイルの命名 .....	12-7
複製用の補助インスタンスの準備 : 基本手順 .....	12-10
タスク 1: 補助インスタンス用の Oracle パスワード・ファイルの作成 .....	12-10
タスク 2: 補助インスタンスへの Oracle Net 接続性の確認 .....	12-10
タスク 3: 補助インスタンス用の初期化パラメータ・ファイルの作成 .....	12-11
タスク 4: 補助インスタンスの起動 .....	12-12
タスク 5: ターゲット・データベースのマウントまたはオープン .....	12-13
タスク 6: 必要なバックアップとアーカイブ REDO ログの確認 .....	12-13
タスク 7: 自動チャンネルが構成されない場合の補助チャンネルの割当て .....	12-13
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 .....	12-15
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 .....	12-15
異なる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 .....	12-16
ローカル・ホストへの複製データベースの作成 .....	12-21
データベースの複製の例 .....	12-21
新しいファイル名の手動による設定 : 例 .....	12-21
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 : 例 .....	12-23
過去の複製データベースの作成 : 例 .....	12-24
クライアント側のパラメータ・ファイルを使用した複製 : 例 .....	12-24

## 13 Recovery Manager によるスタンバイ・データベースの作成

Recovery Manager を使用したスタンバイ・データベースの準備 .....	13-2
Recovery Manager を使用したスタンバイ・データベースの準備 .....	13-2
Recovery Manager を使用したスタンバイ制御ファイルの作成 .....	13-3
スタンバイ制御ファイルのイメージ・コピーの作成 .....	13-6
Recovery Manager 使用時のスタンバイ・データベースのデータ・ファイルの命名 .....	13-7
Recovery Manager 使用時のスタンバイ・データベースのオンライン REDO ログの命名 .....	13-8

<b>Recovery Manager を使用したスタンバイ・データベースの作成: 概要</b> .....	13-9
リカバリなしの Recovery Manager スタンバイの作成 .....	13-10
リカバリありの Recovery Manager スタンバイの作成 .....	13-10
<b>Recovery Manager およびスタンバイ・インスタンスの起動</b> .....	13-12
<b>同じディレクトリ構造を持つリモート・ホストへのスタンバイ・データベースの作成</b> .....	13-13
リカバリを実行しないスタンバイ・データベースの作成 .....	13-13
スタンバイ・データベースの作成とリカバリの実行 .....	13-14
<b>異なるディレクトリ構造を持つリモート・ホストへのスタンバイ・データベースの作成</b> .....	13-15
DB_FILE_NAME_CONVERT を使用したスタンバイ・データベース・ファイルの命名 .....	13-16
SET NEWNAME コマンドを使用したスタンバイ・データベース・ファイルの命名 .....	13-17
CONFIGURE AUXNAME コマンドを使用したスタンバイ・データベース・ファイルの命名 ...	13-19
<b>ローカル・ホストへのスタンバイ・データベースの作成</b> .....	13-22
<b>イメージ・コピーを使用したスタンバイ・データベースの作成</b> .....	13-23
イメージ・コピーを使用したスタンバイ・データベースの作成: 概要 .....	13-23
コピーとデータ・ファイルが同じ名前を使用する場合のスタンバイ・データベースの作成 .....	13-25
コピーとデータ・ファイルが異なる名前を使用する場合のスタンバイ・データベースの作成 ...	13-26
バックアップとイメージ・コピーを使用したスタンバイ・データベースの作成: 例 .....	13-29

## 14 Recovery Manager のチューニング

<b>チューニング概要</b> .....	14-2
<b>チューニングの概念</b> .....	14-2
ディスク・バッファの割当て .....	14-3
テープ・バッファの割当て .....	14-5
同期 I/O と非同期 I/O .....	14-6
テープへのバックアップ速度に影響する要因 .....	14-8
チャネルのチューニング・オプション .....	14-9
バックアップのチューニング・オプション .....	14-10
<b>バックアップ・パフォーマンスの向上</b> .....	14-11
手順 1: RATE パラメータを調整する .....	14-11
手順 2: テープにバックアップする場合は、BACKUP_TAPE_IO_SLAVES を設定する .....	14-12
手順 3: 同期ディスクを使用する場合は、DBWR_IO_SLAVES を設定する .....	14-12
手順 4: 共有メモリーの割当てに失敗する場合は、LARGE_POOL_SIZE を設定する .....	14-12
手順 5: 多重化レベルを調整する .....	14-14
手順 6: ファイルが空であるか変更が少ないかを判断する .....	14-15

手順 7: BLKSIZE パラメータを使用してテープ・バッファ・サイズを変更する .....	14-16
手順 8: V\$ ビューを問い合わせでボトルネックを識別する .....	14-16

## 15    トラブルシューティング

<b>Recovery Manager のメッセージ出力の解釈 .....</b>	<b>15-2</b>
メッセージ出力のタイプの識別 .....	15-2
Recovery Manager のエラー・メッセージ・スタックの認識 .....	15-3
エラー・コードの識別 .....	15-4
Recovery Manager のエラー・スタックの解析 .....	15-7
Recovery Manager のリターン・コードの識別 .....	15-9
<b>Media Management API のテスト .....</b>	<b>15-9</b>
sbttest ユーティリティの取得 .....	15-9
sbttest 用オンライン・マニュアルの取得 .....	15-10
sbttest ユーティリティの使用方法 .....	15-11
<b>Recovery Manager コマンドの終了 .....</b>	<b>15-12</b>
ALTER SYSTEM によるセッションの終了 .....	15-12
オペレーティング・システム・レベルでのセッション終了 .....	15-13
メディア・マネージャで停止した Recovery Manager セッションの終了 .....	15-13
<b>V\$ ビューを介した Recovery Manager の監視 .....</b>	<b>15-16</b>
サーバー・セッションと Recovery Manager チャネルの相関 .....	15-17
Recovery Manager ジョブの進行状況の監視 .....	15-20
Recovery Manager とメディア・マネージャの相互作用の監視 .....	15-23
Recovery Manager ジョブのパフォーマンスの監視 .....	15-24
リカバリを必要とするデータ・ファイルの判断 .....	15-24
<b>トラブルシューティング例 .....</b>	<b>15-26</b>
メディア・マネージャのインストール後に、Recovery Manager のチャネルの割当てに	
失敗した場合：例 .....	15-27
バックアップ・ジョブが停止する場合：例 .....	15-29
Recovery Manager が RPC コールの起動に失敗する場合：例 .....	15-30
無効な RECID エラーでバックアップが失敗する場合：例 .....	15-31
制御ファイルのエンキューが原因でバックアップが失敗する場合：例 .....	15-35
Recovery Manager がすべてのアーカイブ・ログの削除に失敗する場合：例 .....	15-36
Recovery Manager でアーカイブ・ログが見つからないためにバックアップが	
失敗する場合：例 .....	15-37

Recovery Manager がターゲット・データベース・キャラクタ・セットを 設定できない場合 : 例 .....	15-38
Recovery Manager がキャラクタ・セット名を識別しない場合 : 例 .....	15-39
Recovery Manager がターゲット・データベースへのログオンを拒否する場合 : 例 .....	15-40
ログの消失が原因でデータベースの複製が失敗する場合 : 例 .....	15-41
複数の RMAN-06023 エラーが表示されて複製が失敗する場合 : 例 .....	15-42
リカバリ・カタログに UNKNOWN というデータベース名が表示される場合 : 例 .....	15-43

## 第 IV 部 Recovery Manager のリポジトリのメンテナンス

### 16 Recovery Manager のリポジトリの管理

リカバリ・カタログの作成 .....	16-2
リカバリ・カタログ・データベースの構成 .....	16-2
リカバリ・カタログ所有者の作成 .....	16-4
リカバリ・カタログの作成 .....	16-5
リカバリ・カタログ内のターゲット・データベース・レコードの管理 .....	16-6
データベースのリカバリ・カタログへの登録 .....	16-6
リカバリ・カタログからのターゲット・データベースの登録解除 .....	16-9
リカバリ・カタログ内のデータベース・インカネーションのリセット .....	16-11
リカバリ・カタログの再同期化 .....	16-13
再同期化 .....	16-14
再同期化されるレコードのタイプ .....	16-15
再同期化を実行する時期 .....	16-16
リカバリ・カタログの完全再同期化の強制実行 .....	16-17
CONTROLFILE_RECORD_KEEP_TIME 値の設定 .....	16-18
リカバリ・カタログに格納された Recovery Manager のスクリプトの管理 .....	16-18
リカバリ・カタログのストアド・スクリプト .....	16-19
リカバリ・カタログへのスクリプトの格納 .....	16-20
ストアド・スクリプトの実行 .....	16-20
ストアド・スクリプトの置換 .....	16-21
ストアド・スクリプトの削除 .....	16-21
リカバリ・カタログ使用時の制御ファイルの管理 .....	16-22



<b>リカバリ・カタログのバックアップとリカバリ</b> .....	16-23
リカバリ・カタログのバックアップ .....	16-24
リカバリ・カタログのリカバリ .....	16-26
リカバリ・カタログの再作成 .....	16-27
<b>リカバリ・カタログのエクスポート</b> .....	16-28
カタログ・データ移動時の考慮事項 .....	16-28
リカバリ・カタログのエクスポート .....	16-29
リカバリ・カタログのインポート .....	16-29
<b>リカバリ・カタログの可用性の増大</b> .....	16-30
<b>リカバリ・カタログのスキーマのバージョンの判断</b> .....	16-32
<b>リカバリ・カタログのアップグレード</b> .....	16-33
リカバリ・カタログの現行バージョンのチェック .....	16-33
リカバリ・カタログ・スキーマのアップグレード .....	16-34
<b>リカバリ・カタログの削除</b> .....	16-35
<b>リカバリ・カタログを使用しない Recovery Manager リポジトリの管理方法</b> .....	16-36
カタログ専用コマンドの制限 .....	16-36
制御ファイル・レコードの上書きの監視 .....	16-37
制御ファイル・リポジトリのメンテナンス .....	16-38
制御ファイルのバックアップおよびリストア .....	16-39

## 17 Recovery Manager のリポジトリの問合せ

<b>Recovery Manager のメタデータの問合せ</b> .....	17-2
<b>Recovery Manager のバックアップ、コピーおよびデータベース・インカネーションのリスト表示</b> .....	17-3
Recovery Manager のリスト .....	17-4
バックアップごとのリスト表示 .....	17-5
ファイルによるバックアップのリスト表示 .....	17-7
コピーのリスト表示 .....	17-8
サマリー・モードでのバックアップのリスト表示 .....	17-9
制限付きのバックアップおよびコピーのリスト表示 .....	17-10
データベース・インカネーションのリスト表示 .....	17-11
<b>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</b> .....	17-12
Recovery Manager のレポート .....	17-12
バックアップに必要なオブジェクトのレポート .....	17-13
不要なバックアップとコピーのレポート .....	17-15

リカバリ不能なバックアップおよびコピーのレポート .....	17-16
データベース・スキーマのレポート .....	17-17
<b>Recovery Manager の構成設定の表示</b> .....	17-18
すべての Recovery Manager の構成設定の表示 .....	17-18
Recovery Manager の保存方針の構成設定の表示 .....	17-19
自動チャネルの構成設定の表示 .....	17-19
BACKUP コマンドの構成設定の表示 .....	17-21
スナップショット制御ファイルのファイル名の表示 .....	17-23
補助チャネル用に構成されたデフォルトのファイル名の表示 .....	17-24
<b>リカバリ・カタログに格納されたスクリプトの出力</b> .....	17-24
PRINT SCRIPT を使用したストアド・スクリプトのテキスト表示 .....	17-25
RC_STORED_SCRIPT_LINE の問合せによるストアド・スクリプトのテキスト表示 .....	17-25
RC_STORED_SCRIPT の問合せによるストアド・スクリプトのリスト表示 .....	17-26
<b>リカバリ・カタログ・ビューへの問合せ</b> .....	17-27
リカバリ・カタログ・ビューへの問合せ .....	17-27
ターゲット DB_KEY または DBID 値を使用したカタログ・ビューへの問合せ .....	17-28
<b>Recovery Manager リポジトリの問合せの例</b> .....	17-29
制限のあるオブジェクトのリスト表示 : 例 .....	17-30
リカバリ・ウィンドウを必要としないバックアップおよびコピーのレポート : 例 .....	17-30
冗長なバックアップおよびコピーのレポート : 例 .....	17-30
データベース・スキーマの履歴レポートの生成 : 例 .....	17-31
データベース・インカネーションのリスト表示 : 例 .....	17-31

## 18 Recovery Manager のメンテナンスの実行

<b>バックアップおよびコピーのクロスチェック</b> .....	18-2
Recovery Manager のクロスチェック .....	18-2
特定のバックアップおよびコピーのクロスチェック .....	18-3
データベース・ファイルのバックアップおよびコピーのクロスチェック .....	18-4
<b>バックアップおよびコピーの削除</b> .....	18-5
特定のバックアップおよびコピーの削除 .....	18-6
期限切れのバックアップおよびコピーの削除 .....	18-7
不要なバックアップとコピーの削除 .....	18-8
バックアップおよびコピーの削除の強制実行 .....	18-9

<b>複数の Recovery Manager のチャンネルでのクロスチェックおよび削除</b> .....	18-10
メンテナンス・コマンド用の複数の Recovery Manager のチャンネルの割当て .....	18-10
Recovery Manager による複数のチャンネルでのクロスチェックおよび削除の方法 .....	18-11
1 つのコマンドでのディスクおよびテープ・チャンネルのクロスチェック : 例 .....	18-12
複数の Oracle Real Application Clusters ノードでのクロスチェック : 例 .....	18-13
1 つの DELETE コマンドでのディスクおよびテープ・チャンネルの削除 : 例 .....	18-13
複数のチャンネルの割当ての解除 : 例 .....	18-14
<b>バックアップまたはコピー・レコードの可用性のステータスの変更</b> .....	18-15
バックアップまたはコピーを UNAVAILABLE としてマーク .....	18-15
バックアップまたはコピーを AVAILABLE としてマーク .....	18-16
<b>バックアップまたはコピーの保存方針からの除外</b> .....	18-17
<b>アーカイブ・ログおよびユーザー管理コピーのカタログ化</b> .....	18-18
アーカイブ・ログおよびユーザー管理コピーのカタログ化 .....	18-18
ユーザー管理コピーをカタログ化する時期 .....	18-19
ユーザー管理コピーのカタログ化 .....	18-21
<b>Recovery Manager のレコードのカタログからの削除</b> .....	18-21
Recovery Manager のレコードのカタログからの削除 .....	18-22
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削除したファイルの レコードの削除 .....	18-22
ステータス DELETED によるカタログ・レコードの削除 .....	18-23

## 索引



---

# はじめに

ここでは、次の項目について説明します。

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

# 対象読者

『Oracle9i Recovery Manager ユーザーズ・ガイド』は、次の作業を行うデータベース管理者を対象としています。

- Oracle データベースのバックアップ、リストアおよびリカバリ
- データベースのバックアップおよびコピーに対するメンテナンスの実行

このマニュアルを使用するにあたって、次の内容を詳しく知っている必要があります。

- リレーショナル・データベースの概念および基本的なデータベース管理。『Oracle9i データベース概要』および『Oracle9i データベース管理者ガイド』で説明されています。
- 基本的なバックアップとリカバリの概念および方法。『Oracle9i バックアップおよびリカバリ概要』で説明されています。
- Oracle を実行しているオペレーティング・システム環境。

# このマニュアルの構成

このマニュアルには、次のような内容が含まれています。

## 第 I 部「Recovery Manager のスタート・ガイド」

この部では、Recovery Manager を使用するために必要な、基本情報を紹介します。

### 第 1 章「概要」

この章では、Recovery Manager のユーティリティの用途と機能について説明します。

### 第 2 章「Recovery Manager を使用したデータベースへの接続」

この章では、Recovery Manager の起動方法、ターゲット、リカバリ・カタログおよび補助データベースへの接続方法を説明します。

### 第 3 章「クイック・スタート」

この章では、Recovery Manager の基本機能の使用について説明します。

## 第 II 部「アーキテクチャおよび概念」

この部では、Recovery Manager の機能について、詳しい概念を説明します。

### 第 4 章「Recovery Manager のアーキテクチャ」

この章では、Recovery Manager の環境の基本概念およびコマンド・インタフェースの性質について説明します。

### 第 5 章「概念 I: チャンネル、バックアップおよびコピー」

この章では、Recovery Manager の手動および自動のチャンネルについて、また BACKUP および COPY コマンドの使用方法を説明します。

### 第 6 章「概念 II: リストア、リカバリおよび複製」

この章では、RESTORE、RECOVER および DUPLICATE コマンドの使用方法を説明します。

### 第 7 章「概念 III: メンテナンス」

この章では、レポートの生成、バックアップでのメンテナンスの実行、ファイルおよびスクリプトのリカバリ・カタログへの格納などに使用される、様々なコマンドについて説明します。

### 第 8 章「Recovery Manager 環境の構成」

この章では、Recovery Manager 環境の設定と構成を説明します。

## 第 III 部「Recovery Manager を使用したバックアップおよびリカバリの実行」

この部では、バックアップおよびリカバリ操作の実行方法を説明します。

### 第 9 章「Recovery Manager でのバックアップおよびコピーの作成」

この章では、BACKUP および COPY コマンドの使用方法を説明します。

### 第 10 章「Recovery Manager によるリストアとリカバリ」

この章では、RESTORE および RECOVER コマンドの使用方法を説明します。

### 第 11 章「Recovery Manager による表領域の Point-in-Time リカバリの実行」

この章では、データベースの残り部分に影響を与えずに、1 つ以上の表領域を現時点以外の時間にリカバリする方法を説明します。

### 第 12 章「Recovery Manager を使用したデータベースの複製」

この章では、DUPLICATE を使用したターゲット・データベースのコピーの作成方法を説明します。

### **第 13 章「Recovery Manager によるスタンバイ・データベースの作成」**

この章では、DUPLICATE を使用したスタンバイ・データベースの作成方法を説明します。

### **第 14 章「Recovery Manager のチューニング」**

この章では、Recovery Manager によるバックアップとリストアのパフォーマンスを改善するヒントを提供します。

### **第 15 章「トラブルシューティング」**

この章では、Recovery Manager の問題に対する診断と対応について、ヒントを提供します。

## **第 IV 部「Recovery Manager のリポジトリのメンテナンス」**

この部では、Recovery Manager のリポジトリの作成方法を説明します。また、リポジトリへの問合せ方法、リポジトリに記録されたバックアップのメンテナンス方法についても説明します。

### **第 16 章「Recovery Manager のリポジトリの管理」**

この章では、リカバリ・カタログの作成および管理方法を説明します。リカバリ・カタログを使用しない Recovery Manager の実行方法についても説明します。

### **第 17 章「Recovery Manager のリポジトリの問合せ」**

この章では、リポジトリに格納された情報へのアクセス方法を説明します。

### **第 18 章「Recovery Manager のメンテナンスの実行」**

この章では、バックアップおよびコピーに対して操作する CHANGE、CROSSCHECK および DELETE などのメンテナンス・コマンドの使用方法について説明します。



## 関連文書

さらに詳しい情報は、次の Oracle リソースを参照してください。

- 『Oracle9i Recovery Manager リファレンス』
- 『Oracle9i バックアップおよびリカバリ概要』
- 『Oracle9i ユーザー管理バックアップおよびリカバリ・ガイド』
- 『Oracle9i データベース・ユーティリティ』
- <http://www.oracle.com/database/recovery>

次の場所で、Backup Solutions Program に関する情報にアクセスできます。

<http://otn.oracle.com/deploy/availability>

マニュアル・セットの多くでは、Oracle のインストール時にデフォルトでインストールされるシード・データベースのサンプル・スキーマを使用しています。これらのスキーマがどのように作成されているか、およびその使用方法については、『Oracle9i サンプル・スキーマ』を参照してください。

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

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

OTN-J のユーザー名とパスワードを取得済みであれば、次の OTN-J Web サイトの文書セクションに直接接続できます。

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

# 表記規則

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

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

## 本文の表記規則

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

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

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

### コード例の表記規則

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

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

次の表は、コード例の記載上の表記規則と使用例を示しています。

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

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

## Windows オペレーティング・システムの表記規則

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

表記規則	意味	例
「スタート」→を選択	プログラムの起動方法。	Oracle Database Configuration Assistant を起動するには、「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Configuration and Migration Tools」→「Database Configuration Assistant」を選択します。
ファイル名とディレクトリ名	ファイル名とディレクトリ名には、大 / 小文字区別はありません。特殊文字のうち、左山カッコ (<)、右山カッコ (>)、コロン (:)、二重引用符 (")、スラッシュ (/)、パイプ ( ) およびハイフン (-) は使用できません。特殊文字の円記号 (¥) は、引用符で囲まれていても要素のセパレータとして扱われます。¥¥ で始まるファイル名は、Windows では汎用命名規則を使用しているものとみなされます。	c:¥winnnt"¥"system32 は C:¥WINNT¥SYSTEM32 と同じです。

表記規則	意味	例
C:¥>	現行のハード・ディスク・ドライブの Windows コマンド・プロンプトを表します。コマンド・プロンプトのエスケープ文字はカレット (^) です。プロンプトには、作業中のサブディレクトリが反映されます。このマニュアルでは、コマンド・プロンプトと呼んでいます。	C:¥oracle¥oradata>
特殊文字	Windows コマンド・プロンプトでは、特殊文字である円記号 (¥) は、二重引用符 (") のエスケープ文字として必要な場合があります。カッコと一重引用符 (') には、エスケープ文字は不要です。エスケープ文字と特殊文字の詳細は、Windows オペレーティング・システムのマニュアルを参照してください。	C:¥>exp scott/tiger TABLES=emp QUERY=¥"WHERE job='SALESMAN' and sal<1600¥"  C:¥>imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept)
HOME_NAME	Oracle ホーム名を表します。ホーム名は、英数字で 16 文字以内です。ホーム名に使用できる特殊文字は、アンダースコアのみです。	C:¥> net start OracleHOME_NAME_TNSListener

表記規則	意味	例
<code>ORACLE_HOME</code> と <code>ORACLE_BASE</code>	<p>Oracle8 リリース 8.0 以前では、Oracle コンポーネントをインストールすると、すべてのサブディレクトリはトップレベルの <code>ORACLE_HOME</code> ディレクトリの直下にデフォルトで次のいずれかの名前で置かれていました。</p> <ul style="list-style-type: none"><li>■ Windows NT の場合は <code>C:\orant</code></li><li>■ Windows 98 の場合は <code>C:\orawin98</code></li></ul> <p>このリリースは、Optimal Flexible Architecture (OFA) のガイドラインに準拠しています。すべてのサブディレクトリがトップレベルの <code>ORACLE_HOME</code> ディレクトリの直下にあるとはかぎりません。<code>ORACLE_BASE</code> と呼ばれるトップレベル・ディレクトリがあり、デフォルト名は <code>C:\oracle</code> です。最新の Oracle リリースを他の Oracle ソフトウェアがインストールされていないコンピュータにインストールすると、最初の Oracle ホーム・ディレクトリのデフォルト設定は <code>C:\oracle\orann</code> となります。<code>nn</code> は最新リリース番号です。Oracle ホーム・ディレクトリは、<code>ORACLE_BASE</code> の直下にあります。</p> <p>このマニュアルでは、すべてのディレクトリ・パスが OFA の表記規則に従っています。</p>	<p><code>%ORACLE_HOME%\rdbms\admin</code> ディレクトリにアクセスします。</p>

---

# Recovery Manager の新機能

この項では、Oracle9i リリース 2 (9.2) の Recovery Manager の新機能について説明し、追加情報へのポインタを提供します。また、前リリースからの新機能情報も保存されているので、ユーザーが Oracle8i より前のリリースから現行のリリースへ移行するのに役立ちます。

次の項では、Recovery Manager の新機能について説明します。

- [Oracle9i の Recovery Manager の新機能](#)
- [Oracle8i の Recovery Manager の新機能](#)

# Oracle9i の Recovery Manager の新機能

この項では、次の項目について説明します。

- [Oracle9i リリース 2 \(9.2\) の Recovery Manager の新機能](#)
- [Oracle9i リリース 1 \(9.0.1\) の Recovery Manager の新機能](#)

## Oracle9i リリース 2 (9.2) の Recovery Manager の新機能

Oracle9i リリース 2 (9.2) の機能および拡張は、一段と扱いやすくなり、機能も大幅に拡張されています。

### ■ サーバー・パラメータ・ファイルのバックアップ

BACKUP コマンドを実行すると、現行のサーバー・パラメータ・ファイルをバックアップできます。Recovery Manager では、バックアップ・セットに現行の制御ファイルが含まれている場合に、現行のサーバー・パラメータ・ファイルが自動的にバックアップされます。サーバー・パラメータ・ファイルが消失した場合は、それを使用せずにインスタンスを起動し、RESTORE SPFILE を実行してリストアできます。

**関連項目：** BACKUP の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。制御ファイルの自動バックアップ機能を使用可能にする方法は、8-18 ページの「[制御ファイルとサーバー・パラメータ・ファイルの自動バックアップの構成](#)」を参照してください。

### ■ データベース構造を変更した後の制御ファイルの自動バックアップ

CONFIGURE CONTROLFILE AUTOBACKUP を ON にすると（デフォルトでは OFF）、データベース構造が変更された後に、制御ファイルとサーバー・パラメータ・ファイルが自動的にバックアップされます。ターゲット・データベースでは、アラート・ログに自動バックアップが記録されます。

**関連項目：** 制御ファイルの自動バックアップ機能を使用可能にする方法は、8-18 ページの「[制御ファイルとサーバー・パラメータ・ファイルの自動バックアップの構成](#)」を参照してください。



## ■ バックアップを必要とするアーカイブ・ログのバックアップ

BACKUP ARCHIVELOG コマンドの NOT BACKED UP *integer* TIMES 句を使用すると、バックアップ回数が *integer* 回以下のログのみをバックアップできます。ファイルのバックアップ数を計算するときに、Recovery Manager では現行のバックアップと同じデバイス・タイプで作成されたバックアップのみが考慮されます。このオプションを使用すると、指定したメディアにアーカイブ・ログをバックアップできるため便利です (各ログの 3 つ以上のコピーをテープに保存する場合など)。

**関連項目：** BACKUP の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。構文の例は、9-56 ページの「[アーカイブ・ログの週次セカンダリ・バックアップの作成: 例](#)」を参照してください。

## ■ アーカイブ・ログのリストア時の領域管理

RECOVER ... DELETE ARCHIVELOG コマンドの MAXSIZE オプションを使用すると、メディア・リカバリ中にログをリストアするときに Recovery Manager で使用されるディスク領域を制限できます。

**関連項目：** RECOVER の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。構文の例は、10-9 ページの「[デフォルトの場所でのデータベース全体のリストアおよびリカバリ](#)」を参照してください。

## ■ バックアップとコピーの自動タグ

バックアップまたはコピーのタグを指定しない場合は、Recovery Manager によって自動的に割り当てられます。デフォルトの書式には、バックアップ開始日時が含まれます。

**関連項目：** BACKUP コマンドの TAG パラメータのデフォルトの書式については、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ DBNEWID ユーティリティによる DBID またはデータベース名の変更

DBNEWID ユーティリティを使用すると、データベースのデータベース名または DBID を変更できます。Recovery Manager の DUPLICATE コマンドを使用せずにデータベースをコピーすると、オリジナル・データベースとコピーしたデータベースに同じ DBID が共有されます。Recovery Manager では、データベースは DBID で区別されるため、同じ DBID を持つ複数のデータベースを 1 つのリカバリ・カタログに登録することはできません。DBNEWID を使用して、コピーしたデータベースの DBID 値を変更すると、同じリカバリ・カタログに登録できます。

**関連項目：** DBNEWID ユーティリティの詳細は、『Oracle9i データベース・ユーティリティ』を参照してください。

## ■ 新規の V\$ ビュー

V\$DATABASE\_BLOCK\_CORRUPTION ビューでは、最新のバックアップ、バックアップの妥当性チェックまたはファイルのコピー後に、ファイル内の破損ブロックが記録されます。CORRUPTION\_TYPE 列は、破損のタイプを示します。CORRUPTIONLIST 句を指定して BLOCKRECOVER コマンドを実行すると、ビューに記録された破損ブロックがすべてリカバリされます。破損ブロックが修復されても、影響を受けるファイルが Recovery Manager によって次回にバックアップされるまで、破損を記述する行はビューに残ります。V\$DATABASE\_BLOCK\_CORRUPTION ビューには、RC\_DATABASE\_BLOCK\_CORRUPTION 対応するリカバリ・カタログ・ビューがあります。

V\$DATABASE\_INCARNATION ビューには、データベースのすべてのインカネーションがリストされます。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びに、新規インカネーションが作成されます。V\$DATABASE\_INCARNATION ビューに対応するリカバリ・カタログ・ビューには、RC\_DATABASE\_INCARNATION があります。

**関連項目：** 概要については、6-11 ページの「[Recovery Manager を使用したブロック・メディア・リカバリ](#)」を参照してください。BLOCKRECOVER の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ Real Application Clusters 用のデフォルトの自動位置検索

Recovery Manager では、バックアップまたはリストア対象のファイルがアクセスできる Oracle Real Application Clusters 構成のノードが自動的に検出されます。次のファイルの位置が自動的に特定されます。

- バックアップまたはリストア中のバックアップ・ピース
- バックアップ中のアーカイブ REDO ログ
- バックアップまたはリストア中のデータ・ファイルまたは制御ファイルのコピー

割り当てられているチャンネルの PARMS または CONNECT 文字列が異なる場合は、必ず自動位置検索機能が使用可能になります。

リリース 2 (9.2) より前の Oracle9i では、SET AUTOLOCATE を使用してこのオプションを手動で使用可能にする必要があり、このオプションはバックアップ・ピースにのみ適用されていました。

**関連項目：** 6-5 ページ「[Real Application Clusters でリストアする場合の自動的なバックアップ位置検索](#)」

## ■ メディア・マネージャのファンクション・コールの診断

動的パフォーマンス・イベント・ビューを問い合わせ、Recovery Manager からメディア・マネージャへのコールに関する診断データを取得できるようになりました。イベント名は、各メディア管理ファンクションに対応します。これらのイベント名を使用すると、Recovery Manager でのバックアップ、リストアおよびメンテナンス・ジョブの実行中に問題を診断できます。

**関連項目：** 15-23 ページ「[Recovery Manager とメディア・マネージャの相互作用の監視](#)」

## ■ DELETE コマンドの FORCE オプション

Recovery Manager リポジトリ内のオブジェクトの状態に、メディア上のオブジェクトの状態が反映されない場合があります。たとえば、CROSSCHECK を実行する前に、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バックアップ・ピースを削除したとします。オブジェクトは AVAILABLE または EXPIRED としてリストされますが、メディアでの状態は異なり、そのオブジェクトに対して DELETE コマンドを実行しても、Recovery Manager では削除されません。この動作は、DELETE コマンドの FORCE オプションを使用してオーバーライドできます。

**関連項目：** 7-15 ページ「[リポジトリとメディアが対応していない場合の DELETE コマンドの動作](#)」

## ■ デバイスにバックアップ済みのファイルの削除

デバイスへのバックアップが指定した回数に達したファイルを削除できます。たとえば、2 回以上テープにバックアップされたアーカイブ REDO ログをすべて削除できます。

**関連項目：** 7-14 ページ「[バックアップ済みのアーカイブ REDO ログの削除](#)」

## ■ DUPLICATE の拡張

DUPLICATE コマンドの SKIP TABLESPACE オプションを使用すると、複製データベースから表領域のリストを除外できます。また、DUPLICATE コマンドで UNTIL 句を指定すると、複製データベースを現時点以外の時刻までリカバリできます。

**関連項目：** 12-4 ページ「[データベース複製オプション](#)」

## ■ Recovery Manager のエラー出力の改善

Recovery Manager のエラー出力が簡潔でわかりやすくなりました。Recovery Manager コマンドが失敗すると、エラー・スタックに続いて、失敗したコマンドを示す RMAN-03002 または RMAN-03009 が常に出力されます。エラーがターゲット・データベースから生成されたものであっても、Recovery Manager ではそのことは明示的に示されませんが、エラーがカタログまたは補助データベースから生成されたものである場合は、そのことが別のメッセージに示されます。

## Oracle9i リリース 1 (9.0.1) の Recovery Manager の新機能

Oracle9i リリース 1 (9.0.1) の機能および拡張機能は、一段と扱いやすくなり、機能も大幅に拡張しています。

### ■ 永続的な Recovery Manager の構成

CONFIGURE コマンドにより、あらゆるセッションに適用できる、永続的な Recovery Manager の設定が作成されます。自動チャネル、チャネルの平行化、保存方針、バックアップ・オプション、補助ファイル名などを含む、様々な機能を構成できます。

**関連項目：** 構成の手順については、第 8 章「[Recovery Manager 環境の構成](#)」を参照してください。CONFIGURE の構文は『Oracle9i Recovery Manager リファレンス』を参照してください。

### ■ 自動チャネルの構成

CONFIGURE コマンドは、チャネルの永続的な設定を格納します。BACKUP または COPY のようなコマンドを Recovery Manager プロンプトから実行するとき、Recovery Manager はこれらのチャネルを自動的に割り当てます。手動でチャネルを割り当てる必要はありません。

CONFIGURE コマンドを使用して、平行化、二重化、チャネル制御オプションなどのバックアップの属性を定義します。これらの設定は、RUN コマンド内で手動でチャネルを割り当てない場合、チャネルを必要とするコマンドすべてが使用します。

**関連項目：** 概要については、5-4 ページの「[自動チャネルおよび手動チャネルの割当て](#)」を参照してください。CONFIGURE の構文は『Oracle9i Recovery Manager リファレンス』を参照してください。

### ■ Recovery Manager プロンプトから使用可能なコマンドの増加

これまで、バックアップおよびリカバリのコマンドのほとんどは、RUN コマンド内で実行されていました。このバージョンでは、BACKUP、COPY および RESTORE など、多くのコマンドが Recovery Manager プロンプトから実行できます。

## ■ 保存方針

保存方針は、制御ファイルおよびデータ・ファイルの保存期間を指定し、保存されるバックアップ数も指定する構成です。保存方針は、バックアップのリカバリに必要ななくなったために不要と判断されたアーカイブ・ログがどれであるかも判断します。

REPORT OBSOLETE および DELETE OBSOLETE コマンドは、保存方針で指定された基準を使用して、不要とされたログを判断します。保存方針は、次の方法で指定できます。

- － リカバリ・ウィンドウの使用。リカバリ・ウィンドウは、現時点から過去にさかのぼって、どの時点でもリカバリが可能な期間を定義します。
- － 冗長性の使用。冗長性は、指定した数より多いバックアップまたはコピーは維持する必要がないことを定義します。

**関連項目：** 概要については、5-55 ページの「[バックアップの保存方針](#)」を参照してください。CONFIGURE の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ 長期バックアップの保存方針からの除外

BACKUP コマンドの KEEP オプションは、保存方針から除外するマークをバックアップに付けます。REPORT OBSOLETE および DELETE OBSOLETE コマンドは、KEEP オプションに指定した期限が過ぎた場合にのみ、バックアップを不要と判断します。CHANGE ... KEEP コマンドで、バックアップの状態を変更できます。

**関連項目：** 概要については、5-61 ページの「[バックアップの保存方針からの除外](#)」を参照してください。BACKUP コマンドの構文および CHANGE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ 制御ファイルの自動バックアップおよびリストア

CONFIGURE CONTROLFILE BACKUP を ON に設定する場合、Recovery Manager は BACKUP あるいは COPY コマンドの実行後に、必ず自動的に制御ファイルの自動バックアップを作成します。Recovery Manager はこのバックアップに、デフォルト名を付けます (CONFIGURE CONTROLFILE BACKUP FORMAT を使用して変更可能)。これにより、障害時にリカバリ・カタログおよび現行の制御ファイルを消失した場合でも、Recovery Manager はこの制御ファイルをリストアできます。

**関連項目：** 概要については、5-52 ページの「[制御ファイルとサーバー・パラメータ・ファイルの自動バックアップ](#)」を参照してください。

## ■ ブロック・メディア・リカバリ

ブロック・メディア・リカバリは、データ・ファイルをオンラインで改名している間に、データ・ファイルの個別のブロックでメディア・リカバリを実行できます。V\$DATABASE\_BLOCK\_CORRUPTION ビューは、最新のバックアップあるいは各データ・ファイルのイメージ・コピーにある破損ブロックを表示します。

**関連項目：** 概要については、6-11 ページの「[Recovery Manager を使用したブロック・メディア・リカバリ](#)」を参照してください。BLOCKRECOVER の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ アーカイブ REDO ログの改善

次に示す方法で、Recovery Manager はアーカイブ REDO ログに対する Recovery Manager 自体の処理を拡張します。

- BACKUP ... PLUS ARCHIVELOG コマンドは、ログ・スイッチを実行して現行のオンライン・ログをアーカイブし、アーカイブ・ログをバックアップした後、もう一度ログ・スイッチを実行して他のログをバックアップします。この操作によって、バックアップされたデータ・ファイルを一貫性のある状態にリカバリすることが保証されます。
- アーカイブ・ログのバックアップを開始した時点で、Recovery Manager は必要に応じて、自動的に現行のオンライン REDO ログをオフに切り替え、アーカイブします。
- BACKUP ARCHIVELOG コマンドがバックアップするログを検出しなかった場合、Recovery Manager はエラーを発行しません。
- Recovery Manager はアーカイブ・ログのフェイルオーバーを自動的に実行します。バックアップ中に、Recovery Manager がログの破損あるいは消失を検出した場合、Recovery Manager はリポジトリにリストされたすべてのログを、バックアップの代替候補とみなします。
- BACKUP ARCHIVELOG コマンドの DELETE ALL INPUT オプションは、指定された基準に一致するすべてのログを削除します。この方法で、複数のアーカイブ先にあるログを削除できます。
- V\$ARCHIVED\_LOG の BACKUP\_COUNT 列は、アーカイブ・ログのバックアップされた回数を表示します。

**関連項目：** 概要については、5-17 ページの「[アーカイブ・ログのバックアップ](#)」を参照してください。BACKUP の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ LIST の拡張

このバージョンでは、LIST コマンドの構文と、CHANGE、CROSSCHECK および DELETE コマンドの構造が、一段と類似しています。LIST 出力の表示をこれまで以上に制御することが可能になりました。

**関連項目：** 概要については、7-2 ページの「[LIST コマンドの出力](#)」を参照してください。LIST の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ CROSSCHECK の拡張

CROSSCHECK コマンドの機能が拡張され、CHANGE ... CROSSCHECK コマンドは、使用されなくなりました。このバージョンでは、CROSSCHECK コマンドと、DELETE および CHANGE コマンドの構文が類似しています。

**関連項目：** 概要については、7-8 ページの「[Recovery Manager のバックアップおよびコピーのクロスチェック](#)」を参照してください。CROSSCHECK の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ CHANGE の拡張

CHANGE コマンドは、ファイルを使用可能または使用不能に変更します。また、ファイルをカタログに追加したり、カタログから削除することもできます。さらに KEEP の設定も変更できますが、これはバックアップ・リポジトリが制御ファイルカリカバリ・カタログかに関係なく行うことができます（カタログを必要とする CHANGE ... KEEP FOREVER を除く）。CHANGE コマンドが操作するファイルのタイプが増えました。また、このバージョンでは、DELETE および CROSSCHECK コマンドと、構文が類似しています。

**関連項目：** 概要については、7-18 ページの「[Recovery Manager のバックアップおよびコピーの可用性の変更](#)」を参照してください。CHANGE の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ DELETE の拡張

DELETE コマンドには、前バージョンの CHANGE ... DELETE と DELETE EXPIRED コマンドの機能が含まれています。さらに、DELETE OBSOLETE コマンドを実行して、不要になったすべてのファイルを削除することもできます。このバージョンでは、DELETE コマンドと、CHANGE および CROSSCHECK コマンドの構文が類似しています。

デフォルトでは、DELETE EXPIRED および DELETE OBSOLETE を使用すると、ファイルの削除の前に確認のためのプロンプトが表示されますので、注意してください。

Oracle9i より前のリリースでは、DELETE EXPIRED の実行時に、プロンプトが表示されませんでした。

**関連項目：** 概要については、7-10 ページの「[Recovery Manager のバックアップおよびコピーの削除](#)」を参照してください。DELETE の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ バックアップ・セットのバックアップ

このバージョンでは、バックアップ・セットをディスクからテープまたはディスクからディスクにバックアップできます。バックアップ中に、破損ブロックまたはバックアップ・ピースの消失を Recovery Manager が検出した場合、Recovery Manager は自動的に、異常のない既存のコピーにフェイルオーバーします。

**関連項目：** 概要については、5-28 ページの「[バックアップ・セットのバックアップ](#)」を参照してください。BACKUP の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ 二重化機能の拡張

バックアップ・セットを生成するときには、異なる FORMAT 文字列を最大 4 個まで指定できます。そのため、多重化されたコピーは、異なるファイル名を持つことができます。

次のコマンドでは COPIES パラメータを指定して、二重化を使用可能にできます。CONFIGURE、SET および BACKUP コマンド。

**関連項目：**

- 概要については、5-61 ページの「[二重バックアップ・セット](#)」を参照してください。
- BACKUP コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- CONFIGURE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- SET コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。



## ■ バックアップの最適化

Recovery Manager の永続的な設定を変更して、バックアップ済みファイルのバックアップをスキップすることができます。この方法により、アーカイブ・ログなど変更が発生しないファイルを複数回バックアップすることを回避できます。この機能は、BACKUP コマンドの FORCE オプションを使用して変更できます。

**関連項目：** 概要については、5-62 ページの「[バックアップの最適化](#)」を参照してください。CONFIGURE の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ 再起動可能なバックアップ

BACKUP コマンドの NOT BACKED UP SINCE 句を使用して、指定した時間の後にバックアップされなかったファイルのみをバックアップすることができます。したがって、バックアップが途中で失敗した場合、バックアップを再起動して、前回バックアップされなかったファイルのみをバックアップすることができます。

**関連項目：** 概要については、5-68 ページの「[再起動可能なバックアップ](#)」を参照してください。BACKUP の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ リストアの最適化

Recovery Manager は、ファイルをリストアする際に、ディスク上のファイルをチェックして、リストアが必要かどうかを判断します。必要がない場合、Recovery Manager はそのファイルをリストアしません。したがって、リストアが途中で失敗した場合、リストアを再起動して、前回リストアされなかったファイルのみをリストアすることができます。この動作は、FORCE オプションで変更可能です。

**関連項目：** 概要については、6-5 ページの「[リストアの最適化](#)」を参照してください。RESTORE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ 複数のブロック・サイズのサポート

Oracle では、1つのデータベース内で、表領域ごとに、異ったブロック・サイズを使用できるようになっています。Recovery Manager は同じ BACKUP コマンドで、ブロック・サイズの異なる表領域をバックアップします。ただし、1つのバックアップ・セット内では、異なるブロック・サイズを持つデータ・ファイルは混合されません。各バックアップ・セットには、1つのブロック・サイズのファイルのみが含まれています。

## ■ バックアップからの表領域の除外

Recovery Manager を構成して、指定された表領域をデータベースのバックアップ全体から除外することができます。

**関連項目：** 概要については、5-68 ページの「[データベース全体のバックアップから除外する表領域の構成](#)」を参照してください。CONFIGURE の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ デフォルトの NOCATALOG モード

CATALOG または NOCATALOG をコマンドラインで指定しない場合、および CONNECT CATALOG をプロンプトから実行しない場合、リポジトリを必要とするコマンドを実行すると、Recovery Manager はデフォルトの NOCATALOG モードになります。

**関連項目：** 概要については、2-4 ページの「[データベースに接続しない Recovery Manager の起動](#)」を参照してください。

## ■ Recovery Manager のメッセージ出力の改善

Recovery Manager の接頭辞は、非エラー・メッセージから削除されています。

## ■ SBT ライブラリ・アーキテクチャ

新しい SBT\_LIBRARY パラメータは、Oracle Media Management (SBT) API を使用したサードパーティのメディア・マネージャをサポートするプラットフォーム上で、Recovery Manager が DEVICE TYPE sbt のチャンネルで使用するメディア管理ライブラリを制御します。ALLOCATE CHANNEL または CONFIGURE CHANNEL コマンドの PARMS 設定で、SBT\_LIBRARY パラメータに共有ライブラリのファイル名を指定し、ロードされるようにします。SBT\_LIBRARY=oracle.diskssbt を指定して、Oracle のディスク sbt ライブラリ（旧称はダミー API）をサーバーにロードさせることもできます。

SBT\_LIBRARY パラメータが PARMS で指定されていない場合、Oracle サーバーは Windows NT で orasbt.dll、UNIX では libobk.so ダイナミック・ライブラリをロードしようとします。Oracle でライブラリの位置を特定できない場合は、サーバーがエラーを戻します。

**関連項目：** メディア・マネージャへのリンクについては、8-2 ページの「[メディア・マネージャにバックアップを作成する場合の Recovery Manager の構成](#)」を参照してください。PARMS の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

# Oracle8i の Recovery Manager の新機能

この項で説明している Oracle8i の機能および拡張機能は、8.0 の Recovery Manager の機能を大幅に拡張しています。これらの機能の中には、一部の Oracle8i リリースにのみ含まれるものもあります。

- **Recovery Manager によるスタンバイ・データベースのサポート**

Oracle8i リリース 8.1.7 では、Recovery Manager を使用してスタンバイ・データベースを作成し、バックアップすることが可能です。

**関連項目：** [第 13 章「Recovery Manager によるスタンバイ・データベースの作成」](#)

- **テスト・バックアップの実行**

Oracle8i リリース 8.1.7 では、BACKUP VALIDATE コマンドで、実際に出力ファイルは作成せずにバックアップをテストします。

**関連項目：** 概要については、5-75 ページの「[Recovery Manager を使用したバックアップのテスト](#)」を参照してください。構文については、『Oracle9i Recovery Manager リファレンス』を参照してください。

- **Recovery Manager のスナップショット制御ファイルのデフォルトの場所を設定**

Oracle8i リリース 8.1.7 では、SET SNAPSHOT CONTROLFILE LOCATION TO DEFAULT コマンドを使用して、スナップショット制御ファイルに使用されるデフォルトのファイル名を設定できます。デフォルト値はプラットフォーム固有で、Oracle ホームの場所に依存します。

- **複数のチャネルでのクロスチェックおよび削除**

Oracle8i リリース 8.1.7 では、Recovery Manager のメンテナンス・コマンドを発行する前に、複数のメンテナンス・チャネルを割り当てることができます。Recovery Manager は、バックアップの作成に使用されるチャネルと同じデバイス・タイプのチャネルを、すべてチェックします。

**関連項目：** [7-17 ページ「Recovery Manager のメンテナンス・コマンドへの複数チャネルの割当て」](#)

## ■ カタログに必要な少数のメンテナンス・コマンド

Oracle8i リリース 8.1.7 では、リカバリ・カタログを使用する必要があるメンテナンス・コマンドは少数です。リカバリ・カタログを必要とする CHANGE コマンドの唯一のオプションを次に示します。

- CHANGE ... AVAILABLE (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・コピーにはリカバリ・カタログを使用する必要がないことに注意)
- CHANGE ... UNAVAILABLE

Oracle8i リリース 8.1.7 では、CROSSCHECK および DELETE EXPIRED コマンドにリカバリ・カタログは必要ありません。

## ■ 不要なリカバリ・カタログの互換性のレベルの設定

Oracle8i リリース 8.1.6 では、CONFIGURE COMPATIBLE コマンドで、リカバリ・カタログの互換性レベルを設定します。このコマンドは、Recovery Manager がカタログ・レコードをアップデートおよび削除する際に使用した方法に起因する問題の解決に役立ちます。Oracle8i リリース 8.1.7 では、これらの問題が解決しています。そのため、CONFIGURE COMPATIBLE コマンドは、互換性を維持するために使用可能ですが、効果はありません。

## ■ Oracle Real Application Clusters ノードでの自動的なバックアップ位置検索

Oracle8i リリース 8.1.6 では、SET コマンドの AUTOLOCATE オプションで、Oracle Real Application Clusters 構成の、リストアするバックアップを含んでいるノードを自動的に検出できます。

**関連項目：** SET コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ リカバリ・カタログの互換性の設定

Oracle8i リリース 8.1.6 では、CONFIGURE COMPATIBLE コマンドで、Recovery Manager の実行可能ファイルを使用して、リカバリ・カタログ・パッケージの互換性を制御します。

## ■ ALTER DATABASE RESETLOGS コマンド

Oracle8i リリース 8.1.6 では、ALTER DATABASE コマンドの RESETLOGS オプションで、データベースをオープンし、オンライン REDO ログをリセットします。

**関連項目：** ALTER DATABASE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ 削除コマンドによるリポジトリ・レコードの削除

Oracle8i リリース 8.1.6 では、CHANGE ... DELETE、DELETE EXPIRED および BACKUP ... DELETE INPUT コマンドは、カタログ・レコードのステータスを DELETED に更新するのではなく、削除します。

## ■ Media Management API バージョン 2.0

オラクル社は、Media Management API バージョン 2.0 をリリースしました。バージョン 1.1 の Media Management API へのサポートは継続されます。新しい API の機能は、次のとおりです。

- メディア管理ベンダーのソフトウェアは、**プロキシ・コピー**と呼ばれる拡張機能を使用して、バックアップまたはリストアに関連するデータ移動を実行できます。
- 一部のメディア管理ソフトウェアでは、メディア・タイプ、保存期間などの条件に基づいて、バックアップ・メディアをストレージ・プールに配置できます。BACKUP コマンドの POOL パラメータにより、それらの製品と Recovery Manager との統合が可能になります。
- チャンネルが割り当てられると、Recovery Manager は、そのチャンネルでのバックアップ取得に使用するメディア管理製品を識別するメッセージを、ログ内に表示します。メディア・マネージャがエラーを検出すると、そのエラーを説明するエラー・メッセージを返し、Recovery Manager のログに表示します。
- 新しい SEND コマンドを使用して、Recovery Manager セッションからメディア管理ソフトウェアに直接コマンドを送信できます。

**関連項目：** BACKUP の構文については、『Oracle9i Recovery Manager リファレンス』を参照してください。SEND 構文については、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ クロスチェックおよび削除機能

CROSSCHECK BACKUP、CHANGE ... CROSSCHECK および DELETE EXPIRED BACKUP コマンドは、メディア・マネージャのカタログを使用したリカバリ・カタログの同期を可能にします。Recovery Manager は、バックアップおよびコピーがディスクまたはテープ上に存在するかどうかを判断し、存在しない場合は、そのリポジトリ・レコードを更新します。

### 関連項目：

- CHANGE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- CROSSCHECK コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- DELETE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ LIST 出力の改善

LIST BACKUP コマンドは、バックアップ・セットに含まれるバックアップのリストをデータ・ファイルやアーカイブ・ログとは別のレポート・セクションに出力します。

**関連項目：** LIST コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ 必要なバックアップのレポート

新しいコマンドの REPORT NEED BACKUP REDUNDANCY は、データ・ファイルのバックアップ数がユーザー指定の数より少ない場合、新しいバックアップが必要なことをユーザーに警告します。

**関連項目：** REPORT コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ CREATE CATALOG コマンド

CREATE CATALOG コマンドは、リカバリ・カタログを作成します。このコマンドは、`*/dbs/admin` ディレクトリにある `catrman.sql` および関連するスクリプトを置き換えます。

**関連項目：** CREATE CATALOG コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ UPGRADE CATALOG コマンド

以前は、リカバリ・カタログをアップグレードするために、SQL スクリプトを実行する必要がありました。このバージョンでは、UPGRADE CATALOG コマンドを使用して、カタログをアップグレードできます。

**関連項目：** UPGRADE CATALOG コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ■ DROP CATALOG コマンド

DROP CATALOG コマンドは、リカバリ・カタログのスキーマを削除します。

**関連項目：** DROP CATALOG コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

- **STARTUP、SHUTDOWN および ALTER DATABASE コマンド**

Recovery Manager の STARTUP、SHUTDOWN および ALTER DATABASE コマンド (MOUNT および OPEN オプションのみ) には、同等の SQL\*Plus コマンドと同じ構文があります。

**関連項目：**

- STARTUP コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- SHUTDOWN コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- ALTER DATABASE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

- **データベースの複製**

DUPLICATE コマンドにより、別のデータベースのバックアップを使用して新規にデータベースを作成できます。

**関連項目：** [第 12 章「Recovery Manager を使用したデータベースの複製」](#)

- **Oracle Real Application Clusters でのディスク・アフィニティ**

Oracle Real Application Clusters 構成の複数のノードでバックアップを行う場合、一部のディスクにはクラスタ内の特定のノードとのアフィニティがあるため、同じクラスタ内の他のノードからのアクセスより速くなることがあります。Recovery Manager は、ノードに対するアフィニティがあればそれを認識し、データ・ファイルとのアフィニティを持つノードに割り当てられたチャンネルでデータ・ファイルのバックアップを行うようにスケジュールします。

- **二重化バックアップ**

Recovery Manager では、各バックアップ・ピースのコピーを 4 つまで作成できます。

**関連項目：** [5-23 ページ「二重バックアップ・セット」](#)

- **Recovery Manager によって生成されたバックアップ・ピース名**

Recovery Manager で FORMAT パラメータを使用して、バックアップ・ピース名を明示的に指定する必要はありません。Recovery Manager は、デフォルトで各バックアップ・ピースに一意的な名前を選択します。

- **上書きされないバックアップ・ピース**

既存のバックアップ・ピースと同じ名前のバックアップ・ピースを作成しようとした場合、既存のバックアップ・ピースは上書きされません。そのかわりに、Recovery Manager ではエラー・メッセージが出力されます。

- **カタログ不要の TSPITR**

リカバリ・カタログなしで TSPITR（表領域の Point-in-Time リカバリ）を実行できます。

**関連項目：** 11-5 ページ「[リカバリ・カタログを使用しない TSPITR の実行](#)」

- **バックアップの実行を監視する新しい V\$ ビュー**

2 つの新しいビュー、V\$BACKUP\_SYNC\_IO および V\$BACKUP\_ASYNC\_IO を使用して、Recovery Manager のバックアップの進行状況およびパフォーマンスを監視できます。

**関連項目：** 14-16 ページ「[手順 8: V\\$ ビューを問い合わせでボトルネックを識別する](#)」



# 第 I 部

---

## Recovery Manager のスタート・ガイド

第 I 部では、Recovery Manager の一般的な概要および Recovery Manager を使用したターゲット・データベースへの接続方法を説明します。この部では、次の章について説明します。

- [第 1 章「概要」](#)
- [第 2 章「Recovery Manager を使用したデータベースへの接続」](#)
- [第 3 章「クイック・スタート」](#)



この章では、Recovery Manager ユーティリティの定義と利点を説明します。

この章の構成は、次のとおりです。

- [Recovery Manager](#)
- [Recovery Manager を使用する理由](#)
- [Recovery Manager 環境の概要](#)

## Recovery Manager

Recovery Manager は、データベース・ファイルのバックアップ、リストアおよびリカバリを行うことができる Oracle ユーティリティです。この製品は Oracle サーバーの機能であり、個別にインストールする必要はありません。

Recovery Manager は、データベース・サーバー・セッションを使用してバックアップおよびリカバリを実行するクライアント / サーバー・アプリケーションです。Recovery Manager は、操作に関するメタデータをターゲット・データベースの制御ファイルに、また、必要に応じて Oracle データベースのリカバリ・カタログ・スキーマに格納します。

Recovery Manager は、オペレーティング・システムのプロンプトからコマンドライン実行可能ファイルとして起動するか、Enterprise Manager の GUI を介していくつかの Recovery Manager 機能を使用することが可能です。

## Recovery Manager を使用する理由

本番データベース・システムのほとんどに、バックアップおよびリカバリに関する厳しい要求事項があります。バックアップおよびリカバリを管理している DBA の場合、次のような要件があります。

- バックアップおよびリカバリ操作の複雑さを管理する
- 人為的なミスを可能なかぎり最小に押さえる
- バックアップをスケーラブルで信頼性の高いものにする
- 使用可能なメディア・ハードウェアを使用する
- バックアップを、データベースのサイズではなく、トランザクション変更のサイズと均り合うものにする
- リカバリ時間を、リカバリされたデータの量と均り合うものにする

Oracle リリース 8.0 以上のデータベースには、これらのバックアップおよびリカバリ作業を実行するために、2 つの基本方法があります。

- オペレーティング・システムのコマンドを使用して、バックアップおよびリストア操作を実行し、SQL あるいは SQL\*Plus 文でリカバリを実行する
- Recovery Manager を使用して、バックアップ、リストアおよびリカバリを行う

---

---

**注意：** Recovery Manager は Oracle リリース 8.0 で導入されており、リリース 8.0 より前の Oracle データベースとは互換性がありません。

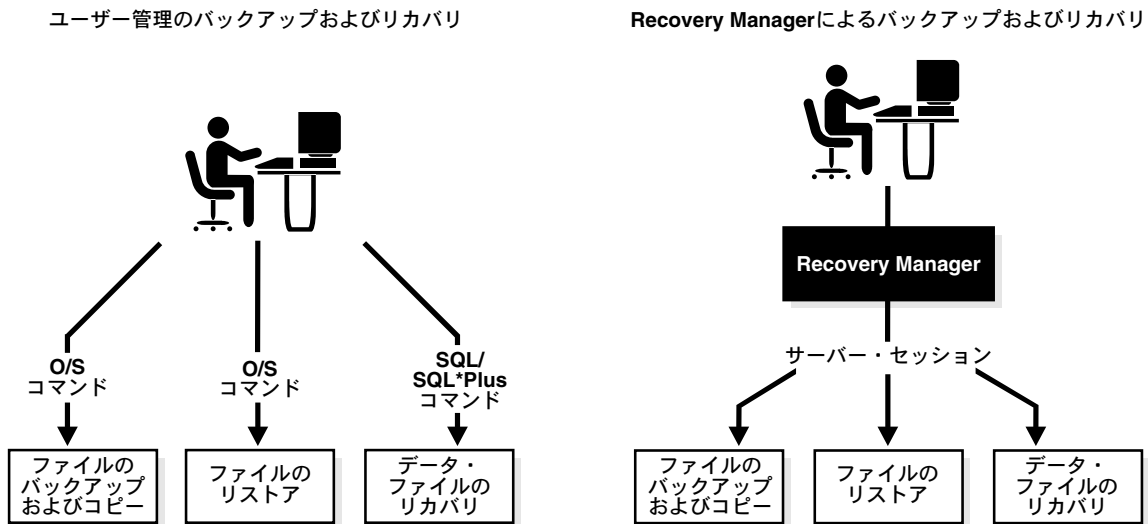
---

---

Recovery Manager での方法を使用する理由を説明します。図 1-1 に示すように、Recovery Manager はサーバー・セッションを使用してバックアップおよびリカバリ操作を実行し、メタデータをリポジトリに格納します。Recovery Manager では、バックアップおよびリカバリは自動化されています。これに対して、ユーザー管理の方法では、ユーザーがすべてのデータベース・ファイルおよびバックアップを追跡し続ける必要があります。たとえば、各データ・ファイルに対応するバックアップの検索、オペレーティング・システムのコマンドを使用した正しい場所へのコピー、適用するログの選択などのタスクを、Recovery Manager は自動的に行います。

Recovery Manager を使用する利点は、Oracle Managed Files を使用すると、特に確実なものになります。Oracle でデータ・ファイル、制御ファイル、オンライン REDO ログの命名および管理を行うと、システムは一段と使用しやすくなります。しかし、ユーザー自身がファイルの命名を行っていないため、様々なデータベース・ファイルのファイル名を追跡し続けることは困難になります。すべての記録は Recovery Manager が維持処理を行うので、Recovery Manager ユーザーがこの問題から被害を受けることはありません。

図 1-1 Recovery Manager の自動化された手順とユーザー管理の手順の比較



自動化による明らかな利点の他に、Recovery Manager は、多くの有用な機能を提供します。  
表 1-1 では、Recovery Manager による方法と、従来のユーザー管理による方法との違いを示しています。

表 1-1 Recovery Manager とユーザー管理の方法の比較

Recovery Manager	ユーザー管理の方法
Media Management API を使用。このため、Recovery Manager は、サード・パーティのメディア管理ソフトウェアとシームレスに動作します。API は 20 以上のベンダーがサポートしています。	公開 API のサポートはありません。
オンラインのファイルをバックアップするとき、Recovery Manager は分裂したデータ・ブロックを再読み込みして、一貫性のある読み込みを行います。バックアップの実行中に、オンライン表領域をバックアップ・モードにする必要はありません。	オンライン表領域をバックアップ・モードにしてから、バックアップを開始する必要があります。また、バックアップの完了後、表領域をこのモードから外す必要があります。オンライン・バックアップの完了後、表領域をバックアップ・モードから外すことを怠ると、データベースのパフォーマンスおよび管理性について、重大な問題が発生する可能性があります。
増分バックアップを実行します。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。増分バックアップを使用して、データベースをリカバリできます。これは、NOARCHIVELOG データベースをリカバリできる、ということです。ただし、一貫性のあるシャットダウンの後でのみ、NOARCHIVELOG データベースの増分バックアップを行うことができます。	変更されたブロックのみでなく、すべてのブロックをバックアップします。NOARCHIVELOG データベースはリカバリできません。
バックアップ中に、各ブロックのチェックサムを計算します。また、バックアップ中あるいはリストア中に、破損ブロックをチェックします。バックアップ中またはリストア中にチェックが行われるときも、実行中の SQL に対して実行される場合も、整合性チェックの多くが正常に実行されます。	エラー・チェックは提供されません。
データ・ファイルのバックアップから、未使用ブロックを除外します。このため、書き込みが行われた経験のあるデータ・ブロックのみが、バックアップに含まれます。	データを含んでいるかどうかに関係なく、すべてのデータ・ブロックを含みます。

表 1-1 Recovery Manager とユーザー管理の方法の比較（続き）

Recovery Manager	ユーザー管理の方法
<p>リポジトリを使用して、重要な情報のレポートを作成します。レポートには、次のものが含まれます。</p> <ul style="list-style-type: none"> <li>■ 指定した時点のデータベース・スキーマ</li> <li>■ バックアップの必要なファイル</li> <li>■ 指定した日数内にバックアップを取っていないファイル</li> <li>■ 冗長なため、またはリカバリで使用不能なため、削除可能なバックアップ</li> <li>■ 現行の永続的な Recovery Manager の設定</li> </ul>	<p>レポートの機能は含みません。</p>
<p>Recovery Manager のスクリプトをリカバリ・カタログへ格納します。</p>	<p>記憶域およびオペレーティング・システム・ベースのスクリプトのメンテナンスが必要です。</p>
<p>テスト目的の本番データベースの複製を簡単に作成できます。または、スタンバイ・データベースを簡単に作成またはバックアップできます。</p>	<p>テストまたはスタンバイ・データベースの作成時には、複雑な手順に従う必要があります。</p>
<p>チェックを実行し、ディスク上のバックアップやメディア・カタログ内のバックアップが引き続き使用可能であるかどうかを確認します。</p>	<p>手動でバックアップの位置を特定し、テストする必要があります。</p>
<p>バックアップおよびリストア操作を自動的にパラレル化します。</p>	<p>バックアップが必要なファイルを判断し、オペレーティング・システムのコマンドをパラレルに発行して、手動でパラレル化する必要があります。</p>
<p>実際にはバックアップやリストアを行わずに、ファイルのバックアップまたはリストアが可能かどうか、テストします。</p>	<p>バックアップのリカバリを試行する前に、実際にバックアップ・ファイルをリストアする必要があります。</p>
<p>アーカイブ・ログのフェイルオーバーを自動的に実行します。バックアップ中に Recovery Manager がログの破損あるいは消失を検出した場合、Recovery Manager はリポジトリにリストされたすべてのログとログのコピーを、バックアップの代替候補とみなします。</p>	<p>バックアップに問題が発生した場合、代替のアーカイブ・ログにフェイルオーバーを実行できません。</p>

## Recovery Manager 環境の概要

Recovery Manager 環境はユーティリティとデータベースから構成され、これらをバックアップ計画やリカバリ計画に活用します。一般的な Recovery Manager の設定では、次のものを利用します。

- Recovery Manager 実行可能ファイル
- 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
- メディア管理ソフトウェア

これらのコンポーネントは、Recovery Manager 実行可能ファイルおよびターゲット・データベースのみを必要とします。Recovery Manager は、ターゲット・データベースの制御ファイルに、自動的にメタデータを格納します。このため、リカバリ・カタログ・データベースはオプションになります。ただし、リカバリ・カタログのメンテナンスは行ってください。別マシン上にカタログを作成し、本番マシンは完全に停止している場合、必要なリストアとリカバリの情報は、すべてカタログに入っています。

この項では、次の項目について説明します。

- [Recovery Manager の実行可能ファイル](#)
- [ターゲット・データベース](#)
- [Recovery Manager リポジトリ](#)
- [Recovery Manager メディア管理インタフェース](#)

## Recovery Manager の実行可能ファイル

Oracle ソフトウェアをインストールすると、Recovery Manager 実行可能ファイルも自動的にインストールされます。インストール場所はプラットフォーム固有であり、一般的には、他の Oracle 実行可能プログラムと同じ場所になります。たとえば、UNIX システムでは、Recovery Manager 実行可能ファイルは \$ORACLE\_HOME/bin にあります。

実行可能ファイルを起動するには、コマンドラインからファイル名を入力します。たとえば、UNIX システムでは、次のように入力します。

```
% rman
```

## ターゲット・データベース

ターゲット・データベースとは、Recovery Manager がバックアップ、リストアまたはリカバリを行うデータベースです。単一のリカバリ・カタログと複数のターゲット・データベースをあわせて使用できます。たとえば、データ・センターがサイズの異なる 10 のデータベースを含んでいると想定します。別のデータ・センターにある単一のリカバリ・カタログを使用して、これらのデータベースすべてのメタデータを管理できます。



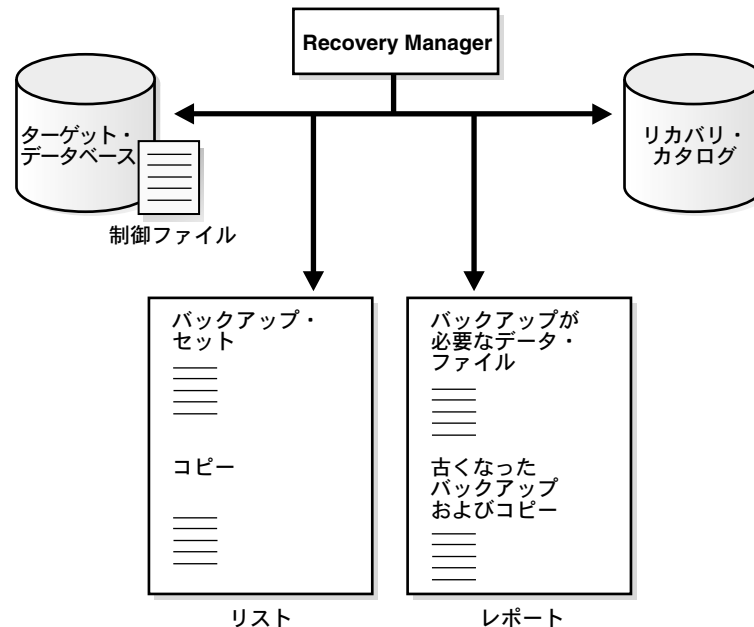
## Recovery Manager リポジトリ

Recovery Manager リポジトリとは、Recovery Manager がターゲット・データベースとそのバックアップおよびリカバリ操作に関する情報を格納するために使用する、メタデータのセットです。特に、Recovery Manager は次のものに関する情報を格納します。

- バックアップ・セットとバックアップ・ピース
- イメージ・コピー（アーカイブ REDO ログを含む）
- プロキシ・コピー
- ターゲット・データベースのスキーマ
- 永続的な構成の設定

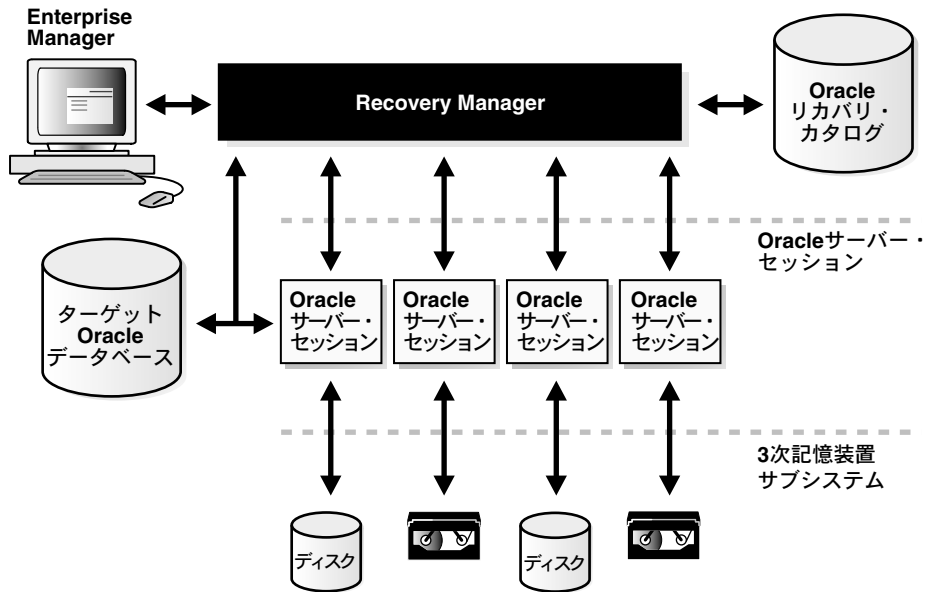
Recovery Manager インタフェースで LIST、REPORT および SHOW コマンドを発行するか、カタログ・ビューで SELECT 文を使用して（リカバリ・カタログを使用している場合のみ）、このメタデータにアクセスできます。Recovery Manager がリストとレポートをどのように出力するかを図 1-2 に示します。

図 1-2 Recovery Manager のリストとレポート



リポジトリは、**リカバリ・カタログ**を作成して格納することも、Recovery Manager によってターゲット・データベースの制御ファイルにのみ格納することもできます。図 1-3 では、Recovery Manager のリカバリ・カタログの使用を示します。

図 1-3 Recovery Manager とオプションのリカバリ・カタログを併用した場合



Recovery Manager は制御ファイルのみを使用して、主要なバックアップおよびリカバリ操作をすべて実行できますが、カタログを使用した場合の利点にも注目してください。

- カタログのみで機能する Recovery Manager コマンドおよび操作機能があります。
- リカバリ・カタログは、制御ファイルにある上書可能なバックアップの履歴情報を保存します。
- リカバリ・カタログは、データベースの異なるインカネーションから、バックアップに関する情報を格納します。

リカバリ・カタログをメンテナンスするのは Recovery Manager 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。Recovery Manager 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・ファイルの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、自動的に伝播させます。RESYNC CATALOG コマンドを使用して、この情報を手動でカタログへ伝播させることもできます。

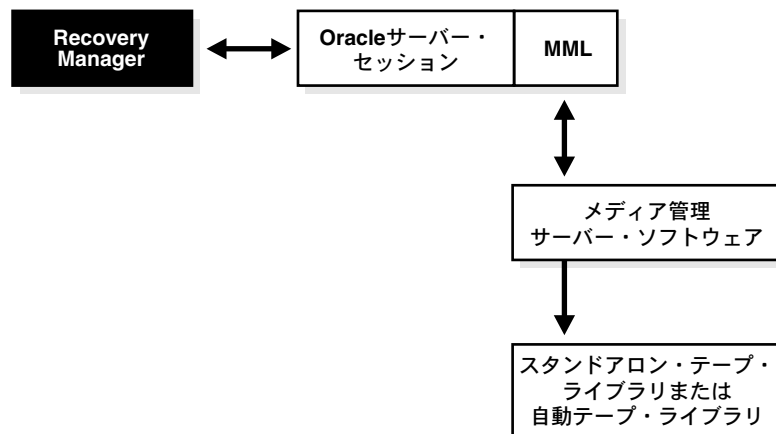
**関連項目：**

- 第 16 章「Recovery Manager のリポジトリの管理」
- 3-14 ページ「Recovery Manager でのリカバリ・カタログの使用・不  
使用の決定」
- 16-36 ページ「カタログ専用コマンドの制限」

## Recovery Manager メディア管理インタフェース

バックアップをテープ上に格納する場合、Recovery Manager ではメディア・マネージャが必要になります。メディア・マネージャとは、データのバックアップおよびリカバリに使用する、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ソフトウェア・プログラムです。図 1-4 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 1-4 Oracle に統合された MML のアーキテクチャ



Oracle サーバー・セッションは、SQL\*Plus などのクライアントがデータベースの接続に使用するセッションと同じタイプのサーバー・セッションです。図 1-4 のメディア・マネージャ・ライブラリ（MML）は、ベンダーが提供するメディア管理ソフトウェア・ライブラリで、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用し、データ・ファイルのバックアップおよびリストアを実行します。

**関連項目：** 8-3 ページ「Oracle とメディア・マネージャの通信方法」



---

# Recovery Manager を使用した データベースへの接続

この章では、Recovery Manager のコマンドライン・インタフェースを起動および停止する方法と、データベース接続の方法を説明します。この章の項目は、次のとおりです。

- [Recovery Manager の起動: 概要](#)
- [データベースに接続しない Recovery Manager の起動](#)
- [リカバリ・カタログを使用しないターゲット・データベースへの接続](#)
- [ターゲット・データベースおよびリカバリ・カタログへの接続](#)
- [Oracle Real Application Clusters 内のターゲット・データベースへの接続](#)
- [補助データベースへの接続](#)
- [データベースへの接続時にパスワードを非表示にする方法](#)
- [パイプを使用した Recovery Manager コマンドの実行](#)
- [Recovery Manager の終了](#)

## Recovery Manager の起動 : 概要

Recovery Manager の起動には、次の基本オプションがあります。

- Recovery Manager 実行可能ファイルをオペレーティング・システムのコマンドラインから起動し、1 つ以上のデータベースへ接続します。次に例を示します。

```
% rman TARGET / CATALOG rman/cat@catdb
% rman TARGET SYS/oracle@trgt NOCATALOG
% rman TARGET / CATALOG rman/cat@catdb AUXILIARY SYS/oracle@auxdb
```

- 接続オプションを指定せずに、Recovery Manager 実行可能ファイルをオペレーティング・システムのコマンドラインから起動します。次に例を示します。

```
% rman
```

Recovery Manager の起動時にコマンドラインからターゲット・データベースに接続した場合は、Recovery Manager プロンプトが表示された後に、コマンドを実行開始できます。

ターゲット・データベースに接続しないで Recovery Manager を起動した場合は、バックアップおよびリカバリ操作を実行開始する前に、Recovery Manager プロンプトから CONNECT TARGET コマンドを発行する必要があります。

## データベース接続のタイプ

接続可能なデータベースのタイプは、次のとおりです。

データベース	接続
ターゲット・データベース	Recovery Manager では、SYSDBA 権限でターゲット・データベースに接続します。この権限がない場合、接続は失敗します。
リカバリ・カタログ・データベース	このデータベースはオプションです。Recovery Manager はデフォルトの NOCATALOG オプションでも使用できます。
補助データベース	スタンバイ・データベース、複製データベースまたは補助インスタンス（スタンバイ・インスタンスまたは表領域の Point-in-Time リカバリのインスタンス）に接続できます。

## データベース接続に必要な認証

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するときには、SYSDBA 権限を持っている必要があります。パスワード・ファイルあるいはオペレーティング・システム認証を使用して、SYSDBA で接続します。

---

---

**注意：** SYSDBA オプションは、Recovery Manager が暗黙的かつ自動的に使用するため、指定する必要はありません。ターゲット・データベースに接続するには、SYSDBA 権限が必要です。

---

---

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は、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。SYSDBA でネット・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。

オペレーティング・システム認証を使用してデータベースに接続する場合は、Oracle SID を指定しているシステム環境変数を設定してください。たとえば、UNIX のコマンドラインから SID を trgt へ設定するには、次のように入力します。

```
% ORACLE_SID=trgt; export ORACLE_SID
```

リカバリ・カタログに接続するときには、SYSDBA 権限は不要ですから、注意してください。RECOVERY\_CATALOG\_OWNER ロールをスキーマの所有者に付与することのみが必要になります。

**関連項目：** データベースでのユーザー認証方法およびパスワード・ファイルの作成について、詳細は『Oracle9i データベース管理者ガイド』の第 1 章を参照してください。

## Recovery Manager 起動時のコマンドライン・オプション

Recovery Manager では、様々なコマンドラインのオプションが提供されており、Recovery Manager の起動時に指定できます。たとえば、Recovery Manager は次のような方法で起動できます。

- 一連の Recovery Manager コマンドで構成されるコマンド・ファイルを指定する、バッチ・モードで起動
- インタラクティブ・モードで起動
- コマンドのパイプを使用して起動
- Recovery Manager の出力をリダイレクトするログ・ファイルを使用して起動

### 関連項目：

- Recovery Manager コマンドライン・オプションの詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- CONNECT の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- パスワード・ファイルの詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

## データベースに接続しない Recovery Manager の起動

データベースに接続しない場合でも、RMAN コマンドを引数なしで発行すると、オペレーティング・システムのコマンドラインから Recovery Manager を起動できます。たとえば、次のように入力します。

```
% rman
```

コマンドラインで LOG オプションを指定しなかった場合、Recovery Manager は Recovery Manager プロンプトを表示します。

```
RMAN>
```

Recovery Manager プロンプトの表示後、さらにコマンドを発行して、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できます。

コマンドラインで CATALOG または NOCATALOG を指定しないで Recovery Manager を起動した場合、Recovery Manager はリポジトリの接続を行いません。初めてリポジトリを必要とするコマンドが発行された時点で、CONNECT CATALOG コマンドが発行されていない場合、Recovery Manager は自動的にデフォルトの NOCATALOG モードで接続を行います。その後、CONNECT CATALOG はセッションで実行できません。



## リカバリ・カタログを使用しないターゲット・データベースへの接続

例では、変数に次の意味があると想定します。

変数	意味
SYS	SYSDBA 権限を持つユーザー
oracle	ターゲット・データベースの orapwd ファイルで指定される、SYSDBA で接続するためのパスワード
trgt	ターゲット・データベースのネット・サービス名

## カタログを使用しないコマンドラインからのターゲット・データベースへの接続

オペレーティング・システムのコマンドラインから接続するには、次の例のように接続オプションを入力します。

```
# example of operating system authentication
% rman TARGET / NOCATALOG
```

```
# example of Oracle Net authentication
% rman TARGET SYS/oracle@trgt NOCATALOG
```

次のように、NOCATALOG あるいは CATALOG を指定しなくても Recovery Manager を起動できます。

```
# example of operating system authentication
% rman TARGET /
```

```
# example of Oracle Net authentication
% rman TARGET SYS/oracle@trgt
```

コマンドラインで NOCATALOG キーワードを指定しない場合や、Recovery Manager の起動後に CONNECT CATALOG を指定しない場合、初めてリポジトリを必要とするコマンドを実行するときに、Recovery Manager はデフォルトの NOCATALOG モードで接続を行います。たとえば、次のように入力します。

```
% rman TARGET /
RMAN> BACKUP DATABASE;      # RMAN defaults to NOCATALOG mode
```

## カタログを使用しない Recovery Manager プロンプトからのターゲット・データベースへの接続

この例のように、Recovery Manager を起動して、Recovery Manager プロンプトからターゲット・データベースに接続する方法もあります。

```
% rman NOCATALOG
RMAN> CONNECT TARGET
```

この例では、パスワード・ファイルを使用してターゲット・データベースに接続します。

```
% rman NOCATALOG
RMAN> CONNECT TARGET SYS/oracle@trgt
```

## ターゲット・データベースおよびリカバリ・カタログへの接続

例では、リカバリ・カタログのメンテナンスが行われていることを前提としています。また、変数には次の意味があります。

変数	意味
SYS	SYSDBA 権限を持つユーザー
oracle	ターゲット・データベースの orapwd ファイルで指定される、SYSDBA で接続するためのパスワード
trgt	ターゲット・データベースのネット・サービス名
rman	RECOVERY_CATALOG_OWNER 権限を持つリカバリ・カタログの所有者
cat	リカバリ・カタログの orapwd ファイル内に指定された、ユーザーの Recovery Manager のパスワード
catdb	リカバリ・カタログ・データベースのネット・サービス名

## コマンドラインからのターゲット・データベースおよびリカバリ・カタログへの接続

ターゲットおよびリカバリ・カタログ・データベースに、オペレーティング・システムのコマンドラインから接続するには、次の例のように接続オプションを入力します。

```
# operating system authentication
% rman TARGET / CATALOG rman/cat@catdb

# Oracle Net authentication
% rman TARGET SYS/oracle@trgt CATALOG rman/cat@catdb
```

## Recovery Manager プロンプトからのターゲット・データベースおよびリカバリ・カタログへの接続

Recovery Manager を起動して、Recovery Manager プロンプトからターゲット・データベースに接続する方法もあります。この例では、オペレーティング・システム認証を使用します。

```
% rman
RMAN> CONNECT TARGET
RMAN> CONNECT CATALOG rman/cat@catdb
```

この例では、Oracle Net 認証を使用します。

```
% rman
RMAN> CONNECT TARGET SYS/oracle@trgt
RMAN> CONNECT CATALOG rman/cat@catdb
```

---

**注意：** CONNECT TARGET を実行し、次に CONNECT CATALOG を実行しなかった場合、初めてリポジトリを必要とするコマンド（たとえば BACKUP）を実行するときに、Recovery Manager はデフォルトの NOCATALOG モードで接続を行います。その後、Recovery Manager セッションでは、CONNECT CATALOG は実行できません。

---

## Oracle Real Application Clusters 内のターゲット・データベースへの接続

Recovery Manager が 1 度に接続できる Oracle Real Application Clusters データベースのインスタンスは、1 つのみです。trgt1、trgt2 および trgt3 を Oracle Real Application Clusters 構成にある 3 つのインスタンスのネット・サービス名と想定します。この場合、これらのネット・サービス名のうちターゲット・データベースの接続に使用できるのは、1 つのサービス名のみです。たとえば、次のように接続できます。

```
% rman TARGET SYS/oracle@trgt2 CATALOG rman/cat@catdb
```

各ネット・サービス名は 1 つのみ指定する必要があります。指定できるインスタンスも 1 つのみです。Oracle Net の機能を使用して、1 つ以上のインスタンスに分散接続しているネット・サービス名は指定できません。

Recovery Manager が初期のターゲット接続用に接続しているインスタンスは 1 つのみですが、このために 3 つのインスタンスを使用したバックアップの実行ができないことはありません。たとえば、自動チャネルを構成して、次のように各クラスタのインスタンスに接続できます。

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;  
CONFIGURE DEVICE TYPE sbt PARALLELISM 3;  
CONFIGURE CHANNEL 1 DEVICE TYPE sbt CONNECT = 'SYS/oracle@trgt1';  
CONFIGURE CHANNEL 2 DEVICE TYPE sbt CONNECT = 'SYS/oracle@trgt2';  
CONFIGURE CHANNEL 3 DEVICE TYPE sbt CONNECT = 'SYS/oracle@trgt3';
```

その後、次のコマンドを実行して、データベース全体のバックアップを行います。

```
BACKUP DATABASE;
```

## 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 Recovery Manager の TSPITR を実行する場合は、補助インスタンスに接続する必要があります。例では、変数に次の意味があると想定します。

変数	意味
SYS	SYSDBA 権限を持つユーザー
oracle	ターゲット・データベースの orapwd ファイルで指定される、SYSDBA で接続するためのパスワード
trgt	ターゲット・データベースのネット・サービス名
rman	RECOVERY_CATALOG_OWNER 権限を持つリカバリ・カタログの所有者
cat	リカバリ・カタログの orapwd ファイル内に指定された、ユーザーの Recovery Manager のパスワード
catdb	リカバリ・カタログ・データベースのネット・サービス名
aux	補助データベースの orapwd ファイルで指定される、SYSDBA で接続するためのパスワード
auxdb	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認証のパスワード・ファイルがある場合、ローカル・アクセスにも、リモート・アクセスにも、パスワードを使用して接続できます。ネット・サービス名を使用して、リモートで接続している場合、パスワード・ファイルを使用した認証が必須になります。

## コマンドラインからの補助データベースへの接続

オペレーティング・システムのコマンドライン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@auxdb
```

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、次のように（すべて1行で）発行します。

```
% rman TARGET SYS/oracle@trgt CATALOG rman/cat@catdb AUXILIARY SYS/aux@auxdb
```

## Recovery Manager プロンプトからの補助データベースへの接続

Recovery Manager を起動し、Recovery Manager プロンプトから補助データベースに接続する方法もあります。

```
% rman
RMAN> CONNECT AUXILIARY SYS/aux@auxdb
```

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、次のように発行します。

```
% rman
RMAN> CONNECT TARGET SYS/oracle@trgt
RMAN> CONNECT CATALOG rman/cat@catdb
RMAN> CONNECT AUXILIARY SYS/aux@auxdb
```

### 関連項目：

- [第 11 章「Recovery Manager による表領域の Point-in-Time リカバリの実行」](#)
- [第 13 章「Recovery Manager によるスタンバイ・データベースの作成」](#)

## データベースへの接続時にパスワードを非表示にする方法

オペレーティング・システムのコマンドラインから Recovery Manager に接続し、認証情報を非表示にするには、最初に Recovery Manager を起動し、次の処理を実行する必要があります。

- CONNECT コマンドを、Recovery Manager プロンプトから実行します。接続文字列でパスワードを指定しない場合は、パスワード入力を求めるプロンプトが表示されます。
- 接続情報を含んだコマンド・ファイルを Recovery Manager プロンプトから実行します。無許可のアクセスを防ぐために、コマンド・ファイルに読み込み権限を設定できます。

たとえば、UNIX 環境で Recovery Manager を実行している場合は、次の手順が使用できます。

1. どのデータベースにも接続しない Recovery Manager の起動

```
% rman
```

2. テキスト・ファイルに接続情報を入れます。たとえば、次の行を `connect.rman` という名前のファイルに記述します。

```
CONNECT TARGET SYS/oracle@trgt  
CONNECT CATALOG rman/cat@catdb
```

3. 接続スクリプトのパーミッションを変更すると、すべてのユーザーがスクリプトを実行できますが、書き込みアクセスができるのは指定のユーザーのみです。たとえば、次のように入力します。

```
% chmod 711 connect.rman
```

4. Recovery Manager プロンプトからスクリプトを実行し、ターゲットおよびカタログ・データベースに接続します。たとえば、次のように入力します。

```
RMAN> @connect.rman
```

## パイプを使用した Recovery Manager コマンドの実行

Recovery Manager パイプ・インタフェースは、Recovery Manager へのコマンド発行および出力の取得を行うための、もう 1 つの方法として選択できます。パイプを使用することで、Recovery Manager は DBMS\_PIPE PL/SQL パッケージとインタフェースを取ることが可能です。ただし、オペレーティング・システムのコマンド・シェルは避けてください。

Recovery Manager では、パブリック・パイプを使用したパイプ・インタフェースを許可していません。これは、セキュリティ上の問題が発生する可能性があるからです。パブリック・パイプを使用すると、そのパイプの名前を知っているユーザーなら誰でも、Recovery Manager にコマンドを送信して、その出力を受信することが可能になります。

パイプが初期化されていない場合、Recovery Manager はこれをプライベート・パイプとして作成します。Recovery Manager を起動する前に入力パイプ上にコマンドを入力する場合、DBMS\_PIPE.CREATE\_PIPE をコールして、最初のパイプを確実に作成してください。パイプを明示的にプライベート・パイプとして作成しない場合、初めてパイプにアクセスする際に、パイプは自動的にパブリック・パイプとして作成されます。このパイプをパブリック・パイプとして使用するよう Recovery Manager に通知されると、Recovery Manager はエラーを戻します。

---

**注意：** ターゲット・データベースに対して複数の Recovery Manager セッションが実行可能な場合は、Recovery Manager の各セッションに一意のパイプ名を使用します。DBMS\_PIPE.UNIQUE\_SESSION\_NAME フังก์ションは、一意のパイプ名を生成するために使用する方法の 1 つです。

---

手順の順序が、必ずしも使用例 1 で示すようになっている必要はありません。使用例 2 では、コマンドをパイプに入力して、データベースに接続します。

### パイプ・モードでの Recovery Manager の使用（使用例 1）

1. ターゲット・データベース（必須）に接続して Recovery Manager を起動し、PIPE オプションを指定します。たとえば、次のように発行します。

```
% rman PIPE abc TARGET SYS/oracle@trgt
```

TIMEOUT オプションも指定できます。このオプションは、指定された秒数内に Recovery Manager が入力パイプから入力を受け付けられない場合、Recovery Manager を自動的に強制終了します。たとえば、次のように入力します。

```
% rman PIPE abc TARGET SYS/oracle@trgt TIMEOUT = 60
```

2. ターゲット・データベースに接続し、DBMS\_PIPE.PACK\_MESSAGE および DBMS\_PIPE.SEND\_MESSAGE を使用して、必要なコマンドを入力パイプに入力します。パイプ・モードでは、標準の Recovery Manager プロンプトを表示するかわりに、入力準備が整ったときに Recovery Manager からメッセージ RMAN-00572 が発行されます。



3. DBMS\_PIPE.RECEIVE\_MESSAGE および DBMS\_PIPE.UNPACK\_MESSAGE を使用して、出力パイプから Recovery Manager の出力を読み込みます。
4. 手順 2 および 3 を繰り返し、手順 1 で起動した、同じ Recovery Manager インスタンスを使用して、後続のコマンドを実行します。
5. EXIT コマンドを送信して、Recovery Manager を強制終了します。TIMEOUT オプションも、同様に使用します。（この場合、指定された長さの時間内に入力があったくないと、その後 Recovery Manager は自動的に終了します。）

### パイプ・モードでの Recovery Manager の使用（使用例 2）

1. ターゲット・データベースへの接続後に、パイプを作成します（ORA\$RMAN\_pipe\_IN という名前で存在しない場合）。
2. 必要なコマンドを入力パイプに入力します。パイプ・モードでは、標準の Recovery Manager プロンプトを表示するかわりに、入力準備が整ったときに Recovery Manager からメッセージ RMAN-00572 が発行されます。
3. Recovery Manager を PIPE オプションを使用して起動し、TIMEOUT = 0 と指定します。たとえば、次のように入力します。

```
% rman PIPE abc TARGET SYS/oracle@trgt TIMEOUT = 0
```

4. Recovery Manager は、DBMS\_PIPE.PACK\_MESSAGE および DBMS\_PIPE.SEND\_MESSAGE を使用して、パイプに入力されたコマンドを読み込み、実行します。入力パイプが空になると、Recovery Manager は即座に終了します。
5. DBMS\_PIPE.RECEIVE\_MESSAGE および DBMS\_PIPE.UNPACK\_MESSAGE を使用して、出力パイプから Recovery Manager の出力を読み込みます。

**関連項目：** DBMS\_PIPE パッケージ上のドキュメントについて、詳細は『Oracle9i PL/SQL パッケージ・プロシージャおよびタイプ・リファレンス』を参照してください。Recovery Manager のパイプに関する概要は、4-11 ページの「[Recovery Manager パイプ・インタフェース](#)」を参照してください。

## Recovery Manager の終了

Recovery Manager プロンプトから EXIT または QUIT と入力して Recovery Manager を終了し、プログラムを終了します。たとえば、次のように入力します。

```
RMAN> exit
```

**関連項目：** EXIT の構文および QUIT の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。



---

## クイック・スタート

この章では、Recovery Manager (RMAN) を使用した作業開始の方法を説明します。この章の項目は、次のとおりです。

- [Recovery Manager のサンプル・スクリプトの使用方法および使用例](#)
- [Recovery Manager 認証方式の選択](#)
- [基本的な Recovery Manager コマンドの使用方法](#)
- [Recovery Manager でのリカバリ・カタログの使用・不使用の決定](#)

# Recovery Manager のサンプル・スクリプトの使用方法および使用例

?/rdbms/demo サブディレクトリ（場所はオペレーティング・システムによって異なります）には、Recovery Manager のサンプル・スクリプトが多数含まれています。スクリプトには、.rcv のファイル名拡張子が接尾辞として付けられています。

これらのファイルには十分な説明を示す事例が記載されており、Recovery Manager のコマンドで使用した機能を理解することができます。独自のバックアップおよびリカバリ計画で使用できるようにカスタマイズしたり、単に Recovery Manager の使用方法の参考として使用できます。

**関連項目：**

- Recovery Manager プロンプトからコマンド・ファイルを実行する方法は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- コマンドラインからコマンド・ファイルを実行する方法は、『Oracle9i Recovery Manager リファレンス』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、16-18 ページの「リカバリ・カタログに格納された Recovery Manager のスクリプトの管理」を参照してください。

## Recovery Manager 認証方式の選択

2-3 ページの「データベース接続に必要な認証」で説明したように、Recovery Manager のコマンドを実行するには、最初に専用サーバー・プロセスに対する SYSDBA ロールを使用して、Recovery Manager をターゲット・データベースに接続する必要があります。接続には、オペレーティング・システム認証または Oracle Net を使用します。Oracle Net で SYSDBA としての接続を行うためには、パスワード・ファイルを作成する必要があります。

**表 3-1 Recovery Manager 認証方式の決定**

接続手段	処理
オペレーティング・システム認証	使用するオペレーティング・システムで SYSDBA 権限を所有しており、初期化パラメータ REMOTE_LOGIN_PASSWORDFILE が NONE に設定されていること。  <b>関連項目：</b> 認証の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。SYSDBA 権限の追加方法は、使用しているオペレーティング・システム固有のドキュメントを参照してください。

表 3-1 Recovery Manager 認証方式の決定（続き）

接続手段	処理
Oracle Net	<p>ORAPWD ユーティリティで、REMOTE_LOGIN_PASSWORDFILE を EXCLUSIVE に設定してパスワード・ファイルを作成し、このパスワード・ファイルにターゲット・データベースに接続するユーザーを加えること。</p> <p>また、ターゲット・データベースのネット・サービス名を tnsnames.ora ファイルに加え、ターゲット・データベースが接続を受け取れるように listener.ora ファイルを構成すること。</p> <p><b>関連項目：</b>『Oracle9i データベース管理者ガイド』および『Oracle9i Net Services 管理者ガイド』を参照してください。</p>

**関連項目：**

- Recovery Manager の起動方法と、ターゲット、リカバリ・カタログおよび補助データベースへの接続方法の詳細は、[第 2 章「Recovery Manager を使用したデータベースへの接続」](#)を参照してください。
- 共有サーバー構成を実行しているデータベース上の専用サーバー・プロセスに Recovery Manager を接続する方法については、8-31 ページの「[Recovery Manager を共有サーバーで使用する場合は設定](#)」を参照してください。

## 基本的な Recovery Manager コマンドの使用方法

ターゲット・データベースへの接続方法がわかれば、バックアップおよびリカバリ操作をただちに実行開始できます。この項の例を使用して、テスト・データベースを使用した基本的なバックアップおよびリストアを演習します。これらの例では、次の項目を前提としています。

- テスト・データベースは、ARCHIVELOG モードであること。
- デフォルトの NOCATALOG モードで実行していること。
- Recovery Manager 実行可能ファイルが、テスト・データベースと同じホストで実行されていること。
- オペレーティング・システム認証を使用してコマンドラインから接続していること（2-3 ページの「[データベース接続に必要な認証](#)」を参照してください）。
- Oracle Real Application Clusters 構成を実行していないこと。

この項では、次の項目について説明します。

- [ターゲット・データベースへの接続](#)
- [データベースの起動および停止](#)
- [ターゲット・データベースの現行スキーマのレポート](#)
- [データベースのバックアップ](#)
- [表領域のバックアップ](#)
- [アーカイブ・ログのバックアップ](#)
- [データ・ファイルのコピー](#)
- [バックアップおよびコピーのリスト表示](#)
- [バックアップのリストアの妥当性チェック](#)
- [データベースのリストアおよびリカバリ](#)
- [表領域のリストアおよびリカバリ](#)
- [Recovery Manager 構成の表示](#)

### ターゲット・データベースへの接続

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・カタログが作成してある場合でも同様に接続できますが、ここで示す例では、デフォルトの NOCATALOG モードで接続していることを前提とします。

オペレーティング・システムのコマンドラインから、次に示すように入力して、ターゲット・データベースにデフォルトの NOCATALOG モードで接続します。

```
% rman TARGET /
```

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、Recovery Manager の出力では次のように表示されます。

```
Recovery Manager: Release 9.2.0.0.0
```

```
connected to target database: RMAN (DBID=1237603294)
```

ここで表示された DBID 値は、ターゲット・データベースのデータベース識別子です。

ターゲット・データベースを起動していない場合、Recovery Manager では次のメッセージが表示されます。

```
connected to target database (not started)
```

```
RMAN> # the RMAN prompt is displayed
```

**関連項目：** 接続オプションの詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## データベースの起動および停止

データベースを起動していない場合には、サーバー・パラメータ・ファイルを使用していない場合にのみ、Recovery Manager プロンプトから初期化パラメータ・ファイルを指定して、STARTUP コマンドを実行します。次の例では、サーバー・パラメータ・ファイルを使用してインスタンスを起動します。

```
RMAN> STARTUP MOUNT
```

```
Oracle instance started  
database mounted
```

データベースがオープンしている場合は、次の Recovery Manager コマンドを実行し、正常にクローズした後にマウントします。

```
RMAN> SHUTDOWN IMMEDIATE
```

```
database closed  
database dismounted  
Oracle instance shut down
```

```
RMAN> STARTUP MOUNT
```

**関連項目：** STARTUP の構文および SHUTDOWN の構文は『Oracle9i Recovery Manager リファレンス』を参照してください。

## ターゲット・データベースの現行スキーマのレポート

この例では、ターゲット・データベースを説明するレポートの生成を行います。REPORT schema コマンドを、次のように実行します。

```
RMAN> REPORT SCHEMA;
```

Recovery Manager では、ターゲット・データベース内で現行のデータ・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。

Report of database schema				
File	K-bytes	Tablespace	RB segs	Datafile Name
1	204800	SYSTEM	***	/oracle/oradata/trgt/system01.dbf
2	20480	UNDOTBS	***	/oracle/oradata/trgt/undotbs01.dbf
3	10240	CWMLITE	***	/oracle/oradata/trgt/cwmlite01.dbf
4	10240	DRSYS	***	/oracle/oradata/trgt/drsys01.dbf
5	10240	EXAMPLE	***	/oracle/oradata/trgt/example01.dbf
6	10240	INDX	***	/oracle/oradata/trgt/indx01.dbf
7	10240	TOOLS	***	/oracle/oradata/trgt/tools01.dbf
8	10240	USERS	***	/oracle/oradata/trgt/users01.dbf

**関連項目：** レポートの作成方法は、[第 17 章「Recovery Manager のリポ  
ジトリの問合せ」](#)を参照してください。REPORT の構文は、『Oracle9i  
Recovery Manager リファレンス』を参照してください。

## データベースのバックアップ

このタスクでは、データベースをデフォルトのディスク位置にバックアップします。デフォルトの位置はポートにより異なります。たとえば、ほとんどの UNIX システムの場合には、その位置は ?/dbs になります。この例では、FORMAT パラメータを指定していないため、Recovery Manager はバックアップを一意のファイル名に割り当てます。チャンネルの割当てを手動で行わない場合、Recovery Manager では、デフォルトで構成済みのディスク・チャンネルが使用されます。

作成できる基本的なバックアップには、全体バックアップと増分バックアップの 2 つのタイプがあります。全体バックアップの場合、Recovery Manager では、ターゲット・データベース・ファイルのすべてのブロックがバックアップされます。増分バックアップの場合、Recovery Manager では、前回のバックアップ以降に変更されたブロックのみがバックアップされます。



## 全体バックアップの作成

Recovery Manager プロンプトから次のように BACKUP コマンドを実行し、データ・ファイル、制御ファイルおよび現行のサーバー・パラメータ・ファイル（インスタンスの起動に使用した場合）の全体バックアップを、デフォルトのデバイス・タイプに作成します。

```
RMAN> BACKUP DATABASE;
```

CONFIGURE コマンドを使用してデフォルトのデバイスをテープに明示的に構成しないかぎり、デフォルト・デバイスはディスクです。

BACKUP コマンドを実行すると、**バックアップ・セット**が生成されます。これは、1 つ以上のバックアップ・ピースを含む論理オブジェクトです。BACKUP コマンドの出力には、バックアップに不可欠な情報が含まれています。この出力例を次に示します。

```
Starting backup at OCT 12 2001 19:09:48
using target database controlfile instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=10 devtype=DISK
channel ORA_DISK_1: starting full datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
including current SPFILE in backupset
including current controlfile in backupset
input datafile fno=00001 name=/oracle/oradata/trgt/system01.dbf
input datafile fno=00002 name=/oracle/oradata/trgt/undotbs01.dbf
input datafile fno=00003 name=/oracle/oradata/trgt/cwmlite01.dbf
input datafile fno=00004 name=/oracle/oradata/trgt/drsys01.dbf
input datafile fno=00005 name=/oracle/oradata/trgt/example01.dbf
input datafile fno=00006 name=/oracle/oradata/trgt/indx01.dbf
input datafile fno=00007 name=/oracle/oradata/trgt/tools01.dbf
input datafile fno=00008 name=/oracle/oradata/trgt/users01.dbf
channel ORA_DISK_1: starting piece 1 at OCT 12 2001 19:09:56
channel ORA_DISK_1: finished piece 1 at OCT 12 2001 19:10:31
piece handle=/oracle/dbs/lvd6dtk1_1_1 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:39
Finished backup at OCT 12 2001 19:10:33
```

### 増分バックアップの作成

増分バックアップは、記憶領域の節約に適した方法です。増分バックアップでは、変更が加えられたデータベース・ブロックのみがバックアップされます。Recovery Manager では、**レベル 0 のバックアップ**とも呼ばれる**基本バックアップ**と現行のデータ・ファイルの比較により、バックアップするブロックを判断します。

たとえば、全体バックアップを基本バックアップとして作成した後、テスト・データベースを更新して、これをコミットします。この後、次のコマンドを実行すると、Recovery Manager は、前回の全体バックアップ以降に変更されたブロックのみをバックアップします。

```
RMAN> BACKUP INCREMENTAL LEVEL 1 DATABASE;
```

次のような行が出力で表示されるので、注意してください。

```
no parent backup or copy of datafile 1 found
```

この行は、問題の発生を示すものではなく、単に基本の LEVEL 0 の増分バックアップが存在しないことを意味しています。Recovery Manager は、自動的に LEVEL 0 のバックアップを作成し、基本の増分バックアップとします。

#### 関連項目：

- 全体バックアップと増分バックアップの違いについては、5-41 ページの「[バックアップ・タイプ](#)」を参照してください。
- バックアップの作成方法は、9-3 ページの「[Recovery Manager を使用したデータベース・ファイルおよびアーカイブ・ログのバックアップ](#)」を参照してください。
- BACKUP コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

### 表領域のバックアップ

データベース全体のバックアップにかぎらず、表領域またはデータ・ファイルをバックアップすることも可能です。この例では、事前構成済みのディスク・チャンネルを使用して 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。

次のように、BACKUP コマンドを Recovery Manager プロンプトから実行します。

```
RMAN> BACKUP TABLESPACE SYSTEM;
```

**関連項目：** 表領域のバックアップ方法は、9-8 ページの「[Recovery Manager を使用した表領域のバックアップ](#)」を参照してください。

## アーカイブ・ログのバックアップ

通常、データベース管理者は、ディスク上のアーカイブ・ログを、サード・パーティのテープなどの記憶メディアにバックアップします。また、アーカイブ・ログはディスクにもバックアップできます。どちらの場合でも、バックアップの完了後には、入力ログを自動的に削除できます。

すべてのアーカイブ・ログをバックアップして、入力ログ（プライマリのアーカイブ先からのみ）を削除するには、次のように、BACKUP コマンドを Recovery Manager プロンプトから実行します。

```
RMAN> BACKUP ARCHIVELOG ALL DELETE INPUT;
```

**関連項目：** アーカイブ REDO ログのバックアップ方法は、9-13 ページの「[Recovery Manager を使用したアーカイブ REDO ログのバックアップ](#)」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 のイメージ・コピーを新しい場所に作成します。イメージ・コピーは、Recovery Manager 固有のフォーマットのバックアップ・セットとは異なるものです。イメージ・コピーは、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用して作成されたコピーと同等のものです。

この例では、自動的に割り当てられたディスク・チャネルを使用して、df1.bak という名前のデータ・ファイルのコピーを作成します。次に示すように、Recovery Manager プロンプトから COPY コマンドをバックアップのパス名を指定して実行します。

```
RMAN> COPY DATAFILE 1 TO '/tmp/df1.cpy'; # specify any filename that you choose
```

Recovery Manager は、作成されたファイルの完全なファイル名を、次のように表示します。

```
Starting copy at OCT 12 2001 19:11:28
using channel ORA_DISK_1
channel ORA_DISK_1: copied datafile 1
output filename=/tmp/df1.cpy recid=141 stamp=442955509
Finished copy at OCT 12 2001 19:11:59
Finished copy at 18-APR-01
```

### 関連項目：

- 概念については、5-69 ページの「[イメージ・コピー](#)」を参照してください。
- イメージ・コピーの作成方法は、9-34 ページの「[Recovery Manager を使用したファイルのコピー](#)」を参照してください。
- COPY の構文は、『Oracle9i Recovery Manager リファレンス』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

作成したバックアップ・セットおよびイメージ・コピーをリスト表示するには、次のように LIST コマンドを実行します。

```
RMAN> LIST BACKUP;
```

Recovery Manager では、作成したバックアップ・セットおよびバックアップ・ピースに加え、これらのセットに含めたデータ・ファイルを、次のように表示します。

BS Key	Type	LV	Size	Device	Type	Elapsed Time	Completion Time
-----							
699	Full	122M		DISK		00:00:30	OCT 12 2001 19:10:23
BP Key: 1293      Status: AVAILABLE      Tag: TAG20011012T191001							
Piece Name: /oracle/dbs/lvd6dtk1_1_1							
Controlfile Included: Ckp SCN: 322170				Ckp time: OCT 12 2001 19:09:53			
SPFILE Included: Modification time: OCT 12 2001 19:09:53							
List of Datafiles in backup set 699							
File	LV	Type	Ckp	SCN	Ckp Time	Name	
-----							
1		Full	322172		OCT 12 2001 19:09:56	/oracle/oradata/trgt/system01.dbf	
2		Full	322172		OCT 12 2001 19:09:56	/oracle/oradata/trgt/undotbs01.dbf	
3		Full	322172		OCT 12 2001 19:09:56	/oracle/oradata/trgt/cwmlite01.dbf	
4		Full	322172		OCT 12 2001 19:09:56	/oracle/oradata/trgt/drsys01.dbf	
5		Full	322172		OCT 12 2001 19:09:56	/oracle/oradata/trgt/example01.dbf	
6		Full	322172		OCT 12 2001 19:09:56	/oracle/oradata/trgt/indx01.dbf	
7		Full	322172		OCT 12 2001 19:09:56	/oracle/oradata/trgt/tools01.dbf	
8		Full	322172		OCT 12 2001 19:09:56	/oracle/oradata/trgt/users01.dbf	

イメージ・コピーのリスト表示には、次のコマンドを実行してください。

```
RMAN> LIST COPY;
```

Recovery Manager では、データ・ファイルおよび制御ファイル以外に、アーカイブ REDO ログの表示も行います（アーカイブ REDO ログは、コピーの一種とみなされます）。

List of Datafile Copies							
Key	File	S	Completion Time	Ckp	SCN	Ckp Time	Name
141	1	A	OCT 12 2001 19:11:49	322204		OCT 12 2001 19:11:29	/tmp/df1.copy
List of Archived Log Copies							
Key	Thrd	Seq	S	Low Time	Name		
1105	1	70	A	OCT 10 2001 17:55:49	/oracle/oradata/trgt/arch/archive1_70.dbf		
1106	1	71	A	OCT 11 2001 05:12:06	/oracle/oradata/trgt/arch/archive1_71.dbf		
1107	1	72	A	OCT 11 2001 14:54:36	/oracle/oradata/trgt/arch/archive1_72.dbf		
1108	1	73	A	OCT 11 2001 14:55:48	/oracle/oradata/trgt/arch/archive1_73.dbf		
1109	1	74	A	OCT 11 2001 15:13:27	/oracle/oradata/trgt/arch/archive1_74.dbf		
1110	1	75	A	OCT 12 2001 06:56:22	/oracle/oradata/trgt/arch/archive1_75.dbf		

**関連項目：**

- リストの概念については、7-2 ページの「[LIST コマンドの出力](#)」を参照してください。
- リストおよびレポートの作成方法は、17-3 ページの「[Recovery Manager のバックアップ、コピーおよびデータベース・インカネーションのリスト表示](#)」を参照してください。
- LIST コマンドの構文および出力される列ヘッダーの詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## バックアップのリストアの妥当性チェック

作成したバックアップを実際にリストアせずに、リストアできるかどうかをチェックします。RESTORE ... VALIDATE コマンドを次のように実行します。

```
RMAN> RESTORE DATABASE VALIDATE;
```

次のような出力が表示されます。

```
Starting restore at 07-DEC-01

allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=9 devtype=DISK
channel ORA_DISK_1: starting validation of datafile backupset
channel ORA_DISK_1: restored backup piece 1
piece handle=/oracle/dbs/07db39tr_1_1 tag=TAG20011012T191001 params=NULL
channel ORA_DISK_1: validation complete
Finished restore at 07-DEC-01
```

エラー・メッセージがない場合は、データベース・バックアップのリストアが可能なことを Recovery Manager が確認したことを意味します。エラーがある場合、Recovery Manager 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、一部のバックアップが存在しない場合にデータベースのリストアを試みると、次のようなエラー・スタックが表示されます。

```
RMAN-00571: =====
RMAN-00569:===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: restore コマンドが 12/07/2001 17:28:24 で失敗しました
RMAN-06026: 見つからないターゲットがあります - リストアを中止します
RMAN-06023: データ・ファイル 8 をリストアするためのバックアップまたはコピーが見つかりません
RMAN-06023: データ・ファイル 7 をリストアするためのバックアップまたはコピーが見つかりません
```

**関連項目：** バックアップおよびコピーのリストア方法は、9-32 ページの「[Recovery Manager を使用したバックアップの妥当性チェックの実行](#)」を参照してください。VALIDATE の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## データベースのリストアおよびリカバリ

バックアップ計画およびリカバリ計画を立てる上で重要なことは、メディア障害が発生した場合の対処方法を理解していることです。この使用例では、メディア障害のシミュレーションを行います。最初に、次のコマンドを実行し、データベースをシャットダウンして、Recovery Manager を終了させます。

```
RMAN> SHUTDOWN IMMEDIATE;  
RMAN> EXIT;
```

データベースをシャットダウンした後、3-6 ページの「[ターゲット・データベースの現行スキーマのレポート](#)」で説明した REPORT SCHEMA コマンドからの出力を使用して、データ・ファイルのファイル名を識別します。オペレーティング・システムのコマンドを使用して、データベース・ファイルの一部またはすべてを一時的に改名します（ただし、制御ファイルは改名しません。注意してください）。この処理でメディア障害をシミュレートできます。これは、Oracle が、起動中にデータ・ファイルを検索するためです。

この、UNIX を使用した例では、tools 表領域内のデータ・ファイル名を変更します。

```
% mv $ORACLE_HOME/oradata/trgt/tools01.dbf $ORACLE_HOME/oradata/trgt/tools01.bak
```

ここで Recovery Manager を起動し、次に示すようにデータベースのオープンを試行します。

```
% rman TARGET /  
RMAN> STARTUP
```

```
RMAN-00571: =====  
RMAN-00569:===== ERROR MESSAGE STACK FOLLOWS =====  
RMAN-00571: =====  
RMAN-03002: startup コマンドが 08/29/2001 13:46:04 で失敗しました  
ORA-01157: データ・ファイル 7 を識別 / ロックできません - DBWR トレース・ファイルを参照してください  
ORA-01110: データ・ファイル 7: '/oracle/oradata/trgt/tools01.dbf'
```

データベースはマウントされていますが、オープンされていません。これは、一部のデータ・ファイルにリカバリが必要なためです。データベースのリストアおよびリカバリを、デフォルト・ディスク・チャネルで行うには、次のコマンドを Recovery Manager プロンプトから実行します。

```
RMAN> RESTORE DATABASE;  
RMAN> RECOVER DATABASE;
```

Recovery Manager では、以前に作成したバックアップおよびコピーを使用して、それらのデフォルトの場所にファイルをリストアします。続いて、Recovery Manager はアーカイブ REDO ログを使用して（必要に応じて）、データベースのリカバリを行います。リカバリの完了後、データベースをオープンします。

```
RMAN> ALTER DATABASE OPEN;
```

## 表領域のリストアおよびリカバリ

メディア障害の影響が、データベース内の全ファイルに及ぶのではなく、一部のファイルに及ぶといった事態は、珍しいことではありません。この使用例では、1つの表領域内のデータ・ファイルに対する、メディア障害のシミュレーションを行います。前述の REPORT SCHEMA コマンドの出力を使用して、SYSTEM 表領域以外の表領域にあるデータ・ファイルのファイル名を識別します。この例では、tools 表領域内のデータ・ファイルを改名します。

データベースがオープンしている間に、表領域内の1つ以上のデータ・ファイルを選択して改名します。この、UNIX を使用した例では、tools 表領域内のデータ・ファイル名を変更します。

```
% mv $ORACLE_HOME/oradata/trgt/tools01.dbf /tmp/tools01.dbf
```

改名したデータ・ファイルの1つで表の位置を特定し、更新を試行すると、データ・ファイルが使用不可能なため、Oracle はエラー・メッセージを生成します。消失したデータ・ファイルに対するリストアおよびリカバリが必要になります。まず Recovery Manager を起動し、SQL コマンドで表領域をオフラインにします。このとき、表領域の名前を指定していることを確認します。

```
RMAN> SQL 'ALTER TABLESPACE tools OFFLINE IMMEDIATE';
```

次に示すコマンドを Recovery Manager プロンプトから実行し、表領域のリストアおよびリカバリをデフォルト・ディスク・チャネルで行います。

```
RMAN> RESTORE TABLESPACE tools;  
RMAN> RECOVER TABLESPACE tools;
```

リカバリが完了した後、次のコマンドを実行して表領域をオンラインにします。

```
RMAN> SQL 'ALTER TABLESPACE tools ONLINE';
```

## Recovery Manager 構成の表示

Recovery Manager にはデフォルトの構成設定が含まれています。デフォルトの設定は、すべての Recovery Manager セッションで適用されます。この設定は、CONFIGURE コマンドで明示的な変更または無効化を行うまで維持されます。CONFIGURE ... CLEAR コマンドを実行すると、いつでもデフォルトの構成設定に戻すことができます。

可能な構成のうち、最も重要なものは、自動チャネルおよび保存方針です。Recovery Manager では、事前に自動ディスク・チャネルの構成が行われているので、ディスクへのバックアップおよびコピーのために手動でチャネルを割り当てる必要がありません。

現行の Recovery Manager 構成を表示するためには、次のコマンドを実行します。

```
RMAN> SHOW ALL;
```

**関連項目：** Recovery Manager 構成の作成方法は、第 8 章「[Recovery Manager 環境の構成](#)」を参照してください。

## Recovery Manager でのリカバリ・カタログの使用・不使用の決定

Recovery Manager は、デフォルトでは NOCATALOG モードでターゲット・データベースに接続します。これは、ターゲット・データベース内の制御ファイルを Recovery Manager メタデータの唯一のリポジトリとして使用するということです。Recovery Manager の使用時に決定する、最も重要と考えられる事項は、通常の運用操作の Recovery Manager リポジトリとして、リカバリ・カタログを作成するかどうかです。リカバリ・カタログとは、個別のデータベースで作成されたスキーマであり、ターゲット制御ファイルから取得したメタデータを含んでいます。

通常、オラクル社では、複数のデータベースを管理する場合に、カタログを使用することをお薦めしています。バックアップを行うデータベースが 2 つ以上ある場合は、システム全体に 1 つのリカバリ・カタログを作成し、このカタログ内にすべてのデータベースのメタデータを格納してください。これにより、複数のデータベースに、それぞれが 1 つのカタログを維持する場合に必要な余分な領域要件や、メモリーのオーバーヘッドを回避します。ただし、カタログをバックアップする際には、追加の予防措置を取る必要があります。これは、カタログが消失すると、複数のターゲット・データベース用のメタデータを失うことになるためです。

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、16-2 ページの「[リカバリ・カタログの作成](#)」にある指示を参照してください。

**関連項目：** Recovery Manager 環境における Recovery Manager リポジトリの機能の概要は、4-12 ページの「[Recovery Manager リポジトリ](#)」を参照してください。



## リカバリ・カタログを Recovery Manager リポジトリとして使用した場合の利点

リカバリ・カタログを使用すると、ターゲット・データベース内の制御ファイルをメタデータの唯一のリポジトリとして使用する場合よりも、Recovery Manager によって自動化されるバックアップおよびリカバリの機能が増えます。次に示す機能は、カタログを使用した場合にのみ使用可能になります。

- 複数のターゲット・データベースに関するメタデータを、1つのカタログ内に格納できます。
- 1つのターゲット・データベースにある、複数のインカネーションに関するメタデータを、カタログ内に格納できます。したがって、どのインカネーションからでも、バックアップをリストアできます。
- リカバリ・カタログの再同期化を `CONTROL_FILE_RECORD_KEEP_TIME` の設定より短い間隔で行うことで、履歴メタデータを保持できます。
- 現在以外の時刻におけるターゲット・データベースのスキーマをレポートできます。
- Recovery Manager のスクリプトをリカバリ・カタログに格納できます。
- データベースに存在するデータベース・ファイルが、マウントした制御ファイルに記録されているファイルと異なるときに、リストアおよびリカバリを行う場合には、リカバリ・カタログが必要とされるファイルを指定します。カタログがない場合には、最初にデータベース・ファイルの正確なセットをリストしている制御ファイルのバックアップをリストアする必要があります。
- 制御ファイルが消失し、バックアップからリストアする必要がある場合に、永続的な構成がテーブル・チャンネル割当てを自動化していると、これらの構成はデータベースがマウントされていないときにも使用できます。

## リカバリ・カタログを Recovery Manager リポジトリとして使用した場合のコスト

カタログ使用時にかかる主なコストは、この補足的なデータベースに必須となる管理上のオーバーヘッドです。たとえば、次のことが必要になります。

- リカバリ・カタログを格納するターゲット・データベース以外のデータベース（カタログが消失した場合のメンテナンスには利点となります）の検索や、新規データベースの作成。
- Recovery Manager メタデータ用に、データベースの十分な領域の作成（16-2 ページの「[リカバリ・カタログ・データベースの構成](#)」を参照してください）。
- リカバリ・カタログのメタデータのバックアップ
- 必要に応じたりカバリ・カタログの更新

これらのことから、データベースのネットワークを管理しない場合にかぎり、オーバーヘッドを回避するために、メタデータの排他的リポジトリとして制御ファイルの使用を選択することになります。制御ファイルを Recovery Manager リポジトリとして使用する場合でも、Recovery Manager は効果的に機能します。カタログを使用しない場合には、16-36 ページの「[リカバリ・カタログを使用しない Recovery Manager リポジトリの管理方法](#)」を参照してください。特に、次の事項を確認してください。

- リカバリ・カタログを使用しない場合のコストについても考慮が必要です。詳細は、16-36 ページの「[カタログ専用コマンドの制限](#)」を参照してください。
- リポジトリのバックアップについて、計画を立てる必要があります。詳細は、16-39 ページの「[制御ファイルのバックアップおよびリストア](#)」を参照してください。

**関連項目：** リカバリ・カタログの管理方法は、[第 16 章「Recovery Manager のリポジトリの管理」](#)を参照してください。

# 第 II 部

---

## アーキテクチャおよび概念

第 II 部では、Recovery Manager 環境のアーキテクチャおよび基本概念の概要を説明します。  
この部では、次の章について説明します。

- 第 4 章「Recovery Manager のアーキテクチャ」
- 第 5 章「概念 I: チャネル、バックアップおよびコピー」
- 第 6 章「概念 II: リストア、リカバリおよび複製」
- 第 7 章「概念 III: メンテナンス」



---

## Recovery Manager のアーキテクチャ

この章では、Recovery Manager インタフェースおよび環境の基本的な要素について説明します。

この章の項目は、次のとおりです。

- [Recovery Manager アーキテクチャの概要](#)
- [Recovery Manager コマンド・インタフェース](#)
- [Recovery Manager リポジトリ](#)
- [メディア管理](#)

---

**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があります。

---

**関連項目：** Recovery Manager の互換性およびアップグレードの詳細は、『Oracle9i データベース移行ガイド』を参照してください。

# Recovery Manager アーキテクチャの概要

この項では、次の項目について説明します。

- [Recovery Manager 環境](#)
- [Recovery Manager セッション・アーキテクチャ](#)
- [Recovery Manager メタデータの格納](#)

## Recovery Manager 環境

Recovery Manager (RMAN) は、バックアップおよびリカバリ操作を実行するクライアント・アプリケーションです。 **Recovery Manager 環境**は、様々なアプリケーションおよびデータベースから構成されており、これらをバックアップ計画やリカバリ計画に活用します。

Recovery Manager の環境には、1 つの Recovery Manager 実行可能ファイルに対し、1 つのターゲット・データベースが接続する単純な環境もあれば、Oracle Enterprise Manager を介してアクセスする多くのメディア・マネージャやターゲット、リカバリ・カタログ、および複数の補助データベースが接続する複雑な環境もあります。[表 4-1](#) では、Recovery Manager 環境で使用可能なコンポーネントを示します。

表 4-1 Recovery Manager 環境のコンポーネント

コンポーネント	説明	必須要件
ターゲット・データベース	制御ファイル、データ・ファイルおよびオプションのアーカイブ REDO ログ。これらを使用して、Recovery Manager はバックアップおよびリストア作業を管理します。Recovery Manager は、データベースの情報収集および操作に関する情報の格納に、ターゲット・データベースの制御ファイルを使用します。バックアップ・ジョブおよびリカバリ・ジョブの実際の処理は、ターゲット・データベース上のサーバー・セッションにより実行されます。	必要
Recovery Manager 実行可能ファイル	ターゲット・データベースのバックアップおよびリカバリ操作を管理するクライアント・アプリケーション。Recovery Manager クライアントは、Oracle Net を使用してターゲット・データベースに接続するため、Oracle Net を使用して接続されたどのホスト上でも位置を指定することが可能です。	必要
リカバリ・カタログ・データベース	リカバリ・カタログのスキーマが含まれるデータベース。これに含まれるメタデータを使用して、Recovery Manager はバックアップおよびリカバリ操作を実行します。	不要

表 4-1 Recovery Manager 環境のコンポーネント（続き）

コンポーネント	説明	必須要件
リカバリ・カタログのスキーマ	Recovery Manager でメンテナンスするメタデータ表を所有するリカバリ・カタログ・データベース内のユーザー。Recovery Manager は、ターゲット・データベースの制御ファイルからのメタデータを、定期的にリカバリ・カタログに伝播します。	不要
スタンバイ・データベース	プライマリ・データベースのコピー。プライマリ・データベースで作成されたアーカイブ・ログを使用して更新されます。Recovery Manager は、スタンバイ・データベースの作成またはバックアップが可能です。	不要
メディア管理アプリケーション	ベンダー固有のアプリケーション。これにより、Recovery Manager は、テープなどのストレージにバックアップを行うことが可能になります。バックアップまたはリストアを行う場合、Recovery Manager クライアントはターゲット・インスタンスに接続し、そのインスタンスがメディア・マネージャと対話するための指示を出します。Recovery Manager クライアントとメディア・マネージャの直接通信が発生することはありません。すべての通信は、ターゲット・インスタンス上で発生します。	不要
メディア管理カタログ	ベンダー固有のリポジトリ。メディア管理アプリケーションに関する情報のリポジトリです。	不要
Oracle Enterprise Manager	GUI ベースのアプリケーション。Recovery Manager へのインタフェースとして使用されます。	不要

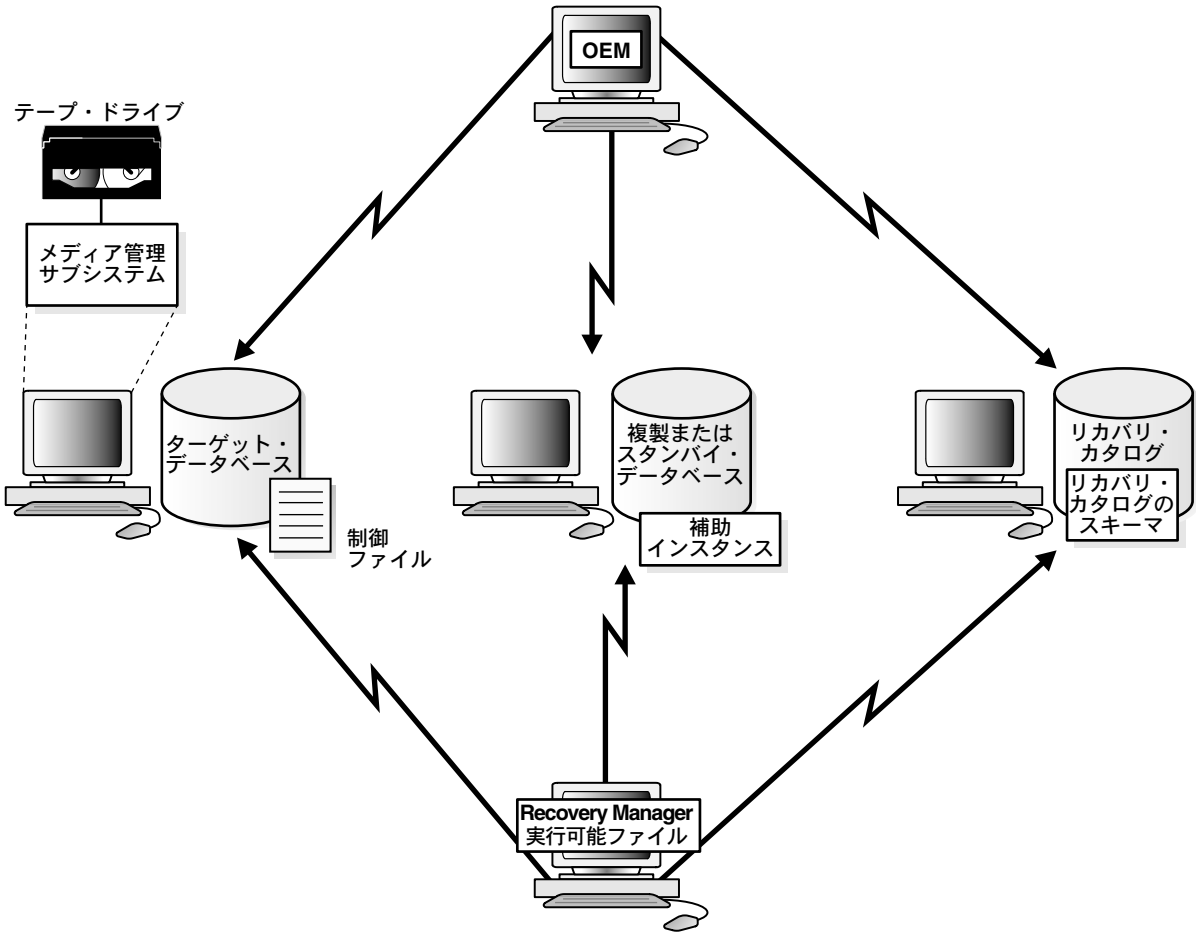
この表が示すように、Recovery Manager 環境に必須のコンポーネントは、ターゲット・データベースおよび Recovery Manager 実行可能ファイルのみです。ただし、実際の構成は、これほど簡潔ではありません。

図 4-1 に示す例は、現実的な Recovery Manager 環境です。この環境では、5 つのノードがネットワークで接続され、それぞれのマシンは異なる目的で使用されています。5 つのノードは、次に示すように役割を分担しています。

- Recovery Manager 実行可能ファイルを実行するクライアント・ノード
- ターゲット・データベースおよびメディア管理サブシステムのホストとなるサーバー・ノード
- 複製データベースまたはスタンバイ・データベースのホストとなるサーバー・ノード
- リカバリ・カタログ・データベースのホストとなるサーバー・ノード
- このシステムのデータベースに GUI インタフェースを提供する Oracle Enterprise Manager アプリケーションを実行するクライアント・ノード

この使用例では、Recovery Manager の実行可能ファイルをクライアント・マシンから実行することで、ターゲット、カタログおよび補助データベースに接続します。その後、バックアップおよびリカバリ・ジョブを実行します。また、Oracle Enterprise Manager のホストを行うクライアントに接続すると、GUI インタフェースを使用して Recovery Manager にアクセスできます。

図 4-1 Recovery Manager 環境の例





## Recovery Manager セッション・アーキテクチャ

Recovery Manager クライアント・アプリケーションは、すべてのバックアップおよびリカバリ・タスクを実行するために、データベースのサーバー・セッションに指示を出します。ここでの「セッション」の意味は、オペレーティング・システムにより異なります。たとえば、UNIX システムでは、サーバー・セッションはサーバー・プロセスに相当します。Windows NT システムでは、サーバー・セッションは Oracle サービスのスレッドに相当します。

Recovery Manager クライアントをターゲット・データベースのサーバーに接続すると、Recovery Manager はサーバー・セッションをターゲット・インスタンスに割り当て、これに指示を出してバックアップおよびリカバリ操作を実行します。Recovery Manager クライアント自体は、バックアップ、リストアまたはリカバリを実行しません。

## Recovery Manager メタデータの格納

Recovery Manager は、バックアップおよびリカバリ操作の管理を行います。このため、データベースについて必要な情報を格納するための領域が必要になります。Recovery Manager は、この情報を常にターゲット・データベースの制御ファイルに格納します。また、Recovery Manager メタデータを、個別のデータベースに含まれるリカバリ・カタログのスキーマに格納することもできます。リカバリ・カタログのスキーマは、ターゲット・データベース以外のデータベースに格納される必要があります。Recovery Manager は、定期的に、制御ファイルからの情報をリカバリ・カタログに移行します。

## Recovery Manager コマンド・インタフェース

Recovery Manager インタフェースを使用して、バックアップおよびリカバリ操作のあらゆる面を管理するためのコマンドを入力します。

---

---

**注意：** Oracle リリース 8.0 以降のすべての Recovery Manager コマンドは、Oracle9i でも機能します。

---

---

この項では、次の項目について説明します。

- [Recovery Manager と PL/SQL パッケージ](#)
- [Recovery Manager コマンドのコンパイルおよび実行方法](#)
- [Recovery Manager コマンドのタイプ](#)
- [Recovery Manager コマンドの実行](#)
- [Recovery Manager パイプ・インタフェース](#)
- [Oracle Enterprise Manager での Recovery Manager のジョブ・スクリプト](#)

**関連項目：** Recovery Manager コマンドとその構文の詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## Recovery Manager と PL/SQL パッケージ

Recovery Manager 実行可能ファイルは、PL/SQL パッケージを使用して、ターゲット・データベースおよびリカバリ・カタログと通信します。これは内部パッケージであり、ドキュメント化されていません。

## Recovery Manager コマンドのコンパイルおよび実行方法

この項で説明するように、Recovery Manager では大部分のコマンドを 2 フェーズで処理します。

- [コンパイル・フェーズ](#)
- [実行フェーズ](#)

### コンパイル・フェーズ

コンパイル・フェーズ中に、Recovery Manager はそのコマンドがアクセスするオブジェクトを判断します。たとえば、TABLESPACE キーワードが、Recovery Manager のコンポーネント・データ・ファイルの名前に変換されます。次に、Recovery Manager は一連のリモート・プロシージャ・コール (RPC) を生成します。これにより、ターゲット・データベースは、データ・ファイルのバックアップ、リストアあるいはリカバリなど、必要な操作を実行するように指示されます。

### 実行フェーズ

実行フェーズ中に、Recovery Manager は RPC コールをターゲット・データベースに送信して、進行状況を監視し、その結果を収集します。2 つ以上のチャンネルが割り当てられている場合、Recovery Manager は一部のコマンドをパラレルに実行します。これは、各チャンネルのターゲット・データベース・セッションのすべてが、RPC コールを同時に実行しているためです。Recovery Manager は、このパラレル実行の調整を行い、実行中の全チャンネルについての進行状況を監視します。

## Recovery Manager コマンドのタイプ

Recovery Manager コマンドは、次のように分類できます。

- Recovery Manager プロンプトからのみ実行可能な、スタンドアロン・コマンド
- RUN コマンドの大カッコ内でのみ実行可能な、ジョブ・コマンド
- Recovery Manager プロンプトからでも、RUN コマンドの大カッコ内からでも実行可能な、コマンド例外

これらのコマンドに加え、Recovery Manager では、Recovery Manager 起動時に指定可能な数多くのコマンドライン引数もサポートしています。

**関連項目：** Recovery Manager コマンドおよびコマンドライン・オプションの詳細は『Oracle9i Recovery Manager リファレンス』を参照してください。

### スタンドアロン・コマンド

ジョブ・コマンドとは異なり、スタンドアロン・コマンドでは、RUN コマンドで使用するサブコマンドのようなものではありません。Recovery Manager プロンプトからのみ使用できるコマンドの一部を次に示します。

- CONNECT
- CONFIGURE
- CREATE CATALOG、DROP CATALOG、UPGRADE CATALOG
- CREATE SCRIPT、DELETE SCRIPT、REPLACE SCRIPT
- LIST
- REPORT

**関連項目：**『Oracle9i Recovery Manager リファレンス』にある構文のエントリで、Recovery Manager プロンプトから実行するコマンドを判断してください。

## 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
- SWITCH

Recovery Manager 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。ブロック内のコマンドに障害が発生すると、Recovery Manager は処理を停止し、それ以降のコマンドは実行されません。実際には、コマンド実行の単位は RUN コマンドが定義しています。RUN ブロック内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：** RUN コマンドの詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## コマンド例外

コマンドには、スタンドアロン・コマンドまたはジョブ・コマンドに専用のものもありますが、その他のコマンドは、プロンプトからでも RUN コマンド内からでも発行できます。これらのコマンドは、プロンプトから発行されても、RUN コマンド内から発行されても、自動チャネルを使用することができます。手動によるチャネル割当ては RUN コマンド内でのみ使用可能です。この場合、手動で割り当てたチャネルが、構成済みのどの自動チャネルよりも優先されることに注意してください。

スタンドアロン・コマンドとジョブ・コマンドの両方に機能するコマンドの例を次に示します。

- BACKUP
- BLOCKRECOVER
- COPY
- RESTORE
- RECOVER
- VALIDATE

## Recovery Manager コマンドの実行

Recovery Manager では、コマンド言語インタプリタ (CLI) が使用されるため、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。また、コマンドラインで LOG オプションを指定すると、Recovery Manager 出力をログ・ファイルに書き込むこともできます。

**関連項目：** Recovery Manager セッションの終了方法は、15-12 ページの「[Recovery Manager コマンドの終了](#)」を参照してください。

### 対話モード

Recovery Manager コマンドを対話形式で実行するには、Recovery Manager を起動してコマンドライン・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、Recovery Manager 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/oracle@trgt CATALOG rman/cat@catdb
```

Recovery Manager プロンプトが表示されると、次に示すようなコマンドが入力できます。

```
RMAN> BACKUP DATABASE;
```

### バッチ・モード

Recovery Manager のコマンドをファイルに入力して、コマンド・ファイルを作成することができます。この**コマンド・ファイル**は、ファイル名をコマンドラインから指定すると実行できます。コマンド・ファイルの内容は、コマンドラインに入力したコマンドと同様に実行されます。

バッチ・モードでは、Recovery Manager 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。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、定期的に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。

この例として、サンプルの Recovery Manager のスクリプトが入力されている `b_10.rcv` というコマンド・ファイルがあります。このファイルを、オペレーティング・システムのコマンドラインから実行して、その出力をログ・ファイル `log.f` に書き込むには、次のようにします。

```
% rman TARGET / CATALOG rman/cat@catdb CMDFILE b_10.rcv LOG log.f
```

**関連項目：** Recovery Manager のコマンドライン・オプションの詳細は、『[Oracle9i Recovery Manager リファレンス](#)』を参照してください。

## ストアド・スクリプト

ストアド・スクリプトとは、リカバリ・カタログに格納されている Recovery Manager のジョブ・コマンドのブロックです。ストアド・スクリプトにより、データベースのバックアップ、リストアおよびリカバリを行うコマンドの、計画、開発およびテストが可能になります。さらに、スクリプトによりユーザー・エラーの可能性が最小限に抑えられます。ストアド・スクリプトは、1つのターゲット・データベースにのみ関連付けられることに注意してください。

ストアド・スクリプトを作成するために、対話形式でスクリプトを Recovery Manager のコマンドライン・インタフェースに入力するか、Recovery Manager のコマンドをコマンド・ファイルに入力して、そのファイルを実行します。この場合、ターゲット・データベースとリカバリ・カタログに接続している必要があります。

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
  BACKUP
    INCREMENTAL LEVEL 0
    TAG b_whole_10
    FILESPERSET 6
    DATABASE PLUS ARCHIVELOG;
}
```

このストアド・スクリプトは、次に示すように Recovery Manager プロンプトから実行できます。

```
RUN { EXECUTE SCRIPT b_whole_10 };
```

ストアド・スクリプトの表示には、リカバリ・カタログのビュー RC\_STORED\_SCRIPT を問い合わせます。

```
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：** ストアド・スクリプトの詳細は、16-18 ページの「[リカバリ・カタログに格納された Recovery Manager のスクリプトの管理](#)」を参照してください。?/rdbms/demo ディレクトリ内のサンプル・スクリプトも参照してください。

## Recovery Manager パイプ・インタフェース

Recovery Manager パイプ・インタフェースは、Recovery Manager へのコマンド発行と、そのコマンドからの出力の取得を行うための、もう 1 つの方法です。このインタフェースでは、Recovery Manager がコマンドの取得および出力の送信を行うために DBMS\_PIPE PL/SQL パッケージを使用します。Recovery Manager は、どのようなデータの読取りあるいは書込みにも、オペレーティング・システムのシェルを使用しません。このインタフェースを使用すると、Recovery Manager に移植性の高いプログラム・インタフェースを記述できます。

パイプ・インタフェースの起動には、PIPE コマンドライン・パラメータを使用します。Recovery Manager が使用するプライベート・パイプは、コマンドの受信用と出力の送信用の 2 つです。パイプの名前は、PIPE パラメータの値から導出されます。たとえば、次に示すコマンドで Recovery Manager を起動できます。

```
% rman PIPE abc TARGET SYS/oracle@trgt
```

Recovery Manager は、ターゲット・データベースに次のパイプをオープンします。

- Recovery Manager が、ユーザーのコマンドを受信するために使用する、ORA\$RMAN\_ABC\_IN
  - Recovery Manager が、出力のすべてを送信するために使用する、ORA\$RMAN\_ABC\_OUT
- 入力および出力パイプ上のすべてのメッセージは、どちらも VARCHAR2 型です。

Recovery Manager では、パブリック・パイプを使用したパイプ・インタフェースを許可していません。これは、セキュリティ上の問題が発生する可能性があるからです。パブリック・パイプを使用すると、そのパイプの名前を知っているユーザーなら誰でも、Recovery Manager にコマンドを送信して、その出力を受信することが可能になります。

**関連項目：** パイプを使用した Recovery Manager コマンドの実行方法は、2-12 ページの「[パイプを使用した Recovery Manager コマンドの実行](#)」を参照してください。

## Oracle Enterprise Manager での Recovery Manager のジョブ・スクリプト

Run RMAN Script 機能を使用すると、Recovery Manager のコマンドラインからもコールできるコマンドやスクリプトを、Oracle Enterprise Manager 内で発行できます。rman スクリプトを発行またはスケジュールすると、Oracle Enterprise Manager Job System を介してジョブとして実行されます。

Oracle Enterprise Manager Job System を使用すると、標準的なタスクと管理タスクを自動化できます。Job System では、ジョブの作成と管理や実行のスケジュールを行ったり、定義済みジョブに関する情報を他の管理者と共有できます。

## Recovery Manager リポジトリ

Recovery Manager リポジトリとは、Recovery Manager がバックアップ、リカバリおよびメンテナンス操作を行うために使用する、ターゲット・データベースに関するメタデータのコレクションのことです。この情報は、**リカバリ・カタログ**を作成して格納することも、Recovery Manager を使用してターゲット・データベースの制御ファイルに格納させることもできます。Recovery Manager は制御ファイルのみで主要なバックアップおよびリカバリ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のみ機能する Recovery Manager コマンドも一部あります。

リカバリ・カタログをメンテナンスするのは Recovery Manager 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。Recovery Manager 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・ファイルの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・カタログの中に伝播させます。

**関連項目：** Recovery Manager リポジトリの管理方法は、[第 16 章「Recovery Manager のリポジトリの管理」](#)を参照してください。カタログ専用コマンドのリストは、16-36 ページの「[カタログ専用コマンドの制限](#)」を参照してください。

## Recovery Manager リポジトリのリカバリ・カタログへの格納

リカバリ・カタログとは、ターゲット・データベースに関する情報のオプションのリポジトリです。リカバリ・カタログの使用およびメンテナンスは Recovery Manager が行います。カタログの格納には、専用のデータベースを使用することをお勧めします。Recovery Manager 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リカバリ操作の実行方法を決定します。

この項では、次の項目について説明します。

- [リカバリ・カタログへのデータベースの登録](#)
- [リカバリ・カタログの内容](#)
- [リカバリ・カタログの再同期化](#)
- [リカバリ・カタログのバックアップ](#)
- [リカバリ・カタログの互換性](#)



## リカバリ・カタログへのデータベースの登録

リカバリ・カタログにデータベースを記録することを、**登録**といいます。同じリカバリ・カタログ内に、2つ以上のターゲット・データベースを登録できます。たとえば、データベース `prod1`、`prod2` および `prod3` を、データベース `catdb` 内で `catowner` が所有する1つのカタログに登録することができます。データベースの登録は、指定のカタログ・スキーマに1回のみ可能です。たとえば、`catowner` のスキーマに `prod1` を登録した後に、再び `catowner` のスキーマに `prod1` を登録することはできません。

Recovery Manager ではデータベースが一意的データベース識別子 (DBID) で区別されるため、指定されたカタログに登録される各データベースには DBID が必要です。同じ DBID を持つ2つのデータベースを、1つのカタログ内に登録することはできません。たとえば、DBID が 862893450 のデータベース `prod1` と、DBID が 862893450 のデータベース `prod2` を、同じカタログ内に登録することはできません。ただし、各データベースのデータベース名が一意である必要はありません。たとえば、DBID が 862893450 のデータベース `prod1` と、DBID が 951781249 の別のデータベース `prod1` を、同じリカバリ・カタログに登録することができます。

オペレーティング・システム・コマンドを使用してデータベースをコピーすると、コピーしたデータベースの DBID はオリジナル・データベースと同じになります。したがって、データベースを手動でコピーした後、このコピーとその親を同じカタログ内に登録することは、コピーの DBID を変更しないかぎり不可能です。このため、`DUPLICATE` コマンドを使用すると、コピーしたデータベースに異なる DBID が自動的に割り当てられるので、コピーを作成するにはこの方法が最も簡単です。また、`DBNEWID` ユーティリティを使用すると、任意の Oracle データベースの DBID (またはデータベース名) を変更できます。`DBNEWID` が役立つのは、同じ DBID を持つ複数のデータベースをすでに作成しており、すべてを同じリカバリ・カタログに登録する必要がある場合です。

**関連項目：** `DBNEWID` ユーティリティを使用してデータベースの DBID を変更する方法は、『Oracle9i データベース・ユーティリティ』を参照してください。

## リカバリ・カタログの内容

リカバリ・カタログには、Recovery Manager 操作に関する情報に加えて、次の内容が含まれます。

- 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 ログとそのコピー
- ターゲット・データベース上の表領域およびデータ・ファイル
- 名前付けされたユーザー作成の一連の Recovery Manager コマンドであるストアド・スクリプト
- 永続的な Recovery Manager の構成の設定

## 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 Recovery Manager 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、Recovery Manager が制御ファイルから取得したメタデータが、現行の状態に保たれます。

再同期化は、完全でも部分でも実行できます。部分再同期化では、Recovery Manager は現行の制御ファイルを読み取って変更されたデータを更新しますが、データベースの物理スキーマに関するメタデータは再同期化されません。更新されないデータは、データ・ファイル、表領域、REDO スレッド、ロールバック・セグメントおよびオンライン REDO ログです。完全再同期化では、Recovery Manager は、スキーマ・レコードも含めてすべての変更済レコードを更新します。

Recovery Manager では、完全再同期化または部分再同期化の必要性が自動的に検出され、必要に応じて再同期化処理が実行されます。また、RESYNC CATALOG コマンドを発行すると、完全再同期化を強制実行することもできます。16-16 ページの「[再同期化を実行する時期](#)」で説明したケースでは、この操作を実行する必要があります。

カタログが現行の状態であることを確実にするために、RESYNC CATALOG コマンドを定期的に行ってください。このコマンドを、少なくとも *n* 日に 1 度を目安にして実行します。この *n* は、初期化パラメータ CONTROL\_FILE\_RECORD\_KEEP\_TIME の設定値になります。制御ファイルは循環再利用システムを採用しているため、バックアップおよびコピーのレコードは上書きされます。カタログの再同期化により、これらのレコードはカタログに格納されるため消失しません。

**関連項目：** 制御ファイルのレコードの詳細は、4-16 ページの「[制御ファイル内のレコードのタイプ](#)」を参照してください。

**スナップショット制御ファイル** Recovery Manager は、リカバリ・カタログを完全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**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、Recovery Manager は制御ファイルのビューに一貫性を保ちます。スナップショット制御ファイルは、Recovery Manager の短期使用を目的としているため、リカバリ・カタログには登録されません。Recovery Manager は、リカバリ・カタログにスナップショット制御ファイルのチェックポイントを記録します。これにより、リカバリ・カタログの新しさが示されます。

Oracle9i Server では、1 つのスナップショット制御ファイルに対してアクセスできる Recovery Manager セッションは、いつでも 1 つのみに制限されています。この保護対策により、2 つの Recovery Manager セッションのそれぞれが、スナップショット制御ファイルの使用で競合しないことを確実にします。

---

**注意：** スナップショット制御ファイルの名前および位置は指定可能です。この方法は、8-29 ページの「[スナップショット制御ファイルの場所の設定](#)」を参照してください。

---

**関連項目：** リカバリ・カタログを再同期化する方法は、16-22 ページの「[リカバリ・カタログ使用時の制御ファイルの管理](#)」を参照してください。構文の詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## リカバリ・カタログのバックアップ

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す。そのため、リカバリ・カタログを消失すると問題が発生する可能性があります。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・カタログが破損した場合に、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ると、カタログの一部をリカバリできます。しかし、常に最新のカタログのバックアップをすることをお勧めします。

**関連項目：** リカバリ・カタログのバックアップ方法は、16-24 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。

## リカバリ・カタログの互換性

リカバリ・カタログと Recovery Manager を使用する場合、Recovery Manager 環境には次のコンポーネントが含まれます。

- Recovery Manager 実行可能ファイル
- リカバリ・カタログ・データベース
- リカバリ・カタログ・データベースにあるリカバリ・カタログのスキーマ
- ターゲット・データベース

これらのコンポーネントには、リリース番号が付けられています。たとえば、リリース 8.0.6.1 の Recovery Manager 実行可能ファイルを、リリース 8.1.6 のターゲット・データベースで使用し、リリース 8.1.6 で作成したカタログ・スキーマを持つリリース 8.1.5 のリカバリ・カタログ・データベースにリポジトリを格納できます。

**関連項目：** Recovery Manager 環境でのコンポーネントの互換性を説明するグラフは、『Oracle9i Recovery Manager リファレンス』を参照してください。

## Recovery Manager リポジトリ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ため、Recovery Manager ではリカバリ・カタログのかわりにターゲット・データベースの制御ファイルのみを使用する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、このモードが適しています。別のリカバリ・カタログ・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。Recovery Manager の機能のうち、NOCATALOG モードでサポートされないのはストアド・スクリプトのみです。

### 制御ファイル内のレコードのタイプ

リカバリ・カタログを使用しない場合、制御ファイルは、他の関連情報とともに、バックアップとコピーに関する排他的情報ソースとなります。制御ファイルには、**循環再利用レコード**および**非循環再利用レコード**の 2 タイプのレコードが含まれています。

**循環再利用レコード** 循環再利用レコードには、重要度の低い情報が含まれ、必要に応じて上書きされます。このレコードには、データベースで継続的に生成される情報が含まれません。循環再利用レコードの情報例は、次のとおりです。

- ログ履歴
- アーカイブ REDO ログ
- Backups
- データ・ファイルのオフライン範囲

循環再利用レコードは、論理的なリングとして配置されています。使用可能なレコード・スロットがいっぱいになると、Oracle は、新規レコード用の領域を確保するために制御ファイルを拡大するか、最も古いレコードを上書きします。CONTROL\_FILE\_RECORD\_KEEP\_TIME 初期化パラメータにより、レコードの再利用が可能になるまでの最低日数を指定します。

**関連項目：** 循環再利用レコードに対する Oracle の処理を管理する方法は、16-37 ページの「[制御ファイル・レコードの上書きの監視](#)」を参照してください。

**非循環再利用レコード** 非循環再利用レコードには、頻繁に変更されることのない重要な情報が含まれ、上書きすることはできません。非循環再利用レコードに含まれる情報の例としては、データ・ファイル、オンライン REDO ログおよび REDO スレッドなどがあります。

## カタログなしのリカバリ

データベースのリストアおよびリカバリに、リカバリ・カタログを使用しない場合、次に示す処理をお勧めします。

- Recovery Manager で制御ファイルを自動的にバックアップすること、および制御ファイルの自動バックアップの際にリポジトリ（リカバリ・カタログか、ターゲット・データベースの制御ファイルかのどちらか）へのアクセスが不要となるように、**制御ファイルの自動バックアップ機能**を使用可能にしてください
- 最低 2 つの多重化またはミラー化した制御ファイルを、別のディスクで使用してください
- Recovery Manager のすべてのバックアップ・ログを保存してください

これらにより、リカバリ・カタログを使用していない場合にも、現行の制御ファイルを消失したときに制御ファイルの自動バックアップをリストアできます。

### 関連項目：

- リカバリ・カタログを使用していない場合に使用不可能となるコマンドのリストは、「[カタログ専用コマンドの制限](#)」を参照してください。
- 制御ファイルの自動バックアップを使用した障害時リカバリの方法は、5-52 ページの「[制御ファイルとサーバー・パラメータ・ファイルの自動バックアップ](#)」を参照してください。

## メディア管理

テープ・ストレージを使用してデータベースのバックアップを行う場合、**Recovery Manager** では**メディア・マネージャ**が必要になります。メディア・マネージャとは、データのバックアップおよびリカバリに使用するテープ・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ユーティリティです。

Oracle が公開している **Media Management API** を使用すると、サード・パーティ・ベンダーが **Recovery Manager** で機能するソフトウェアを作成できます。**Recovery Manager** 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。

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、ディスクなどのストレージ・サブシステムとテープなどのメディア・サブシステム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、このテクノロジーによりほとんど排除できます。

## メディア・マネージャを使用したバックアップおよびリストア操作

次に示す **Recovery Manager** コマンドにより、データ・ファイルをテープにバックアップします。

```
BACKUP DEVICE TYPE sbt DATAFILE 1;
```

**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・セッションでは、出力チャネルがメディア管理デバイスとして識別され、メディア・マネージャに出力の書き込みを要求します。

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。自動テープ・ライブラリを所有しているサイトでは、メディア・マネージャは Oracle からの要求により自動的にテープのロードおよびアンロードを行います。これを所有していないサイトでは、メディア・マネージャはオペレータに対して指定した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リストア処理を行います。ファイルのリストアでは、次のような手順が発生します。

1. 特定のファイルのリストアが Oracle から要求されます。
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、これを読み取ります。
3. メディア・マネージャは、読み取った情報を Oracle サーバー・セッションに戻します。
4. Oracle セッションは、そのファイルをディスクに書き込みます。

## プロキシ・コピー

Oracle は、**プロキシ・コピー**の機能を Media Management API に統合しています。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リストア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。**Recovery Manager** は、バックアップまたはリストア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。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。

## メディア・マネージャのテスト

クライアント・プログラム `sbttest` は、メディア管理ソフトウェアをテストするスタンダードアロンのプログラムであり、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、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リストアができない場合です。`sbttest` プログラムは、必ずオラクル社カスタマ・サポート・センターの指示に従って使用してください。

## Backup Solutions Program

Oracle Partner Program の一部である Oracle の Backup Solutions Program (BSP) は、主要なメディア管理ソフトウェア・ベンダーのグループで、その製品は Oracle の MML に準拠しています。MML インタフェースに準拠したソフトウェアにより、Oracle サーバー・セッションはメディア・マネージャへのバックアップが可能になり、メディア・マネージャにバックアップのリストアを要求することが可能になります。

**Recovery Manager** は、テープのロード、ラベル付けあるいはアンロードに対して特定のコマンドを発行しないことに注意してください。バックアップを行う場合、**Recovery Manager** はバイト・ストリームをメディア・マネージャに与え、このストリームに一意の名前を対応付けます。**Recovery Manager** がバックアップのリストアを必要とする場合、メディア・マネージャに同一のバイト・ストリームを取得するよう要求します。ストリームの格納方法および格納場所の詳細は、メディア・マネージャによりすべて処理されます。

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。現在使用可能な製品のリストは、次に示す Backup Solutions Program の Web サイトにアクセスしてください。

<http://otn.oracle.com/deploy/availability>

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お問い合わせください。

特定のメディア管理製品を使用する場合は、メディア管理のベンダーに直接問い合せて、Oracle の BSP に属する製品かどうかを確認してください。オラクル社は、**Recovery Manager** 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、バージョン間の互換性および機能については、オラクル社ではなく、メディア・ベンダーに直接問い合せてください。





---

## 概念 I: チャネル、バックアップおよびコピー

この章では、Recovery Manager (RMAN) ユーティリティの使用方法に関連する基本概念を説明します。

この章の項目は、次のとおりです。

- [Recovery Manager の自動チャネルおよび手動チャネルの割当て](#)
- [バックアップ・セット](#)
- [バックアップ・オプション: 名前、サイズおよびスピード](#)
- [バックアップ・タイプ](#)
- [バックアップ・エラー](#)
- [制御ファイルとサーバー・パラメータ・ファイルの自動バックアップ](#)
- [バックアップの保存方針](#)
- [バックアップの最適化](#)
- [再起動可能なバックアップ](#)
- [イメージ・コピー](#)
- [バックアップのテストおよび整合性チェック](#)

## Recovery Manager の自動チャネルおよび手動チャネルの割当て

この項では、次の項目について説明します。

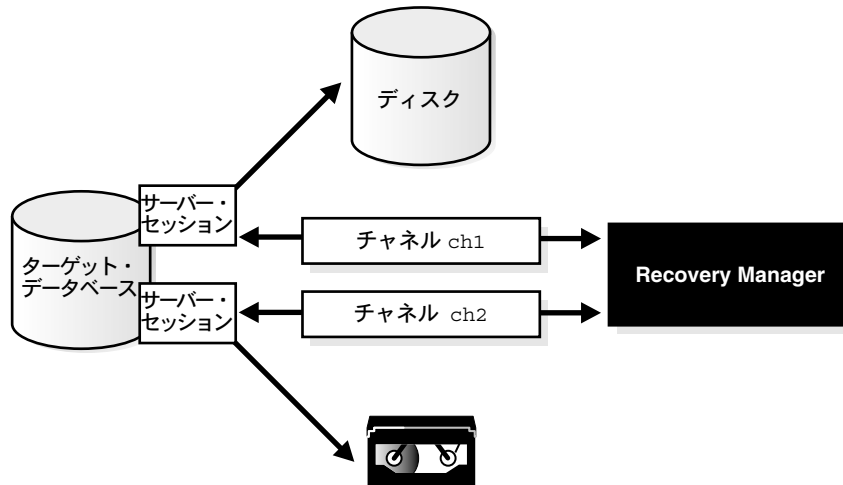
- [Recovery Manager のチャネル](#)
- [自動チャネルおよび手動チャネルの割当て](#)
- [自動チャネルのデバイスの構成](#)
- [自動チャネルのデフォルト・デバイス・タイプ](#)
- [自動チャネルのネーミング規則](#)
- [自動チャネルの一般的な構成](#)
- [特定の自動チャネルの構成](#)
- [自動チャネル設定の消去](#)
- [手動によるチャネル割当ての平行化](#)
- [手動および自動チャネル用のチャネル制御オプション](#)
- [メディア・マネージャによるハードウェアの多重化](#)

**関連項目：** Oracle Real Application Clusters 環境でのチャネル割当てとバックアップの詳細は、『Oracle9i Real Application Clusters 管理』を参照してください。

## Recovery Manager のチャネル

Recovery Manager の**チャネル**とは、デバイス・タイプへのデータのストリームであり、1つのサーバー・セッションに相当します。Recovery Manager のチャネルを 1 つ以上割り当てることが、大部分のバックアップおよびリカバリコマンドを実行するために必要です。[図 5-1](#)に示すように、Recovery Manager 実行可能ファイルからターゲットおよび補助データベースのインスタンスへの接続は、各チャネルがインスタンス上のサーバー・セッションを起動することで確立されます。サーバー・セッションでは、バックアップ、リストアおよびリカバリ操作が実行されます。割り当てられたサーバー・セッションと通信する Recovery Manager セッションは 1 つしかありません。

図 5-1 チャネルの割当て



RUN ブロック内では、手動でチャネルを割り当てることができます。また、構成済みのチャネルは、**自動チャネル割当て**を使用するすべての Recovery Manager セッションで使用できます。Recovery Manager には、構成済みの DISK チャネルを装備しています。これを使用して、ディスクへのコピーおよびバックアップが行えます。また、CONFIGURE CHANNEL コマンドを実行すると、Recovery Manager の自動チャネルをディスクまたはテープに指定できます。この方法により、バックアップ、リストアまたはリカバリをするたびにチャネルを割り当てる必要がなくなります。

チャネルが必要になるコマンドを実行する場合や、チャネルの割当てを手動で行わない場合には、Recovery Manager は CONFIGURE コマンドで指定されたオプションを使用して自動的にチャネルを割り当てます。Recovery Manager が BACKUP コマンドに割り当てるチャネルは、DISK または sbt など単一のタイプのみです。RESTORE コマンドおよび多種のメンテナンス・コマンド（たとえば、DELETE）では、Recovery Manager が要求されたデバイス・タイプを判断し、必要なチャネルのすべてを割り当てます。

手動でチャネルを指定する場合、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（Recovery Manager プロンプトでのみ実行）は、サーバー・セッションがバックアップ、リストアおよびメンテナンス操作の実行に使用する、I/O デバイスのタイプも指定します。

ALLOCATE CHANNEL コマンドまたは CONFIGURE CHANNEL により、メディア・マネージャにリソースが割り当てられるかどうかは、提供ベンダーによって異なります。一部のメディア・マネージャでは、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。読み込み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メディア・マネージャもあります。

**関連項目：** ALLOCATE CHANNEL の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。ALLOCATE CHANNEL FOR MAINTENANCE の詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## 自動チャネルおよび手動チャネルの割当て

自動チャネル割当て機能を使用して、すべての Recovery Manager セッションで使用する永続的な自動チャネルのセットを構成できます。手動チャネル割当て機能を使用すると、RUN ブロック内で使用するコマンドにチャネルを指定することができます。

手動で ALLOCATE CHANNEL コマンドを実行していない場合には、Recovery Manager では次に示すコマンドの設定に応じた自動チャネルを割り当てます。

- CONFIGURE DEVICE TYPE ... PARALLELISM
- CONFIGURE DEFAULT DEVICE TYPE
- CONFIGURE CHANNEL DEVICE TYPE
- CONFIGURE CHANNEL *n* DEVICE TYPE

たとえば、次のコマンドを Recovery Manager プロンプトから発行します。

```
BACKUP DATAFILE 3;  
RUN { RESTORE TABLESPACE users; }
```

次に示す場合に、Recovery Manager では CONFIGURE コマンドで設定した値に応じて自動的にチャネルを割り当てます。

- BACKUP、RESTORE または DELETE などのコマンドを、RUN ブロックの外で使した場合。
- RUN ブロック内でチャネルの割当てを行わずにコマンドを実行した場合。

自動チャネル割当ての設定は、RUN ブロック内で手動によるチャネル割当てを行うとオーバーライドされます。自動チャネルと手動チャネルの混在はできません。手動チャネルは、常に自動チャネルより優先されます。たとえば、次に示すコマンドを発行すると、自動チャネル割当てがオーバーライドされます。

```
RUN  
{  
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt;  
  BACKUP DATABASE PLUS ARCHIVELOG;  
}
```

Recovery Manager では、新しいコマンドが前のコマンドと同じチャネル構成を要求する間は、自動チャネルを保持します。これにより自動チャネル割当てが最適化されています。たとえば、Recovery Manager では次に示す一連のコマンドに同じ割当て済みチャネルを使用します。

```
BACKUP DATAFILE 1;  
BACKUP CURRENT CONTROLFILE;  
BACKUP ARCHIVELOG ALL;
```

ALLOCATE あるいは CONFIGURE のようなコマンドを発行すると、Recovery Manager は自動的に構成済みのチャネルを解放します。

**関連項目：** 自動チャネルの構成方法は、8-11 ページの「[自動チャネルの構成](#)」を参照してください。

## 自動チャネルのデバイスの構成

CONFIGURE DEVICE TYPE ... PARALLELISM コマンドにより、Recovery Manager が使用するチャネルの数を指定します。これは、指定したデバイス・タイプに自動チャネルの割当てを行う場合に使用されます。たとえば、パラレル化を 3 に構成すると、Recovery Manager が自動チャネルを使用するときにはいつでも、Recovery Manager によりデフォルト・デバイス・タイプに 3 つのチャネルが割り当てられます。

パラレル化を行うと、Recovery Manager は常にチャネルを番号順に割り当てます。この番号は 1 から始まりパラレル化の設定値で終わります。たとえば、デバイス・タイプが sbt で、パラレル化を 4 に設定している場合、Recovery Manager は次のような割当てを行います。

```
ORA_SBT_TAPE_1  
ORA_SBT_TAPE_2  
ORA_SBT_TAPE_3  
ORA_SBT_TAPE_4
```

パラレル化の設定を変更するためには、もう 1 度 CONFIGURE DEVICE TYPE ... PARALLELISM コマンドを発行します。この例では PARALLELISM 2 を構成した後、これを 3 に変更します。

```
CONFIGURE DEVICE TYPE DISK PARALLELISM 2;  
CONFIGURE DEVICE TYPE DISK PARALLELISM 3;
```

パラレル化の設定は、Recovery Manager がパラレルに割り当てるデバイスのチャネル数を定義します。この設定を、デバイス用に構成した実際のチャネル数と一致させる必要はありません。たとえば、4 つの異なる sbt チャネルを構成して、sbt の PARALLELISM を、2、1 あるいは 10 に設定することができます。

パラレル化のデフォルトの設定を表示するためには、SHOW DEVICE TYPE コマンドを実行します。デフォルト値の後ろには番号記号 (#) が付きます。たとえば、次のように入力します。

```
RMAN> SHOW DEVICE TYPE;  
RMAN configuration parameters are:  
CONFIGURE DEVICE TYPE DISK PARALLELISM 1; # default
```

次に示す例では、デフォルトのデバイスを sbt に構成して、もう 1 度 SHOW コマンドを実行しています。

```
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO sbt;
new RMAN configuration parameters:
CONFIGURE DEFAULT DEVICE TYPE TO 'sbt';
new RMAN configuration parameters are successfully stored

RMAN> SHOW DEVICE TYPE;
RMAN configuration parameters are:
CONFIGURE DEVICE TYPE SBT PARALLELISM 1; # default
CONFIGURE DEVICE TYPE DISK PARALLELISM 1; # default
```

**関連項目：** 8-13 ページ「[デバイス・タイプ用の一般的な自動チャネルの構成](#)」

## 自動チャネルのデフォルト・デバイス・タイプ

CONFIGURE DEFAULT DEVICE TYPE コマンドを実行して、自動チャネルに対するデフォルトのデバイス・タイプを指定します。たとえば、通常はテープにバックアップを行うが、ディスクにバックアップを行うこともあるという場合があります。この場合、ディスクとテープ・デバイスにチャネルを構成しますが、sbt タイプのデバイスをデフォルトのデバイスにします。

```
CONFIGURE DEVICE TYPE DISK PARALLELISM 1; # configure device disk
CONFIGURE DEVICE TYPE sbt PARALLELISM 2; # configure device sbt
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
```

Recovery Manager は、バックアップ・コマンドを実行すると、sbt チャネルのみを割り当てます。たとえば、次のように発行します。

```
BACKUP TABLESPACE users;
```

Recovery Manager は、バックアップ時に sbt タイプのチャネルのみを割り当てます。これは、sbt がデフォルトのデバイスであるためです。

バックアップおよびコピーに使用するデフォルトのデバイスは、チャネルを使用するコマンドに異なるデバイスを指定すると変更できます。たとえば、バックアップを次のように実行します。

```
BACKUP DEVICE TYPE sbt DATABASE;
```

デフォルトのデバイス・タイプが DISK ならば、この例のコマンドがデフォルトより優先され、sbt チャネルの構成を使用することになります。構成済みの sbt デバイスあるいは構成済みの sbt チャネルがない場合には、このコマンドは失敗することに注意してください。

リストア操作中、Recovery Manager は各デバイス用に構成したパラレル化の設定に応じてすべての自動チャネルを割り当てます。デフォルトのデバイス・タイプの構成は不要です。たとえば、デフォルトの sbt デバイスの PARALLELISM を 3 に構成して、DISK の PARALLELISM を 2 に構成した場合、Recovery Manager は自動的に 3 つの sbt チャネルと 2 つの DISK チャネルをリストア時に割り当てます。

**関連項目：** 8-12 ページ「[デフォルト・デバイス・タイプの変更](#)」

## 自動チャネルのネーミング規則

Recovery Manager では、チャネルのネーミングに次に示す規則が使用されます。ORA\_*devicetype*\_n で示される *devicetype* は、使用しているデバイス・タイプ（DISK や sbt\_tape など）を表し、n は、チャネル番号を表します。

---

**注意：** sbt と sbt\_tape デバイス・タイプは、同じものです。ただし、Recovery Manager の出力表示では、入力が sbt でも sbt\_tape でも、常に sbt\_tape と表示されます。

---

たとえば、Recovery Manager は最初の DISK チャネルを ORA\_DISK\_1、2 番目を ORA\_DISK\_2 というように名前付けします。Recovery Manager は最初の sbt チャネルに ORA\_SBT\_TAPE\_1、2 番目に ORA\_SBT\_TAPE\_2 というように名前付けします。チャネルをパラレル化する場合、Recovery Manager は常にチャネルを番号順に割り当てます。この番号は 1 から始まりパラレル化の設定値（CONFIGURE DEVICE TYPE ... PARALLELISM n）で終わります。この例を次に示します。

```
ORA_SBT_TAPE_1
ORA_SBT_TAPE_2
ORA_SBT_TAPE_3
```

自動チャネル割当ては、メンテナンス・コマンドにも適用されます。Recovery Manager が自動メンテナンス・チャネルを割り当てる場合、自動的に割り当てられる他のチャネルと同じネーミング規則が使用されます。手動でメンテナンス・チャネルを割り当てるために、ALLOCATE CHANNEL FOR MAINTENANCE を使用した場合には、Recovery Manager では次に示すチャネルのネーミング規則が使用されます。ORA\_MAINT\_*devicetype*\_n で示される *devicetype* は、使用しているデバイス・タイプ（たとえば、DISK あるいは sbt）を表し、n は、チャネル番号を表します。たとえば、Recovery Manager では手動で割り当てた 2 つのディスク・チャネルには、次に示す名前が使用されます。

```
ORA_MAINT_DISK_1
ORA_MAINT_DISK_2
```

CONFIGURE DEVICE TYPE コマンドを実行してデバイス・タイプの構成を行っていても、このデバイス・タイプに対する CONFIGURE CHANNEL を実行していない場合には、**Recovery Manager** は他のチャネル制御オプションなしで、すべてのチャネルを割り当てます。たとえば、sbt デバイスの構成とバックアップの実行を想定すると、次のようになります。

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 1;
BACKUP DEVICE TYPE sbt DATABASE;
```

実際には、**Recovery Manager** は次の処理を行います。

```
RUN
{
  ALLOCATE CHANNEL ORA_SBT_TAPE_1 DEVICE TYPE sbt;
  BACKUP DATABASE;
}
```

チャネルは、ORA\_ から始まるように名前付けされていますが、この接頭辞の使用は **Recovery Manager** 専用で予約されています。ALLOCATE CHANNEL コマンドを使用して手動でチャネルの割当てを行い、そのチャネルの名前に接頭辞 ORA\_ を付けることはできません。

## 自動チャネルの一般的な構成

CONFIGURE CHANNEL DEVICE TYPE コマンドにより、指定されたデバイス・タイプのすべての自動チャネルで使用される共通の設定が構成されます。つまり、このコマンドは設定のテンプレートを作成し、**Recovery Manager** では、これをデバイスに割り当てたすべてのチャネルで使用するということです。たとえば、ディスク・チャネルおよびテープ・チャネルを次のように構成します。

```
CONFIGURE CHANNEL DEVICE TYPE sbt PARMS='ENV=(NSR_SERVER=bksvr1)';
CONFIGURE CHANNEL DEVICE TYPE DISK RATE 5M FORMAT='?/oradata/%U' MAXOPENFILES=20;
```

これらのチャネルはチャネル番号が指定されていないため、チャネルの設定は、指定したタイプの自動チャネルのすべてに対して汎用性を持ちます。この構成はテンプレートとして機能します。たとえば、DISK の PARALLELISM を 10 に設定し、デフォルトのデバイス・タイプを DISK に設定します。すると、**Recovery Manager** は CONFIGURE CHANNEL DEVICE TYPE DISK コマンドでの設定を使用して、10 のディスク・チャネルを割り当てます。

**関連項目：** 8-13 ページ「[デバイス・タイプ用の一般的な自動チャネルの構成](#)」



## 特定の自動チャネルの構成

指定した自動チャネルに適用するパラメータを構成することもできます。Oracle Real Application Clusters 構成で実行する場合や、メディア・マネージャを使用していて、各チャネルに異なる設定が必要な場合に、個々のチャネルの構成が役立ちます。

たとえば、Oracle Real Application Clusters 環境では、次のように入力します。

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;  
CONFIGURE DEVICE TYPE sbt PARALLELISM 2;  
CONFIGURE CHANNEL 1 DEVICE TYPE sbt CONNECT 'node1' PARMS='ENV=(NSR_SERVER=bksvr1)';  
CONFIGURE CHANNEL 2 DEVICE TYPE sbt CONNECT 'node2' PARMS='ENV=(NSR_SERVER=bksvr1)';
```

この例では、Recovery Manager は ORA\_SBT\_TAPE\_1 および ORA\_SBT\_TAPE\_2 のバックアップ・ジョブに、それぞれ 2 つのチャネルを割り当てます。ORA\_SBT\_TAPE\_1 チャネルは、CHANNEL 1 の設定を使用し、ORA\_SBT\_TAPE\_2 チャネルは、CHANNEL 2 の設定を使用します。

共通の構成を作成するための CONFIGURE CHANNEL コマンドと、特定の構成を作成するための CONFIGURE CHANNEL コマンドを混在させることができます。汎用の自動チャネルは、明示的に構成されていないあらゆるチャネルで使用できる構成を簡単に作成します。たとえば、次のコマンドを実行すると仮定します。

```
CONFIGURE DEVICE TYPE DISK PARALLELISM 3;  
CONFIGURE CHANNEL DEVICE TYPE DISK MAXPIECESIZE = 2M;  
CONFIGURE CHANNEL 3 DEVICE TYPE DISK MAXPIECESIZE = 900K;
```

この例では、Recovery Manager で ORA\_DISK\_1 および ORA\_DISK\_2 に、MAXPIECESIZE = 2M オプションを使用して割当てを行います。このとき、番号を付けない DISK チャネルの設定が使用されます。Recovery Manager は、ORA\_DISK\_3 の割当てに MAXPIECESIZE = 900K を使用します。これは、このチャネルに手動でチャネル番号を割り当てているためです。Recovery Manager では、常にパラレル化のパラメータで指定したチャネル数の割当てを行います。

**関連項目：** 8-13 ページ「[デバイス・タイプ用の一般的な自動チャネルの構成](#)」

## 自動チャネル設定の消去

CONFIGURE コマンドではどれにでも、CLEAR オプションの指定ができます。CLEAR オプションは、指定した構成をデフォルト値に戻します。たとえば、次のコマンドを実行したと仮定します。

```
CONFIGURE DEVICE TYPE DISK CLEAR;           # returns DISK to default parallelism
CONFIGURE DEFAULT DEVICE TYPE CLEAR;         # returns to default device type of DISK
CONFIGURE CHANNEL DEVICE TYPE sbt CLEAR;     # removes all the options for sbt channel
CONFIGURE CHANNEL 3 DEVICE TYPE DISK CLEAR; # removes all configurations for 3rd channel
```

各 CONFIGURE コマンドでは、ユーザー入力の設定が削除され、構成がデフォルト値に戻されます。デフォルトの構成の設定は、独自の構成を入力する前に、SHOW ALL コマンドを実行すると表示されます。デフォルトには、番号記号 (#) の接頭辞が付きます。

**関連項目：** 各 CONFIGURE コマンドのデフォルト設定の詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。17-18 ページの「[すべての Recovery Manager の構成設定の表示](#)」も参照してください。

## 手動によるチャネル割当ての平行化

自動チャネル割当てが不要な場合には、RUN コマンド内で、複数のチャネルを手動で割り当てることができます。これにより、単一の Recovery Manager コマンドで、複数のバックアップやイメージ・コピーの読み込みまたは書き込みを平行で行うことができます。したがって、割り当てるチャネル数によって、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、必要な数のチャ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。

手動で割り当てた各チャネルは、ターゲットまたは補助データベースへの個別の接続を使用します。チャネルごとに異なる CONNECT 文字列を指定して、ターゲット・データベースの様々なインスタンスに接続できます。これは、ノード間で作業を分散する Oracle Real Application Clusters 構成で役立ちます。

Recovery Manager では、BACKUP、COPY および RESTORE コマンドの平行化が内部処理されます。指定する必要があるのは、次のとおりです。

- 多重化 ALLOCATE CHANNEL コマンド
- バックアップ、コピーまたはリストアの対象となるオブジェクト

Recovery Manager は、コマンドを順次実行します。つまり、現行のコマンドが完了した後で、次のコマンドが開始されます。平行化は、1 つのコマンド内のみ適用されます。したがって、データ・ファイルのバックアップを 3 つ作成するためには、BACKUP コマンドを個別に 3 回発行するのではなく、3 つのデータ・ファイルをすべて指定して、単一の BACKUP コマンドを発行します。

次に示すスクリプトでは、バックアップの作成にシリアルライズ化を使用しています。BACKUP コマンドを個別に 3 回使用して、それぞれ 1 つのファイルをバックアップしています。バックアップの進行するファイルは 1 つに限られるため、アクティブになるチャネルは毎回 1 つのみです。

```
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt;
  ALLOCATE CHANNEL c2 DEVICE TYPE sbt;
  ALLOCATE CHANNEL c3 DEVICE TYPE sbt;
  BACKUP DATAFILE 5;
  BACKUP DATAFILE 6;
  BACKUP DATAFILE 7;
}
```

次に示す文では同じ例で**パラレル化**を使用します。Recovery Manager の BACKUP コマンド 1 回で、3 つのデータ・ファイルを 3 つのチャネルすべてを使用してバックアップします。3 つのチャネルは**同時アクティブ**になり、各サーバー・セッションでそれぞれ 1 つのデータ・ファイルを別のテープ・ドライブにコピーします。

```
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt;
  ALLOCATE CHANNEL c2 DEVICE TYPE sbt;
  ALLOCATE CHANNEL c3 DEVICE TYPE sbt;
  BACKUP DATAFILE 5,6,7;
}
```

**関連項目：** Oracle Real Application Clusters 構成でのパラレル化の詳細は、『Oracle9i Real Application Clusters 概要』を参照してください。

## 手動および自動チャネル用のチャネル制御オプション

チャネル割当てを手動で行うか、自動で行うかにかかわらず、チャネル制御コマンドおよびオプションが使用できます。これにより、次のことを行います。

- Recovery Manager の操作時に使用するオペレーティング・システム・リソースの制御
- バックアップまたはリストアの並列度の調整 (BACKUP コマンドの FILESPERSET パラメータとの関係)
- I/O 帯域幅の使用の制限を KB、MB または GB 単位で設定 (ALLOCATE CHANNEL ... RATE および CONFIGURE CHANNEL ... RATE)
- バックアップ・ピースのサイズの制限を設定 (CONFIGURE CHANNEL および ALLOCATE CHANNEL コマンドで MAXPIECESIZE パラメータを指定)
- バックアップ・セットのサイズの制限を設定 (BACKUP および CONFIGURE コマンドで MAXSETSIZE パラメータを指定)

- 同時にオープンするファイル数の制限を設定 (ALLOCATE CHANNEL ... MAXOPENFILES および CONFIGURE 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 (SEND)
- メディア・マネージャにベンダー固有のパラメータを指定 (ALLOCATE CHANNEL ... PARMS および CONFIGURE CHANNEL ... PARMS)
- 操作を実行するインスタンスの指定 (ALLOCATE CHANNEL ... CONNECT および CONFIGURE CHANNEL ... CONNECT)

一部のプラットフォームでは、チャネル割当ておよびチャネル制御コマンドに、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、使用するオペレーティング・システムのアクセス方法または I/O ドライバ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メカニズムにより制御されます。

Oracle9i では、ALLOCATE CHANNEL コマンドにより、Recovery Manager はタイプ指定が DISK 以外の場合、常にメディア・マネージャとコンタクトを取ります。Oracle9i より前のリリースでは、ALLOCATE CHANNEL コマンドにより、Recovery Manager がメディア・マネージャとコンタクトを取ることはありませんでした。Recovery Manager は、BACKUP、RESTORE または RECOVER コマンドが発行されないかぎり、メディア・マネージャを呼び出しません。

---

---

**注意：** Recovery Manager はリリースにかかわらず、DEVICE TYPE DISK を指定したときに、サーバー・セッションの作成以外にオペレーティング・システムのリソースを割り当てることはありません。また、メディア・マネージャが呼び出されることもありません。

---

---

Recovery Manager には、構成済みの自動 DISK チャネルがあるため、ディスク上にのみ存在するファイル (ARCHIVELOG、DATAFILECOPY または CONTROLFILECOPY) に対して CHANGE ... AVAILABLE、CROSSCHECK または DELETE を実行する場合に、手動でメンテナンス・チャネルを割り当てる必要はありません。メンテナンス・チャネルはメンテナンス・タスクにのみ使用します。このチャネルをバックアップまたはリストア・ジョブの入力チャネルまたは出力チャネルとして使用することはできません。

**関連項目：** ALLOCATE CHANNEL の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。CONFIGURE の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## メディア・マネージャによるハードウェアの多重化

ハードウェアの多重化は、単一のシーケンシャル・デバイス（テープ・ドライブなど）に、Recovery Manager のバックアップを複数書き込むと発生します。オラクル社では、Recovery Manager のバックアップに対してハードウェアの多重化を行うことは、お薦めしていません。

## バックアップ・セット

BACKUP コマンドを実行する場合には、1 つ以上のバックアップ・セットを作成します。この項では、次の項目について説明します。

- [バックアップ・セット](#)
- [プロキシ・コピー](#)
- [バックアップ・セットの格納](#)
- [バックアップ・セットの圧縮](#)
- [アーカイブ・ログのバックアップ](#)
- [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・セットのバックアップ](#)

**関連項目：** バックアップの作成方法は、第 9 章「[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、『Oracle9i Recovery Manager リファレンス』を参照してください。

## バックアップ・セット

**バックアップ・セット**とは、1つ以上の物理**バックアップ・ピース**が含まれる論理オブジェクトです。デフォルトの状態では、1つのバックアップ・セットに含まれるバックアップ・ピースは1つです。バックアップ・ピースとは、バックアップしたデータ・ファイル、制御ファイルまたはアーカイブ REDO ログが含まれている、オペレーティング・システム・ファイルです。たとえば、10 個のデータ・ファイルを、単一のバックアップ・ピースを含む単一のバックアップ・セットに、バックアップできます（つまり、単一の出力ファイル）。1 つのファイルを複数のバックアップ・セットに分割することや、アーカイブ・ログとデータ・ファイルを1つのバックアップ・セットに混在させることはできません。

バックアップ・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から成り立つバックアップ・ピースの完全なセットです。バックアップ・セットは、RECOVERY MANAGER 固有の形式です。**イメージ・コピー**とは、単一のデータ・ファイル、制御ファイルまたはアーカイブ・ログの完全なイメージです。これは、Recovery Manager 固有の形式ではありません。

データ・ファイル、制御ファイル、アーカイブ REDO ログおよび現行のサーバー・パラメータ・ファイルをバックアップできます。また、他のバックアップ・セットやイメージ・コピーをバックアップすることもできます。これにより、ディスク上のバックアップをテープにバックアップすることができます。たとえば、次に示すようなコマンドを発行できます。ここに示す各コマンドでは、自動チャネルの構成が使用されています。

```
BACKUP DATABASE;  
BACKUP TABLESPACE users, tools;  
BACKUP (SPFILE) (CURRENT CONTROL FILE);  
BACKUP BACKUPSET 12;  
BACKUP DATAFILECOPY '/tmp/system01.dbf';
```

データ・ファイルのバックアップを行う場合は、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 ARCHIVELOG モードの場合、ターゲットがオープンしていても、クローズしていてもバックアップが可能です。データベースをクローズする必要もありません。NOARCHIVELOG モードのデータベースでは、バックアップを作成する前に一度クローズする必要があります。

---

**注意：** 移動された表領域は、読み込み / 書き込みが指定されるまでは、バックアップの作成ができません。

---

**関連項目：** 自動チャネルの詳細は、5-8 ページの「[自動チャネルの一般的な構成](#)」を参照してください。バックアップの作成方法は、9-3 ページの「[Recovery Manager を使用したデータベース・ファイルおよびアーカイブ・ログのバックアップ](#)」を参照してください。

## プロキシ・コピー

**プロキシ・コピー**は、特別なタイプのバックアップです。これにより、Recovery Manager はデータ転送の制御を、この機能をサポートするメディア・マネージャに委譲します。BACKUP コマンドに PROXY オプションを指定すると、そのバックアップはプロキシ・コピーになります。

バックアップを行おうとする各ファイルに、BACKUP PROXY コマンドを使用すると、Recovery Manager はメディア・マネージャに問合せを行い、プロキシ・コピーの実行が可能かどうかを判断します。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、Recovery Manager 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。PROXY ONLY オプションが、プロキシ・コピーの不可能な場合に使用されると例外が発生し、Oracle がエラー・メッセージを発行します。

Oracle では、プロキシ・コピーが行われた各ファイルを、制御ファイルに記録します。Recovery Manager は、このデータを使用してリカバリ・カタログの再同期化を行います。V\$PROXY\_DATAFILE ビューを使用することで、プロキシ・コピーの情報を取得します。CHANGE PROXY コマンドあるいは DELETE PROXY コマンドを使用することで、プロキシ・バックアップのステータスの変更や削除を行います。

---

**注意：** プロキシ機能は Oracle8i リリース 8.1.5 で導入されました。プロキシ・コピーをサポートする Recovery Manager 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、Recovery Manager はバックアップ・セットの作成にプロキシ・コピーを使用しません。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、Recovery Manager はリストアにプロキシ・コピー・バックアップを使用しません。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、Recovery Manager 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、『Oracle9i データベース・リファレンス』を参照してください。

## バックアップ・セットの格納

Recovery Manager 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。DEVICE TYPE DISK を指定する場合には、ランダム・アクセスのディスクにバックアップを行うことになります。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。つまり、CREATE TABLESPACE tablespace\_name DATAFILE 'filename' 文が機能するならば、'filename' は有効なバックアップ・パス名であるということです。

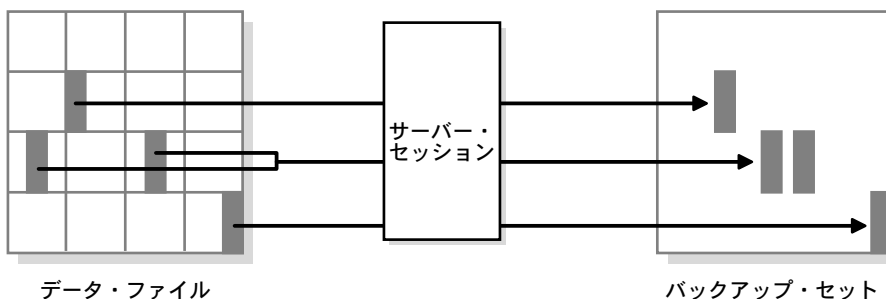
メディア管理システムが、使用中のオペレーティング・システムでサポートされていて使用可能な場合、バックアップ・セットを磁気テープなどの順次出力メディアに書き込むことができます。デバイス・タイプに sbt など（正確には、DISK 以外）を指定した場合には、メディア管理ソフトウェアでサポートされるどのメディアにでもバックアップを行うことができます。

**関連項目：** 不要とするバックアップを判断する方針の構成方法は、5-55 ページの「[バックアップの保存方針](#)」を参照してください。

## バックアップ・セットの圧縮

図 5-2 に示すように、Recovery Manager はバックアップの**圧縮**を実行します。つまり、使用されたことのないデータ・ファイルのブロックについては、バックアップを行いません。ただし、データ・ファイルのイメージ・コピーには、常にすべてのデータ・ブロックが含まれています。

図 5-2 バックアップ・セットの圧縮



**関連項目：** Recovery Manager がバックアップをバッファする方法は、14-3 ページの「[ディスク・バッファの割当て](#)」および 14-5 ページの「[テープ・バッファの割当て](#)」を参照してください。



## アーカイブ・ログのバックアップ

Recovery Manager では、アーカイブ REDO ログのバックアップのための付加機能を提供します。この項では、次の項目について説明します。

- バックアップ後のアーカイブ・ログの削除
- アーカイブ REDO ログのバックアップ・フェイルオーバー
- アーカイブ・ログのバックアップ時に切り替える自動オンライン REDO ログ
- バックアップが必要なアーカイブ・ログのバックアップ

### バックアップ後のアーカイブ・ログの削除

ログを直接テープにアーカイブすることはできませんが、Recovery Manager を使用して、アーカイブ・ログをディスクから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、Recovery Manager は、指定されたログ順序番号について、それぞれ 1 つのコピーをバックアップします。続いて、バックアップされたコピーをディスクから削除します。

DELETE ALL INPUT オプションを指定すると、Recovery Manager はそれぞれログ順序番号で指定した 1 つのコピーをバックアップします。続いて、指定した基準と一致するコピーを削除します。たとえば、LIKE パラメータを指定すると、Recovery Manager は文字列に一致したログのすべてを削除します。Recovery Manager では自動的に、必要なアーカイブ・ログをリカバリ時にテープからディスクへリストアします。

### アーカイブ REDO ログのバックアップ・フェイルオーバー

Oracle9i より前のリリースでは、Recovery Manager はアーカイブ REDO ログのバックアップを行うときに、最初のアーカイブ先のみを検索しました。Oracle9i では、Recovery Manager はアーカイブ REDO ログのフェイルオーバーを実行します。Recovery Manager では、次のことが行えます。

- 少なくとも 1 つのログが指定されたログ順序と一致していて、どのアーカイブ先でもスレッドが使用可能な場合、Recovery Manager は使用可能なログをバックアップします。
- Recovery Manager がアクセスしているログに破損ブロックが存在する場合、Recovery Manager は破損ブロックのない他のディレクトリを検索します。

Recovery Manager は、それぞれ個別のログ順序番号について常に 1 つのコピーのみをバックアップします。たとえば、ログ 121 から 124 を 2 つのアーカイブ先 /arch1 および /arch2 にアーカイブすることを想定します。制御ファイルには、次に示すアーカイブ・ログのレコードが含まれています。

順序番号	ファイル名
121	/arch1/archive1_121.arc
121	/arch2/archive1_121.arc
122	/arch1/archive1_122.arc
122	/arch2/archive1_122.arc
123	/arch1/archive1_123.arc
123	/arch2/archive1_123.arc
124	/arch1/archive1_124.arc
124	/arch2/archive1_124.arc

ただし、あるユーザーが誤ってログ 122 および 124 を /arch1 ディレクトリから削除したことは、Recovery Manager には通知されていません。この場合、次のバックアップを実行します。

BACKUP ARCHIVELOG FROM SEQUENCE 121 UNTIL SEQUENCE 125;

この場合、Recovery Manager は次の表に示すようにフェイルオーバー機能を使用してアーカイブ・ログをバックアップできます。

順序番号	Recovery Manager の検索対象	Recovery Manager の検索結果およびバックアップ
121	/arch1/archive1_121.arc	/arch1/archive1_121.arc
122	/arch1/archive1_122.arc (消失)	/arch2/archive1_122.arc
123	/arch1/archive1_123.arc	/arch1/archive1_123.arc
124	/arch1/archive1_124.arc (消失)	/arch2/archive1_124.arc

## アーカイブ・ログのバックアップ時に切り替える自動オンライン REDO ログ

UNTIL 句が指定されていない BACKUP ... ARCHIVELOG コマンドの開始時に、Recovery Manager は自動的に切替えを行って、現行のオンライン REDO ログをアーカイブします。この方法により、Recovery Manager はバックアップ・セットに現行の REDO ログを含めることができます。

データベースがオープンしている場合、アーカイブ・ログのバックアップ開始時に、Recovery Manager は切替えを試行して、次に示す規則に応じて現行のオンライン・ログをアーカイブします。

- Recovery Manager は、ALTER SYSTEM ARCHIVE LOG CURRENT を実行します。
- UNTIL 句または SEQUENCE パラメータが指定されている場合、Recovery Manager は切替えもオンライン・ログのアーカイブも行いません。

## バックアップが必要なアーカイブ・ログのバックアップ

バックアップを作成する場合には、BACKUP ... PLUS ARCHIVELOG コマンドを実行して、アーカイブ・ログのバックアップと同時にオンライン・ログのアーカイブも行います。この機能の目的は、バックアップしたデータ・ファイルが、一貫した状態でリカバリすることを保証するためです。

PLUS ARCHIVELOG を指定すると、Recovery Manager は次に示す処理を順番に実行します。

1. ALTER SYSTEM ARCHIVE LOG CURRENT を実行します。
2. BACKUP ARCHIVELOG ALL を実行します。
3. BACKUP コマンドで指定したファイルをバックアップします。
4. ALTER SYSTEM ARCHIVE LOG CURRENT を実行します。
5. バックアップ中に生成した、残りのアーカイブ REDO ログをすべてバックアップします。

明示的にアーカイブ・ログをバックアップする場合には、PLUS ARCHIVELOG を指定することはできません。ただし、データベースなど他のオブジェクトをバックアップする場合にのみ指定することができます。たとえば、次のようにコマンドを実行できます。

```
BACKUP DEVICE TYPE sbt DATABASE PLUS ARCHIVELOG;
```

---

---

**注意：** バックアップの最適化が使用可能な場合、Recovery Manager はすでに割当て済みのデバイスにバックアップされたアーカイブ・ログのバックアップをスキップします。

---

---

### 関連項目：

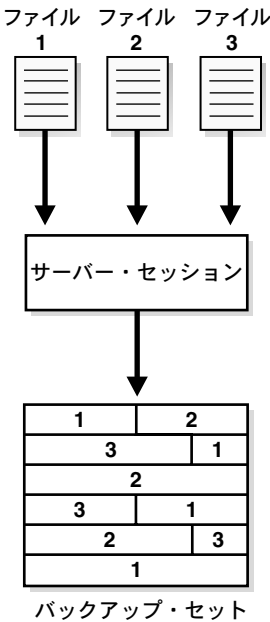
- PLUS ARCHIVELOG の使用方法は、9-15 ページの「[BACKUP ... PLUS ARCHIVELOG を使用したログのバックアップ](#)」を参照してください。
- 5-55 ページ「[バックアップの保存方針](#)」
- 5-62 ページ「[バックアップの最適化](#)」

## 多重バックアップ・セット

Recovery Manager の**多重化**の技法では、ディスク上の複数ファイルを同時に読み取って、これを同一のバックアップ・ピースに書き込みます。たとえば、Recovery Manager は 2 つのデータ・ファイルを同時に読み込んで、このデータ・ファイルからのブロックを単一のバックアップ・ピースの中に結合します。この使用例での多重化は、**二重化**とはまったく異なる点に注意してください。

図 5-3 に示すように、Recovery Manager は 1 つのバックアップ・ピースのみが含まれたバックアップ・セットに 3 つのデータ・ファイルをバックアップすることができます。このバックアップ・ピースには、混合された 3 つの入力ファイルのデータ・ブロックが含まれています。

図 5-3 データ・ファイルの多重化



多重化は、表 5-1 の要因によって、影響を受けます。

表 5-1 Recovery Manager の多重化

概念	定義
各バックアップ・セット内のファイル数	FILESPERSET の設定値、または各チャネルで読み込むファイル数の最小値
多重化	MAXOPENFILES の設定値、または各バックアップ・セット内のファイル数の最小値

FILESERSET パラメータは、各バックアップ・セット内に含めるデータ・ファイルの数を決定します。これに対し、MAXOPENFILES は、Recovery Manager が同時に読み込むことができるデータ・ファイルの数を定義します。

1 つの Recovery Manager のチャンネルを使用して、6 つのデータ・ファイルをバックアップすると仮定します。FILESERSET が 6 で、MAXOPENFILES が 1 の場合、チャンネルは 6 つのデータ・ファイルを 1 つのセットに含めますが、このファイルを多重化しません。これは、Recovery Manager では、ファイルからの複数の読み込みが行われないためです。チャンネルは 1 度に 1 つのファイルを読み取り、バックアップ・ピースに書き込みます。この場合、多重化の程度は 1 です。

次に、FILESERSET が 6 で、MAXOPENFILES が 3 の場合を想定します。この場合、チャンネルは読み込みと書き込みを次の順序で行います。

1. データ・ファイル 1、2 および 3 から同時に読み込みが行われ、バックアップ・ピースへの書き込みが行われます。
2. データ・ファイル 4、5 および 6 から同時に読み込みが行われ、バックアップ・ピースへの書き込みが行われます。

この例では、多重化の程度は 3 です（6 と 3 の小さい方）。

多重化したファイルの場合には、次のように実行できます。

- 特定のデータ・ファイルに割り当てたチャンネルで、明示的にデータ・ファイルをバックアップ・セットに分割するか、Recovery Manager が自動的に分割を選択するようにします。
- 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します。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に対してバックアップを行う場合に重要です。
- 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ます。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。
- データ・ファイルまたはアーカイブ・ログを格納するバックアップ・セットを作成します。ただし、両方を同時に格納することはできません。データ・ファイルとアーカイブ・ログを同じバックアップ・セットに書き込むことはできません。多重化バックアップ内のオブジェクトに対する Oracle の論理ブロック・サイズは、同じである必要があります。

多重化するファイルが多すぎると、リストアのパフォーマンスが低下することがあります。できるかぎり、同時に同じバックアップ・セットにリストアされるファイルをグループにしてください。Recovery Manager で 17 個のファイルをバックアップするために、FILESERSET = 64 および MAXOPENFILES = 16 を指定すると仮定します。データ・ファイル 17 の data17.f をリストアすると、Recovery Manager は最初の 16 個のファイルの多重化したデータを読み込み、その後、data17.f のデータの読み込みを開始します。この場合、data17.f のバックアップの開始点への移動に、このファイルをリストアするよりも多くの時間がかかることがあります。

関連項目：

- バックアップ時に、多重化がディスク・バッファに与える影響については、14-3 ページ「ディスク・バッファの割当て」を参照してください。
- バックアップを多重化する方法は、9-45 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：例」を参照してください。
- BACKUP コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

二重バックアップ・セット

Recovery Manager は、バックアップ・セット内の各バックアップ・ピースの複数のコピーを作成するための、効率的な機能を提供します。この機能は、バックアップ・セットの**二重化**とも呼ばれています。

バックアップ・セット内の各ピースと同一のコピーを 4 つまで作成することができます。これには、優先順位に従いリストした次のいずれかのコマンドを発行します。複数のコマンドが同時に影響する場合には、次の表で上にリストされているコマンドが、下にあるコマンドより優先されます。

コマンド	コマンドの制限事項	オブジェクトの制限事項
BACKUP COPIES	COPIES が指定されている BACKUP コマンドのみ	あらゆるオブジェクトが BACKUP コマンドで指定される
SET BACKUP COPIES	RUN ブロック内の、すべての BACKUP コマンド	あらゆるオブジェクトが BACKUP コマンドで指定される
CONFIGURE ... BACKUP COPIES	すべての BACKUP コマンド	データ・ファイル、制御ファイル、サーバー・パラメータ・ファイルおよびアーカイブ・ログのみ

**注意：** ディスクでの制御ファイルの自動バックアップは、特殊ケースであり、二重化されません。Recovery Manager では、常にコピーが 1 つのみ書き込まれます。

FORMAT のオプションには、4 つまでの値が指定できます。Recovery Manager では、BACKUP COPIES、SET BACKUP COPIES または CONFIGURE ... BACKUP COPIES が指定された場合にのみ、2 番目、3 番目そして 4 番目の値を使用します。この例では、データ・ファイル 7 のバックアップに 3 つのコピーを作成しています。

```
BACKUP DEVICE TYPE DISK COPIES 3 DATAFILE 7 FORMAT '/tmp/%U','?/oradata/%U','?/%U';
```

Recovery Manager は、各バックアップ・ピースの最初のコピーを /tmp に配置し、2 番目を ?/oradata に、3 番目を Oracle ホームに配置します。Recovery Manager は、それぞれに一意のバックアップ・セット・キーを個別に持つ、3 つのバックアップ・セットを作成することはありません。Recovery Manager は一意のキーを持つ 1 つのバックアップ・セットを作成し、セット内の各バックアップ・ピースの 3 つの同一のコピーを生成します。この例の LIST 出力を次に示します。

```
List of Backup Sets
=====

BS Key  Type LV Size
-----
1       Full  64K
List of Datafiles in backup set 1
File LV Type Ckp SCN    Ckp Time  Name
-----
7       Full  98410      08-FEB-02 /oracle/oradata/trgt/tools01.dbf

Backup Set Copy #1 of backup set 1
Device Type Elapsed Time Completion Time Tag
-----
DISK        00:00:01      08-FEB-02      TAG20020208T152314

List of Backup Pieces for backup set 1 Copy #1
BP Key  Pc# Status      Piece Name
-----
1       1   AVAILABLE  /tmp/01dg9tb2_1_1

Backup Set Copy #2 of backup set 1
Device Type Elapsed Time Completion Time Tag
-----
DISK        00:00:01      08-FEB-02      TAG20020208T152314
```



```

List of Backup Pieces for backup set 1 Copy #2
BP Key  Pc# Status      Piece Name
-----  -
2        1  AVAILABLE  /oracle/oradata/01dg9tb2_1_2

Backup Set Copy #3 of backup set 1
Device Type Elapsed Time Completion Time Tag
-----
DISK        00:00:01      08-FEB-02      TAG20020208T152314

List of Backup Pieces for backup set 1 Copy #3
BP Key  Pc# Status      Piece Name
-----  -
3        1  AVAILABLE  /oracle/01dg9tb2_1_3

```

各バックアップ・ピースに使用する `FORMAT` の値を選択するときに、**Recovery Manager** では、最初のフォーマット値をコピー番号 1 に、2 番目のフォーマット値をコピー番号 2 に、というように使用されます。フォーマット値の数がコピー数を超えた場合には、余分なフォーマット値は使用されません。フォーマット値の数がコピー数に満たない場合には、**Recovery Manager** は最初の値から再利用を始めます。

#### 関連項目：

- バックアップを二重化する方法は、9-18 ページの「[二重化バックアップ](#)」を参照してください。
- `CONFIGURE` コマンドの構文は、『*Oracle9i Recovery Manager リファレンス*』を参照してください。
- `SET` コマンドの構文は、『*Oracle9i Recovery Manager リファレンス*』を参照してください。

## バックアップの平行化

`PARALLELISM` を 2 以上に構成する場合や、複数チャネルを手動で割り当てた場合に、**Recovery Manager** は複数のバックアップ・セットを平行で書き込みます。サーバー・セッションは、指定ファイルのバックアップ作業を分割します。

---

---

**注意：** 1 つのバックアップ・セットを複数のチャネルにストライプすることはできません。

---

---

デフォルトの状態では、**Recovery Manager** は、どのチャンネルでどのデータベース・ファイルをバックアップするかを判断します。**BACKUP ... CHANNEL** コマンドを使用すると、指定したファイルのバックアップを行うチャンネルを手動で割り当てることができます。また、**FILESERSET** パラメータを使用して、バックアップ・セットに含まれるデータ・ファイル数を制限することもできます。この例では、デフォルトの自動割当て **DISK** チャンネルを使用して、ディスクへのパラレル化バックアップを示します。

```
BACKUP
(DATAFILE 1,2,3
  FILESERSET = 1
  CHANNEL ORA_DISK_1)
(DATAFILECOPY '/tmp/system01.dbf', '/tmp/tools01.dbf'
  FILESERSET = 2
  CHANNEL ORA_DISK_2)
(ARCHIVELOG FROM SEQUENCE 100 UNTIL SEQUENCE 102 THREAD 1
  FILESERSET = 3
  CHANNEL ORA_DISK_3);
```

また、次の例で示すように、手動でチャンネルの割当てを行うこともできます。

```
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt PARMS="ENV=(BACKUP_SERVER=tape_server1)";
  ALLOCATE CHANNEL c2 DEVICE TYPE sbt PARMS="ENV=(BACKUP_SERVER=tape_server2)";
  ALLOCATE CHANNEL c3 DEVICE TYPE sbt PARMS="ENV=(BACKUP_SERVER=tape_server3)";
  BACKUP
    (DATAFILE 1,2,3
      FILESERSET = 1
      CHANNEL c1)
    (DATAFILECOPY '/tmp/system01.dbf', '/tmp/tools01.dbf'
      FILESERSET = 2
      CHANNEL c2)
    (ARCHIVELOG FROM SEQUENCE 100 UNTIL SEQUENCE 102 THREAD 1
      FILESERSET = 3
      CHANNEL c3);
}
```


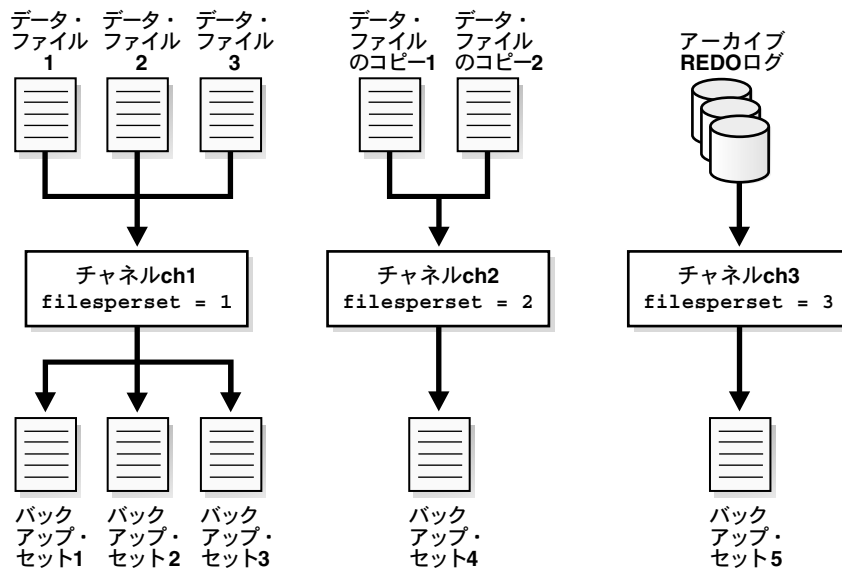
 **5-4** にパラレル化の例を示します。この例では、データ・ファイルのバックアップが **ch1** で行われています。**ch2** ではデータ・ファイルのコピーのバックアップを行い、**ch3** ではログのバックアップを行っています。

図 5-4 バックアップの平行化

**関連項目：**

- チャンネル割当てと平行化の関係の概要は、5-10 ページの「[手動によるチャンネル割当ての平行化](#)」を参照してください。
- バックアップの平行化の方法は、9-50 ページの「[チャンネルによるバックアップ作業負荷分散の決定：例](#)」を参照してください。
- BACKUP コマンドの CHANNEL パラメータの資料は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## バックアップ・セットのバックアップ

Recovery Manager の BACKUP BACKUPSET コマンドは、実際のデータベース・ファイルではなく、バックアップ・セットをバックアップします。このコマンドは、ディスクからディスクへのバックアップや、ディスクからテープへのバックアップをサポートしますが、テープからテープへのバックアップはサポートしていません。

BACKUP BACKUPSET コマンドは、デフォルトのディスク・チャンネルを使用して、ディスクからディスクへバックアップ・セットをコピーします。ディスクからテープへのバックアップを行う場合には、DEVICE TYPE sbt でチャンネルを手動で割り当てるか、自動 sbt チャンネルを構成する必要があります。

### バックアップ・セットのバックアップの使用方法

BACKUP BACKUPSET コマンドは、複数のメディアにバックアップを分散させるために役立ちます。たとえば、次に示す BACKUP コマンドを、本番バックアップ・スケジュールの一部として毎週実行します。

```
BACKUP DEVICE TYPE sbt BACKUPSET ALL;
```

---

**注意：** 自動チャンネルを使用する sbt デバイスへのバックアップには、最初に CONFIGURE DEVICE TYPE sbt コマンドを実行する必要があります。

---

これにより、すべてのバックアップが、ディスクとテープの両方に確実に存在させることができます。次の例のように、バックアップ・セットの各バックアップも二重化できることに注意してください（ただし、制御ファイルの自動バックアップは、BACKUP BACKUPSET では二重化されません）。

```
BACKUP COPIES 2 DEVICE TYPE sbt BACKUPSET ALL;
```

さらに、BACKUP BACKUPSET を使用すると、バックアップの領域割当てを管理できます。たとえば、新しいバックアップをディスク上に、古いバックアップはテープに保存することが必要だが、同時にディスクとテープの両方にバックアップが存在する必要はない、という状況を想定します。この場合、次のコマンドを定期的に行います。

```
BACKUP BACKUPSET COMPLETED BEFORE 'SYSDATE-7' DELETE INPUT;
```

このコマンドでは、1 週間以上前に作成されたバックアップ・セットをディスクからテープにバックアップします。その後、ディスクからバックアップ・セットを削除します。ここに示した DELETE INPUT は、DELETE ALL INPUT と同じ意味を持ちます。Recovery Manager は、存在するすべてのバックアップ・セットのコピーを削除します。4 つの場所にバックアップを二重化した場合、Recovery Manager はバックアップ・セット内のピースのコピーを、4 つともすべて削除します。

## バックアップ・セットのバックアップ実行時のバックアップ最適化

バックアップ・セットのバックアップを行うコマンドを発行したときに、**バックアップの最適化**が有効になっていて、同一のバックアップ・セットが同じデバイス・タイプにすでにバックアップされている場合、Recovery Managerはこのバックアップ・セットのバックアップをスキップします。たとえば、次のコマンドでは、すべてのバックアップ・セットをテープにバックアップします。このバックアップ・セットは、まだデバイス・タイプ `sbt` に存在していません。

```
BACKUP DEVICE TYPE sbt BACKUPSET ALL;
```

## バックアップ・セットのバックアップ実行時のバックアップ・フェイルオーバー

バックアップ・セットのバックアップを行うときに、バックアップを行うコピーが破損していたり消失している場合には、Recovery Managerはすべての使用可能なバックアップのコピーを検索します。この動作は、複数のアーカイブ先にあるアーカイブ REDO ログのバックアップを行うときの Recovery Manager の動作と同じです。

たとえば、バックアップ・セット 123 に 3 つのバックアップ・ピースが含まれていて、各バックアップ・ピースに 3 つのコピーが存在するため、`BACKUP COPIES 3` を発行したとします。バックアップ・ピースのコピーは、それぞれ異なるファイル・システムにあります。次の表では、ピースのコピーの一部が破損または消失しており、それ以外は異常がないことを表しています。

バックアップ・ピースの番号	ピースのコピー番号	コピーの状態
1	1	破損
1	2	正常
1	3	破損
2	1	消失
2	2	破損
2	3	正常
3	1	正常
3	2	破損
3	3	消失

次のコマンドを実行すると仮定します。

```
BACKUP BACKUPSET 123;
```

Recovery Manager は、自動フェイルオーバーを実行し、正常なコピーのみをバックアップ・セットに取り込みます。次の表では、Recovery Manager が取り込んだコピーを示します。

バックアップ・ピースの番号	ピースのコピー番号	コピーの状態
1	2	正常
2	3	正常
3	1	正常

**関連項目：** 5-23 ページの「[二重バックアップ・セット](#)」および 5-62 ページの「[バックアップの最適化](#)」を参照してください。

## バックアップ・オプション: 名前、サイズおよびスピード

Recovery Manager は、バックアップを制御するために、多数のオプションを提供しています。

この項では、次の項目について説明します。

- [バックアップ・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・セットの数とサイズ: 基本アルゴリズム](#)
- [バックアップ・セットの数とサイズ: 拡張アルゴリズム](#)
- [バックアップの I/O 読取りレート](#)

### 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には、Recovery Manager に一意の名前を自動的に判断させるか、FORMAT パラメータを使用します。ファイル名を指定しないと、Recovery Manager は %U 置換変数を使用することで、一意の名前を保証します。たとえば、次のように入力します。

```
BACKUP TABLESPACE users;
```

Recovery Manager は、バックアップ・ピースに一意の名前を自動的に生成します。

FORMAT パラメータでは、一意のファイル名を生成するために使用する置換変数を提供します。たとえば、次のようなコマンドを実行します。

```
BACKUP TABLESPACE users FORMAT = '/tmp/users_%u%p%c';
```

5-23 ページの「[二重バックアップ・セット](#)」で説明したように、FORMAT の値を 4 つまで指定できます。Recovery Manager では、BACKUP COPIES、SET BACKUP COPIES または CONFIGURE ... BACKUP COPIES を実行する場合にのみ、2 番目、3 番目および 4 番目の値を使用します。

---

**注意：** メディア・マネージャを使用する場合には、ベンダーのマニュアルで、名前のサイズやネーミング規則などの FORMAT に関する制限事項をチェックしてください。

---

**関連項目：** FORMAT パラメータおよび置換変数の詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## バックアップ・ピースのサイズ

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。バックアップ・ピースのサイズを制限しない場合、それぞれのバックアップ・セットに含まれるバックアップ・ピースは1つのみになります。各バックアップ・ピースのサイズを制限するには、CONFIGURE CHANNEL または ALLOCATE CHANNEL コマンドに MAXPIECESIZE オプションを指定します。このオプションでは、バックアップ・ピースのサイズを、指定されたバイト数までに制限します。

たとえば、ディスク・バックアップ用のバックアップ・ピースのサイズを 2GB に制限するには、次に示すように自動ディスク・チャンネルを構成してからバックアップを実行します。

```
CONFIGURE CHANNEL DEVICE TYPE DISK MAXPIECESIZE = 2G;
BACKUP DATAFILE 1;
```

LIST BACKUP コマンドにより、Recovery Manager は MAXPIECESIZE のサイズ制限にあわせて、1つのバックアップ・ピースではなく、5つのバックアップ・ピースを作成したことがわかります。

BS Key	Type	LV Size	Device	Type	Elapsed Time	Completion Time
-----						
29	Full	9728M	DISK		00:00:35	NOV 02 2001 18:29:26
List of Datafiles in backup set 29						
File	LV	Type	Ckp	SCN	Ckp Time	Name
-----						
1		Full	177590		NOV 02 2001 18:28:51	/oracle/oradata/trgt/system01.dbf
Backup Set Copy #1 of backup set 29						
Device	Type	Elapsed Time	Completion Time	Tag		
-----						
DISK		00:00:35	NOV 02 2001 18:29:26	TAG20011102T152701		
List of Backup Pieces for backup set 29 Copy #1						
BP Key	Pc#	Status	Piece Name			
-----						
53	1	AVAILABLE	/oracle/dbs/10d85733_1_1			
54	2	AVAILABLE	/oracle/dbs/10d85733_2_1			
55	3	AVAILABLE	/oracle/dbs/10d85733_3_1			
56	4	AVAILABLE	/oracle/dbs/10d85733_4_1			
57	5	AVAILABLE	/oracle/dbs/10d85733_5_1			

関連項目：

- ALLOCATE CHANNEL コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- CONFIGURE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。



## バックアップ・セットの数とサイズ: 基本アルゴリズム

BACKUP コマンドの *backupSpec* 句を使用すると、バックアップするオブジェクトを指定したり、他のオプションを指定できます。それぞれの *backupSpec* 句では、少なくとも 1 つのバックアップ・セットが生成されます。バックアップ・セットの合計数とサイズは、Recovery Manager で**基本アルゴリズム**と**拡張アルゴリズム**のどちらが使用されるかに応じて異なります。

Recovery Manager では、次のいずれかの条件に該当する場合は、基本アルゴリズムが使用されます。

- データ・ファイルと制御ファイル以外のファイルをバックアップする場合。
- オペレーティング・システムが、どのファイルがどのディスクにあるか、またはクラスタのどのノードにどのディスクとのアフィニティがあるかに関するデータを提供しない場合。
- BACKUP コマンドの DISKRATIO パラメータを手動で 0 に設定する場合。

前述の条件のいずれにも該当しない場合、Recovery Manager では拡張アルゴリズムが使用されます。拡張アルゴリズムについては、5-39 ページの「[バックアップ・セットの数とサイズ: 拡張アルゴリズム](#)」で説明しています。基本アルゴリズムは、DISKRATIO=0 に設定することでいつでも使用できます。

## バックアップ・セットの数とサイズに影響する要因

Recovery Manager のバックアップ・セットの特性を判断する際に、基本アルゴリズムは次の要素の影響を受けます。

- 指定した *backupSpec* 句の数
- 各 *backupSpec* 句内に指定または暗黙的に指定された入力ファイルの数
- 割り当てるチャンネルの数
- バックアップ・セットのファイル数を制限する、FILESERSET パラメータ
- バックアップ・セットの最大サイズを指定する、MAXSETSIZE パラメータ (CONFIGURE および BACKUP コマンドで指定)

バックアップ・セットを作成するためのアルゴリズムで最も重要なルールは次のとおりです。

- バックアップ・ジョブで作業を実行する各割り当てチャンネル（アイドル状態ではない各チャンネル）では、最低 1 つのバックアップ・セットが生成されます。デフォルトでは、このバックアップ・セットに 1 つのバックアップ・ピースが含まれています。

---

---

**注意：** Recovery Manager は、バックアップ・ピースを順番に書き込みます。1 つのバックアップ・ピースを複数の出力デバイスにストライプ化することはできません。たとえば、Recovery Manager では、バックアップ・ピースの半分をディスクに、もう半分を他のディスクに同時に書き込むことはありません。

---

---

- Recovery Manager は、常にバックアップの負荷を分割しようとするため、割り当てられた全チャンネルの作業量はほぼ同じです。
- 各バックアップ・セット内の最大データ・ファイル数は、BACKUP コマンドの FILESPERSET パラメータにより決定されます。
- バックアップ・セットのデータ・ファイルの多重化数は、FILESPERSET および MAXOPENFILES の小さい方に制限されます。
- バックアップ・セットの最小サイズは、CONFIGURE あるいは BACKUP コマンドの MAXSETSIZE パラメータで決定されます。

## FILESPERSET パラメータの使用方法

FILESPERSET パラメータは、バックアップ・セットのファイル数（制御ファイル、データ・ファイルあるいはアーカイブ・ログ）を制限します。デフォルト値は、次に示すように Recovery Manager により計算されます。Recovery Manager はファイル数をチャンネル数で除算して切り上げた値と 64 を比較し、小さい方の値を FILESPERSET に設定します。たとえば、70 ファイルを 1 つのチャンネルでバックアップする場合、Recovery Manager では、70 を 1 で除算した結果の 70 と 64 を比較して、小さい方の 64 を FILESPERSET に設定します。

バックアップ・セットの数は、データ・ファイルの数を FILESPERSET で除算して切り上げた数が、Recovery Manager により生成されます。たとえば、70 データ・ファイルをバックアップして、FILESPERSET が 64 の場合、Recovery Manager では 2 つのバックアップ・セットが生成されます。

Recovery Manager は、合計ブロック数を合計セット数で除算した値とほぼ同じサイズで、各バックアップ・セットを作成しようとします。バックアップでの合計ブロック数は、バックアップされる各ファイルのブロック数と同じです。たとえば、それぞれが 50 個のブロックを含む 70 ファイルをバックアップする場合に、セット数が 2 であれば、Recovery Manager は各バックアップ・セットをほぼ 3500/2、つまり 1750 ブロックにします。

**FILESERSET のデフォルト値の使用法: 例** 次の例では、Recovery Manager で、3 チャンネルを使用して 8 ファイルをバックアップすることを想定します。FILESERSET が指定されていないため、Recovery Manager は、64 と 3 (8 を 3 で除算して切上げ) を比較して、FILESERSET に 3 を設定します。Recovery Manager は、3 つのバックアップ・セットを作成して、各セットがほぼ同じサイズになるように、ファイルをセットにグループ分けします。Recovery Manager の LIST 出力の例を次に示します。

List of Backup Sets

=====

BS Key	Type	LV	Size	Device	Type	Elapsed Time	Completion Time
6	Full		320K	DISK		00:00:05	08-FEB-02
BP Key: 17 Status: AVAILABLE Tag: TAG20020208T153359							
Piece Name: /oracle/dbs/06dg9tv9_1_1							
List of Datafiles in backup set 6							
File	LV	Type	Ckp	SCN	Ckp Time	Name	
3		Full	98410		08-FEB-02	/oracle/oradata/trgt/cwmlite01.dbf	
4		Full	98410		08-FEB-02	/oracle/oradata/trgt/drsys01.dbf	
6		Full	98410		08-FEB-02	/oracle/oradata/trgt/indx01.dbf	
8		Full	98410		08-FEB-02	/oracle/oradata/trgt/users01.dbf	
BS Key	Type	LV	Size	Device	Type	Elapsed Time	Completion Time
7	Full		312K	DISK		00:00:07	08-FEB-02
BP Key: 18 Status: AVAILABLE Tag: TAG20020208T153359							
Piece Name: /oracle/dbs/07dg9tv9_1_1							
List of Datafiles in backup set 7							
File	LV	Type	Ckp	SCN	Ckp Time	Name	
2		Full	98410		08-FEB-02	/oracle/oradata/trgt/undotbs01.dbf	
5		Full	98410		08-FEB-02	/oracle/oradata/trgt/example01.dbf	
7		Full	98410		08-FEB-02	/oracle/oradata/trgt/tools01.dbf	
BS Key	Type	LV	Size	Device	Type	Elapsed Time	Completion Time
8	Full		196M	DISK		00:00:36	08-FEB-02
BP Key: 19 Status: AVAILABLE Tag: TAG20020208T153359							
Piece Name: /oracle/dbs/08dg9tv9_1_1							
SPFILE Included: Modification time: 08-FEB-02							
List of Datafiles in backup set 8							
File	LV	Type	Ckp	SCN	Ckp Time	Name	
1		Full	98410		08-FEB-02	/oracle/oradata/trgt/system01.dbf	

**FILESERSET の指定: 例** 3つのチャネルを使用して、8つのデータ・ファイルをバックアップするとき FILESERSET = 2 を指定した場合、Recovery Manager は2以下のデータ・ファイルを、各バックアップ・セット内に配置します。その結果、最低（8を2で除算して）4つのバックアップ・セットが Recovery Manager により作成されます。データ・ファイルのサイズやその他の要素によって、さらに多くのバックアップ・セットが作成される場合があります。Recovery Manager の LIST 出力の例を次に示します。

RMAN> list backup;

List of Backup Sets  
=====

BS Key	Type	LV	Size	Device	Type	Elapsed Time	Completion Time
-----							
9	Full		144K	DISK		00:00:03	08-FEB-02
	BP Key: 20 Status: AVAILABLE Tag: TAG20020208T153550						
	Piece Name: /oracle/dbs/0adg9u2n_1_1						
	List of Datafiles in backup set 9						
	File	LV	Type	Ckp	SCN	Ckp Time	Name
	-----						
	4		Full	98410		08-FEB-02	/oracle/oradata/trgt/drsys01.dbf
	7		Full	98410		08-FEB-02	/oracle/oradata/trgt/tools01.dbf

BS Key	Type	LV	Size	Device	Type	Elapsed Time	Completion Time
-----							
10	Full		144K	DISK		00:00:04	08-FEB-02
	BP Key: 21 Status: AVAILABLE Tag: TAG20020208T153550						
	Piece Name: /oracle/dbs/0bdg9u2n_1_1						
	List of Datafiles in backup set 10						
	File	LV	Type	Ckp	SCN	Ckp Time	Name
	-----						
	5		Full	98410		08-FEB-02	/oracle/oradata/trgt/example01.dbf
	8		Full	98410		08-FEB-02	/oracle/oradata/trgt/users01.dbf

BS Key	Type	LV	Size	Device	Type	Elapsed Time	Completion Time
-----							
11	Full		144K	DISK		00:00:06	08-FEB-02
	BP Key: 22 Status: AVAILABLE Tag: TAG20020208T153550						
	Piece Name: /oracle/dbs/09dg9u2n_1_1						
	List of Datafiles in backup set 11						
	File	LV	Type	Ckp	SCN	Ckp Time	Name
	-----						
	3		Full	98410		08-FEB-02	/oracle/oradata/trgt/cwmlite01.dbf
	6		Full	98410		08-FEB-02	/oracle/oradata/trgt/indx01.dbf

BS Key	Type	LV	Size	Device	Type	Elapsed Time	Completion Time
-----							
12	Full		152K	DISK		00:00:02	08-FEB-02

```

BP Key: 23   Status: AVAILABLE   Tag: TAG20020208T153550
Piece Name: /oracle/dbs/0cdg9u2r_1_1
List of Datafiles in backup set 12
File LV Type Ckp SCN    Ckp Time  Name
-----
2          Full 98410      08-FEB-02 /oracle/oradata/trgt/undotbs01.dbf

BS Key  Type LV Size      Device Type Elapsed Time Completion Time
-----
13      Full 196M      DISK          00:00:31      08-FEB-02
BP Key: 24   Status: AVAILABLE   Tag: TAG20020208T153550
Piece Name: /oracle/dbs/0ddg9u2t_1_1
SPFILE Included: Modification time: 08-FEB-02
List of Datafiles in backup set 13
File LV Type Ckp SCN    Ckp Time  Name
-----
1          Full 98410      08-FEB-02 /oracle/oradata/trgt/system01.dbf

```

データ・ファイルまたはデータ・ファイルの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ム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、個々のファイルのリストアが速くなります。これは、スキップできるはずのデータが他のデータ・ファイルと関連しているために少なくなるためです。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リストアする必要がある場合が多いためです。

**関連項目：** `backupSpec` 句の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。Recovery Manager のバッファ管理について、詳細は第 14 章「Recovery Manager のチューニング」を参照してください。

## MAXSETSIZE パラメータの使用方法

1 つのチャネルで、1 つ以上のバックアップ・セットが生成されます。バックアップ・セットの最大バイト数を指定するために、CONFIGURE あるいは BACKUP コマンドで MAXSETSIZE パラメータを使用します。このパラメータでバックアップ・セットの全体サイズを制限することにより、セット内のファイル数を間接的に制限し、Recovery Manager が追加のバックアップ・セットを作成するように強制します。

MAXSETSIZE パラメータを使用して、バックアップ・セットの合計サイズを、メディア・マネージャまたはオペレーティング・システムでサポートする最大のファイル・サイズに制限します。これにより、ディスク・バックアップが領域制限のために失敗することも、テープ・バックアップが複数ボリュームに及ぶことも防げます。

**MAXSETSIZE と FILESPERSET の関係** 次の表では、MAXSETSIZE パラメータと FILESPERSET パラメータを比較しています。

パラメータ	意味	設定する内容	Recovery Manager で 設定される内容
MAXSETSIZE	バックアップ・セットの最大ファイル数を設定せずに、バックアップ・セットの最大バイト数を設定します。	バックアップ・セットの最大サイズ	パラメータの設定に応じた各セット内のファイル数。
FILESPERSET	バックアップ・セットの最大バイト数を設定せずに、バックアップ・セットの最大ファイル数を設定します。	バックアップ・セットに含める最大ファイル数。	パラメータの制限に応じたバックアップ・セットのバイト数。

FILESPERSET にはデフォルト値があるため、MAXSETSIZE を設定する場合は、FILESPERSET の動作も考慮に入れる必要があります。両方のパラメータを使用する場合は、次の点を考慮します。

- バックアップ・セットの数は、次の数より大きくなります。
  - － ブロックの合計数を MAXSETSIZE で除算した値
  - － データ・ファイルの合計数を FILESPERSET で除算した値
- Recovery Manager では、FILESPERSET と MAXSETSIZE の両方の制限が施行されます。必要に応じて、Recovery Manager は前述の計算より多くのバックアップ・セットを作成します。
- Recovery Manager は最適な複合アルゴリズムを使用するため、大部分のバックアップ・セットは、できるかぎり MAXSETSIZE の制限に近いサイズで、FILESPERSET の制限に近いファイル数になります。

**MAXSETSIZE の指定: 例** それぞれ 1000 ブロックが含まれる 50 ファイルのデータ・ファイルをバックアップするとします。データベースをバックアップするために、バックアップ・セットの最大サイズを 10MB に設定するには、次のようにコマンドを発行します。

```
BACKUP DATABASE MAXSETSIZE = 10M;
```

FILESERSET を設定していないので、Recovery Manager はデフォルト値を計算するために 64 と 50/2 を比較します。その結果、FILESERSET = 25 が設定されます。Recovery Manager は次に示す値を比較して、大きな値を選択します。

- 50000/10000 (ブロックの合計数 / MAXSETSIZE) = 5
- 50/25 (ファイルの合計数 / FILESPERSET) = 2

この結果、Recovery Manager は 5 つのバックアップ・セットを作成し、各バックアップ・セットには 25 以下のファイルが含まれ、合計サイズは 10MB 以下になります。

MAXSETSIZE の設定値が、入力ファイルの最大サイズよりも小さいと、RMAN-06183 エラーが表示されます。

```
RMAN-06183: データ・ファイルまたはデータ・ファイル・コピーが MAXSETSIZE を超えています: ファイル番号 # 1
/oracle/oradata/trgt/system01.dbf
```

**関連項目:** MAXSETSIZE パラメータの詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## バックアップ・セットの数とサイズ: 拡張アルゴリズム

拡張アルゴリズムでは、5-33 ページの「バックアップ・セットの数とサイズ: 基本アルゴリズム」で説明したのと同じ要因を使用して、バックアップ・セットの数とサイズが決定されます。ただし、拡張アルゴリズムは、DISKRATIO パラメータの影響も受けます。DISKRATIO=n の場合、各バックアップ・セットは少なくとも n 個のディスク・ドライブからデータを読み取る必要があります。Recovery Manager は、データベース・サーバーから取得したファイル位置情報を使用して、どのデータ・ファイルがどのディスク・ドライブにあるかを判断します。

FILESERSET を設定し、DISKRATIO を設定しなければ、DISKRATIO はデフォルトで FILESERSET と同じ値に設定されます。どちらのパラメータも指定しない場合、DISKRATIO のデフォルト値は 4 です。Recovery Manager は、DISKRATIO の値をバックアップに関係するデバイスの実際の数と比較して、最小値を使用します。たとえば、DISKRATIO=4 で、データ・ファイルが 3 つのディスクにある場合、Recovery Manager はデータ・ファイルを 3 つのバックアップ・セットに分散させます。

データベースに含まれる 50 個のデータ・ファイルが 6 つのディスクに分散しており、オペレーティング・システムではこのディスク競合情報をサーバーに配信できるとします。1 つの sbt チャンネルを構成し、次のコマンドを実行します。

```
BACKUP DATABASE;
```

Recovery Manager では拡張アルゴリズムが使用されます。基本アルゴリズムでは、FILESERSET も MAXSETSIZE も指定されていないため、Recovery Manager ではバックアップ・セットを 1 つ生成する必要があることが示されます。拡張アルゴリズムでも、DISKRATIO が検査されますが、この場合はデフォルトの 4 になっています。そのため、各バックアップ・セットには少なくとも 4 つのディスクからのデータ・ファイルを含める必要があります。Recovery Manager は 6 つのディスクの全データ・ファイルを含む 1 つのバックアップ・セットしか生成しないため、DISKRATIO を指定しても結果は変わりません。

BACKUP コマンドを次のように変更するとします。

```
BACKUP DATABASE FILESPERSET 5;
```

この場合、基本アルゴリズムでは 10 個のバックアップ・セットが生成されます。拡張アルゴリズムでは、DISKRATIO も考慮されます。ここでは、DISKRATIO はデフォルトの 5 (FILESERSET と同じ値) に設定されています。そのため、拡張アルゴリズムでは、10 個のバックアップ・セットにそれぞれ 6 つのディスクのうち少なくとも 5 つからのデータ・ファイルを含める必要があることが示されます。

拡張アルゴリズムでは、ノードのアフィニティも認識されることに注意してください。ノードのアフィニティが使用されるのは、Real Application Clusters 構成で異なるノードでチャネルを割り当てる場合のみで、Recovery Manager では、あるノードより別のノードから特定のデータ・ファイルを読み取る方が効率的であることが検出されます。

## バックアップの I/O 読取りレート

バックアップのスピードを制限するには、ALLOCATE CHANNEL または CONFIGURE CHANNEL コマンドに RATE オプションを指定します。RATE オプションは、Recovery Manager がチャネルを読み取る、1 秒間当りの最大バイト数を指定します。

たとえば、自動チャネルを構成して、チャネル c1 の読込みを 700KB 毎秒に、チャネル c2 の読込みを 1MB 毎秒に制限します。

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 2;
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE CHANNEL 1 DEVICE TYPE sbt RATE 700K;
CONFIGURE CHANNEL 2 DEVICE TYPE sbt RATE 1M;
```

```
BACKUP
  (TABLESPACE system, undotbs
    CHANNEL ORA_SBT_TAPE_1)
  (TABLESPACE users, tools, indx
    CHANNEL ORA_SBT_TAPE_2);
```

実質的に RATE オプションは、バックアップ・ジョブがコンピュータの帯域幅を過度にコンシュームしないように Recovery Manager を調整します。



**関連項目：** Recovery Manager のパフォーマンスを最適化する方法のヒントは、14-11 ページの「バックアップ・パフォーマンスの向上」を参照してください。

## バックアップ・タイプ

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](#)
- [全体バックアップ](#)
- [増分バックアップ](#)
- [Recovery Manager がバックアップ可能なファイルのタイプ](#)

## バックアップ・タイプ

作成するバックアップのタイプを Recovery Manager で制御します。表 5-2 で説明するように、Recovery Manager のバックアップは次の種類に分類されます。

- 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

表 5-2 バックアップ・タイプ

バックアップ・タイプ	定義
全体	増分でないバックアップ。全体バックアップは、データ・ファイル内のすべての使用されたデータ・ブロックを含みます。制御ファイルおよびアーカイブ・ログの全体バックアップは、常にファイル内のすべてのブロックを含みます。 <b>注意：</b> 全体バックアップは、データベース全体のバックアップとは異なります。データベース全体のバックアップとは、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。
増分	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・ファイルのバックアップ。増分バックアップには、ベースとなる全体バックアップまたは増分レベル 0 のバックアップが必要です。
オンライン	オンラインのバックアップ。データ・ファイルの読み込み / 書き込み時にデータベースはオープンしています。 <b>注意：</b> ALTER TABLESPACE BEGIN BACKUP 文を使用して、表領域をバックアップ・モードにしないでください。Recovery Manager では、別の方法を使用してオンライン・バックアップの一貫性を保証します。

表 5-2 バックアップ・タイプ（続き）

バックアップ・タイプ	定義
クローズ状態	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。
一貫性	データベースを、クラッシュまたは ABORT オプションでシャットダウンしていない状態でマウントして（ただし、オープンしないで）取ったバックアップ。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一致しており、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、リカバリを実行することなくリストアできます。
非一貫性	次に示す状態のターゲット・データベースの任意部分のバックアップ。 <ul style="list-style-type: none"><li>■ オープンした状態</li><li>■ マウントする前に、クラッシュまたは SHUTDOWN ABORT が実行された状態。</li></ul> 非一貫性バックアップを一貫性のあるものにするには、リカバリ操作を行う必要があります。

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、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。ただし、未使用のデータ・ファイル・ブロックはスキップされます。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には、Recovery Manager はブロックをスキップしません。

**注意：** 全体バックアップとデータベース全体のバックアップは異なります。「全体バックアップ」とは、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影響を与えず、増分バックアップ方法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の内容は変わりません。

Recovery Manager では、データ・ファイル、制御ファイルおよびアーカイブ REDO ログの全体バックアップの作成とリストアが行えます。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

この項では、次の項目について説明します。

- [増分バックアップ](#)
- [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](#)

### 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。Recovery Manager を使用することで、データ・ファイル、表領域あるいはデータベース全体の増分バックアップが行えます。Recovery Manager は、制御ファイルを増分バックアップ・セットに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。

メディア・リカバリ時、Recovery Manager はリストアしたファイルを調査して、このファイルが増分バックアップを使用してリカバリ可能かどうかを判断します。Recovery Manager では、常にアーカイブ・ログに対して増分バックアップが選択されます。Recovery Manager ではリカバリ時に、増分バックアップをデータ・ファイルに適用するための、基本増分バックアップをリストアする必要がありません。たとえば、データベースにあるデータ・ファイルの非増分イメージ・コピーをリストアできます。Recovery Manager は、このデータ・ファイルをリカバリするために増分バックアップを使用します。

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・マネージャを使用している場合にはテープを節約し、ディスク・バックアップを作成している場合にはディスク領域を節約するため。
- 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するため。
- テープ書込み I/O に使用可能なテープ帯域幅の合計を、ディスク読取り I/O に使用されるディスク帯域幅の合計より大幅に少なくするため。
- NOLOGGING オプションで作成したオブジェクトをリカバリできるように変更するため（ダイレクト・ロード・インサートはログを再実行しませんが、データ・ブロックを変更するため、増分バックアップで獲得されます）。

- NOARCHIVELOG データベースのバックアップのサイズを減少させるため。毎回、データベース全体のバックアップを作成するかわりに、増分バックアップを作成します。NOARCHIVELOG データベースの増分バックアップは、一貫したシャットダウンの後でのみ有効です。

効果的な方針として、増分バックアップをディスクに作成した後、BACKUP BACKUPSET を実行してバックアップをメディア・マネージャにコピーします。これにより、増分バックアップを直接テープに作成すると発生する場合がある、テープ・ストリームの継続の問題を回避できます。増分バックアップは完全バックアップほど大きくないため、簡単にディスクに作成できます。

### マルチレベル増分バックアップ

Recovery Manager では、**マルチレベル増分バックアップ**が作成できます。それぞれの増分レベルは、整数で表されます。たとえば、0、1、2 というようになります。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ベースとなります。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 0 のバックアップと全体バックアップの唯一の違いは、全体バックアップには増分方針が含まれていない、ということです。レベル 0 のバックアップが存在しない場合に、レベル 1 以降のバックアップを実行すると、Recovery Manager はベースとして使用できるレベル 0 のバックアップを自動的に作成します。

レベル  $n$  の増分バックアップで、 $n$  が 1 以上の場合には、次のいずれかでバックアップが行われます。

- レベル  $n$  以下の、最新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**と呼びます）。
- レベル  $n-1$  以下の、最新のバックアップ以降に変更された全ブロック（これを**累積バックアップ**と呼びます）。

---

**注意：** 多くの状況で、差分バックアップより累積バックアップの方が適切な選択となります。これは、リカバリ時に適用する必要がある増分バックアップが少ないためです。

---

マルチレベル増分バックアップを行う利点は、Recovery Manager で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。Recovery Manager 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要する時間と増分バックアップに要する時間はほぼ同じです。

1 以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。

## 増分バックアップの仕組み

データ・ファイルの各データ・ブロックにはシステム変更番号（SCN）が含まれています。この SCN は、そのブロックが最後に変更された時点の SCN です。増分バックアップ時、Recovery Manager は入力ファイルの各データ・ブロックの SCN を読み取り、これを親の増分バックアップのチェックポイント SCN と比較します。Recovery Manager は、毎回ファイル全体を読み込み、ブロックが使用されているかどうかを確認します。

親のバックアップは、Recovery Manager が SCN を比較するために使用するバックアップです。現行の増分がレベル *n* の差分バックアップの場合、その親は、レベル *n* 以下の最新の増分です。現行の増分がレベル *n* の累積バックアップの場合、その親は、レベル *n-1* 以下の最新の増分です。入力データ・ブロックの SCN が親のチェックポイント SCN 以上の場合、Recovery Manager はこのブロックをコピーします。

Recovery Manager はこのメカニズムを、リカバリ中に変更されたデータが含まれている、すべてのブロックに適用します。たとえ、そのオブジェクトが NOLOGGING オプションを使用して作成されたものでも同じ結果になります。したがって、増分バックアップを作成することは、NOLOGGING 属性が設定されている表への変更内容が失われないための保護対策となります。

**関連項目：** NOLOGGING モードの詳細は、『Oracle9i データベース概要』を参照してください。

## 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、Recovery Manager は、レベル *n* 以下の最新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 Recovery Manager が判断し、新しい方のバックアップ以後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、Recovery Manager はベースとなるレベル 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。使用可能なレベル 0 のバックアップが存在しない場合、Recovery Manager はこのファイル用に新規にベースとなるレベル 0 のバックアップを作成します。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 5-5 差分増分バックアップ（デフォルト）

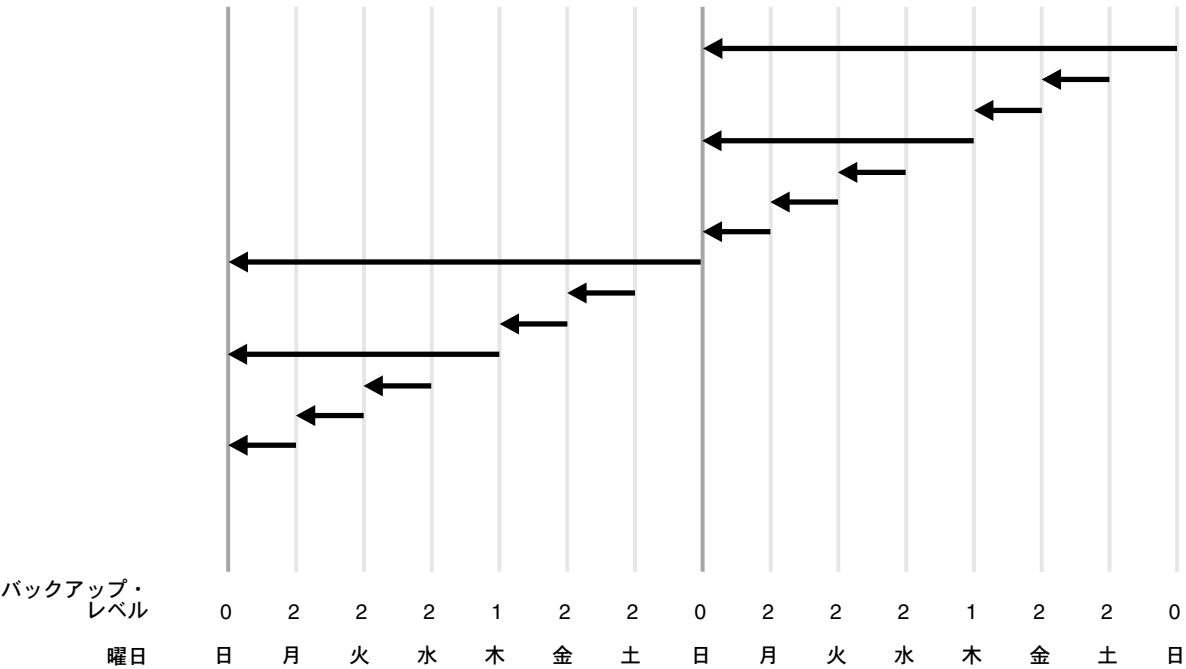


図 5-5 に示す例では、バックアップは次のようになります。

- 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用さ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 以下の最新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 以下の最新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 以下の最新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 以下の最新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 の差分増分バックアップでは、レベル  $n$  以下の最新の増分バックアップ以降に変更された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 以下の最新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 の差分増分バックアップでは、レベル  $n$  以下の最新の増分バックアップ以降に変更された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 以下の最新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。

- 金曜日

レベル 2 の差分増分バックアップでは、レベル  $n$  以下の最新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 以下の最新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 の差分増分バックアップでは、レベル  $n$  以下の最新の増分バックアップ以降に変更された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 以下の最新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。

- このサイクルが次の週も繰り返されます。

## 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、Recovery Manager 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 Recovery Manager が判断し、最後のレベル 1 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。利用できるレベル 1 のバックアップがない場合、Recovery Manager 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用すると、特定レベルからの増分バックアップが 1 つで済むため、リストアに必要な作業が軽減されます。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ため、必要な領域と時間が差分バックアップより増加します。

図 5-6 累積増分バックアップ

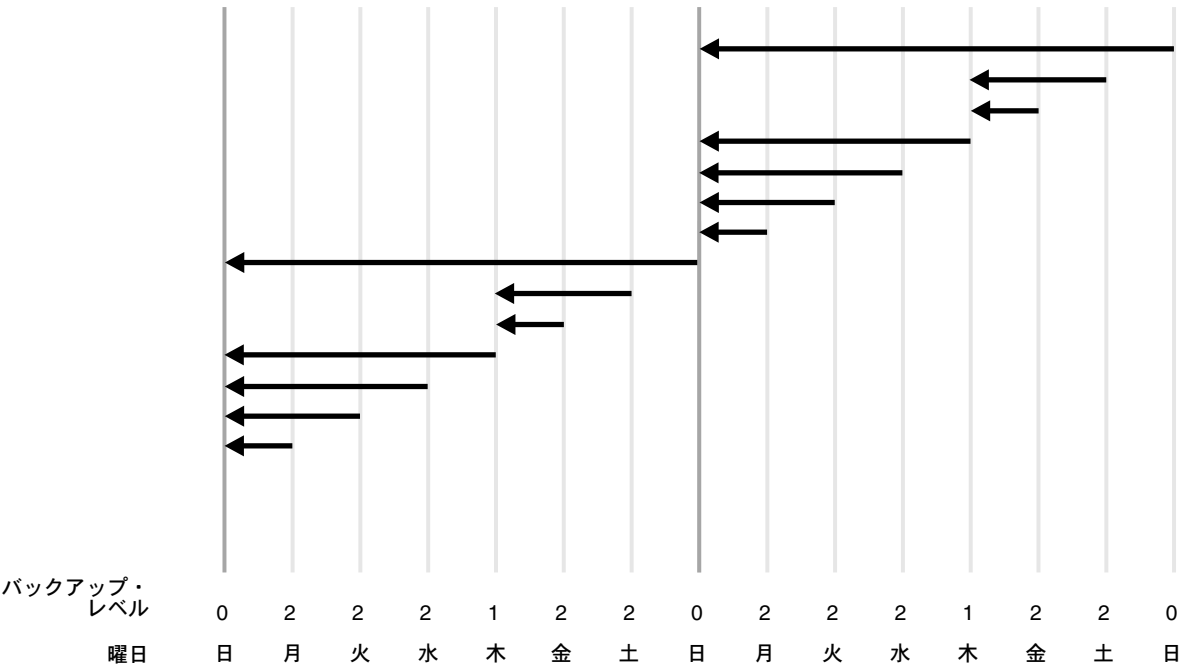


図 5-6 に示す例では、バックアップは次のようになります。

- 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用さ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  $n-1$  以下の最新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 (2-1) 以下の最新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  $n-1$  以下の最新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 1 (2-1) 以下の最新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  $n-1$  以下の最新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 (2-1) 以下の最新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、月曜日と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  $n-1$  以下の最新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 0 (1-1) 以下の最新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。

- 金曜日

レベル 2 の累積増分バックアップでは、レベル  $n-1$  以下の最新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 (2-1) 以下の最新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すべての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  $n-1$  以下の最新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 (2-1) 以下の最新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すべてのブロックのバックアップがとられます。

- このサイクルが次の週も繰り返されます。

## 増分バックアップ計画

バックアップのスキームは、許容可能な MTTR（平均リカバリ時間）に応じて選択します。たとえば、全体またはレベル 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方式を実装できます。この方式では、完全リカバリを実行する際に、1 日分の量を超える REDO を適用する必要はありません。

全体またはレベル 0 のバックアップをとる頻度の決定は、データの 50% 以上が変更された場合に必ずレベル 0 の新しいバックアップをとるようにするのがおおよその目安です。データの変更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ると、レベル 0 の新しいバックアップを行う時期を適切に判断できます。次の問合せを行うと、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を表示できます。

```
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、ベースとなるレベル 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のみを作成した場合、レベル 1 の最新のバックアップが、ベースとなるレベル 0 のバックアップのほぼ半分のサイズであるときに、レベル 0 の新しいバックアップを作成します。

**関連項目：** 9-20 ページ「[Recovery Manager を使用した増分バックアップの作成](#)」

## Recovery Manager がバックアップ可能なファイルのタイプ

シングル・インスタンス・データベースの場合、Recovery Manager 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またはオープンされている場合のみです。これは、バックアップ情報を記録するための制御ファイルが使用可能である必要があるためです。

BACKUP コマンドを実行すると、次のタイプのファイルをバックアップできます。

- すべてのデータ・ファイル、現行の制御ファイルおよび現行のサーバー・パラメータ・ファイルを含むデータベース
- 表領域（ローカル管理の一時表領域を除く）
- 現行のデータ・ファイル
- 現行の制御ファイル
- アーカイブ REDO ログ
- 現行のサーバー・パラメータ・ファイル
- バックアップ・セット

Recovery Manager でバックアップできないものは、次のとおりです。

- オンライン REDO ログ
- 読み込み / 書き込みが行われる前に移動された表領域
- クライアント側の初期化パラメータ・ファイルまたは非現行サーバー・パラメータ・ファイル

**関連項目：**

- Oracle Real Application Clusters のバックアップの制約の詳細は、『Oracle9i Real Application Clusters 管理』を参照してください。

## バックアップ・エラー

Recovery Manager では、主に 2 つのタイプのバックアップ・エラーを処理します。I/O エラーと破損ブロックです。ファイルの読み取り時またはバックアップ・ピースへの書き込み時に Recovery Manager 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終了します。たとえば、Recovery Manager ではバックアップ対象のデータ・ファイルがディスク上に存在しなかった場合、Recovery Manager によりバックアップが終了させられます。

Recovery Manager のジョブが、複数のバックアップ・セットを個別に作成中にエラーが発生した場合、Recovery Manager はエラーが発生した時点で書き込んでいたバックアップ・セットをリライトする必要があります。ただし、終了する前に正常に書き込まれていたバックアップ・セットは、維持されています。BACKUP コマンドの NOT BACKED UP SINCE オプションでは、部分的に完了しているバックアップが再起動され、バックアップが未済のファイルのみをバックアップします。

Recovery Manager は、破損しているとすで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。Recovery Manager は、破損していることをまだ識別されていないデータ・ファイル・ブロックを見つけた場合、メディア破損を示す情報をヘッダーに付加して、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。このレコードにアクセスするには、V\$BACKUP\_CORRUPTION および V\$COPY\_CORRUPTION ビューを使用します。

SET MAXCORRUPT コマンドを使用すると、指定したデータ・ファイル内で事前に検出されていないブロックの破損を一定数まで許可します。BACKUP 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、そのコマンドは終了します。デフォルトは 0（ゼロ）で、Recovery Manager が破損ブロックを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、5-72 ページの「[バックアップのテストおよび整合性チェック](#)」を参照してください。
- NOT BACKED UP SINCE 句の詳細は、5-68 ページの「[再起動可能なバックアップ](#)」を参照してください。
- V\$BACKUP\_CORRUPTION の説明は、『Oracle9i データベース・リファレンス』を参照してください。
- SET MAXCORRUPT コマンドの資料は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## 制御ファイルとサーバー・パラメータ・ファイルの自動バックアップ

CONFIGURE CONTROLFILE AUTOBACKUP を ON にすると（デフォルトでは OFF）、次の場合に制御ファイルと現行のサーバー・パラメータ・ファイル（使用している場合）が自動的にバックアップされます。

- Recovery Manager プロンプトで BACKUP または COPY コマンドが発行された後。
- RUN ブロック内に BACKUP または COPY コマンドがあり、後に続くコマンドが BACKUP でも COPY でもない場合
- ブロック内の最後のコマンドが BACKUP または COPY の場合の各 RUN ブロックの最後
- 新しい表領域の追加、表領域またはデータ・ファイルの状態の変更（オンライン化など）、新しいオンライン REDO ログの追加、ファイル名の変更、新しい REDO スレッドの追加など、データベース構造に変更があった後。前述の場合に発生する自動バックアップとは異なり、このタイプの自動バックアップはディスクにのみ作成されます。CONFIGURE CONTROLFILE AUTOBACKUP FOR DEVICE TYPE DISK を実行すると、デフォルト以外のディスク位置を設定できます。自動バックアップによって、それに関連する構造変更が失敗することはありません。たとえば、データ・ファイルを追加したために自動バックアップが失敗しても、データ・ファイルの追加は正常終了します。

---

**注意：** Oracle Enterprise Manager を使用する場合は、メンテナンス・ウィザードを使用して自動バックアップ機能を構成できます。

---

バックアップまたはコピー・ジョブで最初に割り当てられたチャンネルによって自動バックアップが作成され、専用バックアップ・セットに格納され、構造の事後自動バックアップ用にデフォルトのディスク・チャンネルによってバックアップが作成されます。制御ファイルとサーバー・パラメータ・ファイルは、同じバックアップ・ピースに書き込まれます。制御ファイルの自動バックアップが完了すると、バックアップ・ピースのフルパスとデバイス・タイプを含むメッセージが Oracle によってアラート・ログに書き込まれます。

次の表に示すように、BACKUP コマンドにデータ・ファイル 1 が含まれている場合の Recovery Manager の動作は、CONFIGURE CONTROLFILE AUTOBACKUP が ON であるか OFF であるかによって異なります。

CONTROLFILE AUTOBACKUP	BACKUP コマンドの動作
ON	<p>バックアップにデータ・ファイル 1 が含まれている場合、現行の制御ファイルがデータ・ファイルのバックアップ・セットに自動的に組み込まれることはありません。かわりに、Recovery Manager は制御ファイルとサーバー・パラメータ・ファイルを別の自動バックアップ・ピースに書き込みます。</p> <p><b>注意：</b>自動バックアップは、BACKUP DATABASE INCLUDE CURRENT CONTROLFILE など、BACKUP または COPY コマンドで現行の制御ファイルが明示的に指定されているかどうかに関係なく発生します。</p>
OFF	<p>バックアップにデータ・ファイル 1 が含まれている場合は、現行の制御ファイルとサーバー・パラメータ・ファイルがデータ・ファイルのバックアップ・セットに自動的に組み込まれます。制御ファイルとサーバー・パラメータ・ファイルを含む別個の自動バックアップ・ピースは作成されません。</p>

これらのコマンドの実行中に行われる現行の制御ファイルのバックアップの他にも、制御ファイルの自動バックアップが発生します。自動バックアップ機能を有効化または無効化するためには、次のコマンドを実行します。

```
CONFIGURE CONTROLFILE AUTOBACKUP ON;
CONFIGURE CONTROLFILE AUTOBACKUP OFF;
```

制御ファイルの自動バックアップの目的は、制御ファイルに含まれるバックアップ・リポジトリをリストアする手段を提供することです。これにより、制御ファイルが消失し、リカバリ・カタログが消失または使用されていない場合にも、リストアが可能になります。リカバリ・カタログまたはターゲット制御ファイルは、制御ファイルの自動バックアップをリストアするためには必要ありません。たとえば、次のように発行できます。

```
RESTORE CONTROLFILE FROM AUTOBACKUP;
```

制御ファイルのリストアおよびマウントを行った後に、これを使用してデータベースのリストアおよびリカバリに必要なバックアップ情報を取得できます。たとえば、リカバリ・カタログ・データベースとターゲット・データベースが破損した場合には、新規にカタログを作成してターゲット・データベースを登録できます。新しいカタログには、リストアされた制御ファイルからのバックアップ情報が移入されます。また、ターゲット・インスタンスに NOCATALOG モードで接続して、簡単にデータベースのリストアを行うこともできます。

制御ファイルの自動バックアップのファイル名には、すべてのデバイス・タイプに対して %F のデフォルトの書式があるため、**Recovery Manager** はこれをリストアするためにリポジトリを必要としません。置換変数 %F は、CONFIGURE コマンドの記述で定義されます。デバイス・タイプに異なる書式を指定するためには、CONFIGURE CONTROLFILE AUTOBACKUP FORMAT コマンドを使用します。有効な書式には、文字列の一部として %F 変数が必ず含まれます。

SET CONTROLFILE AUTOBACKUP FORMAT コマンドを、RUN ブロック内または **Recovery Manager** プロンプトで指定すると、構成済みの自動バックアップ書式をセッション専用の書式に変更できます。優先順位は次のとおりです。

1. RUN ブロック内の SET
2. Recovery Manager プロンプトの SET
3. CONFIGURE

CONFIGURE CONTROLFILE AUTOBACKUP の設定が OFF の場合にも、自動バックアップ書式を構成することが可能です。ただし、この場合には、**Recovery Manager** は自動バックアップを生成しません。**Recovery Manager** で自動バックアップを作成するには、CONFIGURE CONTROLFILE AUTOBACKUP を ON に設定する必要があります。

### 関連項目：

- 9-37 ページ [「制御ファイルの自動バックアップ書式の上書き」](#)
- BACKUP の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- RESTORE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## バックアップの保存方針

この項では、次の項目について説明します。

- バックアップの保存方針について
- リカバリ・ウィンドウ
- バックアップ冗長性
- 不要なバックアップとコピーのバッチ削除
- バックアップの保存方針からの除外

## バックアップの保存方針について

CONFIGURE RETENTION POLICY コマンドを使用して、永続および自動のバックアップの保存方針を作成します。バックアップ保存方針が有効な場合、Recovery Manager は、CONFIGURE コマンドで指定した基準に従い、データ・ファイルと制御ファイルの、バックアップおよびコピーを不要（つまり、メディア・リカバリの必要がなくなった）とみなします。このため、定期的に REPORT OBSOLETE コマンド発行して不要ファイルを確認し、このファイルを削除するために DELETE OBSOLETE コマンドを発行します。

保存方針は継続します。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が長時間経過したときに、Recovery Manager は、どのファイルが保持され、どれが不要かを追跡します。Recovery Manager はユーザー・ファイルの自動的な削除は行わずに、レコードに保持します。

不要という用語は、期限切れと同じ意味ではありません。バックアップあるいはコピーが不要となるのは、ユーザー定義の保存方針に基づいて REPORT OBSOLETE または DELETE OBSOLETE を行い、メディア・リカバリを必要としないと判断された場合です。バックアップまたはコピーが期限切れとなるのは、Recovery Manager がクロスチェックを実行したときに、ファイルがディスクから消失しているか、そのファイルが見つからないというメッセージがメディア・マネージャから返された場合のみです。つまり、「不要」とは「必要ではない」という意味であり、「期限切れ」とは「見つからない」という意味になります。

保存方針の視点に立つと、バックアップとは、個々のデータ・ファイルまたは制御ファイルのバックアップまたはコピーのことです。バックアップが、データ・ファイルのコピーであるか、プロキシ・コピーまたは他のバックアップ・セットの一部であるかは、問題ではありません。データ・ファイルのコピーおよびプロキシ・コピーは、Recovery Manager で必要ではないと判断された場合は削除されます。バックアップ・セット内のデータ・ファイルでは、バックアップ・セット内の個々のデータ・ファイルのバックアップがすべて不要とされるまで、Recovery Manager がバックアップ・セットを削除することはできません。

一方で、操作対象のデータ・ファイルおよび制御ファイルのバックアップでは、保存方針がアーカイブ・ログおよびアーカイブ・ログのバックアップに影響を及ぼします。最初に、**Recovery Manager** はどのデータ・ファイルおよび制御ファイルのバックアップを不要とするかを決定します。次に、**Recovery Manager** は保存する必要がある最も古いデータ・ファイルまたは制御ファイルのバックアップより古いすべてのアーカイブ・ログを不要と判断します。この動作は、保存方針がリカバリ・ウィンドウまたは冗長性に構成されているかどうかに関係なく発生します。

---

---

**注意：** **Recovery Manager** では、メディア・マネージャのテープ保存方針を介するなど、**Recovery Manager** 以外の方法でバックアップが削除された場合の自動保存方針が実装されていません。メディア・マネージャは、テープにあるすべての **Recovery Manager** のバックアップがメディア・マネージャのカタログから削除されるまでは、テープを期限切れにすることはありません。

---

---

相互排他的な 2 つのオプションが、保存方針を実装するためにあります。これには、**リカバリ・ウィンドウ** または **冗長性**（保存方針のデフォルトのタイプ）を指定してください。保存方針のコマンドを次に示します。

- `CONFIGURE RETENTION POLICY TO RECOVERY WINDOW`
- `CONFIGURE RETENTION POLICY TO REDUNDANCY`

保存方針を完全に無効化することもできます。これには、次のコマンドを実行します。これにより、**Recovery Manager** はすべてのバックアップまたはコピーを不要とみなしません。

```
CONFIGURE RETENTION POLICY TO NONE;
```

## リカバリ・ウィンドウ

リカバリ・ウィンドウとは、現時点から過去方向にさかのぼった**リカバリ可能ポイント**までの期間のことです。リカバリ可能ポイントは **Point-in-Time** リカバリが行えると仮定した場合に最も古い時点、つまりメディア障害後にリカバリ可能な最も古い時点を示します。たとえば、リカバリ・ウィンドウを 1 週間で実装した場合、バックアップをリストアして、この時点までリカバリするには、期間を現在からさかのぼってちょうど 7 日間にする必要があります。この保存方針の実装を次に示します。

```
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;
```

このコマンドにより、各データ・ファイルではリカバリ可能ポイントより古いバックアップを確実に保持する必要があります。たとえば、リカバリ・ウィンドウが 7 の場合、次の条件を満たす各データ・ファイルのバックアップが常に存在する必要があります。

```
SYSDATE - (SELECT CHECKPOINT_TIME FROM V$DATAFILE) >= 7
```

この条件を満たす最新のバックアップより前のバックアップはすべて不要です。



図 5-7 に示す保存方針に従うものと仮定します。保存方針には次の特徴があります。

- リカバリ・ウィンドウは 7 日間。
- データベースのバックアップのスケジュールは次の日付で 2 週間に 1 回。
  - 1 月 1 日
  - 1 月 14 日
  - 1 月 28 日
  - 2 月 4 日
- データベースを ARCHIVELOG モードで実行し、アーカイブ・ログは保存方針に従って、必要な期間ディスクのみに保存。

図 5-7 リカバリ・ウィンドウ：1

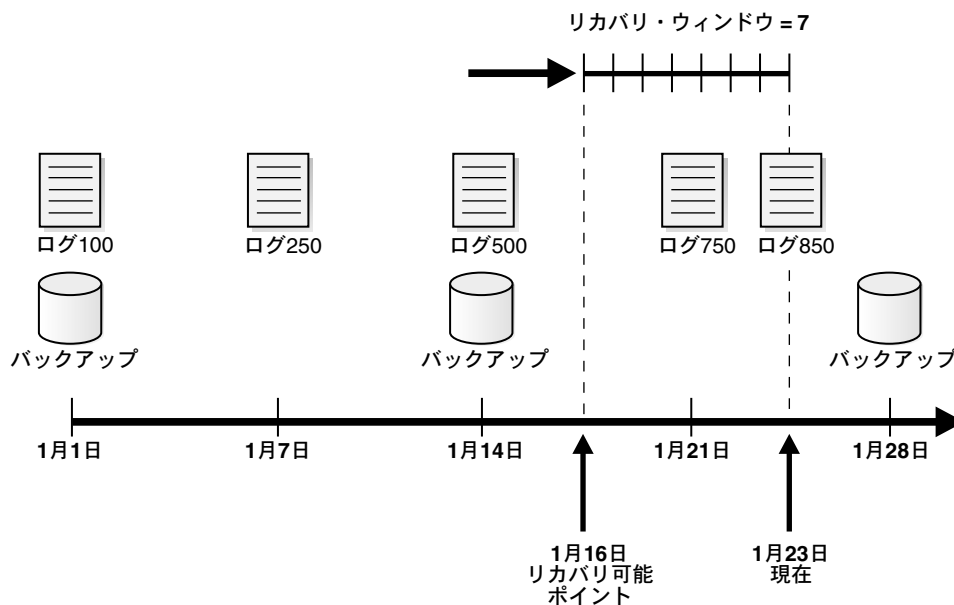
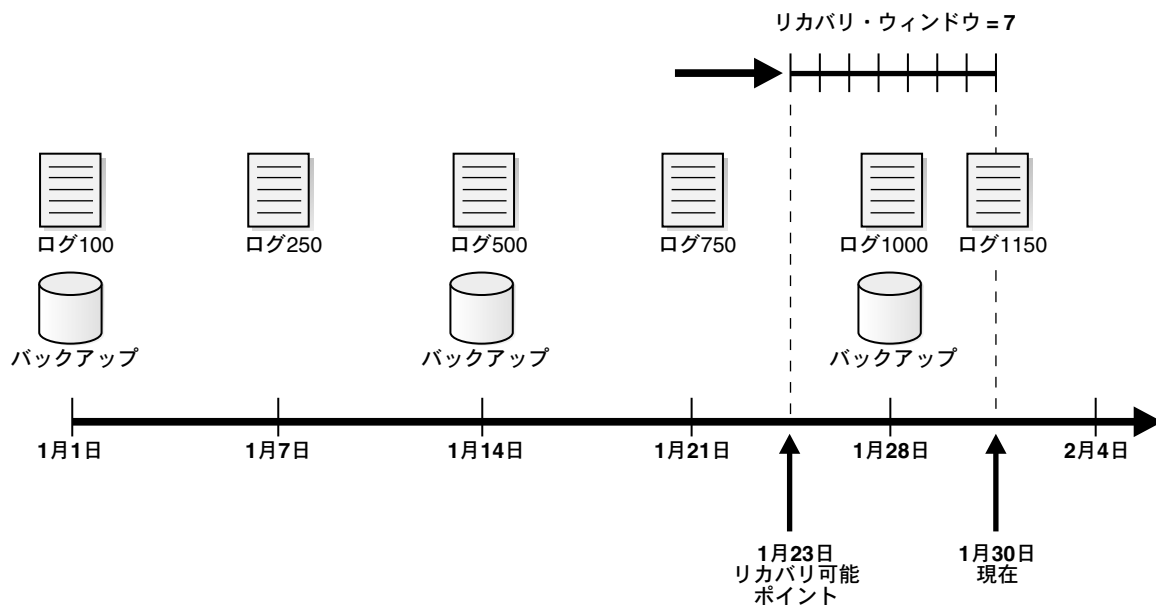


図 5-7 のように、現時点は 1 月 23 日で、リカバリ可能ポイントは 1 月 16 日になります。したがって、リカバリを行うには、1 月 14 日のバックアップおよびログ順序 500 から 850 までのアーカイブ・ログが必要になります。ログ順序 500 より前のログおよび 1 月 1 日のバックアップは、リカバリ・ウィンドウ内の時点のリカバリに必要なため不要となります。

図 5-8 に図示するように、同じ例の 1 週間後を想定します。

図 5-8 リカバリ・ウィンドウ：2



この例では、現時点は1月30日で、リカバリ可能ポイントは1月23日になります。リカバリ・ウィンドウ内により新しいバックアップ（1月28日）があるにもかかわらず、1月14日のバックアップが不要とならないことに注意してください。これは、1月28日のバックアップをリストアした場合、リカバリ・ウィンドウの最初の時点である1月23日のリカバリができないためです。リカバリ・ウィンドウ内のどの時点でも確実にリカバリ可能とするために、1月14日のバックアップに加えてログ順序500から1150までのすべてのアーカイブ REDO ログを保存する必要があります。

**関連項目：** 8-21 ページ「リカバリ・ウィンドウのための保存方針の構成」

## バックアップ冗長性

RECOVERY WINDOW 以外にも CONFIGURE コマンドで REDUNDANCY パラメータを使用することで、バックアップの保存期間を制御するメカニズムが得られます。REDUNDANCY パラメータで、保存する必要があるバックアップおよびコピーの最大数を指定します。たとえば、次のように指定できます。

```
CONFIGURE RETENTION POLICY TO REDUNDANCY 2;
```

リカバリ・ウィンドウは保存方針を指定する最適な方法ですが、リカバリ・ウィンドウにより保存されるバックアップの数は一定ではなく、バックアップ・スケジュールに依存するため、好ましい選択ではない場合があります。その場合は CONFIGURE RETENTION POLICY TO REDUNDANCY コマンドを使用して、各データ・ファイルのバックアップ数を一定に保つ保存方針を実装します。RECOVERY WINDOW オプションは REDUNDANCY オプションと相互排他的です。

デフォルトでは、保存方針が REDUNDANCY = 1 に構成されています。これは以前の Recovery Manager リリースの REPORT OBSOLETE と動作の互換性を保つためです。次のコマンドを実行すると、Recovery Manager は保存方針をデフォルト値に戻します。

```
CONFIGURE RETENTION POLICY CLEAR;
```

また、次のコマンドを実行して、保存方針を完全に無効化することもできます。

```
CONFIGURE RETENTION POLICY TO NONE;
```

保存方針を NONE に構成すると、REPORT OBSOLETE および DELETE OBSOLETE は、どのバックアップも不要と判断することはありません。

**関連項目：** 8-22 ページ [「冗長性のための保存方針の構成」](#)

## 不要なバックアップとコピーのバッチ削除

REPORT OBSOLETE コマンドを実行して、どのバックアップとコピーが保存方針に従って現在不要とされているか判断します。たとえば、CONFIGURE コマンドを使用して、リカバリ・ウィンドウを7日に設定した場合、次のコマンドを実行することでどのバックアップおよびアーカイブ REDO ログが不要とされているか判断します。

```
REPORT OBSOLETE;
```

保存方針を NONE に構成している場合、Recovery Manager はどのバックアップまたはコピーも不要と判断しません。そのため、他のオプションを指定せずに REPORT OBSOLETE を実行すると Recovery Manager はエラーを発行します。

これと対になるコマンドの DELETE OBSOLETE では、保存方針により不要とされたファイルが削除されます。このコマンドを実行すると、不要なバックアップとコピーを1度に削除できます。たとえば、DELETE OBSOLETE を毎週実行するスクリプトの中で実行します。

DELETE OBSOLETE コマンドを実行することで、不要なファイルを格納することによるディスク領域の浪費がなくなります。

さらに、REPORT または DELETE コマンドに、REDUNDANCY あるいは RECOVERY WINDOW オプションを指定できます。この設定は、コマンドの実行時に不要とするものを判断し、CONFIGURE RETENTION POLICY の設定を上書きします。

### 関連項目：

- 17-12 ページ「[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](#)」
- 18-5 ページ「[バックアップおよびコピーの削除](#)」
- DELETE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- REPORT コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## バックアップの保存方針からの除外

ユーザーの保存方針で指示された時間よりも、はるかに長期間、別サイトに**長期バックアップ**が必要になることがあります。たとえば、記録に残すのみの目的で毎年1月1日にデータベースのバックアップを作成することがあります。このバックアップを、バックアップおよびリカバリ計画に使用する予定はありませんが、レポートを生成するために、ある時点でのリストアが必要になる場合があります。たとえば、5年前のデータベースのユーザー数を調査したい場合などです。

長期バックアップを、本番バックアップ・スケジュールに含める必要性はないが、これをリカバリ・カタログに記録することが必要になる場合があります。長期バックアップは保存方針に含めないでください。**Recovery Manager** は、すぐにこれを不要とマーク付けします。その結果、**DELETE OBSOLETE** コマンドがこのバックアップを削除することになります。

**BACKUP** または **COPY** コマンドに **KEEP** オプション（あるいは **CHANGE** コマンド **KEEP** にオプション）を使用することで、バックアップを保存方針から除外します。ただし、そのバックアップは完全に有効なバックアップであるため他の **Recovery Manager** のバックアップと同様にリストア可能です。**NOKEEP** オプション（デフォルト）は、構成した保存方針の影響を受けることを表します。**CHANGE** コマンドを使用すると、バックアップの状態を **KEEP** と **NOKEEP** の間で切り替えることができます。

**LOGS** オプションを指定すると、長期バックアップの不完全リカバリのためのアーカイブ・ログを保存できます。**LOGS** を指定すると、バックアップより新しいすべてのログが、バックアップの保存と同じ期間保存されますつまり、**KEEP FOREVER LOGS** の意味は、長期バックアップ後に生成されたログを **Recovery Manager** は不要と判断しないということです。**NOLOGS** を指定すると、**Recovery Manager** はバックアップのリカバリに必要となるログを保存しません。この場合、その長期バックアップでは一貫性が保たれている必要があります。

バックアップが保存され続ける **FOREVER** を指定するか、**UNTIL** 句を使用して保存される期間を指定することができます。**UNTIL** を指定すると、**Recovery Manager** は **UNTIL** に指定した日を過ぎるまで、バックアップを不要としてマークしません。

次に示すコマンドは、長期バックアップの例です。

```
# exempts the backup from retention policy until last day of 2002
BACKUP DATABASE KEEP UNTIL TIME "TO_DATE('31-DEC-2002', 'dd-mon-yyyy')" NOLOGS;

# alters status of backup set 231 from KEEP to NOKEEP
CHANGE BACKUPSET 231 NOKEEP;

# specifies that this backup and the logs required to recover it are indefinitely exempt
# from the retention policy
BACKUP TABLESPACE users KEEP FOREVER NOLOGS;
```

**関連項目：** **CHANGE** コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

# バックアップの最適化

Recovery Manager のバックアップ最適化とは、バックアップ実行時に特定の条件と一致したファイルをスキップする機能を指します。

この項では、次の項目について説明します。

- バックアップの最適化のアルゴリズム
- バックアップの最適化を有効化および無効化するための要件
- 保存方針がバックアップの最適化に与える影響

## バックアップの最適化のアルゴリズム

バックアップの最適化を使用可能にすると、BACKUP コマンドでは、割り当てられたデバイス・タイプで同一ファイルがバックアップ済みの場合に、ファイルのバックアップをスキップします。表 5-4 で説明する基準を使用して、Recovery Manager はすでにバックアップを行ったファイルと同一のファイルかどうかを判断します。

表 5-3 同一ファイルの判断基準

ファイルのタイプ	同一ファイルの判断基準
データ・ファイル	同一の DBID、チェックポイント SCN および RESETLOGS の SCN と時間。データ・ファイルは、NORMAL モードでオフラインにされているか、読取り専用または正常にクローズされていることが必要。
アーカイブ REDO ログ	同一のスレッド、順序番号および RESETLOGS SCN と時間。
バックアップ・セット	同一のバックアップ・セット RECID およびスタンプ。

Recovery Manager は、ファイルが同一であり、そのファイルはすでにバックアップ済みであると判断すると、これをスキップの候補とします。ただし、Recovery Manager ではそのファイルをスキップするかどうかを判断するために、さらにチェックを行う必要があります。これは、保存方針とバックアップの二重化機能の両方が、指定されたデバイス・タイプに条件を満たすバックアップを Recovery Manager が持っているかどうかを判断するアルゴリズムに影響を及ぼすためです。

表 5-4 で説明するアルゴリズムは、同一ファイルをスキップする場合を判断するために、バックアップ最適化で使用されます。

表 5-4 バックアップの最適化のアルゴリズム

同一ファイル のタイプ	バックアップの最適化のアルゴリズム
データ・ ファイル	<p>リカバリ・ウィンドウが構成してある場合、Recovery Manager は、最後のファイルのバックアップがリカバリ・ウィンドウの最初の時点より前の場合のみデータ・ファイルをスキップします。</p> <p>リカバリ・ウィンドウを構成していない場合、Recovery Manager は <math>n=1</math> を設定し、次に示す優先順位（リストの番号の大きいものが、番号の小さいものより優先される）に従い <math>n</math> の値を検索します。</p> <ol style="list-style-type: none"> <li>1. CONFIGURE RETENTION POLICY TO REDUNDANCY <math>r</math> が有効な場合、Recovery Manager は <math>n=r+1</math> のバックアップが存在する場合のみデータ・ファイルをスキップ。</li> <li>2. BACKUP ... COPIES <math>n</math></li> <li>3. SET BACKUP COPIES <math>n</math></li> <li>4. CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE ... TO <math>n</math></li> </ol> <p>Recovery Manager は、指定されたデバイスに少なくとも <math>n</math> の同一ファイルのバックアップが存在した場合のみバックアップをスキップします。Recovery Manager がバックアップをスキップしない場合には、指定に従いバックアップを作成します。</p>
アーカイブ・ ログ	<p>デフォルトでは、<math>n=1</math> が設定されています。Recovery Manager は、次に示す優先順位（リストの番号の大きいものが、番号の小さいものより優先される）に従い <math>n</math> の値を検索します。</p> <ol style="list-style-type: none"> <li>1. BACKUP ... COPIES <math>n</math></li> <li>2. SET BACKUP COPIES <math>n</math></li> <li>3. CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE ... TO <math>n</math></li> </ol> <p>Recovery Manager は、指定されたデバイスに少なくとも <math>n</math> の同一ファイルのバックアップが存在した場合のみバックアップをスキップします。Recovery Manager がバックアップをスキップしない場合には、指定に従いバックアップを作成します。</p>
バックアップ・ セット	<p>デフォルトでは、<math>n=1</math> が設定されています。Recovery Manager は、次に示す優先順位（リストの番号の大きいものが、番号の小さいものより優先される）に従い <math>n</math> 以外の値を検索します。</p> <ol style="list-style-type: none"> <li>1. BACKUP ... COPIES <math>n</math></li> <li>2. SET BACKUP COPIES <math>n</math></li> </ol> <p>Recovery Manager は、指定されたデバイスに少なくとも <math>n</math> の同一ファイルのバックアップが存在した場合のみバックアップをスキップします。Recovery Manager がバックアップをスキップしない場合には、指定に従いバックアップを作成します。</p>

たとえば、午前 9 時に、既存の全アーカイブ・ログの 3 つのコピーをテープにバックアップした場合を想定します。

```
BACKUP DEVICE TYPE sbt COPIES 3 ARCHIVELOG ALL;
```

その後、次に示す構成の設定を使用可能にします。

```
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE sbt TO 4;  
CONFIGURE BACKUP OPTIMIZATION ON;
```

正午に、次に示すアーカイブ・ログのバックアップを実行します。

```
BACKUP DEVICE TYPE sbt COPIES 2 ARCHIVELOG ALL;
```

この場合、BACKUP ... COPIES の設定は、CONFIGURE ... COPIES の設定をオーバーライドします。このため、**Recovery Manager** は  $n=2$  を設定します。**Recovery Manager** は、少なくとも 2 つのログのコピーが sbt デバイスに存在する場合のみログのバックアップをスキップします。各ログの 3 つのコピーが、午前 9 時以前に生成されたすべてのログがある sbt に存在するため、**Recovery Manager** はそれらのログをスキップします。ただし、**Recovery Manager** は午前 9 時以降に生成されたすべてのログの 2 つのコピーをバックアップします。これらのログは、まだテープにバックアップされていないためです。

この時点で、午前 9 時以前に作成したログの 3 つのコピーがテープに存在し、午前 9 時以降に作成したログの 2 つのコピーもテープに存在しています。午後 3 時に、次のバックアップを実行することを想定します。

```
RUN  
{  
    SET BACKUP COPIES 3;  
    BACKUP DEVICE TYPE sbt ARCHIVELOG ALL;  
}
```

この場合、**Recovery Manager** は  $n=3$  を設定するため、午前 9 時以前に作成したログのバックアップは行われません。3 つのコピーはすでにテープに存在しているためです。ただし、午前 9 時以降に作成したログの 2 つのコピーのみがテープに存在するため、これらのファイルのバックアップに **Recovery Manager** は最適化を適用しません。したがって、**Recovery Manager** は午前 9 時以降に作成したログの 3 つのコピーをバックアップします。



## バックアップの最適化を有効化および無効化するための要件

バックアップの最適化は、次に示す条件が真の場合に有効化されます。

- CONFIGURE BACKUP OPTIMIZATION ON コマンドを実行している。
- BACKUP DATABASE, BACKUP ARCHIVELOG を、ALL または LIKE オプションで実行している。あるいは BACKUP BACKUPSET ALL を実行している。
- 1 つのタイプのチャネルのみが割り当てられている。つまり、DISK と sbt タイプのチャネルを混在させていない。

たとえば、次のコマンドを実行すると仮定します。

```
BACKUP DEVICE TYPE sbt DATABASE PLUS ARCHIVELOG;  
BACKUP DEVICE TYPE sbt BACKUPSET ALL;
```

最後のバックアップからどのファイルも変更されていない場合、Recovery Manager はこれらのファイルに対して再度バックアップを行いません。また、コマンドで指定したすべてのファイルをスキップした場合にも、エラーの通知は行われません。

---

---

**注意：** バックアップの最適化が有効かどうかに関係なく、アーカイブ・ログが存在しない場合に BACKUP ARCHIVELOG を実行しても、Recovery Manager はエラーの通知を行いません。前回実行された BACKUP ARCHIVELOG ALL DELETE INPUT コマンド以降、ログが新規に生成されていないことが考えられます。

---

---

バックアップの最適化を更新して、すべてのファイルを変更されたかどうかに関係なくバックアップするには、BACKUP コマンドに FORCE オプションを指定します。バックアップの最適化を無効化するには、CONFIGURE BACKUP OPTIMIZATION コマンドに OFF オプションを指定します。CLEAR オプションにより、バックアップの最適化をデフォルト値の OFF に戻します。

## 保存方針がバックアップの最適化に与える影響

保存方針は、バックアップの最適化に影響を与えます。保存方針では REDUNDANCY=1 をデフォルトとしているため、CONFIGURE RETENTION POLICY TO NONE による明示的な無効化が行われていない場合には、いつでも保存方針を使用できます。

---

---

**注意：** バックアップの最適化を有効化する際、使用しているメディア・マネージャに期限切れ方針がある場合には注意が必要です。メディア・マネージャでは、バックアップの含まれるテープを期限切れにすることがあります。この場合、最適化の要因から、**Recovery Manager** は新しいバックアップを作成できません。CROSSCHECK を定期的に行って、リポジトリとメディア・マネージャの再同期化を行ってください。

---

---

## バックアップの最適化とリカバリ・ウィンドウ

最適化が有効であり、リカバリ・ウィンドウの保存方針が構成されている場合、**Recovery Manager** では、最後にバックアップしたのがリカバリ・ウィンドウよりも前の、古いデータ・ファイルは、必ずバックアップされます。たとえば、次のように仮定します。

- 今日 は 2 月 21 日。
- リカバリ・ウィンドウは 7 日間。
- 表領域 tools のテープへの最新のバックアップは 1 月 3 日。
- 表領域 tools は読取り専用。

2 月 21 日、表領域 tools をテープにバックアップするコマンドを発行すると、**Recovery Manager** はこれをバックアップします。この表領域に 1 月 3 日以降の変更がない場合にもバックアップは行われます（読取り専用のため）。**Recovery Manager** がバックアップを作成するため、表領域のバックアップは 7 日間のリカバリ・ウィンドウ内に存在します。

この動作により、メディア・マネージャは古いテープを期限切れとすることができます。それ以外の場合、メディア・マネージャは、表領域 tools の 1 月 3 日のバックアップを、無期限に保存することを強制されます。2 月 21 日に表領域 tools の新しいバックアップを作成することで、**Recovery Manager** は不要な 1 月 3 日のバックアップを含むテープをメディア・マネージャが期限切れにすることを許可します。

バックアップの最適化と冗長性

冗長性の保存方針を構成することを想定します。この場合、 $r+1$  ( $r$ は CONFIGURE RETENTION POLICY TO REDUNDANCY  $r$  で設定) のバックアップが存在すると、Recovery Manager はオフラインまたは読取り専用のデータ・ファイルのバックアップのみをスキップします。

バックアップの最適化を有効にして、次に示す保存方針を設定すると仮定します。

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;  
CONFIGURE BACKUP OPTIMIZATION ON;  
CONFIGURE RETENTION POLICY TO REDUNDANCY 2;
```

これにより、Recovery Manager は同一のファイルがすでに 3 つバックアップされている場合にのみ、バックアップをスキップします。さらに、読込み / 書込み用の users 表領域がバックアップされたことがない場合に、表 5-5 で説明する約 1 週間の手順を処理したと仮定します。

表 5-5 冗長性セットがバックアップの最適化に与える影響

曜日	処理	結果	冗長バックアップ
月曜日	表領域 users を NORMAL モードでオフラインにする。		
火曜日	BACKUP DATABASE	users 表領域がバックアップされる。	
水曜日	BACKUP DATABASE	users 表領域がバックアップされる。	
木曜日	BACKUP DATABASE	users 表領域がバックアップされる。	火曜日のバックアップ
金曜日	BACKUP DATABASE	users 表領域はバックアップされない。	火曜日のバックアップ
土曜日	BACKUP DATABASE	users 表領域はバックアップされない。	火曜日のバックアップ
日曜日	DELETE OBSOLETE	火曜日のバックアップが削除される。	
月曜日	BACKUP DATABASE	users 表領域がバックアップされる。	水曜日のバックアップ

オフラインの `users` 表領域をバックアップした火曜日、水曜日および木曜日のバックアップにより、3つのバックアップが存在する（冗長性セットより1大きい）という条件が満たされます。金曜日および土曜日のバックアップは、バックアップの最適化のために `users` 表領域をバックアップしません。火曜日の `users` のバックアップは、木曜日の始まりに不要になることに注意してください。

日曜日、`users` の火曜日のバックアップを削除した不要なバックアップをすべて消去します。火曜日のバックアップは、保存方針の設定により不要となります。月曜日のデータベース全体のバックアップに加えて `users` 表領域をバックアップすることで、3つのバックアップが存在する（冗長性セットより1大きい）という条件が満たされます。この方法で、テープを繰り返しリサイクルさせます。

**関連項目：** 9-31 ページの「[バックアップの最適化を使用したファイルのバックアップ](#)」および 8-24 ページの「[バックアップの最適化の構成](#)」を参照してください。

## 再起動可能なバックアップ

再起動可能なバックアップ機能を使用すると、Recovery Manager は指定日以降にバックアップされていないファイルのみをバックアップすることができます。これは、途中まで進行了バックアップが失敗した場合に、データベースのバックアップが終了していない部分のみのバックアップに適した機能です。

再起動可能な単位は、1 バックアップ・セットです。データベース全体を1つのバックアップ・セットにバックアップしていて、そのバックアップが失敗した場合には、バックアップ作業全体を再実行する必要があります。バックアップを複数のバックアップ・セットに生成した場合、正常終了しているバックアップについては再実行の必要がありません。このため、`FILESERSET` にデフォルトよりも十分小さな値を設定します。これにより、Recovery Manager は各バックアップ・セットに含めるファイル数を制限します。

たとえば、毎日のデータベースのバックアップを次のように実行します。

```
BACKUP DATABASE FILESPERSET = 8;
```

データベース内で、この 24 時間以内にバックアップしていないすべてのファイルをバックアップするために、次のコマンドを発行します。

```
BACKUP DATABASE NOT BACKED UP SINCE TIME 'SYSDATE-1';
```

`SINCE TIME` パラメータを設定していないと、Recovery Manager はこれまで1度もバックアップしていないファイルのみをバックアップします。

ファイルがバックアップ済みかどうかを判断するときに、Recovery Manager は `SINCE TIME` に指定した日付と最新のファイルのバックアップの完了時刻を比較します。バックアップ・セット内のファイルの完了時刻は、バックアップ・セット全体の完了時刻です。つまり、同じバックアップ・セット内のすべてのファイルは、同一の完了時刻になります。`SINCE TIME` が、完了時刻より後だった場合に、Recovery Manager はそのファイルをバックアップします。

**関連項目：** BACKUP の構文の詳細は、9-32 ページの「部分的に完了したバックアップの再起動」および『Oracle9i Recovery Manager リファレンス』を参照してください。

## イメージ・コピー

イメージ・コピーには、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 1 つ含まれ、そのままの状態ではリカバリの実行に使用できません。イメージ・コピーを作成するには、Recovery Manager の COPY コマンドを使用するか、UNIX の cp コマンドなどのオペレーティング・システムのコマンドを使用します。

Recovery Manager 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いては、単一ファイルのオペレーティング・システム・コピーと同じです。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記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ため、バックアップ・セットとは異なります。Recovery Manager は、ディスクにのみイメージ・コピーの書き込みを行います。

この項では、次の項目について説明します。

- [Recovery Manager のイメージ・コピー](#)
- [ユーザー管理のイメージ・コピー](#)
- [バックアップおよびイメージ・コピーのタグ](#)
- [長期間の Recovery Manager コピー](#)

## Recovery Manager のイメージ・コピー

Recovery Manager の COPY コマンドを使用して、ディスクにイメージ・コピーを作成します。COPY を使用して DISK 以外のデバイスに、コピーを作成することはできません。RESTORE コマンドを実行すると、デフォルトの状態では Recovery Manager はデータ・ファイルあるいは制御ファイルのイメージ・コピーを元の場所にリストアします。

現行のデータ・ファイルを置換するときにデータ・ファイルのイメージ・コピーがある場合は、そのコピーをリストアする必要はありません。かわりに、SWITCH コマンドを使用して制御ファイルがコピーを指すようにして、コピーが切り替えられたことを指示するためにリポジトリを更新します。この場合、SWITCH コマンドを発行することと、SQL 文 ALTER DATABASE RENAME DATAFILE を発行することは同じです。その後、そのコピーを使用したリカバリが実行できます。

Recovery Manager は既存のイメージ・コピーを調査し、そのメタデータを制御ファイルおよびリカバリ・カタログに入力できます。この操作は、リカバリ・カタログが失われ、障害時リカバリを実行する必要がある場合に、重要な役割を果たします。カタログに登録できるのは、イメージ・コピーとアーカイブ・ログのみです。

## ユーザー管理のイメージ・コピー

Oracle では、Recovery Manager 以外のメカニズムで作成したイメージ・コピーをサポートします。これは、**ユーザー管理のコピー**または**オペレーティング・システム・コピー**と呼ばれています。たとえば、UNIX の dd コマンドで作成したデータ・ファイルのコピーは、オペレーティング・システム・コピーです。Recovery Manager を使用して、オペレーティング・システム・コピーをカタログに追加した後、このコピーを RESTORE または SWITCH コマンドで使します。

オペレーティング・システム・コピーは、データベースのオープン時またはクローズ時に作成できます。データベースがオープンしていてデータ・ファイルがオンラインで読み込み / 書き込みモードの場合は、表領域を**バックアップ・モード**にする必要があります。つまり、SQL 文 ALTER TABLESPACE BEGIN BACKUP を発行してコピーを作成します。

---

**注意：** オンライン・ユーザー管理バックアップを作成する前に表領域をバックアップ・モードにしていない場合、バックアップには分裂ブロックが含まれることがあり、リカバリ不可能になります。5-74 ページの「[論理的なブロック破損の検出](#)」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、**ミラー化の解除**によってイメージ・コピーの作成が可能になります。ミラー化を解除した後、Recovery Manager に新しいオペレーティング・システム・コピーの存在を通知すると、このコピーをリストア操作に使用できるようになります。そのコピーがリストアに使用できなくなった場合は、CHANGE ... UNCATALOG コマンドを使用して、Recovery Manager に通知する必要があります。この例では、**ミラーを復元**（解除されたミラーの他のコピーは含めない）した場合、CHANGE ... UNCATALOG コマンドを使用してリカバリ・カタログを更新し、このコピーが使用可能でなくなったことを通知する必要があります。

### 関連項目：

- コピーをカタログに追加する方法は、18-18 ページの「[アーカイブ・ログおよびユーザー管理コピーのカタログ化](#)」を参照してください。
- ミラーの分割によるバックアップの詳細は、9-21 ページの「[Recovery Manager を使用したミラーの分割によるバックアップの作成](#)」を参照してください。
- CHANGE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』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（Recovery Manager あるいはオペレーティング・システムのユーティリティで作成したコピー）には、**タグ**と呼ばれるユーザー指定の文字列を割り当てることができます。タグとは、`weekly_backup` というような、バックアップ・セットまたはファイル・コピーのシンボル名で、大 / 小文字区別はありません。ファイル名ではなくタグを指定して、RESTORE コマンドまたは CHANGE コマンドを実行できます。タグの最大長は 30 文字です。

タグ名を指定しない場合、Recovery Manager ではデフォルトでバックアップとコピー（制御ファイルの自動バックアップ以外）用のタグが書式 `TAGYYYYMMDDTHHMMSS` を使用して作成されます。YYYY は年、MM は月、DD は日、HH は時間（24 時間形式）、MM は分、SS は秒です。たとえば、データ・ファイル 1 のバックアップのタグは、`TAG20020208T133437` などとなります。日時は、Recovery Manager によってバックアップが開始された時点を示します。1 つの BACKUP コマンドで複数のバックアップ・セットが作成される場合、各バックアップ・ピースには同じデフォルト・タグが割り当てられます。

バックアップ・セットに適用される場合、タグはバックアップ・セットの特定のコピーに適用されます。バックアップ・セットを二重化しない場合、つまりその同一コピーを複数作成する場合は、タグとバックアップ・セットの間に 1 対 1 の対応関係が存在します。たとえば、`BACKUP COPIES 1 DATAFILE 7 TAG foo` を実行すると、タグ `foo` を持つ 1 つのバックアップ・セットが作成されます。ただし、このバックアップ・セットをバックアップし、新規コピーにタグ `bar` を与えることができます。その結果、バックアップ・セットには、一方にはタグ `foo`、他方にはタグ `bar` が付いた 2 つの同一コピーができます。

タグは一意にする必要はありません。そのため、複数のバックアップ・セットまたはイメージ・コピーに、`weekly_backup` などの同じタグ名が付けられます。あるタグが一意でない場合、特定のデータ・ファイルについては、そのタグは、最も適正な現行のファイルを指します。デフォルトの状態では、Recovery Manager は、タグまたは `SET UNTIL` コマンド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リストアします。Point-in-Time リカバリの場合、指定のファイルを含む最も適正な現行のバックアップが、最新のバックアップではないことがあります。

たとえば、毎週月曜日の夜に作成したデータ・ファイルのコピーには、常に `mondaypmcopy` というタグを付けていた場合、そのタグは最新のコピーを指しています（リストアおよびリカバリの前に、`SET UNTIL` コマンドが発行されないと仮定）。そのため、完全リカバリが必要な場合、次のコマンドで `datafile 3` を、月曜日の夜の最新のコピーに切り替えます。

```
SWITCH DATAFILE 3 TO DATAFILECOPY TAG mondaypmcopy;
```

タグによって、バックアップまたはファイル・コピーの各種クラスの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・ファイルのコピーには、RESTORE でのみ使用されるファイル・コピーとは異なるタグを付けることができます。

---

**注意：** RESTORE または SWITCH コマンドの入力ファイルの指定にタグを指定すると、**Recovery Manager** は使用するバックアップ・セットまたはイメージ・コピーを選択する際に、一致しているタグが付いたバックアップ・セットのみをその候補とみなします。

---

**関連項目：** SWITCH コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。RESTORE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## 長期間の Recovery Manager コピー

COPY ... KEEP コマンドを使用して、イメージ・コピーが現行の保存方針から除外されるように指定します。このコマンドの動作は、BACKUP ... KEEP の動作と同じです。

**関連項目：** 保存方針の説明とファイルの除外および非除外の方法は、5-61 ページの「[バックアップの保存方針からの除外](#)」を参照してください。

## バックアップのテストおよび整合性チェック

この項では、次の項目について説明します。

- [テストおよび整合性チェック](#)
- [物理的なブロック破損の検出](#)
- [論理的なブロック破損の検出](#)
- [オープン・バックアップ時の分裂ブロックの検出](#)
- [Recovery Manager を使用したバックアップのテスト](#)



## テストおよび整合性チェック

Oracle では、バックアップが使用不可になったり、リストア済のデータ・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。整合性チェックを使用することで、Oracle サーバーは次に示す内容を自動的に実行します。

- データベースの古いインカネーションからバックアップを適用し、リストア操作でデータベースが破損しないようにします。
- 増分バックアップが正しい順序で適用されるようにします。
- リストアまたはリカバリ中のデータ・ファイルへのアクセスを禁止します。
- 各データ・ファイルについて 1 度に 1 つのリストア操作のみ許可します。
- 特別なエラー・チェック情報をバックアップに格納し、破損したバックアップ・ファイルを検出します。

BACKUP VALIDATE コマンドを使用して、実際には出力ファイルを作成しないで、バックアップのテスト実行を行います。これにより、データベースの問題発生の可能性をチェックできます。

## 物理的なブロック破損の検出

バックアップ操作およびコピー操作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・セッションは多くのタイプの物理的な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。デフォルトの状態、物理的な破損のエラー・チェックが可能です。

Recovery Manager は、バックアップ操作の完了時に破損情報を問い合わせ、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 ビューおよび V\$COPY\_CORRUPTION ビューを使用します。

サーバー・セッションがバックアップ中に、データベースで破損と識別されているデータ・ファイル・ブロックを見つけると、そのサーバー・セッションは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としてこの破損を制御ファイルに記録します。Recovery Manager は、そのブロックをコピーして、ブロックの内容の修正が必要になった場合に備えます。

破損ヘッダーが付いているが、まだ破損のマークが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 Recovery Manager が見つけた場合は、メディア破損があることを示すヘッダーを付け直して、そのブロックをバックアップに書き込みます。

---

**注意：** Recovery Manager は、すべてのタイプの破損ブロックを検出できるわけではありません。

---

## 論理的なブロック破損の検出

Recovery Manager は物理的な破損チェックを通過したデータ・ブロックと索引ブロックについて、論理的な破損がないことをテストします。たとえば、行ピースまたは索引エントリの破損がないか調べます。Recovery Manager は論理的な破損を発見すると、そのブロックを alert.log とサーバー・セッションのトレース・ファイルに記録します。デフォルトの状態では、論理的な破損のエラー・チェックは使用禁止です。

1 つのファイルに対する物理的および論理的な破損の合計数が MAXCORRUPT の設定値以下の場合、Recovery Manager 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移入します。MAXCORRUPT を超過した場合、このコマンドはビューへの移入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 パラメータの設定は、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。

---

---

**関連項目：** BACKUP ... MAXCORRUPT の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。COPY ... MAXCORRUPT の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## オープン・バックアップ時の分裂ブロックの検出

オンライン・ブロック作成の危険性の 1 つとして、ブロック内でデータに一貫性がない可能性があります。たとえば、100 ブロックあるデータ・ファイル users.dbf をバックアップしていると仮定します。さらに、コピー・ユーティリティが全ブロックを読み込んでいる間に、データベース・ライターがそのブロックの中間を更新しているとします。この場合、コピー・ユーティリティはブロックの前半分で古いデータを、後半分で新しいデータを読み取ることになります。この場合、ブロックは、**分裂ブロック**になります。これは、このブロック内に含まれるデータには一貫性がないということです。

Recovery Manager を使用しないでオープン・バックアップを実行するときには、表領域を **バックアップ・モード**にする必要があります。これは、データベース・ライターによって書込みが行われているバックアップのブロックを、オペレーティング・システムが読み込めるようにするためです。バックアップ・モードでない場合、Oracle では、ブロック全体ではなく変更された REDO ストリーム内のバイトのみを記録します。表領域がバックアップ・モードの場合、Oracle では、表領域にある変更された各ブロックの前イメージを、変更後の REDO ログに書き込みます。次に、Oracle は REDO ログ内のブロックに変更部分の記録も行います。SQL\*Plus を使用してリカバリを行う場合、Oracle では、ブロックおよび変更部分をリカバリ時に適用します。これにより、バックアップ内のブロックの分裂が問題となりません。

Recovery Manager によるバックアップの実行中は、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。サーバー・セッションは Oracle ブロック全体を読み取り、各ブロックのヘッダーおよびフッターを比較して、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、データの全体像が一貫するまで、そのブロックは再度読み込まれます。これらの理由から、Recovery Manager では、表領域をバックアップ・モードにすることを要求しません。Recovery Manager を起動する前に表領域をバックアップ・モードにすることは、必要なことでも正しいことでもありません。

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

## Recovery Manager を使用したバックアップのテスト

BACKUP ... VALIDATE コマンドを実行することで、次のことを行います。

- 物理破損および論理破損のチェック
- データベース・ファイルが正しい場所に存在していることの確認

Recovery Manager は実際にはバックアップ・セットを作成するのではなく、指定されたファイルをスキャンしてバックアップが可能かどうかを判断します。この意味から、BACKUP VALIDATE コマンドは、リストア・ジョブ用ではなくバックアップ用である点を除いて、RESTORE VALIDATE コマンドと同じです。

たとえば、すべてのデータベース・ファイルおよびアーカイブ REDO ログのバックアップが可能かどうかを検査するために、次のようにコマンドを発行します。

```
BACKUP VALIDATE DATABASE ARCHIVELOG ALL;
```

VALIDATE オプションとともに MAXCORRUPT または PROXY パラメータを使用することはできません。

**関連項目：** BACKUP の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。



---

## 概念 II: リストア、リカバリおよび複製

この章では、**Recovery Manager** を使用したデータベースのリストア、リカバリおよび複製に関する基本概念を説明します。

この章の項目は、次のとおりです。

- **Recovery Manager** を使用したファイルのリストア
- **Recovery Manager** を使用したデータ・ファイルのメディア・リカバリ
- **Recovery Manager** を使用したブロック・メディア・リカバリ
- **Recovery Manager** を使用したデータベースの複製
- **Recovery Manager** を使用したスタンバイ・データベースの作成

## Recovery Manager を使用したファイルのリストア

この項では、次の項目について説明します。

- [RESTORE コマンド](#)
- [リストア操作の仕組み](#)
- [リストア処理時のファイル選択](#)
- [リストアの最適化](#)
- [Real Application Clusters](#) でリストアする場合の自動的なバックアップ位置検索

### RESTORE コマンド

Recovery Manager の RESTORE コマンドを使用して、ディスク上のイメージ・コピーやディスクまたは他のメディア上のバックアップから、次のタイプのファイルをリストアできます。

- データベース（全データ・ファイル）
- 表領域
- 制御ファイル
- アーカイブ REDO ログ
- サーバー・パラメータ・ファイル

バックアップ・セットは Oracle 特有の形式で書かれているため、オペレーティング・システム・ユーティリティで作成されたバックアップ・データベース・ファイルのように単にコピーすることはできません。Recovery Manager 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、Recovery Manager の COPY コマンドで作成したイメージ・コピーは、追加処理をしなくても Oracle で使用できます。

---

**注意：** 通常、アーカイブ・ログに対してリストアを行うことはありません。これは、リカバリ中に Recovery Manager が、必要に応じてこの処理を自動的に実行するためです。ただし、リカバリに必要なアーカイブ REDO ログのバックアップ・セットを手動でリストアしておくと、リカバリのパフォーマンスを改善できます。

---

**関連項目：** リストア方法は、10-2 ページの「[Recovery Manager を使用したリストアおよびリカバリ：概要](#)」を参照してください。RESTORE の構文および前提条件は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## リストア操作の仕組み

Recovery Manager では、ファイル・リストアの手順が自動化されています。オペレーティング・システム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必要はありません。RESTORE コマンドを発行すると、Recovery Manager 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リストアします。リストア先は次のどれかになります。

- デフォルトの場所。存在する同じ名前のファイルを上書き。
- 新規の場所。SET NEWNAME コマンドで指定。

データ・ファイルをリストアするには、データベースをマウントするか、オープン状態のままリストア対象のデータ・ファイルをオフラインにします。Recovery Manager がリストアを実行すると、RESTORE コマンドによりリストアされたファイルがデータ・ファイルのコピーとして作成され、リポジトリに記録されます。次の表に、RESTORE、SET NEWNAME および SWITCH コマンドの動作を示します。

SET NEWNAME を実行するかどうか	RESTORE の動作	SWITCH を実行するかどうか
実行しない	ファイルは現行のパス名にリストアされ、リストア中に作成されたデータ・ファイルのコピーに関するリポジトリ・レコードは即時に削除されます。	N/A
実行する	ファイルは SET NEWNAME で指定したパス名にリストアされ、リストア中に作成されたデータ・ファイルのコピーに関するリポジトリ・レコードは削除されません。	実行すると、制御ファイル内のデフォルトの名前がリストアされたファイルの名前に更新されます。実行しない場合、制御ファイル内のファイル名は更新されず、リストアされたファイルはデータ・ファイルのコピーとなります。

たとえば、データ・ファイル ?/oradata/trgt/tools01.dbf をデフォルトの場所にリストアする場合、Recovery Manager はファイル ?/oradata/trgt/tools01.dbf をリストアし、同じファイル名で検出したファイルをすべて上書きします。ファイルをリストアする前に、SET NEWNAME コマンドを実行する場合、Recovery Manager は指定した名前でデータ・ファイルのコピーを作成します。たとえば、次のコマンドを実行します。

```
SET NEWNAME FOR DATAFILE '?/oradata/trgt/tools01.dbf' TO '/tmp/tools01.dbf';
RESTORE DATAFILE '?/oradata/trgt/tools01.dbf';
```

この場合、Recovery Manager では、`?/oradata/trgt/tools01.dbf` のデータ・ファイルのコピーが `/tmp/tools01.dbf` という名前で作成され、リポジトリに記録されます。データ・ファイル `?/oradata/trgt/tools01.dbf` の名前を制御ファイル内で `/tmp/tools01.dbf` に変更するには、Recovery Manager でリストアされたファイルが現在のデータベース・ファイルとみなされるように、`SWITCH` コマンドを実行します。たとえば、次のように入力します。

```
SWITCH DATAFILE '/tmp/tools01.dbf' TO DATAFILECOPY '?/oradata/trgt/tools01.dbf';
```

`SWITCH` コマンドは、SQL 文の `ALTER DATABASE RENAME FILE` と同じです。

**関連項目：** `SET NEWNAME` の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。`SWITCH` の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## リストア処理時のファイル選択

Recovery Manager はリポジトリを使用して、使用可能な最善のバックアップ・セットまたはイメージ・コピーをリストア操作用に選択します。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、Recovery Manager 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、`RESTORE` コマンドの `UNTIL` 句が指定されているかどうか、選択基準の中に含まれます。

Recovery Manager がバックアップ・セットまたはファイル・コピーをリストアする前に、`RESTORE` コマンドのすべての設定を満たしておく必要があります。`DEVICE TYPE` 句で制限しないかぎり、`RESTORE` コマンドはバックアップを検索し、構成されたチャネルのすべてのデバイス・タイプをコピーします。

リポジトリ内に、指定したすべての基準を満たす使用可能なバックアップまたはコピーがない場合、Recovery Manager はリストア・ジョブのコンパイル・フェーズで、エラーを戻します。手動でチャネルを割り当て、ジョブで割り当てられたデバイス・タイプの中にバックアップ・セットまたはデータ・ファイルのコピーがないためにファイルをリストアできない場合は、新しいジョブを作成し、既存のバックアップ・セットまたはコピーを含むデバイスのチャネルを指定してください。自動チャネルを構成している場合、この問題は発生しません。

**関連項目：** 自動チャネルの構成方法は、8-11 ページの「[自動チャネルの構成](#)」を参照してください。



## リストアの最適化

デフォルトでは、リストアされるファイルが正しい場所にあり、ファイルのヘッダーに予想どおりの情報が入っている場合、Recovery Manager はリストアを実行しません。Oracle9i 以前のリリースでは、Recovery Manager は常に、要求されたファイルをリストアしていました。Oracle9i では、Recovery Manager はヘッダーのチェックが成功しなかった場合のみ、ファイルをリストアしますが、RESTORE コマンドの FORCE オプションを使用すると、要求されたファイルの無条件のリストアが可能になります。

---

**注意：** リストアの最適化は、データ・ファイルのヘッダーのみチェックします。データ・ファイル本体で破損ブロックをスキャンすることはありません。

---

リストアの最適化は、リストアが部分的にのみ完了している場合に、特に便利です。たとえば、全データベースのリストア中に、1つのデータ・ファイル以外はすべてリストアされた状態で停電が発生したと仮定します。同じリストアを再び開始すると、Recovery Manager は、前回のリストア中にリストアされなかった1つのデータ・ファイルのみリストアします。

## Real Application Clusters でリストアする場合の自動的なバックアップ位置検索

Recovery Manager では、リストア対象となるバックアップ、制御ファイルのコピーまたはデータ・ファイルのコピーが、Oracle Real Application Clusters 構成のどのノードに含まれているかが自動的に検出されます。クラスタの各ノードに接続するチャンネルを構成するか手動で割り当てていれば、Recovery Manager はすべてのチャンネルでファイルを検索し、テープ上またはローカル・ファイル・システム上でバックアップまたはコピーがあるチャンネルからのみファイルをリストアします。

たとえば、3 ノード・クラスタを構成するとします。ノード 1 に接続しているチャンネルがアーカイブ・ログをローカル・テープ・ドライブにバックアップすると、Recovery Manager は、このログのリストアにはノード 2 またはノード 3 に接続しているチャンネルの使用を試みません。

**関連項目：** Real Application Clusters 構成における RESTORE の動作の詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## Recovery Manager を使用したデータ・ファイルのメディア・リカバリ

データ・ファイルのメディア・リカバリの概念は、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リストアされたデータ・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に更新することです。Recovery Manager の RECOVER コマンドを使用して、メディア・リカバリを実行し、ログまたは増分バックアップを自動的に適用します。

この項では、次の項目について説明します。

- [Recovery Manager メディア・リカバリ：基本手順](#)
- [リカバリの仕組み：増分バックアップおよび REDO ログ](#)
- [不完全リカバリ](#)
- [表領域の Point-in-Time リカバリ](#)
- [制御ファイルの自動バックアップを使用した障害時リカバリ](#)

### Recovery Manager メディア・リカバリ：基本手順

メディア・リカバリを実行するには、できるかぎりリカバリ・カタログを使用可能にしてください。使用できない場合、Recovery Manager はターゲット・データベースの制御ファイルのメタデータを 사용합니다。制御ファイルとリカバリ・カタログの両方が消失した場合でも、データ・ファイルのバックアップおよび（少なくとも 1 つの）制御ファイルの自動バックアップがあると仮定して、データベースをリカバリできます。

Recovery Manager を使用したメディア・リカバリの基本的な手順は、次のとおりです。

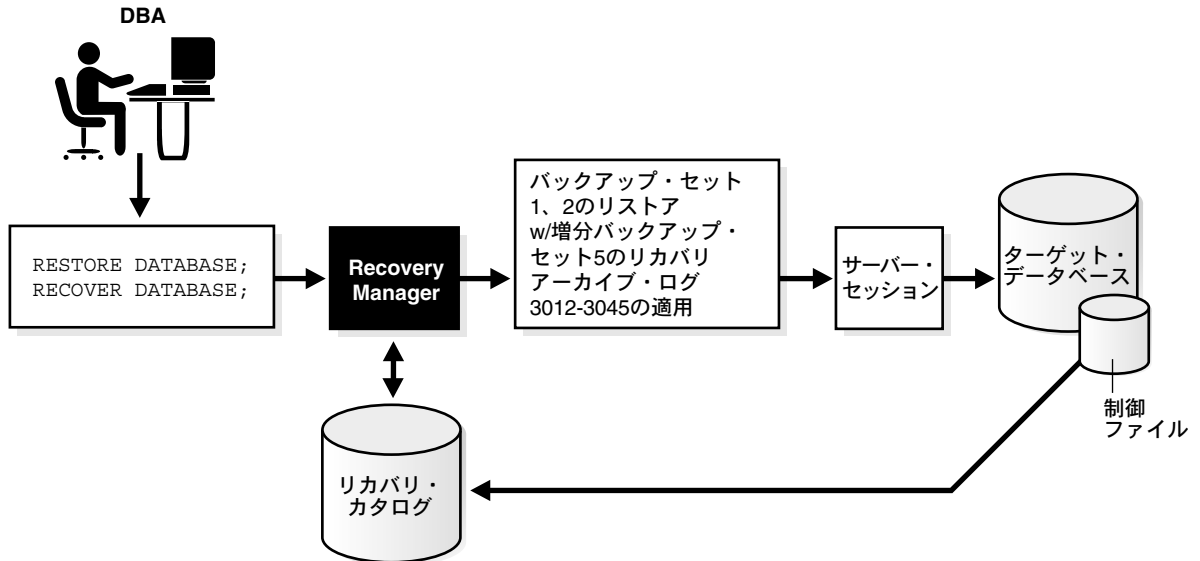
1. データベースをマウントやオープンなど、適切な状態にします。たとえば、データベース全体のリカバリを実行する場合はデータベースをマウントし、オンライン表領域のリカバリを実行する場合はデータベースをオープンします。
2. 不完全リカバリを実行するには、SET UNTIL コマンドを使用して、リカバリを終了する時刻、SCN またはログ順序番号を指定します。同様に、RESTORE および RECOVER コマンドにも UNTIL 句を指定できます。
3. RESTORE コマンドを使用して、必要なファイルをリストアします。
4. RECOVER コマンドを使用して、データ・ファイルをリカバリします。
5. データベースを、通常の状態に戻します。たとえば、データベースをオープンしたり、リカバリされた表領域をオンライン化します。

図 6-1 では、Recovery Manager のメディア・リカバリの例を示します。DBA は、次のコマンドを実行します。

```
RESTORE DATABASE;  
RECOVER DATABASE;
```

Recovery Manager がリポジトリに問合せを行います。リポジトリは、この例ではリカバリ・カタログです。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、ファイルのメタデータを取得します。次に、Recovery Manager はリストアするバックアップ・セットを決定し、どの増分バックアップとアーカイブ・ログをリカバリに使用するかも決定します。ターゲット・データベース・インスタンスのサーバー・セッションは、リストアおよびリカバリの作業を実際に行います。

図 6-1 Recovery Manager メディア・リカバリの実行



**関連項目：**

- リストアおよびリカバリの手順の詳細は、[第 10 章「Recovery Manager によるリストアとリカバリ」](#)を参照してください。
- 5-52 ページ「[制御ファイルとサーバー・パラメータ・ファイルの自動バックアップ](#)」
- RESTORE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- RECOVER コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## リカバリの仕組み：増分バックアップおよび REDO ログ

リストアされたデータ・ファイルに対し、増分バックアップと REDO のどちらを適用するか選択が必要になった場合、Recovery Manager は常に増分バックアップの使用を選択します。増分バックアップのオーバーラップしているレベルが使用可能な場合、Recovery Manager は、最も長い期間をカバーしているレベルを自動的に選択します。

Recovery Manager では、リストアされたレベル 0 の増分バックアップに、他の増分バックアップを適用する必要はありませんので注意してください。アーカイブ・ログも適用できません。Recovery Manager は必要なデータ・ファイルを、使用可能なバックアップやコピーからリストアし、可能な場合でログを適用しない場合は、増分バックアップをデータ・ファイルに適用します。

### リカバリ時における Recovery Manager のアーカイブ REDO ログの検索方法

増分バックアップが見つからない場合、Recovery Manager はリポジトリでアーカイブ REDO ログの名前を検索し、リカバリに使用します。次の作業を行う場合、Oracle は必ずアーカイブ・ログを制御ファイルに記録します。

- アーカイバ・プロセスで REDO ログをアーカイブする
- Recovery Manager でアーカイブ・ログをリストアする
- Recovery Manager の COPY コマンドでログをコピーする
- Recovery Manager の CATALOG コマンドで、アーカイブ・ログのユーザー管理バックアップをカタログ化する

再同期化の間に、Recovery Manager はアーカイブ・ログのデータをリカバリ・カタログ内に伝播させ、アーカイブ・ログをイメージ・コピーとして分類します。次のものを使用して、ログ情報を参照できます。

- LIST コマンド
- V\$ARCHIVED\_LOG 制御ファイル・ビュー
- RC\_ARCHIVED\_LOG リカバリ・カタログ・ビュー

リカバリ中に、Recovery Manager は V\$ARCHIVED\_LOG ビューで指定されたファイル名を使用して、必要なログを検索します。複数の宛先でログが作成された場合や COPY、CATALOG または RESTORE コマンドによってログが生成された場合、各ログ順序番号の複数の同一コピーがディスクに存在します。Recovery Manager には、リカバリ中にコピーを繰り返すといった作業環境はありません。AVAILABLE としてリストされたログ順序番号のコピーはすべて候補です。ある意味では、ログが別の宛先や別の方法で生成されたという事実を、Recovery Manager では認識していません。

Recovery Manager リポジトリにおいて、ログが削除またはカタログから削除されている場合、Recovery Manager は、そのログをリカバリに使用不可能だとみなします。たとえば、データベースがログ 100 を、ディレクトリ /dest1 および /dest2 にアーカイブするとし、Recovery Manager リポジトリは /dest1/log100.arc および /dest2/log100.arc の存在を示します。DELETE コマンドで /dest1/log100.arc を削除した場合、リポジトリは /dest2/log100.arc のみを、リカバリに使用可能と示します。

必要なログ順序番号のコピーがディスクに存在しないと Recovery Manager リポジトリが示した場合、Recovery Manager はバックアップを検索し、アーカイブ REDO ログをリストアして、メディア・リカバリを実行します。デフォルトでは、Recovery Manager は初期化パラメータ・ファイルで指定された、最初のローカルのアーカイブ先に、アーカイブ REDO ログをリストアします。SET ARCHIVELOG DESTINATION コマンドを実行して、別のリストア場所を指定できます。RECOVER で DELETE ARCHIVELOG オプションを指定すると、アーカイブ・ログはリストアされて適用された後に削除されます。また、RECOVER コマンドで MAXSIZE integer を指定すると、integer より多くのディスク領域が一度に使用されないようにリストアが調整されます。

## リポジトリが同期化されていないときの Recovery Manager の動作

アーカイブ・ログがディスクから削除され、リポジトリがこの事実を反映しない場合、Recovery Manager はリカバリ中に自動フェイルオーバーを実行しません。たとえば、実際には /dest1/log100.arc がオペレーティング・システムのコマンドですでに削除されているときに、リポジトリがこのログをディスク上に存在すると示し、Recovery Manager がリカバリ中にこのログ・ファイルを適用しようとした場合、エラーが発生してリカバリは終了します。Recovery Manager は、リポジトリに使用可能としてリストされている、ログ 100 の他のコピーを、自動的に適用しようとはしません。

この状況は、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アーカイブ・ログを削除した後、CROSSCHECK を実行してリポジトリの同期化に失敗した場合に、発生することがあります。リポジトリの同期化のために CROSSCHECK を実行した場合、ログの使用可能なコピーを適用する、または使用可能なディスク・コピーがない場合はログのバックアップをリストアすることで、リカバリを実行できます。

**関連項目：** SET コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## 不完全リカバリ

Recovery Manager は、完全リカバリも不完全リカバリも実行できます。SET UNTIL コマンドを使用した不完全リカバリの制限値として、また、RESTORE および RECOVER コマンドの UNTIL 句で指定されたディレクトリを使用した不完全リカバリの制限値として、時間、SCN またはログ順序番号を指定できます。最も簡単な方法は、SET UNTIL コマンドを、RESTORE および RECOVER コマンドを発行する前に実行することです。不完全リカバリを実行した後は、RESETLOGS オプションでデータベースをオープンしてください。

**関連項目：** UNTIL 句の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## 表領域の Point-in-Time リカバリ

Recovery Manager の、自動化された表領域の Point-in-Time リカバリ（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リカバリすることができます。Recovery Manager の TSPITR は、次のような場合に最も有効です。

- 誤って実行した、表の削除または切捨て操作からリカバリする場合
- 論理的に破損した表をリカバリする場合
- 不正確なバッチ・ジョブ、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 文からリカバリする場合
- 1 つの物理データベースの別の表領域に、複数の論理スキーマがあり、そのうちの 1 つのスキーマを、残りの物理データベースとは異なる時点にリカバリする場合
- VLDB（大規模データベース）で、データベース全体の Point-in-Time リカバリが可能であっても、バックアップからデータベース全体をリストアしてロールフォワードを実行するのではなく、表領域の Point-in-Time リカバリののみを実行する場合

表のエクスポートと同様に、Recovery Manager の TSPITR によって一貫性のあるデータ・セットをリカバリできます。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：** Recovery Manager を使用した TSPITR の実行方法は、[第 11 章「Recovery Manager による表領域の Point-in-Time リカバリの実行」](#)を参照してください。

## 制御ファイルの自動バックアップを使用した障害時リカバリ

ターゲット・データベースも、リカバリ・カタログも消失したと仮定します。唯一残っているのは、ターゲット・データベースおよびアーカイブ REDO ログの Recovery Manager のバックアップが入ったテープです。それでも、データベースはリカバリできるでしょうか？ 答えは、可能です。制御ファイルの自動バックアップ機能を使用可能にした、と仮定してください。障害時リカバリの状況下では、使用可能なリポジトリがなくても、Recovery Manager は制御ファイルの自動バックアップの名前を判断できます。制御ファイルのリストア、データベースのマウント、メディア・リカバリが実行可能です。

**関連項目：** 10-37 ページの「障害時リカバリの実行」および 8-19 ページの「制御ファイルの自動バックアップの有効化と無効化」を参照してください。

## Recovery Manager を使用したブロック・メディア・リカバリ

この項では、次の項目について説明します。

- [ブロック・メディア・リカバリ](#)
- [ブロック・メディア・リカバリを使用する場合](#)
- [REDO が存在しない場合のブロック・メディア・リカバリ](#)

## ブロック・メディア・リカバリ

リカバリの主な形式はデータ・ファイルのメディア・リカバリですが、Recovery Manager の BLOCKRECOVER コマンドを使用して、**ブロック・メディア・リカバリ**を実行することもできます。ブロック・メディア・リカバリは、個別の破損データ・ブロックまたはデータ・ファイルの範囲内でデータ・ブロックのセットを記録します。少数のブロックにメディア・リカバリが必要な場合、データベース全体ではなく、選択的に破損したブロックをリストアおよびリカバリできます。

ブロック・メディア・リカバリは、データ・ファイルのメディア・リカバリに様々な利点を与えます。たとえば、ブロック・メディア・リカバリには、次のような利点があります。

- リカバリを必要としているブロックのみがリストアされ、また、必要な破損ブロックのみがリカバリされるので、**平均リカバリ時間 (MTTR)** を短縮できます。ブロック・メディア・リカバリは、リカバリ中の REDO アプリケーションの時間を最小限に押さえ、I/O のオーバーヘッドを回避します。
- ブロックのリカバリ中、影響を受けるデータ・ファイルをオンライン状態のままにしておくことができます。ブロックレベルのリカバリを使用しない場合、たとえ単一ブロックが破損している状況でも、完全なデータ・ファイルのバックアップをリストアする必要があり、バックアップが作成された後には、そのファイルに対して生成される、すべての REDO を適用する必要があります。

次に示すブロック・メディア・リカバリの制限事項に注意してください。

- ブロック・メディア・リカバリは、**Recovery Manager** を使用してのみ実行できます。**SQL\*Plus** のリカバリ・インタフェースは使用できません。
- 個別のブロックの、完全リカバリのみ実行できます。つまり、すべての REDO がブロックに適用されるまではリカバリを停止できません。
- リカバリは、メディア破損のマークが付けられたブロックに対してのみ実行できます。**V\$DATABASE\_BLOCK\_CORRUPTION** ビューには、ファイルに対する最後の **BACKUP**、**BACKUP ... VALIDATE** または **COPY** コマンドの実行後に破損マークが設定された、ファイル内のブロックが表示されます。
- **Recovery Manager** の全体バックアップが必要です。増分バックアップは許可されません。
- リカバリが完了するまで、ユーザーは、メディア破損のマークが付けられたブロックにアクセスすることはできません。メディア・リカバリを受けているブロックを、どんな形で使用しようとしても、ブロックのメディア破損を示すエラー・メッセージが表示されるだけです。

**関連項目：** **BLOCKRECOVER** コマンドの構文は、10-40 ページの  
「[Recovery Manager を使用したデータ・ブロックのリカバリ](#)」および  
『[Oracle9i Recovery Manager リファレンス](#)』を参照してください。

## ブロック・メディア・リカバリを使用する場合

ブロック・メディア・リカバリは、データ消失あるいは破損の範囲が不明で、データ・ファイル全体がリカバリを必要としている場合を対象にはしていません。この場合は、データ・ファイルのメディア・リカバリが最善の解決策になります。ブロック・メディア・リカバリは、従来のデータ・ファイルのメディア・リカバリの代替ではなく、補助だということです。

ほとんどの場合、**Oracle** はブロックに破損メディアのマークを付け、インスタンス（または **Oracle Real Application Clusters** 構成で使用可能な、すべてのインスタンス）でブロックを無効にし、最初に破損を検出したときに、これをディスクに書き込みます。ブロックがリカバリされるまで、そのブロック以降の読み込みは成功しません。実行できるのは、破損のマークを付けられたブロックのブロック・リカバリのみです。この破損状態は、リカバリの間効果的に、すべてのデータベース・インスタンスでブロックをオフラインにし、ユーザー・アクセスを防ぎます。

ブロック・メディア・リカバリは、特定のブロックに影響を与えるデータ消失の場合に、最も役立ちます。ブロックレベルのデータ消失は、通常、断続的でランダムな I/O エラーが原因で発生します。このエラーはディスクに書き込まれるメモリー破損と同様に、広範囲にわたるデータの損害を引き起こすことはありません。通常、このようなタイプのブロック破損は、次の場所でレポートされます。

- 標準出力にある **Oracle** エラー・メッセージ



- alert.log
- ユーザー・トレース・ファイル
- SQL コマンド ANALYZE TABLE および ANALYZE INDEX の結果
- DBVERIFY ユーティリティの結果
- サード・パーティのメディア管理の出力

たとえば、ユーザー・トレース・ファイルでは、次のメッセージを検出できます。

```
ORA-01578:Oracle データ・ブロックに障害が発生しました (ファイル番号 7、ブロック番号 3)
ORA-01110: データ・ファイル 7: '/oracle/oradata/trgt/tools01.dbf'
ORA-01578:Oracle データ・ブロックに障害が発生しました (ファイル番号 2、ブロック番号 235)
ORA-01110: データ・ファイル 2: '/oracle/oradata/trgt/undotbs01.dbf'
```

破損ブロックは、次のように BLOCKRECOVER コマンド内で指定できます。

```
BLOCKRECOVER DATAFILE 7 BLOCK 3 DATAFILE 2 BLOCK 235;
```

## REDO が存在しない場合のブロック・メディア・リカバリ

ブロック・メディア・リカバリは、データ・ファイルのメディア・リカバリのように、欠落あるいはアクセス不可能なアーカイブ・ログを存続させることはできません。ただし、ブロック・メディア・リカバリでは、欠落あるいは破損した REDO レコードがリカバリ中のブロックに影響を与えない場合には、REDO ストリームにギャップを存続させることができます。データ・ファイル・リカバリでは、リカバリの開始時から終了時に至るまで、破損のない一連の REDO の変更が必要になりますが、これに対してブロック・メディア・リカバリでは、リカバリ中のブロックに関連する、破損のない REDO の変更のみが必要になります。

---

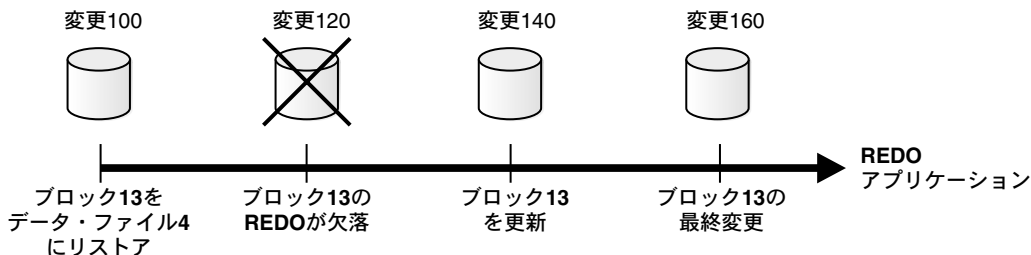
**注意：** ブロック・メディア・リカバリの間、ブロックはそれぞれが単独でリカバリされるので、ブロックのサブセットに対するリカバリも成功することがあります。

---

ブロック・メディア・リカバリの間、Recovery Manager が最初に欠落または破損した REDO レコードを検出したときにも、エラーが即座に表示されるわけではありません。これは、リカバリを受けているブロックが、後に REDO ストリームで**更新ブロック**になる可能性があるためです。ブロックが新たに更新されると、ブロックに関する以前の REDO はすべて無関係となります。これは、以前の REDO はブロックの古いインカンセーションに適用されるためです。たとえば、ユーザーがブロックに記録されたすべての行を削除したり、表を削除するとき、Oracle はブロックを更新することができます。

図 6-2 に示すように、ブロック 13 上でメディア・リカバリが実行されるとします。

図 6-2 Recovery Manager メディア・リカバリの実行



ブロック・リカバリの開始後、Recovery Manager は変更 120 の欠落を検出します。ブロック 13 が後に REDO ストリームで更新されることを予想して、Recovery Manager はリカバリを終了しません。変更 140 では、ブロック 13 に格納された表 EMPLOYEE をユーザーが削除します。この時点で、Oracle はブロック 13 を更新ブロックとしてフォーマットします。変更 120 のブロック 13 に対する REDO は EMPLOYEE 表と関連付けられており、EMPLOYEE 表が変更 140 で削除されたので、Recovery Manager は、この欠落した変更をスキップして、変更 140 および 160 間に REDO を適用することができます。

## Recovery Manager を使用したデータベースの複製

Recovery Manager の DUPLICATE コマンドを使用して、テストに使用できるようにターゲット・データベースのコピーを作成します。このコマンドは、プライマリ・データベースのファイルのバックアップおよびコピーをリストアし、新しいデータベースを作成します。

複製操作の一環として、Recovery Manager は次の処理を行います。

- ターゲット・データ・ファイルを複製データベースにリストアし、使用可能なアーカイブ・ログおよび増分バックアップをすべて使用して不完全リカバリを実行します。
- 不完全リカバリの実行後、RESETLOGS オプションを指定して複製データベースをオープンし、オンライン REDO ログを作成します。
- 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します。

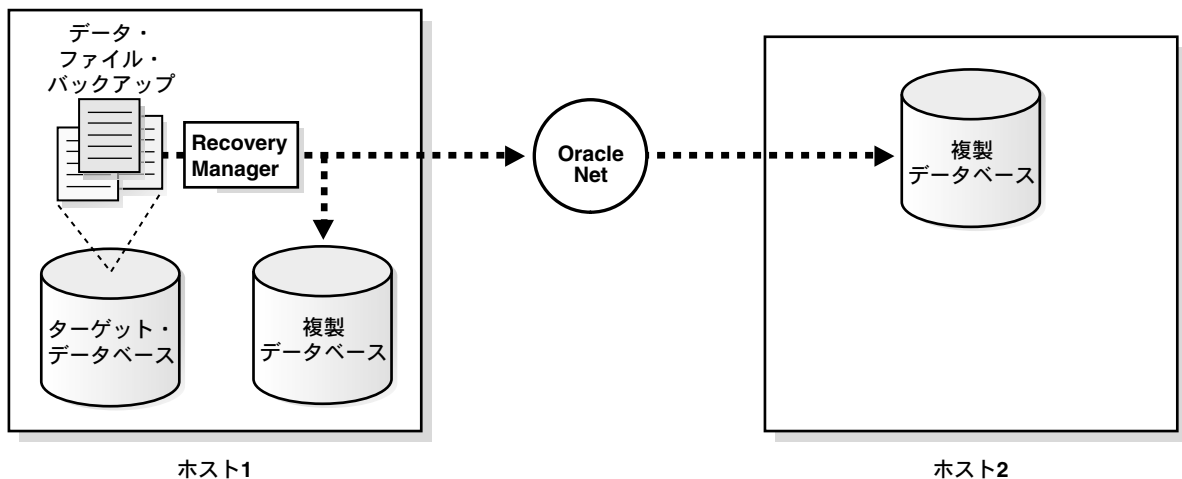
Recovery Manager による複製には次のような特徴もあります。次に示す操作が可能です。

- SKIP READONLY 句を使用して、読取り専用表領域をスキップできます（デフォルトで読取り専用表領域が含まれます）。SYSTEM 表領域にロールバック・データや UNDO データが含まれないように、SKIP TABLESPACE 句を使用して表領域を除外することもできます。表領域を削除しても後で追加できます。

- 複製データベースを新しいホストに作成できます。同じディレクトリ構造の場合は、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・ファイルでも再利用できます。
- SET UNTIL コマンド、または DUPLICATE コマンドの UNTIL 句を使用して複製データベースを作成すると、現在以外の状態にリカバリできます。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リカバリするように設定されています。
- 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できます。
- 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・カタログに登録できます。このオプションが可能なのは、複製時に、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。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 6-3 は、データベースの複製を示しています。この例では、Recovery Manager はデータ・ファイルのバックアップを 1 セット使用して、2 つの複製データベースをローカル・ホスト上に 1 つ、リモート・ホスト上に 1 つ作成します。

図 6-3 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方法は、作成するホストが同じか別か、複製ディレクトリ構造がターゲット・データベースのディレクトリ構造と同じか違うかで決まります。たとえば、同じディレクトリ構造とファイル名を複製データベースで使用できる場合もあれば、ファイルを改名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方法は、[第 12 章「Recovery Manager を使用したデータベースの複製」](#)を参照してください。  
DUPLICATE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## Recovery Manager を使用したスタンバイ・データベースの作成

スタンバイ・データベースの作成には、Recovery Manager の DUPLICATE コマンドが使用できます。Recovery Manager は次の作成手順を自動化します。

1. スタンバイ制御ファイルをリストアします。
2. プライマリ・データ・ファイルとコピーをリストアします。
3. 必要な場合は、スタンバイ・データベースを（制御ファイルがマウントされた後で）、指定された時間または生成された最新のアーカイブ REDO ログまでリカバリします。
4. Recovery Manager はマウントされたデータベースをそのままの状態にします。これによりユーザーは、このデータベースをアクティブ化する、手動または管理リカバリ・モードで配置する、または読取り専用モードでオープンすることができます。

Recovery Manager では、スタンバイ・データベースの作成を完全に自動化することはできません。理由は、ユーザーはスタンバイ・データベース用の初期化パラメータ・ファイルを手動で作成する必要があり、スタンバイ・インスタンスは、制御ファイルをマウントせずに起動する必要があり、また、スタンバイを作成する前に、必要な Oracle Net の設定をすべて行う必要があるからです。また、スタンバイ制御ファイルとして使用可能な制御ファイルのバックアップとともに、すべてのデータ・ファイルのバックアップも使用可能にする必要があります。

Recovery Manager では、スタンバイ・データベースおよびアーカイブ REDO ログのバックアップが可能です。これらのスタンバイ・バックアップは、プライマリ・バックアップで完全に置き換えることができます。つまり、スタンバイ・データ・ファイルのバックアップをプライマリ・データベースにリストアすることも、プライマリ・データ・ファイルのバックアップをスタンバイ・データベースにリストアすることも可能です。

### 関連項目：

- Recovery Manager を使用したスタンバイ・データベースの作成方法は、[第 13 章「Recovery Manager によるスタンバイ・データベースの作成」](#)を参照してください。
- スタンバイ・データベースのバックアップ方法は、[9-23 ページ「Recovery Manager を使用したスタンバイ・データベース・サイト上のファイルのバックアップ」](#)を参照してください。
- Recovery Manager を使用しないスタンバイ・データベースの作成方法は、『Oracle9i Data Guard 概要および管理』を参照してください。

---

## 概念 III: メンテナンス

この章では、Recovery Manager (RMAN) ユーティリティの使用方法に関連する基本概念を説明します。

この章の項目は、次のとおりです。

- [Recovery Manager のレポート機能](#)
- [Recovery Manager のバックアップおよびコピーのクロスチェック](#)
- [Recovery Manager のバックアップおよびコピーの削除](#)
- [Recovery Manager のメンテナンス・コマンドへの複数チャネルの割当て](#)
- [Recovery Manager のバックアップおよびコピーの可用性の変更](#)
- [Recovery Manager のバックアップおよびコピーの保存ステータスの変更](#)

**関連項目：** Recovery Manager のメンテナンス・ジョブの実行方法は、[第 18 章「Recovery Manager のメンテナンスの実行」](#)を参照してください。

## Recovery Manager のレポート機能

Recovery Manager リポジトリとは、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のことです。このリポジトリには、バックアップおよびコピーに関する情報に加え、データベース・スキーマや構成の設定など多くの有用なメタデータが格納されています。Recovery Manager コマンドの LIST、REPORT および SHOW を使用すると、リポジトリの情報にアクセスできます。

この項では、次の項目について説明します。

- [LIST コマンドの出力](#)
- [REPORT コマンドによる出力](#)
- [SHOW コマンドによる出力](#)
- [PRINT SCRIPT コマンドによる出力](#)

### LIST コマンドの出力

LIST コマンドは、Recovery Manager リポジトリ（リカバリ・カタログまたは制御ファイル）に問合せを行い、リポジトリのコンテンツのレコードを作成します。このコマンドにより取得されるデータは次のとおりです。

- Recovery Manager によって生成されたファイル（バックアップ・セット、プロキシ・コピーおよびイメージ・コピー）
- Recovery Manager によって生成されたファイル（アーカイブ・ログ、データ・ファイルおよび制御ファイル）の中に含まれるオブジェクト
- 指定したデータベースのインカネーションまたはリカバリ・カタログが認識するすべてのデータベースのインカネーション

Recovery Manager では、出力を標準出力またはメッセージ・ログに記録します。ただし、同時に両方に記録することはありません。また、出力の編成方法を制御することも、出力の詳細レベルを制御することもできます。多くの場合、LIST コマンドを実行することが、V\$ の問合せやリカバリ・カタログ・ビューよりも適切な選択となります。

LIST コマンドにより表示されるファイルは、CROSSCHECK および DELETE コマンドの操作対象になるファイルと同じものです。これにより、LIST を発行してリポジトリの内容を確認した後で、CROSSCHECK を実行すると、ディスクまたはテープに存在するファイルを確認できます。

**関連項目：**

- リストの生成方法は、17-3 ページの「[Recovery Manager のバックアップ、コピーおよびデータベース・インカネーションのリスト表示](#)」を参照してください。
- LIST コマンドの代替手段としてビューを使用する方法については、17-27 ページの「[リカバリ・カタログ・ビューへの問合せ](#)」および 17-27 ページの「[リカバリ・カタログ・ビューへの問合せ](#)」を参照してください。
- LIST コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- LOG のコマンドラインの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## REPORT コマンドによる出力

この項では、次の項目について説明します。

- [Recovery Manager のレポート](#)
- [不要なバックアップのレポート](#)
- [孤立したバックアップのレポート](#)

### Recovery Manager のレポート

Recovery Manager のレポートは、バックアップおよびリカバリ状況の分析を行うために使用します。Recovery Manager 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。
- 冗長であるため、またはリカバリ・ウィンドウ内のリカバリには必要ないために、不要とするバックアップまたはコピーはどれか。
- リカバリ不能操作が原因でリカバリ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。
- データベースのインカネーションに属してはいるが、現行のインカネーションの直接の親ではないために、**孤立した**（リストア操作に使用できない）バックアップはどれか。

REPORT NEED BACKUP および REPORT UNRECOVERABLE コマンドを定期的に行うことで、メディア・リカバリの実行に必要なバックアップが使用可能になり、メディア・リカバリが適切な時間内で実行できるようになります。さらに、REPORT OBSOLETE を定期的に行うことで、不要なファイルが削除されディスク領域を節約することができます。

**注意：** バックアップを持たないデータ・ファイルは、リカバリ不能とはみなされません。そのようなデータ・ファイルでも、そのファイルが作成された最初の時点からの REDO ログが残存している場合に、CREATE DATAFILE 文を使用するとリカバリが可能になります。

不要なバックアップのレポート

REPORT OBSOLETE コマンドでは、不要となり削除可能な、データ・ファイルのバックアップおよびコピー、制御ファイルおよびアーカイブ REDO ログを表示します。どのファイルを不要とするかは、次に示す相互排他的な方法で定義できます。

パラメータ	意味
REDUNDANCY 整数値	整数値より最近のバックアップまたはイメージ・コピーが、このファイルに対して存在します。
RECOVERY WINDOW 整数値	バックアップまたはコピーは、整数値の日数のリカバリ・ウィンドウ内で任意の時点へのリカバリには不要。それぞれのデータ・ファイルに対して、リカバリ・ウィンドウより古いバックアップが必ず1つ存在します。つまり、各データ・ファイルに対して1つのバックアップが、条件 <code>SYSDATE - CHECKPOINT_TIME &gt;= RECOVERY WINDOW</code> を必ず満たしています。この条件を満たす最新のバックアップより古いバックアップは、すべて不要となります。

不要なデータ・ファイルのバックアップに加え、Recovery Manager では不要なアーカイブ・ログおよび不要なアーカイブ・ログのバックアップをレポートします。指定されたパラメータにかかわらず、Recovery Manager ではこの設定を使用して、すでに不要となったデータ・ファイルのバックアップおよびコピーを判断し、その結果からアーカイブ・ログ（およびアーカイブ・ログのバックアップ）がいつ不要になるかを判断します。データ・ファイルの作成は、保存するログを判断しているときにはバックアップとみなされます。

REDUNDANCY および RECOVERY WINDOW パラメータ・オプションを、次に示すコマンドに指定することができます。

- REDUNDANCY または RECOVERY WINDOW 句を指定した REPORT OBSOLETE コマンド
- REDUNDANCY または RECOVERY WINDOW 句を指定した CONFIGURE RETENTION POLICY コマンド



レポートの生成時には、REPORT OBSOLETE のオプションが CONFIGURE RETENTION POLICY のオプションをオーバーライドします。たとえば、保存方針をリカバリ・ウィンドウ 7 日に構成した場合、REPORT OBSOLETE REDUNDANCY 2 を実行すると、REPORT OBSOLETE コマンドでは、REDUNDANCY 2 に該当するため不要となるバックアップとコピーが表示されます。構成済みの保存方針の効果は保持されていることに注意してください。REPORT コマンドでは、単に指定したオプションにより不要となるものを表示します。REPORT OBSOLETE の実行時に、REDUNDANCY オプションも RECOVERY WINDOW オプションも指定しない場合には、Recovery Manager は構成済の保存方針によって不要とされるものを表示します。

保存方針を無効化するために CONFIGURE RETENTION POLICY TO NONE を実行していた場合、Recovery Manager ではどのバックアップおよびコピーも不要とみなすことはありません。この場合に、オプションを指定しないで REPORT OBSOLETE を実行すると、Recovery Manager はエラー・メッセージを発行します。これは、ファイルを不要とマークする保存方針が構成されていないためです。

---

**注意：** 不要なバックアップと、期限切れのバックアップは別のものです。不要なバックアップとは、設定した保存方針に基づいて必要がないということです。期限切れバックアップとは、CROSSCHECK コマンドが指定したメディア・デバイスでファイルの検索に失敗することです。

---

**関連項目：** CONFIGURE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## 孤立したバックアップのレポート

REPORT OBSOLETE ORPHAN コマンドでは、**孤立したバックアップ**を表示します。孤立したバックアップとは、データベースのインカネーションには属しているが、このインカネーションが現行のインカネーションの直接の親でないため、使用できないバックアップのことを指します。

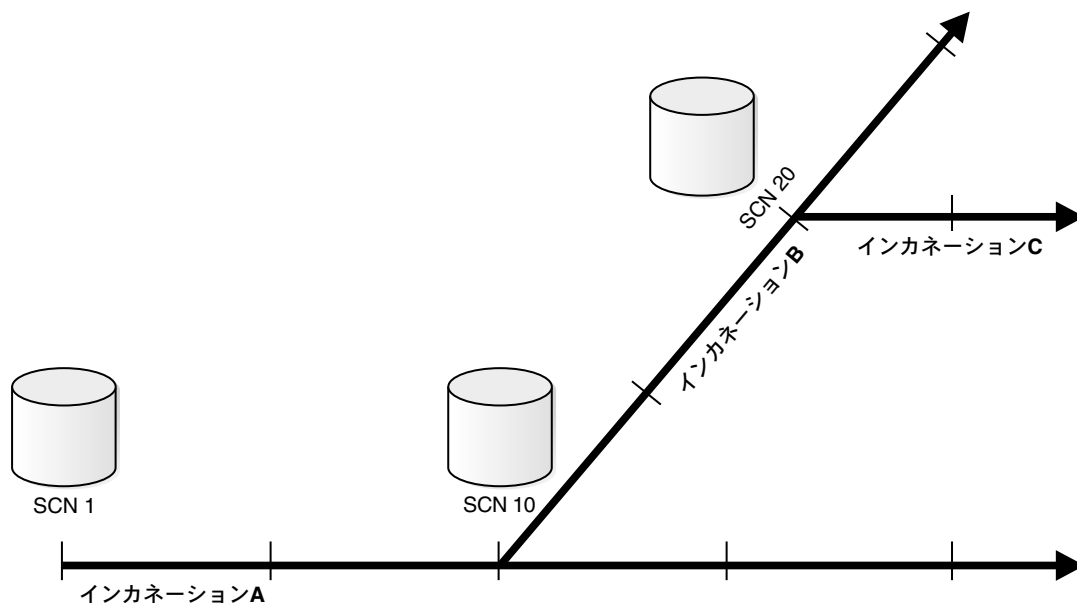
データベースの新しいインカネーションは、データベースのオープン時に、現行の REDO ログのログ順序を 1 にリセットする RESETLOGS オプションを使用すると必ず発生します。RESETLOGS オプションにより、Oracle は新しい RESETLOGS SCN を制御ファイルおよびデータ・ファイルのヘッダーに書き込みます。REDO の破損する可能性をなくすために、ある RESETLOGS SCN を持つデータベースのアーカイブ REDO ログは、別の RESETLOGS SCN を持つデータベースには適用できません。データベースが同じ（つまり、DBID が同じ）でも、データベースは、RESETLOGS 操作の実行頻度に応じて異なるインカネーションを紹介します。

Recovery Manager では、前のインカネーションにまでデータベースをリセットできます。たとえば、火曜日にデータベース prod1 に対して RESETLOGS 操作を実行した場合、Recovery Manager を使用して現行のデータベースのインカネーションを月曜日のインカネーションにまでリセットすることができます。その後、これに関連したバックアップおよ

びアーカイブ REDO ログのリストアを行うことができます。このように、前のインカネーションのバックアップが、後に役に立つ場合があります。

状況が複雑になる可能性があるのは、データベースが複数のインカネーションを介する場合です。このような場合、使用できなくなるバックアップがあります。これが、孤立です。孤立したバックアップは、ディスクやテープの領域を節約するために削除する必要があります。図 7-1 に図示するインカネーションの使用例では、3 つのインカネーションを介するデータベースを示しています。

図 7-1 孤立したバックアップ



データベースのインカネーション A は SCN1 から開始されています。SCN10 で RESETLOGS 操作を実行してインカネーション B を作成し、SCN20 で、インカネーション B に再度 RESETLOGS 操作を実行して、新しいインカネーション C を作成したと仮定します。

現行のインカ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。その関係を次の表に示します。

現行のインカネーション	使用可能なバックアップ (孤立していないバックアップ)	孤立したバックアップ
インカネーション A	インカネーション A から作成した全バックアップ	インカネーション B および C から作成した全バックアップ
インカネーション B	<ul style="list-style-type: none"> <li>■ SCN10 より前にインカネーション A から作成した全バックアップ</li> <li>■ インカネーション B から作成した全バックアップ</li> </ul>	<ul style="list-style-type: none"> <li>■ SCN10 より後にインカネーション A から作成したバックアップ</li> <li>■ インカネーション C から作成した全バックアップ</li> </ul>
インカネーション C	<ul style="list-style-type: none"> <li>■ SCN10 より前にインカネーション A から作成した全バックアップ</li> <li>■ SCN20 より前にインカネーション B から作成した全バックアップ</li> <li>■ インカネーション C から作成した全バックアップ</li> </ul>	<ul style="list-style-type: none"> <li>■ SCN10 より後にインカネーション A から作成した全バックアップ</li> <li>■ SCN20 より後にインカネーション B から作成した全バックアップ</li> </ul>

**関連項目：** レポートの生成方法は、17-12 ページの「[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](#)」を参照してください。REPORT の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## SHOW コマンドによる出力

SHOW コマンドでは、CONFIGURE コマンドで設定した構成を任意に表示します。たとえば、CONFIGURE CHANNEL の設定を表示するには、SHOW CHANNEL を実行します。また、SHOW ALL を実行すると、現行の構成をすべて表示できます。

**関連項目：** Recovery Manager の構成の表示方法は、17-18 ページの「[Recovery Manager の構成設定の表示](#)」を参照してください。SHOW コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## PRINT SCRIPT コマンドによる出力

PRINT SCRIPT コマンドでは、リカバリ・カタログ内に格納してある、指定した Recovery Manager のスクリプトのテキストを表示します。このコマンドは、リカバリ・カタログ内に格納してあるすべてのスクリプトをリスト表示するものではありません。カタログの全スクリプトを表示するには、リカバリ・カタログ・ビュー RC\_STORED\_SCRIPT\_LINE に問合せを行います。

**関連項目：** ストアド・スクリプトの表示方法は、17-24 ページの「[リカバリ・カタログに格納されたスクリプトの出力](#)」を参照してください。

PRINT SCRIPT の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## Recovery Manager のバックアップおよびコピーのクロスチェック

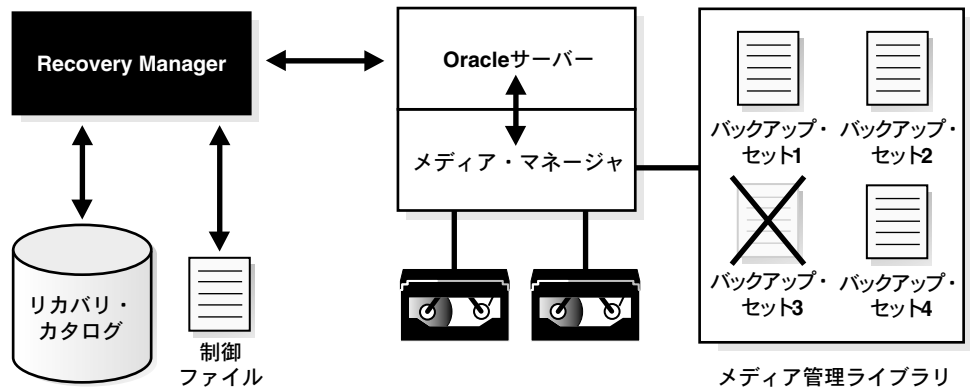
バックアップおよびコピーがディスクから消失したり、メディア管理ライブラリのテープが使用不可能になることがあります。たとえば、あるユーザーが不用意にディスクからバックアップ・ピースを削除する、またはメディア・マネージャで使用しているテープが破損することがあります。

リカバリ・カタログや制御ファイル内のバックアップ・セットおよびイメージ・コピーに関するデータが、ディスク上やメディア管理カタログ内の実際のファイルとの間で同期化されていることを確認するために、**クロスチェック**を実行します。CROSSCHECK コマンドは、Recovery Manager リポジトリに記録されているファイルのみを操作します。

[図 7-2](#) では、メディア・マネージャのクロスチェックを図示します。Recovery Manager では、チェックを行う 4 つのバックアップ・セットの名前と場所についてリポジトリ（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）に問合せを行います。

Recovery Manager は、この情報をターゲット・データベースのサーバーに送信します。このサーバーでは、バックアップについてメディア管理ソフトウェアに問合せを行います。メディア管理ソフトウェアでは、メディア・カタログをチェックして、バックアップ・セット 3 が消失していることをサーバーにレポートします。Recovery Manager は、バックアップ・セット 3 の状態をリポジトリ内で EXPIRED に更新します。これで、DELETE EXPIRED を実行すると、バックアップ・セット 3 のレコードは削除されます。

図 7-2 クロスチェック



クロスチェックは、次の操作に役立ちます。

- ディスクやテープから消失、破損したバックアップおよびコピーの期限切れの情報の更新
- アーカイブ REDO ログなどのファイルがオペレーティング・システムにより削除された場合のリポジトリの更新

クロスチェック機能を使用して、ディスクまたはテープ上のバックアップまたはコピーの状態をチェックします。バックアップまたはコピーがディスク上に存在する場合、CROSSCHECK では、ファイルのヘッダーが有効かどうかをチェックします。バックアップがテープに存在する場合、このコマンドはバックアップの存在をチェックします。

バックアップ・セット、バックアップ・ピースおよびコピーの状態は、AVAILABLE、EXPIRED または UNAVAILABLE のいずれかです。状態情報は、LIST コマンドの出力とリカバリ・カタログ・ビューで確認できます。

DELETE EXPIRED コマンドを発行することで、期限切れのすべてのバックアップおよびコピーを削除することに注意してください。Recovery Manager は、リポジトリから期限切れファイルのレコードを削除します。なんらかの理由でファイルがメディアに残存していた場合、Recovery Manager は警告を発行し、削除できない不一致になったオブジェクトをリストします。

---

**注意：** CROSSCHECK コマンドでは、オペレーティング・システムのファイルやリポジトリ・レコードを削除できません。これらの操作には、DELETE コマンドを使用する必要があります。

---

### 関連項目：

- クロスチェックの実行方法は、18-2 ページの「[バックアップおよびコピーのクロスチェック](#)」を参照してください。
- CROSSCHECK の構文と可能な状態値は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- DELETE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## Recovery Manager のバックアップおよびコピーの削除

この項では、次の項目について説明します。

- [Recovery Manager による削除](#)
- [Recovery Manager による削除方法のまとめ](#)
- [DELETE コマンドによるバックアップおよびコピーの削除](#)
- [リポジトリとメディアが対応していない場合の DELETE コマンドの動作](#)
- [BACKUP ... DELETE INPUT コマンドによるバックアップおよびコピーの削除](#)

## Recovery Manager による削除

Recovery Manager では、バックアップまたはコピーのそれぞれに対応するレコードを制御ファイルに（使用している場合にはリカバリ・カタログにも）作成します。たとえば、全データベースのバックアップ・セットを生成する場合、このバックアップ・セットのレコードは `V$BACKUP_SET` 制御ファイル・ビューで表示できます。リカバリ・カタログを使用している場合には、`RC_BACKUP_SET` カタログ・ビューでレコードを表示させることもできます。

`V$` 制御ファイル・ビューとカタログ表では情報の格納方法が異なります。これは、Recovery Manager がリポジトリのレコードを処理する方法に影響します。制御ファイル内に格納された Recovery Manager の情報は、`V$` ビューでは表形式で表示されますが、実際には表として格納されているわけではありません。このデータは、制御ファイルの循環再利用セクションの内部データ構造に格納されます。それとは逆に、リカバリ・カタログのデータは、実際にデータベースの表に格納されます。

Recovery Manager コマンドを使用してバックアップまたはコピーの削除を行うと、次の手順が実行されます。

- オペレーティング・システムから物理ファイルを削除
- 制御ファイル内のバックアップ・レコードまたはコピー・レコードのステータスを `DELETED` に更新
- カタログ表から（カタログを使用している場合）バックアップ・レコードまたはコピー・レコードを削除

制御ファイル・データが格納される方法によって、Recovery Manager は制御ファイルからレコードを削除できません。レコードのステータスを `DELETED` に更新するのみです。ただし、カタログ表は通常のデータベース表であるため、Recovery Manager はカタログ表から行を削除します。

Recovery Manager による削除方法のまとめ

表 7-1 では、様々な Recovery Manager の削除コマンドとスクリプトの機能を説明しています。表中のコマンドまたはスクリプトは、リカバリ・カタログを必要としません。

表 7-1 メンテナンス・コマンドとスクリプト

コマンドまたはスクリプト	用途
DELETE	<p>物理バックアップおよびイメージ・コピーを削除するために、制御ファイルのレコードのステータスを DELETED に更新して、このレコードをカタログから（カタログを使用している場合）削除します。</p> <p>DELETE で削除する必要があるバックアップおよびコピーを EXPIRED または OBSOLETE で指定します。存在するオブジェクトに対して DELETE EXPIRED を実行すると、警告が発行され、そのオブジェクトは削除されません。DELETE FORCE を実行すると、この動作をオーバーライドしてオブジェクトを削除できます。</p>
BACKUP ... DELETE [ALL] INPUT	<p>アーカイブ・ログ、データ・ファイルのコピーまたはバックアップ・セットのバックアップを行った後、バックアップが正常に完了した入力ファイルをオペレーティング・システムから削除します。さらに、Recovery Manager では削除した入力ファイルのリポジトリ・レコードを更新します。</p> <p>DELETE INPUT を（ALL を指定せずに）指定した場合、Recovery Manager ではバックアップを行ったコピーのみを削除します。ALL を指定した場合、Recovery Manager では指定したログのすべてのコピーを V\$ARCHIVED_LOG ビューから検索して削除します。</p>
CHANGE ... UNCATALOG	<p>指定されたバックアップおよびコピーのリカバリ・カタログ・レコードを削除して、それらの制御ファイル・レコードのステータスを DELETED に変更します。CHANGE ... UNCATALOG コマンドはオペレーティング・システムのファイルは削除しないことに注意してください。</p>
prgrmanc.sql	<p>ステータスが DELETED のバックアップまたはコピーのレコードを、リカバリ・カタログからすべて削除します。リリース 8.1.6 より前の Recovery Manager では、リカバリ・カタログのレコードに対して削除を行わずに（現行リリースでは削除を行う）、ステータスを DELETED に更新する場合があります。</p> <p><b>注意：</b>リカバリ・カタログのバージョンが Recovery Manager の現行リリースと同一の場合、リカバリ・カタログのレコードに DELETED マークが設定される可能性があるのは、カタログがリリース 8.1.6 より前のカタログからアップグレードされた場合、またはリリース 8.1.6 より前に作成されたバックアップ制御ファイルから再同期化された場合のみです。</p>



**関連項目：** 7-8 ページ「[Recovery Manager のバックアップおよびコピーのクロスチェック](#)」

## DELETE コマンドによるバックアップおよびコピーの削除

DELETE コマンドでは、LIST および CROSSCHECK コマンドの操作対象になる任意のファイルを削除できます。たとえば、バックアップ・セット、アーカイブ REDO ログおよびデータ・ファイルのコピーを削除することができます。DELETE コマンドは、物理ファイルと、そのファイルに対するカタログ・レコードを両方とも削除します。

### オペレーティング・システムのコマンドのかわりに DELETE を使用する利点

Recovery Manager のバックアップおよびコピーの削除には、オペレーティング・システムのユーティリティではなく、常に DELETE を使用してください。これを行わないと、Recovery Manager リポジトリは、ファイル・システムまたはテープに存在するものを同期しません。ファイルの削除にオペレーティング・システムのユーティリティを使用する場合は、次のようになります。

- 削除するファイルのステータスを EXPIRED に変更するために、CROSSCHECK を実行した後で、DELETE EXPIRED を実行
- オペレーティング・システムのユーティリティで削除したファイルに対して DELETE を実行
- CHANGE ... UNCATALOG を実行してカタログ・レコードを削除

### 不要なバックアップとコピーを削除

DELETE OBSOLETE コマンドにより提供される適切な方法により、不要になったバックアップおよびコピーを物理的に削除し、それらのカタログ・レコードを削除します。このコマンドでは、REPORT OBSOLETE コマンドと同様に REDUNDANCY、RECOVERY WINDOW および ORPHAN オプションを使用します。

保存方針が構成してある場合には、DELETE OBSOLETE を定期的に行うことで、この保存方針で不要と判断されたバックアップおよびコピーのすべてを削除します。たとえば、DELETE OBSOLETE をスクリプトにして、スケジュール・ユーティリティにより毎晩実行できます。この方法により、不要なすべてのファイルを削除し、ディスク・ドライブを整理できます。

**関連項目：** 7-4 ページ「[不要なバックアップのレポート](#)」

### 期限切れバックアップの削除

CROSSCHECK コマンドでは、バックアップまたはコピーをバックアップした位置で、それらの位置を特定できなかった場合には、期限切れのマークを付けます。つまり、期限切れと

は、「見つからない」ということです。期限切れのバックアップおよびコピーを表示させるためには、CROSSCHECK コマンドを実行します。次に例を示します。

```
RMAN> CROSSCHECK BACKUP;  
  
crosschecked backup piece: found to be 'AVAILABLE'  
backup piece handle=0ad8d32i_1_1 recid=10 stamp=445025363  
crosschecked backup piece: found to be 'AVAILABLE'  
backup piece handle=c-1334876723-20011105-00 recid=11 stamp=445025367  
crosschecked backup piece: found to be 'EXPIRED'  
backup piece handle=0cd8d36l_1_1 recid=12 stamp=445025473  
crosschecked backup piece: found to be 'AVAILABLE'  
backup piece handle=c-1334876723-20011105-01 recid=13 stamp=445025475  
Crosschecked 4 objects
```

DELETE EXPIRED コマンドでは、期限切れのバックアップおよびコピーのリカバリ・カタログ・レコードを削除し、それらの制御ファイル・レコードのステータスを DELETED に更新します。

このコマンドが特に役立つのは、Recovery Manager のバックアップまたはアーカイブ・ログを、オペレーティング・システムのユーティリティを不用意に使用してディスクから削除した場合です。これを行うと、Recovery Manager リポジトリはディスク上の物理的な実体と同期化されることがありません。CROSSCHECK コマンドを実行することで、Recovery Manager は検索できなかったバックアップおよびコピーを期限切れとしてマークします。その後、DELETE EXPIRED を実行することで、それらのファイルに応じたレコードを削除します。

### バックアップ済みのアーカイブ REDO ログの削除

アーカイブ・ログなどのファイルのうち、テープへのバックアップが指定した回数に達したもののみを削除するようにできます。DELETE コマンドは、この動作をサポートします。次の例では、2 回以上テープにバックアップされたバックアップとコピーをすべて削除します（アーカイブ・ログはコピーとみなされます）。

```
DELETE ARCHIVELOG ALL BACKED UP 2 TIMES TO DEVICE TYPE sbt;
```

## リポジトリとメディアが対応していない場合の DELETE コマンドの動作

オブジェクトのリポジトリ・レコードに、オブジェクトの物理的なステータスが反映されない場合があります。たとえば、ログをディスクにバックアップしてから、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オブジェクトを削除するとします。CROSSCHECK コマンドを実行してリポジトリを更新せず、オブジェクトに対して DELETE を実行すると、リポジトリはオブジェクトが AVAILABLE であることを示しますが、実際には消失しています。次の表に、このような場合の DELETE の動作を示します。

リポジトリのステータス	物理的なステータス	DELETE コマンドの動作
AVAILABLE	メディア上で検出されない	オブジェクトは削除されず、ジョブの終了時に不一致のオブジェクトのリストがレポートされます。リポジトリのステータスは更新されません。
EXPIRED	メディア上で検出される	オブジェクトは削除されず、ジョブの終了時に不一致のオブジェクトのリストがレポートされます。リポジトリのステータスは更新されません。
UNAVAILABLE	すべて	リポジトリ・レコードが削除され、存在する場合はオブジェクトが削除されます。すべての I/O エラーは無視されます。
FORCE	すべて	リポジトリ・レコードが削除され、存在する場合はファイルが削除されます。すべての I/O エラーは無視されます。ジョブの終了時に、削除されたオブジェクトの数が表示されます。

## BACKUP ... DELETE INPUT コマンドによるバックアップおよびコピーの削除

BACKUP ... DELETE INPUT コマンドでは、アーカイブ REDO ログ、データ・ファイルのコピーおよびバックアップ・セットのバックアップを行った後で、これらを削除します。この機能が特に役立つのは、アーカイブ・ログをディスクからテープにバックアップする場合です。Recovery Manager では、各ログ順序番号のコピー 1 つに対してバックアップを 1 回のみ行います。その後、バックアップしたファイルを削除します。たとえば、次のコマンドを発行したとします。

```
BACKUP ARCHIVELOG ALL DELETE INPUT;
```

このコマンドにより、Recovery Manager は使用可能な各順序番号を持つそれぞれのログのコピーを 1 つずつバックアップします。その後、実際にバックアップを行ったファイルのみを削除します。

DELETE ALL INPUT オプションを指定すると、Recovery Manager では指定した基準と一致するファイルであれば、同一のログ順序番号を持つファイルが複数存在していたとしても、すべて削除します。たとえば、異なる 3 つのディレクトリにアーカイブを行うことを想定します。この場合、次のコマンドを実行します。

```
BACKUP ARCHIVELOG ALL FROM SEQUENCE 1200 DELETE ALL INPUT;
```

この場合、Recovery Manager では 1200 から最新の順序までの各ログ順序に対して 1 つのコピーのみバックアップを行い、その順序番号を持つ 3 つのアーカイブ先に含まれるログは、すべて削除します。

アーカイブ・ログのフェイルオーバー機能とは、Recovery Manager がログ順序番号を持つ正常なコピーを使用可能なすべてのアーカイブ先で検索するということです。たとえば、使用可能なアーカイブ先が /log1 と /log2 のみで、そこに同一の順序番号が含まれているとします。このとき、次のコマンドを実行します。

```
BACKUP ARCHIVELOG FROM SEQUENCE 123 DELETE ALL INPUT;
```

Recovery Manager は、使用可能なすべてのアーカイブ・ディレクトリから読み込みを始めます。たとえば、Recovery Manager がディレクトリ /log1 から開始して、このディレクトリで log\_123.f を検索したと仮定します。その後、Recovery Manager が log\_124.f の破損を検出した場合、このログの正常なコピーを /log2 から検索します。DELETE ALL INPUT を指定しているので、ディスク上にあるログ順序 123 以上のコピーをすべて削除します。

### 関連項目：

- バックアップおよびコピーの削除方法は、18-5 ページの「[バックアップおよびコピーの削除](#)」を参照してください。
- CHANGE ... UNCATALOG コマンドの使用方法是、18-17 ページの「[バックアップまたはコピーの保存方針からの除外](#)」を参照してください。
- CHANGE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- DELETE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## Recovery Manager のメンテナンス・コマンドへの複数チャネルの割当て

メンテナンス・コマンドには CROSSCHECK や DELETE などがあり、これらのコマンドは構成したすべてのデバイス・タイプで使用できます。バックアップを作成したすべてのデバイス用に自動チャネルが構成済みの場合には、CROSSCHECK または DELETE を、Recovery Manager プロンプトから次のように実行するのみです（出力例も含む）。

```
CROSSCHECK BACKUP;

allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: sid=14 devtype=SBT_TAPE
channel ORA_SBT_TAPE_1: WARNING: Oracle Test Disk API
using channel ORA_DISK_1
crosschecked backup piece: found to be 'AVAILABLE'
backup piece handle=/ade/lashdown_seeddb/oracle/dbs/c-1337987235-20011210-01 recid=6
stamp=448109988
Crosschecked 1 objects

crosschecked backup piece: found to be 'AVAILABLE'
backup piece handle=01dbb78r_1_1 recid=1 stamp=448109852
crosschecked backup piece: found to be 'AVAILABLE'
backup piece handle=02dbb791_1_1 recid=2 stamp=448109857
crosschecked backup piece: found to be 'AVAILABLE'
backup piece handle=03dbb7af_1_1 recid=3 stamp=448109904
Crosschecked 3 objects
```

Recovery Manager では、デフォルトで構成済みの DISK チャネルに加え、sbt デバイス用に構成されたチャネルも使用します。Recovery Manager は、バックアップの作成に使用したチャネルと同じデバイス・タイプを持つすべてのチャネルについて、各バックアップを検索します。複数チャネルのクロスチェック機能は、次の使用例に向けて設計されています。

- テープとディスクの両方に存在する、すべてのバックアップ・ピースまたはプロキシ・コピーのクロスチェックと削除を単一のコマンドで可能にします。
- 各バックアップ・ピースまたはプロキシ・コピーが 1 つのノードのみに存在する Oracle Real Application Clusters 構成で、クロスチェックおよび削除作業を正しく実行させます。

1 つのデバイス・タイプのみが存在するファイルのメンテナンスを実行するためには、メンテナンス・コマンドにデバイス・タイプを指定します。たとえば、次のように入力します。

```
DELETE EXPIRED BACKUP DEVICE TYPE sbt;
```

自動チャネルを構成していない場合には、メンテナンス・チャネルを手動で割り当てます。たとえば、次のようにコマンドを実行できます。

```
ALLOCATE CHANNEL FOR MAINTENANCE TYPE sbt;
CROSSCHECK BACKUP;
```

### 関連項目：

- 複数のメンテナンス・チャネルの割当て方法は、18-10 ページの「[複数の Recovery Manager のチャネルでのクロスチェックおよび削除](#)」を参照してください。
- `ALLOCATE CHANNEL FOR MAINTENANCE` の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## Recovery Manager のバックアップおよびコピーの可用性の変更

Recovery Manager では、バックアップおよびコピーが使用可能であるか使用不可能であるかを示すためにリポジトリを更新できます。たとえば、いくつかのバックアップがあるテープ・ドライブをアップグレードする場合があります。それらのバックアップおよびコピーに対して、`CHANGE ... UNAVAILABLE` コマンドを使用して、ドライブのメンテナンス継続期間は使用不可能であるとしてマークします。Recovery Manager では、UNAVAILABLE なバックアップおよびコピーをバックアップおよびリカバリ操作で考慮しません。

メンテナンスが完了した場合、`CHANGE ... AVAILABLE` コマンドを発行して、Recovery Manager にバックアップおよびコピーが再び使用可能になったことを通知します。ファイルの設定をステータス `AVAILABLE` に戻した後、再び `CROSSCHECK` を実行して Recovery Manager がファイルにアクセスできるようにするために二重チェックを行います。

### 関連項目：

- 18-15 ページ「[バックアップまたはコピー・レコードの可用性のステータスの変更](#)」
- `CHANGE` コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## Recovery Manager のバックアップおよびコピーの保存ステータスの変更

Recovery Manager では、バックアップおよびコピーのステータスを変更するために、KEEP および NOKEEP オプションが使用できます。KEEP オプションでは、現行の保存方針からバックアップおよびコピーを除外します。除外期間は無期限、または UNTIL で指定した時間までに設定できます。Recovery Manager は、保存方針では不要と判断されるファイルでも、不要のマークを付けません。このようなバックアップは、**長期バックアップ**と呼ばれます。NOKEEP オプションは、ファイルの除外ステータスを削除します。

CHANGE ... KEEP または CHANGE ... NOKEEP コマンドでは、バックアップおよびコピーの保存ステータスを変更します。たとえば、次のコマンドでは保存方針により、タグ SAVE\_BACKUP が付いているバックアップ・セットを不要とすることができます。

```
CHANGE BACKUPSET TAG save_backup NOKEEP;
```

### 関連項目：

- 18-17 ページ「[バックアップまたはコピーの保存方針からの除外](#)」
- CHANGE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。





# 第 III 部

---

## Recovery Manager を使用したバックアップ およびリカバリの実行

第 III 部では、Recovery Manager ユーティリティを使用したバックアップおよびリカバリ操作の実行方法、複製およびスタンバイ・データベースの作成方法を説明します。また、Recovery Manager のチューニング方法、一般的な問題に対するトラブルシューティングも説明します。この部では、次の章について説明します。

- 第 8 章「Recovery Manager 環境の構成」
- 第 9 章「Recovery Manager でのバックアップおよびコピーの作成」
- 第 10 章「Recovery Manager によるリストアとリカバリ」
- 第 11 章「Recovery Manager による表領域の Point-in-Time リカバリの実行」
- 第 12 章「Recovery Manager を使用したデータベースの複製」
- 第 13 章「Recovery Manager によるスタンバイ・データベースの作成」
- 第 14 章「Recovery Manager のチューニング」
- 第 15 章「トラブルシューティング」



---

## Recovery Manager 環境の構成

この章では、設定および構成作業の実行方法を説明します。この章の項目は、次のとおりです。

- [メディア・マネージャにバックアップを作成する場合の Recovery Manager の構成](#)
- [自動チャネルの構成](#)
- [制御ファイルとサーバー・パラメータ・ファイルの自動バックアップの構成](#)
- [バックアップの保存方針の構成](#)
- [バックアップ・セットの最大サイズの設定](#)
- [バックアップの最適化の構成](#)
- [バックアップ・コピー数の設定](#)
- [データベース全体のバックアップから除外する表領域の構成](#)
- [Recovery Manager のグローバリゼーション・サポートの環境変数の設定](#)
- [スナップショット制御ファイルの場所の設定](#)
- [Recovery Manager を共有サーバーで使用する場合の設定](#)
- [リカバリ・カタログの設定](#)

## メディア・マネージャにバックアップを作成する場合の Recovery Manager の構成

ほとんどのプラットフォームでは、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用して、バックアップやリストアを実行する場合は、Oracle にメディア・マネージャを統合する必要があります。メディア・マネージャは Oracle 製品ではなく、サード・パーティ・ベンダーから入手する必要があります。Recovery Manager とメディア・マネージャを一緒に使用する場合は、ベンダーから製品固有の情報をすべて入手する必要があります。

この項では、メディア・マネージャとともに使用するための Recovery Manager 構成の基本的な手順について説明します。実際の手順は、インストールするメディア管理製品と Oracle を実行するプラットフォームにより異なります。

メディア・マネージャを構成する場合は、次の項をお読みください。

1. [Recovery Manager でメディア・マネージャを使用する場合の前提条件](#)
2. [Oracle とメディア・マネージャの通信方法](#)
3. [Recovery Manager とメディア・マネージャの統合: 基本手順](#)
4. [メディア・マネージャ・ライブラリが正常に統合されているかどうかのテスト](#)
5. [自動チャネルをメディア・マネージャで使用する場合の構成](#)

**関連項目：** メディア管理ソフトウェアの概要および Recovery Manager に対する含意は、4-18 ページの「[メディア管理](#)」を参照してください。

## Recovery Manager でメディア・マネージャを使用する場合の前提条件

Recovery Manager をメディア・マネージャとともに使用する前に、Recovery Manager をインストールし、Recovery Manager とメディア・マネージャが通信できるようにする必要があります。この手順は、メディア・マネージャ・ベンダーのソフトウェアのドキュメントに記載されています。

インストールする製品に依存しますが、基本的には次の手順を行います。

1. ターゲット・ホストまたは本番ネットワーク上にメディア管理ソフトウェアをインストールおよび構成します。この段階では、Recovery Manager と統合する必要はありません。
2. ターゲット・データベースのホスト上で、オペレーティング・システム・ファイルについて Recovery Manager 以外のバックアップを作成できることを確認します。この手順により、後のトラブルシューティングが非常に容易になります。メディア・マネージャにファイルをバックアップする方法は、メディア管理ソフトウェアのドキュメントを参照してください。

3. Oracle サーバーと統合するサード・パーティ・メディア管理モジュールを入手してインストールします。このモジュールには、Oracle がメディア・マネージャにアクセスするときにロードするライブラリが含まれている必要があります。

前述の手順を実行していない場合は、メディア管理の構成には進まないでください。

## Oracle とメディア・マネージャの通信方法

メディア・マネージャにバックアップを作成する場合、Recovery Manager ではメディア管理ライブラリとの通信が必要になります。Oracle9i で Recovery Manager とメディア・マネージャを使用する手順は、以前のリリースから変更されていません。

メディア・マネージャ・ライブラリのロード手順は、オペレーティング・システム固有です。チャンネルの割当て時に、Recovery Manager では次のアルゴリズムを使用してロード対象のライブラリを判断します。

1. ALLOCATE CHANNEL または CONFIGURE CHANNEL コマンドの SBT\_LIBRARY パラメータにより示されているライブラリのロードを試みます。SBT\_LIBRARY パラメータが指定されていない場合は、Oracle が次の手順に進みます。
2. デフォルトのメディア管理ライブラリのロードを試みます。デフォルトのライブラリのファイル名は、オペレーティング・システム固有です。UNIX では、ライブラリ・ファイル名は \$ORACLE\_HOME/lib/libobk.so で、拡張子名は .so、.sl、.a など、プラットフォームに応じて異なります。Windows NT では、ライブラリ名は %ORACLE\_HOME%\bin\orasbt.dll です。

---

**注意：** デフォルトのメディア管理ライブラリ・ファイルは以前のリリースと同様に Oracle 標準インストールには含まれていないので注意してください。このファイルは、サード・パーティのメディア管理ソフトウェアをインストールしないと作成できません。

---

3. Oracle が前述の手順でデフォルト・ライブラリを検出できない場合、Recovery Manager は ORA-27211 エラーを発行して終了します。

チャンネル割当てに失敗すると、トレース・ファイルが USER\_DUMP\_DEST ディレクトリに書き込まれます。サンプルの出力を次に示します。

```
SKGFQ OSD: Error in function sbtinit on line 2278
SKGFQ OSD: Look for SBT Trace messages in file /oracle/rdbms/log/sbtio.log
SBT Initialize failed for /oracle/lib/libobk.so
```

## Recovery Manager とメディア・マネージャの統合 : 基本手順

メディア管理ソフトウェアをインストールすると、メディア管理ライブラリは Oracle サーバーに自動的に統合されます。これ以上の統合作業を行う必要はありません。ただし、この項の手順を使用して、メディア・マネージャが正しく統合されていることを確認できます。

メディア管理と Oracle との統合に関する仕様は、メディア管理製品およびオペレーティング・システムにより異なるため、この項ですべての場合を網羅することはできません。この項では、UNIX および Windows NT 上でメディア管理ライブラリに名前を付けて Oracle でロードできるようにする基本手順を説明します。

### UNIX 上での Recovery Manager とメディア・マネージャの統合 : 基本手順

UNIX 上では、Oracle は UNIX 共有ライブラリ `libobk.so` を介してメディア管理ライブラリにアクセスします。このファイルが、システム・パスのいずれかの場所にある必要があります。`libobk.so` は、Oracle が最初に検索する `$ORACLE_HOME/lib` に入れることをお勧めします。

メディア管理ライブラリのインストール時に、インスタンスを起動またはシャットダウンする必要はありません。

#### UNIX 上でメディア・マネージャを統合する方法 :

1. 古い `libobk.so` シンボリック・リンクが `$ORACLE_HOME/lib` にある場合は、メディア・マネージャをインストールする前に削除します。たとえば、次のように入力します。

```
% rm $ORACLE_HOME/lib/libobk.so
```

2. インストールした後、メディア・マネージャのドキュメントを調べて、メディア管理ライブラリがどこにインストールされるかを確認します。たとえば、ライブラリが `/vendor/lib/oracle_lib.so` としてインストールされるとします。

3. インストールされたメディア管理ライブラリの名前を `$ORACLE_HOME/lib/libobk.so` に変更するか、`libobk.so` という名前のライブラリへのシンボリック・リンクを作成します。たとえば、ライブラリへのシンボリック・リンクは次のように作成できます。

```
% ln -s /vendor/lib/oracle_lib.so $ORACLE_HOME/lib/libobk.so
```

または、ライブラリの名前を `libobk.so` に変更することもできます。たとえば、次のようにします。

```
% mv /vendor/lib/oracle_lib.so $ORACLE_HOME/lib/libobk.so
```

## Windows NT 上での Recovery Manager とメディア・マネージャの統合： 基本手順

Windows NT の場合、Oracle はライブラリ `orasbt.dll` を介してメディア管理ライブラリにアクセスします。このファイルが、システム・パスのいずれかの場所にある必要があります。通常、このファイルは、Oracle ホームの `%ORACLE_HOME%\bin` フォルダにあります。

メディア管理ライブラリのインストール時に、インスタンスを起動またはシャットダウンする必要はありません。

### Windows NT 上で Recovery Manager にメディア・マネージャを統合する方法：

1. システム・パスに `orasbt.dll` がすでにある場合は、メディア・マネージャをインストールする前にこれを削除します。たとえば、次のように入力します。

```
D:\> del %ORACLE_HOME%\bin\orasbt.dll
```

2. インストールした後、メディア・マネージャのドキュメントを調べて、メディア管理ライブラリがどこにインストールされるかを確認します。たとえば、ライブラリが `D:\vendor\lib\oracle_lib.dll` としてインストールされるとします。

3. インストールされたメディア管理ライブラリが `orasbt.dll` という名前でない場合は、このライブラリの名前を `%ORACLE_HOME%\bin\orasbt.dll` に変更します。たとえば、インストールしたライブラリを次のようにコピーできます。

```
D:\> copy D:\vendor\lib\oracle_lib.dll %ORACLE_HOME%\bin\orasbt.dll
```

`orasbt.dll` ファイルは、ライブラリを含むフォルダがシステムの PATH 変数設定内にある場合は、`%ORACLE_HOME%\bin` フォルダにある必要はありません。PATH 変数設定をチェックするには、「スタート」→「設定」→「コントロール パネル」→「システム」→「環境」を選択します。

### 関連項目：

- 使用プラットフォーム上でメディア・マネージャの統合を行う方法は、オペレーティング・システム固有の Oracle ドキュメントおよびメディア・ベンダーが提供するドキュメントを参照してください。
- メディア・マネージャに関するトラブルシューティングの例は、15-27 ページの「[メディア・マネージャのインストール後に、Recovery Manager のチャンネルの割当てに失敗した場合：例](#)」を参照してください。

## メディア・マネージャ・ライブラリが正常に統合されているかどうかのテスト

Oracle サーバーがメディア管理ライブラリをロードできることを確認した後、Recovery Manager がメディア・マネージャにバックアップできることを確認するテストを行います。このテストは、次の使用例で説明されている手順に沿って行います。

- メディア管理ソフトウェアの Recovery Manager のバックアップのための構成
- メディア・マネージャのチャンネルの割当てのテスト
- メディア・マネージャに対するバックアップのテスト

### メディア管理ソフトウェアの Recovery Manager のバックアップのための構成

メディア管理ソフトウェアをインストールした後、このソフトウェアに必要な構成を実行し、ソフトウェアが Recovery Manager のバックアップを受け入れられるようにします。インストールしたメディア管理ソフトウェアの種類によっては、メディア・プールの定義、ユーザーの構成、クラスの構成などが必要になることがあります。

ALLOCATE CHANNEL または CONFIGURE CHANNEL コマンドに必要な PARMS 設定と、(必要な場合は) BACKUP コマンドのための推奨 FORMAT 文字列を判断する必要があります。PARMS パラメータは、メディア・マネージャに指示を送信します。たとえば、次のベンダー固有の PARMS 設定は、メディア・マネージャに対して `oracle_tapes` というボリューム・プールへのバックアップを指示します。

```
PARMS='ENV=(NSR_DATA_VOLUME_POOL=oracle_tapes)'
```

適切な設定に関しては、サード・パーティ・ベンダーのドキュメントを参照してください。

Recovery Manager が提供する置換変数を使用すると、バックアップをメディア・マネージャに書き込むときに、一意のバックアップ・ピース名を生成できます。バックアップ・ピース名は、BACKUP コマンド、CONFIGURE CHANNEL コマンドまたは ALLOCATE CHANNEL コマンドに指定される FORMAT 文字列によって決定されます。

FORMAT パラメータを指定しない場合、Recovery Manager は %U 置換変数を使用して、一意の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・ファイル名として認識するため、この名前はメディア・マネージャのカタログ内で一意の名前にしてください。

---

**注意：** メディア・マネージャの中には、14 文字のバックアップ・ピース名のみをサポートするものや、特殊な FORMAT 文字列を必要とするものがあります。使用するメディア管理ソフトウェアのドキュメントを参照して、使用しているメディア・マネージャの文字列の文字制限を確認してください。

---



一部のメディア・マネージャでは、バックアップまたはリストアできるファイルの最大サイズに制限があります。Recovery Manager は複数のデータ・ファイルを 1 つのファイルに多重化しますが、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る状況では、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ドキュメントを参照し、運用時のファイル・サイズ制限を確認してください。Recovery Manager 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・ピースのサイズを制限するには、CONFIGURE CHANNEL および ALLOCATE CHANNEL コマンドの MAXPIECESIZE パラメータを使用します。システム上の demo ディレクトリにある \*.rcv スクリプトを参照してください。このスクリプトは、オペレーティング・システム指定の場所（UNIX では \$ORACLE\_HOME/rdbms）にあります。

#### 関連項目：

- ALLOCATE CHANNEL の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- チャネル制御オプションについては、『Oracle9i Recovery Manager リファレンス』を参照してください。
- BACKUP コマンドで使用できる変数のリストは、『Oracle9i Recovery Manager リファレンス』を参照してください。

## メディア・マネージャのチャネルの割当てのテスト

Oracle にメディア管理ライブラリを統合した後、Recovery Manager がこのライブラリをロードできるかどうかをテストします。

#### メディア・マネージャ上のチャネル割当てをテストする方法：

1. Recovery Manager を起動し、ターゲット・データベースに接続します。たとえば、次のように入力します。

```
% rman TARGET /
```

2. メディア管理ソフトウェアが必要とする PARMS を指定した ALLOCATE CHANNEL コマンドを実行します。たとえば、次のコマンドを実行します。

```
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt
    PARMS='ENV=(NSR_SERVER=tape_srv,NSR_GROUP=oracle_tapes)';
}
```

エラー・メッセージが返されない場合は、共有ライブラリが正常にロードされています。ただし、チャネル割当ては ORA-27211 エラーで失敗することがあります。

```
RMAN-00571: =====
RMAN-00569:===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03009: allocate コマンド (c1 チャンネル上) が 11/30/2001 13:57:18 で失敗しました
ORA-19554: デバイスの割当て中にエラーが発生しました。デバイス型: SBT_TAPE、デバイス名:
ORA-27211: メディア管理ライブラリのロードに失敗しました
Additional information: 25
```

ORA-27211 のエラーは、インストール済みのメディア管理ライブラリを Oracle がロードできないことを示しています。この場合は、メディア管理ライブラリのインストールを調べて、ライブラリが正常にインストールされていることを確認し、8-5 ページの「[Windows NT 上での Recovery Manager とメディア・マネージャの統合: 基本手順](#)」の手順を再実行する必要があります。

ORA-27211 エラーが返されない場合は、ライブラリが正常にロードされています。その他のエラーの場合、詳細は USER\_DUMP\_DEST ディレクトリにあるトレース・ファイルを調べてください。

**関連項目:** トラブルシューティングの例については、15-27 ページの「[メディア・マネージャのインストール後に、Recovery Manager のチャンネルの割当てに失敗した場合: 例](#)」を参照してください。

### メディア・マネージャに対するバックアップのテスト

メディア・マネージャ上のチャンネル割当てをテストした後は、バックアップ・テストを実行します。たとえば、テープに正常にバックアップされることをテストするには、次のようなコマンドを実行します。

```
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt
    PARMS='ENV=(NSR_SERVER=tape_srv,NSR_GROUP=oracle_tapes)';
  BACKUP DATAFILE 1;
}
```

PARMS および FORMAT の実際の設定は、使用するメディア管理ソフトウェアに依存します。次の表は、考えられる結果および失敗した場合の対処を示します。

結果	対処
バックアップがハングする。	バックアップのハングは、通常はメディア・マネージャがテープのマウントを待機していることを示します。「テープ・マウント要求」モードになっているメディア・マネージャ・ジョブがあるかどうかを確認し、問題を修正します。  8-4 ページの「 <a href="#">Recovery Manager とメディア・マネージャの統合：基本手順</a> 」に記載の手順が正しい順序で実行されていることを確認します。問題が繰り返し発生する場合は、15-29 ページの「 <a href="#">バックアップ・ジョブが停止する場合：例</a> 」を参照してください。
ORA-19511 エラーまたは ORA-70xxx エラーでバックアップが失敗する。	このエラーは、メディア管理ソフトウェアが正しく構成されていないことを示します。8-4 ページの「 <a href="#">Recovery Manager とメディア・マネージャの統合：基本手順</a> 」に記載の手順が正しい順序で実行されていることを確認します。メディア管理ソフトウェアが必要とする PARMS および FORMAT 文字列が正しく設定されていることも確認します。
バックアップが正常に終了する。	この場合は、Recovery Manager を使用して sbt バックアップを実行できます。

**関連項目：** Recovery Manager でメディア・マネージャを使用する場合のトラブルシューティングの詳細は、15-9 ページの「[Media Management API のテスト](#)」および 15-26 ページの「[トラブルシューティング例](#)」を参照してください。

## 自動チャネルをメディア・マネージャで使用する場合の構成

この項では、メディア・マネージャで使用するための自動チャネルを構成する方法について説明します。自動チャネルの概要とその使用方法は、8-11 ページの「[自動チャネルの構成](#)」を参照してください。次の設定手順では、「[自動チャネルの構成](#)」の該当する項を参照しています。

メディア・マネージャで使用する自動チャネルの構成：

1. 8-13 ページの「[デバイス・タイプ用の一般的な自動チャネルの構成](#)」に説明されているように、DEVICE TYPE sbt の一般的なチャネルを構成します。この構成では、8-8 ページの「[メディア・マネージャに対するバックアップのテスト](#)」でテストしたすべてのパラメータを入力します。たとえば、メディア・ベンダーでは次のような PARMS 設定を必要としているとします。

```
CONFIGURE CHANNEL DEVICE TYPE sbt
PARMS='ENV=(NSR_SERVER=tape_svr, NSR_CLIENT=oracleclnt, NSR_GROUP=oracle_tapes)'
FORMAT "BACKUP_%U";
```

2. チャンネルを構成した後は、次のコマンドを実行してバックアップをテストします。

```
BACKUP DEVICE TYPE sbt DATAFILE 1;
```

3. 次のコマンドを実行して構成を調べます。

```
SHOW CHANNEL DEVICE TYPE sbt;
```

4. 8-12 ページの「[デフォルト・デバイス・タイプの変更](#)」に説明されているように、デフォルト・デバイスを `sbt` に構成します。デバイスを `sbt` に構成することにより、**Recovery Manager** ではバックアップをすべて自動的にメディア・マネージャに送信します。たとえば、次のように入力します。

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
```

5. デフォルト・デバイスを構成した後、テスト・バックアップを作成し、このバックアップが実際にメディア・マネージャに送られるかどうかを判別します。

```
BACKUP DATAFILE 1;
```

6. 次のコマンドを実行して構成を調べます。

```
SHOW DEFAULT DEVICE TYPE;
```

7. メディア・マネージャを複数使用する場合は、8-12 ページの「[自動チャンネルの平行化の構成](#)」に説明されているように、チャンネルの平行化を指定する必要があります。2つのテープ・ドライブを平行に使用して、メディア・マネージャにバックアップするとします。この場合は、次のコマンドを実行できます。

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 2;          # two channels (tape drives)
BACKUP DATABASE; # backup goes to tape but in two streams in parallel (two
tapes)
```

## 自動チャネルの構成

この項では、次の項目について説明します。

- [自動チャネルの構成 : 概要](#)
- [デフォルト・デバイス・タイプの変更](#)
- [自動チャネルの平行化の構成](#)
- [デバイス・タイプ用の一般的な自動チャネルの構成](#)
- [デバイス・タイプ用の特定のチャネルの構成](#)
- [チャネルおよびデバイス設定の消去](#)
- [制御ファイルとサーバー・パラメータ・ファイルの自動バックアップの構成](#)

**関連項目：** 自動および手動チャネルの概説は、5-2 ページの「[Recovery Manager の自動チャネルおよび手動チャネルの割当て](#)」を参照してください。構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

### 自動チャネルの構成 : 概要

チャネルのパラメータ、平行化およびデフォルト・デバイス・タイプなどの永続的な構成情報は、Recovery Manager リポジトリに保存できます。このため、バックアップごとにチャネルを手動で割り当てする必要がありません。かわりに、バックアップ、リストア、リカバリおよびメンテナンスのジョブで使用する自動チャネルを構成できます。

ALLOCATE CHANNEL を使用してチャネルを手動で割り当てると、自動チャネルはいつでもオーバーライドできます。自動チャネル機能と手動チャネル機能は相互に排他的です。Recovery Manager では各ジョブに対してこの 2 つのいずれかを使用します。

Recovery Manager にはデフォルトでディスク・チャネルが事前に構成されているため、ディスクへのバックアップを手動で構成する必要はありません。このため、メディア・マネージャにではなくディスクにバックアップする場合は、即座にディスクへのバックアップを開始できます。変更が必要になる可能性がある項目は、8-12 ページの「[自動チャネルの平行化の構成](#)」に説明されているチャネルの平行化のみです。

## デフォルト・デバイス・タイプの変更

デフォルト・デバイスとは、明示的にチャネルを割り当てていない場合に **Recovery Manager** が **BACKUP** コマンドで使用するデバイスです。たとえば、デフォルト・デバイスが **sbt**（メディア・マネージャ）の場合、**BACKUP DATABASE** コマンドは構成済みの **sbt** デバイスにバックアップします。デフォルト・デバイス・タイプが必要になる理由は、チャネルを手動で割り当てない場合に、**Recovery Manager** がどのデバイスを使用するかを認識するためです。デフォルト・デバイス・タイプを **NULL** などの未指定値に設定することはできません。

事前構成のデフォルト・デバイス・タイプは、**DISK** です。デフォルトのデバイス・タイプを **sbt** に変更するには、次のコマンドを実行する必要があります。

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
```

デフォルトを **DISK** に戻すには、次のいずれかのコマンドを実行します。

```
CONFIGURE DEFAULT DEVICE TYPE TO DISK;  
CONFIGURE DEFAULT DEVICE TYPE CLEAR; # returns to default of DISK
```

次に示す例では、デフォルトのデバイス・タイプを **sbt** に設定してから、バックアップを実行します。

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;  
BACKUP DATABASE; # backup goes to sbt
```

## 自動チャネルの平行化の構成

構成済みの各デバイスのチャネルの平行化は、デフォルトでは **1** に設定されます。指定されたデバイス・タイプ上で **Recovery Manager** が各ジョブに割り当てるチャネル数を指定するには、**CONFIGURE DEVICE TYPE ... PARALLELISM integer** コマンドを実行します。**Integer** は、**254** 以下の正の整数です。たとえば、**DISK** デバイス上で各ジョブに **3** つのチャネルを割り当てるには、次のコマンドを実行します。

```
CONFIGURE DEVICE TYPE DISK PARALLELISM 3;
```

デバイス・タイプの **PARALLELISM** は、次のコマンドを実行すると変更できます。次の例では平行化を **3** に設定し、**3** から **4** に変更し、さらに **2** に変更します。

```
CONFIGURE DEVICE TYPE DISK PARALLELISM 3;  
CONFIGURE DEVICE TYPE DISK PARALLELISM 4;  
CONFIGURE DEVICE TYPE DISK PARALLELISM 2;
```

**2** つのテープ・ドライブを平行に使用して、メディア・マネージャにバックアップするとします。次のコマンドを実行します。

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;      # make the tape drive the default backup device  
CONFIGURE DEVICE TYPE sbt PARALELLISM 2;    # configure two channels (tape drives)  
BACKUP DATABASE; # backup goes to tape but in two parallel streams (two tapes)
```

この場合は、構成済みの sbt チャネルのそれぞれが全体のデータのおよそ半分ずつをバックアップします。

関連項目： 5-10 ページ「[手動によるチャネル割当ての平行化](#)」

## デバイス・タイプ用の一般的な自動チャネルの構成

Recovery Manager は、デフォルトでは、事前構成の DISK チャネルをオプションなしで自動的に割り当てます。ただし、特殊なオプション（PARMS、FORMAT、MAXPIECESIZE など）を必要とするメディア・マネージャを使用したり、デフォルトの DISK 設定の変更が必要になる場合があります。チャネルを構成することにより、Recovery Manager で自動的にチャネルを割り当てるときにどのパラメータを使用するかを定義します。

使用可能なデバイス・タイプ（DISK および sbt）に自動チャネル・オプションを構成するには、CONFIGURE CHANNEL コマンドを使用します。CONFIGURE CHANNEL には ALLOCATE CHANNEL と同じオプションを使用できます。オプションのうちの少なくとも 1 つを指定する必要があります。たとえば、一般的なディスク・チャネルおよびテープ・チャネルは次の例のように構成できます。

```
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT = '?/bkup_%U';  
CONFIGURE CHANNEL DEVICE TYPE sbt PARMS='ENV=(NSR_SERVER=tape_svr,NSR_CLIENT=oracleclnt,  
NSR_GROUP=oracle_tapes)';
```

**一般的なチャネル**、つまりすべての平行化チャネルに使用されるテンプレートを構成する場合は、チャネルの数は割り当てないでください。デバイスに平行化を設定してからこのデバイスをデフォルトにすると、Recovery Manager では各平行化チャネルにこのチャネル構成を使用します。

指定されたデバイス・タイプに新しい一般的なチャネル設定を構成するには、そのデバイス・タイプに対して新しいコマンドを実行します。次の例では、デフォルトの DISK チャネルを MAXPIECESIZE 2G に構成し、次にこの設定を削除して FORMAT を設定します。

```
CONFIGURE CHANNEL DEVICE TYPE DISK MAXPIECESIZE 2G;  
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT = '/tmp/%U';
```

## 構成済みのチャネルとデフォルト・デバイス・タイプ

Recovery Manager がバックアップ用に割り当てる自動チャネルは、デフォルトのデバイス・タイプにより異なります。デフォルトのデバイス・タイプが DISK の場合は、DISK チャネルのみが使用されます。デフォルトのデバイス・タイプが sbt の場合は、sbt チャネルのみが使用されます。Recovery Manager では、バックアップ・ジョブで、複数のデバイス・タイプに対して同時にチャネルを自動的に割り当てることはできません。

次の例では、すべてのバックアップが 2 つのテープに平行で行われるような構成を作成します。さらに、メディア管理ソフトウェアには次の追加パラメータが必要です。ENV=(NSR\_DATA\_VOLUME\_POOL=oracle\_tapes)、および FORMAT は %U\_backup です。

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;           # by default, backup goes to media manager
CONFIGURE DEVICE TYPE sbt PARALLELISM 2;        # two tapes in parallel
CONFIGURE CHANNEL DEVICE TYPE                   # sets parameters for all channels
  PARMS 'ENV=(NSR_DATA_VOLUME_POOL=oracle_tapes)' FORMAT '%U_backup';
BACKUP DATABASE;                                # backs up database
```

### 構成済みチャネルの手動オーバーライド

ジョブの最中にチャネルを手動で割り当てると、自動チャネル設定はすべて無視されます。たとえば、デフォルト・デバイス・タイプが `sbt` に構成されているときに、次のコマンドを実行するとします。

```
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE DISK;
  BACKUP TABLESPACE users;
}
```

この場合、**Recovery Manager** では `RUN` ブロック内で手動で割り当てられたチャネルのみを使用します。`CONFIGURE DEVICE TYPE` 設定、`CONFIGURE DEFAULT DEVICE` 設定または `CONFIGURE CHANNEL` 設定は、いずれもこのジョブには適用されません。

#### 関連項目：

- 自動チャネルの詳細は、5-13 ページの「[バックアップ・セット](#)」を参照してください。
- `ALLOCATE` コマンドの構文は、『*Oracle9i Recovery Manager リファレンス*』を参照してください。
- `CONFIGURE` コマンドの構文は、『*Oracle9i Recovery Manager リファレンス*』を参照してください。



## デバイス・タイプ用の特定のチャネルの構成

デバイスに対する一般的なチャネルの構成に加えて、チャネルに独自のチャネル番号を手動で割り当てることにより、各デバイス・タイプに対して1つ以上の特定のチャネルを構成することもできます。特定のチャネルを構成するには、CONFIGURE CHANNEL *n* コマンドを実行します (*n* は 254 以下の正の整数)。チャネルに手動で番号を付けるときは、1つ以上のチャネル・オプション (MAXPIECESIZE や FORMAT など) を指定する必要があります。

チャネルの構成を変更する場合は、特定のチャネルを使用します。通常、これを行う必要があるのは、Oracle Real Application Clusters 構成を実行する場合と、異なる PARMS 設定を必要とする複数のテープ・ドライブでメディア・マネージャを使用する場合のみです。

### 特定チャネルの構成 : 例

たとえば、2 台のテープ・ドライブがあり、そのうちの 1 台は最初のテープ・プールからのテープを使用し、2 台目では 2 番目のテープ・プールからのテープを使用するとします。次のコマンドを実行します。

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;      # backup goes to sbt
CONFIGURE DEVICE TYPE sbt PARALLELISM 2; # two tapes used in parallel
# configure first stream to go to data volume pool named first_pool
CONFIGURE CHANNEL 1 DEVICE TYPE sbt PARMS 'ENV=(NSR_DATA_VOLUME_POOL=first_pool)';
# configure second stream to go to data volume pool named second_pool
CONFIGURE CHANNEL 2 DEVICE TYPE sbt PARMS 'ENV=(NSR_DATA_VOLUME_POOL=second_pool)';
BACKUP DATABASE; # first stream goes to 'first_pool' and second to 'second_pool'
```

次の例では、1 台のディスクでは領域が足りないため、2 台のディスクにバックアップします。このため、次のように実行します。

```
CONFIGURE DEFAULT DEVICE TYPE TO disk;      # backup goes to disk
CONFIGURE DEVICE TYPE sbt PARALLELISM 2;    # two channels used in parallel
CONFIGURE CHANNEL 1 DEVICE TYPE DISK FORMAT '/disk1/%U' # first channel goes to disk1
CONFIGURE CHANNEL 2 DEVICE TYPE DISK FORMAT '/disk2/%U' # second channel goes to disk2
BACKUP DATABASE; # backup - first channel goes to disk1 and second to disk2
```

### 一般的なチャネルと特定チャネルの混在

パラレル化を行う場合、Recovery Manager は常にチャネルを順番に割り当てます。この順番は CHANNEL 1 から始まりパラレル化の設定値と等しいチャネル番号で終わります。このため、Recovery Manager では特定の構成が設定されたチャネルの場合はその構成を使用し、それ以外の場合は一般的な構成を使用します。

次のチャネル構成を入力するとします。

```
# disk channel configuration
CONFIGURE DEVICE TYPE DISK PARALLELISM 4;
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT = '/tmp/backup_%U';
CONFIGURE CHANNEL 2 DEVICE TYPE DISK MAXPIECESIZE = 2M;
CONFIGURE CHANNEL 4 DEVICE TYPE DISK MAXPIECESIZE = 4M;
```

```
# sbt channel configuration
CONFIGURE DEVICE TYPE sbt PARALLELISM 3;
CONFIGURE CHANNEL DEVICE TYPE sbt PARMS='SBT_LIBRARY=oracle.disksbt,
ENV=(BACKUP_DIR=?/oradata)';
CONFIGURE CHANNEL 3 DEVICE TYPE sbt PARMS='SBT_LIBRARY=oracle.disksbt,
ENV=(BACKUP_DIR=/tmp)';
```

次の表は、デフォルト・デバイスが DISK で、DISK のパラレル化が 4 に設定されている場合に、Recovery Manager が割り当てるチャネル名とチャネル設定を示します。

チャネル名	設定
ORA_DISK_1	FORMAT = '/tmp/backup_%U'
ORA_DISK_2	MAXPIECESIZE = 2M
ORA_DISK_3	FORMAT = '/tmp/backup_%U'
ORA_DISK_4	MAXPIECESIZE = 4M

次の表は、デフォルト・デバイスが sbt で、sbt のパラレル化が 3 に設定されている場合に、Recovery Manager が割り当てるチャネル名とチャネル設定を示します。

チャネル名	設定
ORA_SBT_TAPE_1	PARMS='ENV=(BACKUP_DIR=/disk2/oracle/backup) '
ORA_SBT_TAPE_2	PARMS='ENV=(BACKUP_DIR=/disk2/oracle/backup) '
ORA_SBT_TAPE_3	PARMS='ENV=(BACKUP_DIR=/store) '

CONFIGURE CHANNEL とパラレル化設定の関係

PARALLELISM 設定は明確に構成されたチャネルの数で制限されることはありません。たとえば、20 台のテープ・デバイスにバックアップする場合は、20 個の sbt チャネルを構成できます。それぞれのチャネルに手動で番号（1 ～ 20）を割り当て、それぞれに別々のチャネル・オプションを設定できます。パラレル化は任意の値に設定できます。

Recovery Manager では常にパラレル・チャネルに番号を付けます。この番号は 1 から始まりパラレル化の設定値で終わります。たとえば、デフォルト・デバイスが sbt で、sbt のパラレル化が 3 に設定してある場合、Recovery Manager は次のようにチャネルに名前を付けます。

```
ORA_SBT_TAPE_1
ORA_SBT_TAPE_2
ORA_SBT_TAPE_3
```

DEVICE TYPE sbt を構成してある場合でも、Recovery Manager では常に ORA\_SBT\_TAPE\_*n* という名前（シノニムの sbt\_tape ではなく）を使用します。Recovery Manager では、常にパラレル化で指定されたチャネル数を割り当てます。この場合、手動チャネルが構成してある場合は手動チャネルを使用し、構成されていない場合は一般的なチャネルを使用します。

**関連項目：** 手動で番号を付けたチャネルの概要は、5-9 ページの「[特定の自動チャネルの構成](#)」を参照してください。8-15 ページの「[特定チャネルの構成: 例](#)」も参照してください。

## チャネルおよびデバイス設定の消去

構成を消去するということは、デフォルトの設定に戻すということです。チャネルおよびデバイスの設定は、次のコマンドで消去できます。

- CONFIGURE DEVICE TYPE ...CLEAR
- CONFIGURE DEFAULT DEVICE TYPE CLEAR
- CONFIGURE CHANNEL DEVICE TYPE ...CLEAR
- CONFIGURE CHANNEL *n* DEVICE TYPE ...CLEAR (*n* は整数)

それぞれの CONFIGURE ...CLEAR コマンドは、それ自体のみを消去します。たとえば、CONFIGURE DEVICE TYPE ...CLEAR は、CONFIGURE DEFAULT DEVICE TYPE を消去しません。CONFIGURE DEVICE TYPE ...CLEAR コマンドは、指定されたデバイス・タイプの構成を削除し、これをデフォルト（PARALLELISM 1）に戻します。

---

**注意：** デバイス・タイプを消去するときは、他のオプションは指定できません。

---

CONFIGURE DEFAULT DEVICE TYPE ...CLEAR コマンドは構成済みのデフォルト・デバイスを消去し、これを DISK（デフォルト設定）に戻します。

CONFIGURE CHANNEL DEVICE TYPE ...CLEAR コマンドは、指定されたデバイス・タイプのチャネル構成を消去します。PARALLELISM は別の CONFIGURE コマンドで指定されているため、Recovery Manager ではこのデバイス・タイプのパラレル化は変更しません。

自動チャネルに手動で番号を割り当てている場合は、これらのチャネルのオプションを個別に消去します。これには、CONFIGURE CHANNEL *n* DEVICE TYPE ...CLEAR にチャネル番号を指定します。たとえば、次のコマンドを実行するとします。

```
CONFIGURE CHANNEL DEVICE TYPE DISK MAXOPENFILES = 10 MAXPIECESIZE = 1800K;
CONFIGURE CHANNEL 3 DEVICE TYPE DISK FORMAT = /tmp/%U;
CONFIGURE CHANNEL 3 DEVICE TYPE DISK CLEAR;
```

この場合、Recovery Manager は CHANNEL 3 の設定を消去しますが、一般的な DISK チャネル（番号が手動で割り当てられていないチャネル）の設定はそのまま残します。

**関連項目：** 5-10 ページ「[自動チャネル設定の消去](#)」

## 制御ファイルとサーバー・パラメータ・ファイルの自動バックアップの構成

Recovery Manager では、制御ファイル内でデータベース構造のメタデータに変更があったり、バックアップまたはコピーのレコードが追加されるたびに、制御ファイルとサーバー・パラメータ・ファイルを自動的にバックアップできます。

CONFIGURE CONTROLFILE AUTOBACKUP を ON にすると（デフォルトでは OFF）、次の場合に制御ファイルと現行のサーバー・パラメータ・ファイル（使用している場合）が自動的にバックアップされます。

- Recovery Manager プロンプトで BACKUP または COPY コマンドが発行された後。
- RUN ブロック内に BACKUP または COPY コマンドがあり、後に続くコマンドが BACKUP でも COPY でもない場合。
- ブロック内の最後のコマンドが BACKUP または COPY の場合の各 RUN ブロックの最後。
- 新しい表領域の追加、表領域またはデータ・ファイルの状態の変更（オンライン化など）、新しいオンライン REDO ログの追加、ファイル名の変更、新しい REDO スレッドの追加など、データベース構造に変更があった後。前述の場合に発生する自動バックアップとは異なり、このタイプの自動バックアップはディスクにのみ作成されます。CONFIGURE CONTROLFILE AUTOBACKUP FOR DEVICE TYPE DISK を実行すると、デフォルト以外のディスク位置を設定できます。自動バックアップによって、それに関連する構造変更が失敗することはありません。たとえば、データ・ファイルを追加したために自動バックアップが失敗しても、データ・ファイルの追加は正常終了します。

---

**注意：** Oracle Enterprise Manager を使用する場合は、メンテナンス・ウィザードを使用して自動バックアップ機能を構成できます。

---

バックアップまたはコピー・ジョブで最初に割り当てられたチャネルによって自動バックアップが作成され、専用バックアップ・セットに格納され、構造の事後自動バックアップ用にデフォルトのディスク・チャネルによってバックアップが作成されます。制御ファイルとサーバー・パラメータ・ファイルは、同じバックアップ・ピースに書き込まれます。制御ファイルの自動バックアップが完了すると、バックアップ・ピースのフルパスとデバイス・タイプを含むメッセージが Oracle によってアラート・ログに書き込まれます。

次の表に示すように、BACKUP コマンドにデータ・ファイル 1 が含まれている場合の Recovery Manager の動作は、CONFIGURE CONTROLFILE AUTOBACKUP が ON であるか OFF であるかによって異なります。

CONTROLFILE AUTOBACKUP	BACKUP コマンドの動作
ON	バックアップにデータ・ファイル 1 が含まれている場合、現行の制御ファイルがデータ・ファイルのバックアップ・セットに自動的に組み込まれることはありません。かわりに、 <b>Recovery Manager</b> は制御ファイルとサーバー・パラメータ・ファイルを別の自動バックアップ・ピースに書き込みます。  <b>注意：</b> 自動バックアップは、BACKUP DATABASE INCLUDE CURRENT CONTROLFILE など、BACKUP または COPY コマンドで現行の制御ファイルが明示的に指定されているかどうかに関係なく発生します。
OFF	バックアップにデータ・ファイル 1 が含まれている場合は、現行の制御ファイルとサーバー・パラメータ・ファイルがデータ・ファイルのバックアップ・セットに自動的に組み込まれます。制御ファイルとサーバー・パラメータ・ファイルを含む別個の自動バックアップ・ピースは作成されません。

自動バックアップの目的は、現行の制御ファイル、リカバリ・カタログおよびサーバー・パラメータ・ファイルにアクセスできない場合でも、**Recovery Manager** がデータベースをリカバリできるようにすることです。自動バックアップでは一般的な書式が使用されるため、**Recovery Manager** ではリポジトリにアクセスしなくてもサーバー・パラメータ・ファイルを検索してリストアできます。

リストアされたサーバー・パラメータ・ファイルを使用してインスタンスを起動した後、**Recovery Manager** で制御ファイルを自動バックアップからリストアできます。制御ファイルのマウント後に、マウントされた制御ファイル内で **Recovery Manager** リポジトリを使用して、データ・ファイルをリストアします。

## 制御ファイルの自動バックアップの有効化と無効化

自動バックアップ機能を有効にするには、次のコマンドを実行します。

```
CONFIGURE CONTROLFILE AUTOBACKUP ON;
```

自動バックアップ機能を無効にするには、次のコマンドを実行します。

```
CONFIGURE CONTROLFILE AUTOBACKUP OFF;
```

## 制御ファイルの自動バックアップ書式の構成

デフォルトでは、構成済みのすべてのデバイスにおける自動バックアップ・ファイルの書式は置換変数 %F です。この変数の書式は、c-*IIIIIIIIII*-*YYYYMMDD*-*QQ*に変換されます。

- *IIIIIIIIII* は DBID を示します。DBID は、データベースの一意の数値 ID です。DBID は十進数で出力されるため、ターゲット・データベースへの対応付けが容易です。
- *YYYYMMDD* は、バックアップの生成日をグレゴリオ暦で表したタイムスタンプです。
- *QQ* は、00 で始まる 16 進数で表した順番で、最大値は FF (256) です。

デフォルトの書式は次のコマンドを使用して変更できます。*deviceSpecifier* は DISK や sbt などの有効なデバイスで、'*string*' には変数 %F が含まれ、指定されたデバイスの有効なハンドルを示します。

```
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE deviceSpecifier TO 'string';
```

たとえば、次のコマンドを実行できます。

```
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '?/oradata/cf_%F';
```

---

**注意：** 書式文字列には %F 変数を含める必要があります。書式文字列に %F 変数が含まれていないと、エラーが発生します。

---

次のコマンドを実行すると、デフォルトの自動バックアップ書式を %F に戻すことができます。*deviceSpecifier* は DISK や sbt などの有効なデバイスです。

```
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE deviceSpecifier CLEAR;
```

RUN ブロック内または Recovery Manager プロンプトに指定できる SET CONTROLFILE AUTOBACKUP FORMAT コマンドは、現在のセッションのみで構成済みの自動バックアップ書式をオーバーライドできます。優先順位は次のとおりです。

1. SET CONTROLFILE AUTOBACKUP FORMAT (RUN 内)
2. SET CONTROLFILE AUTOBACKUP FORMAT (Recovery Manager プロンプト)
3. CONFIGURE CONTROLFILE AUTOBACKUP FORMAT

### 関連項目：

- 概説は、5-52 ページの「[制御ファイルとサーバー・パラメータ・ファイルの自動バックアップ](#)」を参照してください。
- CONFIGURE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- SET コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## バックアップの保存方針の構成

保存方針は、Recovery Manager で作成するバックアップおよびコピーをいつ不要とみなすかを指定します。方針を構成する前に、5-55 ページの「バックアップの保存方針」に記載の概要説明をお読みください。

CONFIGURE コマンドを使用して保存方針を設定します。Oracle Enterprise Manager を使用すると、メンテナンス・ウィザードで方針を設定することもできます。

この項では、次の項目について説明します。

- [リカバリ・ウィンドウのための保存方針の構成](#)
- [冗長性のための保存方針の構成](#)
- [保存方針の無効化](#)
- [保存方針をデフォルト設定に戻す方法](#)

### 関連項目：

- CONFIGURE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- Recovery Manager のリストアおよびリカバリの詳細は、『Oracle Enterprise Manager 管理者ガイド』を参照してください。

## リカバリ・ウィンドウのための保存方針の構成

CONFIGURE コマンドの RECOVERY WINDOW パラメータは、現時点から最初のリカバリ可能ポイントまでの日数を指定します。Recovery Manager では、このリカバリ・ウィンドウ内にあるバックアップまたはコピーは不要とはみなしません。さらに、期間内の任意の時点にリカバリするために必要なアーカイブ・ログおよび増分バックアップをすべて保持します。

Recovery Manager プロンプトで CONFIGURE コマンドを実行します。次の例は、最近 7 日間の任意の時点までデータベースをリカバリできるようにします。

```
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;
```

Recovery Manager では、リカバリ・ウィンドウにより不要とみなされたバックアップおよびコピーを自動的に削除しません。かわりに、これらを REPORT OBSOLETE 出力に OBSOLETE として表示します。DELETE OBSOLETE を実行するとこれらが削除されます。

保存方針の設定を変更するには、新しい設定を指定して CONFIGURE RETENTION POLICY コマンドを実行します。たとえば、次のように入力します。

```
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 4 DAYS;  
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 11 DAYS;
```

**関連項目：** [リカバリ・ウィンドウの概要](#)は、5-56 ページの「[リカバリ・ウィンドウ](#)」を参照してください。

## 冗長性のための保存方針の構成

CONFIGURE RETENTION POLICY コマンドの REDUNDANCY パラメータは、各データ・ファイルおよび制御ファイルに対してバックアップおよびコピーを Recovery Manager でいくつ保持するかを指定します。つまり、特定のデータ・ファイルまたは制御ファイルのバックアップおよびコピーの数が、REDUNDANCY 設定を超過すると、Recovery Manager では超過したバックアップおよびコピーを不要とみなします。

バックアップを作成するたびに、Recovery Manager ではどのファイルを保持しどれが不要かを追跡します。さらに、不要ではないバックアップのリカバリに必要なアーカイブ・ログおよび増分バックアップをすべて保持します。

特定のデータ・ファイルを月曜日、火曜日、水曜日および木曜日にバックアップするとします。このため、このデータ・ファイルにはバックアップが 4 つあります。REDUNDANCY が 2 の場合は、月曜日と火曜日のバックアップが不要になります。金曜日にもう 1 つバックアップを作成した場合は、水曜日のバックアップが不要になります。

次の例のように、CONFIGURE RETENTION POLICY コマンドを Recovery Manager プロンプトで実行します。

```
CONFIGURE RETENTION POLICY TO REDUNDANCY 3;
```

設定を変更するには、新しい REDUNDANCY 設定を指定してこのコマンドを実行します。たとえば、次のように実行します。

```
CONFIGURE RETENTION POLICY TO REDUNDANCY 4;  
CONFIGURE RETENTION POLICY TO REDUNDANCY 2;
```

**関連項目：** バックアップ冗長性の概要は、5-59 ページの「[バックアップ冗長性](#)」を参照してください。

## 保存方針の無効化

保存方針を無効にするには、次のコマンドを実行します。

```
CONFIGURE RETENTION POLICY TO NONE;
```

このコマンドは、Recovery Manager がどのバックアップまたはコピーも不要と判断しないことを意味します。この結果、他のオプションを指定しないで REPORT OBSOLETE または DELETE OBSOLETE を実行するとエラーが発生します。これはどのバックアップおよびコピーが不要かを判断する保存方針がないためです。

---

---

**注意：** 保存方針を NONE に構成することと、保存方針を消去することは同じではありません。保存方針を消去するとデフォルト設定に戻りますが、NONE を指定すると保存方針が完全に無効になります。

---

---



## 保存方針をデフォルト設定に戻す方法

保存方針を消去する（保存方針をデフォルト設定に戻す）には、次のコマンドを実行します。

```
CONFIGURE RETENTION POLICY CLEAR;
```

このコマンドは、保存方針をデフォルト設定（REDUNDANCY = 1）に戻します。

## バックアップ・セットの最大サイズの設定

CONFIGURE MAXSETSIZE コマンドは、チャンネルで作成されるバックアップ・セットのサイズを制限します。手動で割り当てられたか自動的に割り当てられたかにかかわらず、BACKUP コマンドの実行時にすべてのチャンネルに対して CONFIGURE 設定が適用されます。

MAXSETSIZE は、バイト単位（デフォルト）、KB 単位（K）、MB 単位（M）および GB 単位（G）で設定できます。デフォルト値はバイト単位で表され、最も近い最小 KB 値に切り下げられます。たとえば、最大セット・サイズを 2000 に設定した場合、この値は 1KB（1024 バイト）に切り下げられます。最大セット・サイズを 2049 に設定した場合、この値は 2KB（2048 バイト）に切り下げられます。

MAXSETSIZE は次の 2 つのコマンドに指定できます。コマンドは優先順位の高い方が先に示されています（高い方が低い方をオーバーライドします）。

1. BACKUP
2. CONFIGURE

次の例には設定の階層が示されています。次の一連のコマンドを、Recovery Manager プロンプトから発行するとします。

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE CHANNEL DEVICE TYPE sbt PARMS 'ENV=(NSR_DATA_VOLUME_POOL=first_pool)';
CONFIGURE MAXSETSIZE TO 7500K;
BACKUP TABLESPACE users;
BACKUP TABLESPACE tools MAXSETSIZE 10M;
```

この結果、Recovery Manager の動作は次のようになります。

- hr 表領域のバックアップには、自動 sbt チャンネルと MAXSETSIZE のデフォルト設定の 7500K が使用されます。
- orders 表領域のバックアップには、BACKUP コマンドで使用されている MAXSETSIZE 設定の 10M が使用されます。

**関連項目：** BACKUP コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## バックアップの最適化の構成

バックアップの最適化を有効および無効にするには、CONFIGURE コマンドを実行します。バックアップの最適化では、まったく同一のファイルまたはファイルのまったく同一のバージョンがすでにバックアップ済みの場合、ファイルのバックアップをスキップします。最適化を構成する前に、5-62 ページの「[バックアップの最適化](#)」に記載されている、バックアップの最適化アルゴリズムの説明をお読みください。

バックアップの最適化は、次に示すコマンドにのみ適用されます。

- BACKUP DATABASE
- BACKUP ARCHIVELOG (ALL オプションまたは LIKE オプションを指定したもの)
- BACKUP BACKUPSET ALL

BACKUP コマンドに FORCE オプションを指定することで、最適化はいつでもオーバーライドできます。たとえば、次のように実行します。

```
BACKUP DATABASE FORCE;  
BACKUP ARCHIVELOG ALL FORCE;
```

バックアップの最適化は、デフォルトでは OFF に構成されています。バックアップの最適化を有効にするには、次のコマンドを実行します。

```
CONFIGURE BACKUP OPTIMIZATION ON;
```

バックアップの最適化を無効にするには、次のコマンドを実行します。

```
CONFIGURE BACKUP OPTIMIZATION OFF;
```

バックアップの最適化の現在の設定を消去する（バックアップの最適化をデフォルト設定の OFF に戻す）には、次のコマンドを実行します。

```
CONFIGURE BACKUP OPTIMIZATION CLEAR;
```

---

---

**注意：** Oracle Enterprise Manager を使用する場合は、メンテナンス・ウィザードを使用してバックアップの最適化を構成できます。

---

---

### 関連項目：

- ファイルがまったく同一かどうかの判断基準およびバックアップの最適化が実行される条件の詳細は、5-62 ページの「[バックアップの最適化](#)」を参照してください。
- Recovery Manager のバックアップの最適化の例は、「[バックアップの最適化を使用した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・コピー数の設定

指定したデバイス・タイプ上で、指定したファイル・タイプの各バックアップ・ピースのコピーをいくつ作成するかを指定するには、CONFIGURE ... BACKUP COPIES コマンドを使用します。この機能は、**二重化**とも呼ばれます。CONFIGURE 設定は、データ・ファイル（現行の制御ファイルを含む）とアーカイブ REDO ログのバックアップにのみ適用されます。

---

**注意：** ディスクでの制御ファイルの自動バックアップは、特殊ケースであり、二重化されません。Recovery Manager では、常にコピーが 1 つのみ作成されます。

---

sbt にバックアップする場合は、初期化パラメータ BACKUP\_TAPE\_IO\_SLAVES=true を設定する必要があります。設定しない場合、次のエラーが発生します。

```
RMAN-10035: 例外が RPC で発生しました :ORA-19565: シーケンシャル・デバイスへのコピー時は BACKUP_TAPE_IO_SLAVES は使用できません。
```

バックアップ・コピー数を構成するには、整数を指定します。次の例は、可能な構成を示します。

```
# Makes 2 copies of every datafile and control file backup (autobackups excluded) to disk
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 2;
# Makes 3 copies of every archived redo log backup to tape
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE sbt TO 3;
```

複数の FORMAT 文字列と二重化機能と一緒に使用する場合は、個々のバックアップ・コピーに名前を付けられます。たとえば、BACKUP COPIES を 3 に構成するとします。次のように発行できます。

```
BACKUP DATABASE FORMAT '/tmp/%U', '?/dbs/%U', '?/oradata/%U';
```

Recovery Manager はバックアップ・セット内の各バックアップ・ピースにまったく同じコピーを 3 つ生成し、指定された FORMAT 文字列に従ってそれぞれのピースに名前を付けます。最初のコピーを /tmp ディレクトリに配置し、2 番目を ?/dbs ディレクトリに、3 番目を ?/oradata ディレクトリに配置します。FORMAT 文字列は、BACKUP コマンド、CONFIGURE CHANNEL コマンドおよび ALLOCATE CHANNEL コマンドに指定できます。

BACKUP COPIES 構成をデフォルト値に戻すには、次の例のように、CLEAR オプションを指定して同じ CONFIGURE コマンドを実行します。

```
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE sbt CLEAR;
```

CONFIGURE ... BACKUP COPIES は、デフォルトでは各デバイス・タイプに対して 1 に設定されています。

---

**注意：** 永続的なコピーの構成を作成しない場合は、BACKUP COPIES および SET BACKUP COPIES コマンドでコピーを指定できます。

---

### 関連項目：

- 概要については、5-23 ページの「[二重バックアップ・セット](#)」を参照してください。
- BACKUP コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- CONFIGURE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- SET コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## データベース全体のバックアップから除外する表領域の構成

指定した表領域を BACKUP DATABASE コマンドの対象から除外するには、CONFIGURE EXCLUDE FOR TABLESPACE を指定できます。除外条件は、この表領域に対して後で追加するデータ・ファイルにもすべて適用されます。

この表領域除外機能は、次の場合のように、特定の表領域を定期的なバックアップ・スケジュールに含めない場合に役立ちます。

- 表領域が簡単に再作成できるため、毎日バックアップするよりも再作成した方が費用対効果が高い場合。
- バックアップする必要のない一時データまたはテスト・データが表領域に含まれている場合。
- 表領域が頻繁には変更されないため、他のバックアップとは別のスケジュールでバックアップする場合。

たとえば、次のように、テスト表領域の `cwmlite` および `example` をデータベース全体のバックアップから除外できます。

```
CONFIGURE EXCLUDE FOR TABLESPACE cwmlite;  
CONFIGURE EXCLUDE FOR TABLESPACE example;
```

次のコマンドを実行すると、データベースの中で `cwmlite` および `example` 以外のすべての表領域がバックアップされます。

```
BACKUP DATABASE;
```

構成済みの表領域でもバックアップできます。これには、BACKUP コマンドに表領域を明示的に指定するか、BACKUP DATABASE コマンドに NOEXCLUDE オプションを指定します。たとえば、次のコマンドのいずれかを入力します。

```
BACKUP DATABASE NOEXCLUDE;      # backs up the whole database, including cwmllite and example
BACKUP TABLESPACE cwmllite, example; # backs up only cwmllite and example
```

cwmllite および example を除外する機能を、次のように無効化できます。

```
CONFIGURE EXCLUDE FOR TABLESPACE cwmllite CLEAR;
CONFIGURE EXCLUDE FOR TABLESPACE example CLEAR;
```

これ以降のデータベース全体のバックアップには、これらの表領域が含まれます。

---

---

**注意：** Oracle Enterprise Manager を使用する場合は、メンテナンス・ウィザードを使用して表領域をバックアップから除外できます。

---

---

**関連項目：**

- BACKUP コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- CONFIGURE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- メンテナンス・ウィザードの詳細は、『Oracle Enterprise Manager 管理者ガイド』を参照してください。

## Recovery Manager のグローバリゼーション・サポートの環境変数の設定

この項では、Recovery Manager でのグローバリゼーション・サポート変数の設定に関して、次の項目を説明します。

- [NLS\\_DATE\\_FORMAT および NLS\\_LANG の設定](#)
- [データベース・キャラクタ・セットの指定](#)

### NLS\_DATE\_FORMAT および NLS\_LANG の設定

Recovery Manager を起動する前に、環境変数 `NLS_DATE_FORMAT` と環境変数 `NLS_LANG` を設定します。これらの変数によって、`RESTORE`、`RECOVER`、`REPORT` などの、Recovery Manager コマンドの時間を指定するパラメータで使用す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'
```

### データベース・キャラクタ・セットの指定

Recovery Manager を使用してマウント状態でないデータベースに接続し、Recovery Manager 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に `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・セットの `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、Recovery Manager 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に `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できます。

```
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` および `NLS_DATE_FORMAT` パラメータの詳細は、『Oracle9i データベース・リファレンス』を参照してください。『Oracle9i Database グローバリゼーション・サポート・ガイド』も参照してください。

# スナップショット制御ファイルの場所の設定

Recovery Manager では、制御ファイルの読み込み一貫性バージョンからの再同期化が必要な場合、一時的なスナップショット制御ファイルを作成します。Recovery Manager でスナップショット制御ファイルが必要になるのは、リカバリ・カタログと再同期化する場合か、現行の制御ファイルのバックアップを作成する場合のみです。

スナップショット制御ファイルのデフォルト値はプラットフォーム固有で、Oracle ホームの場所に依存します。たとえば、Oracle9i では、ある UNIX プラットフォームの場合、デフォルトのファイル名が \$ORACLE\_HOME/dbs/snapcf\_@.f になります。

通常、制御ファイルの場所の設定が必要になるのは、8.1.7 より前のリリースから現行リリースにアップグレードする場合のみです。このような以前のリリースでは、スナップショット制御ファイルのデフォルトの場所は Oracle ホームに依存していましたが、現行のリリースではデフォルトの場所は Oracle ホームに依存します。

## スナップショット制御ファイルの場所をデフォルト値に設定する方法

場合によっては、スナップショット制御ファイルをデフォルトの場所に設定することがあります。CONFIGURE SNAPSHOT CONTROLFILE LOCATION CLEAR コマンドを実行します。デフォルト値の動作を次の表に示します。

状況	動作
現行リリースでデータベースを作成します。	スナップショット制御ファイルの場所には、DEFAULT 値が使用されます。この場合、Oracle ホームを変更すると、スナップショット制御ファイルのデフォルトの場所が変更されます。
8.1.7 より前のリリースから現行リリースにアップグレードします。	スナップショット制御ファイルの場所は、DEFAULT 値には設定されません。かわりに、Recovery Manager では制御ファイルにすでに格納されているスナップショットの場所を使用します。この場合、Oracle ホームを変更しても、スナップショット制御ファイルのデフォルトの場所は変更されません。スナップショット制御ファイルの場所をデフォルト値に設定するには、CONFIGURE SNAPSHOT CONTROLFILE NAME CLEAR を実行します。

スナップショットの現在の場所は、SHOW コマンドで表示できます。次の例では、スナップショットのデフォルトの場所が表示されます。

```
RMAN> SHOW SNAPSHOT CONTROLFILE NAME;  
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/oracle/dbs/snapcf_trgt.f'; # default
```

次の例では、デフォルト以外のファイル名を持つスナップショット制御ファイルが表示されます。

```
RMAN> SHOW SNAPSHOT CONTROLFILE NAME;  
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/tmp/snapcf_trgt.f'; # default
```

## スナップショット制御ファイルの場所をデフォルト以外の値に設定する方法

スナップショット制御ファイルの名前をデフォルト以外の値に変更するには、`CONFIGURE SNAPSHOT CONTROLFILE NAME TO 'filename'` コマンドを使用します。この後、**Recovery Manager** が作成するスナップショット制御ファイルには、指定されたファイル名が使用されます。

たとえば、**Recovery Manager** を起動し次のように入力します。

```
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/oracle/oradata/trgt/snap_trgt.ctl';
```

スナップショット制御ファイル名を RAW デバイスに設定することもできます。**Oracle Real Application Clusters** 構成では、全インスタンスでスナップショット制御ファイルを共有する必要はありませんが、任意のインスタンスがスナップショット制御ファイルを作成できる場所にスナップショット制御ファイル名を設定する必要があります。各インスタンスでスナップショット制御ファイルが必要な場合は、各インスタンス専用のファイル・システム内にスナップショット制御ファイルを作成できます。次の例では、スナップショット制御ファイル名を RAW デバイスに設定します。

```
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/dev/vgd_1_0/rlvt5';
```

ある **Recovery Manager** 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ることに注意してください。

```
waiting for snapshot controlfile enqueue
```

通常、ジョブが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：** `CONFIGURE` コマンドの構文は、15-35 ページの「[制御ファイルのエンキューが原因でバックアップが失敗する場合：例](#)」および『[Oracle9i Recovery Manager リファレンス](#)』を参照してください。



## Recovery Manager を共有サーバーで使用する場合の設定

Recovery Manager では共有サーバー・ディスパッチャを使用してターゲット・データベースに接続することはできません。専用サーバー・プロセスが必要です。ただし、ターゲットが共有サーバー用に構成されている場合でも、指定したセッションを専用サーバーに接続できます。

ターゲット・データベースが共有サーバー用に構成されている場合に Recovery Manager がディスパッチャに接続しないようにするには、Recovery Manager で使用するネット・サービス名は、接続文字列の CONNECT\_DATA 属性に (SERVER=DEDICATED) を含んでいる必要があります。

Oracle Net 構成は、システムによって大きく異なります。次の手順は、方法の 1 つを説明しています。この使用例では、tnsnames.ora ファイルの次のサービス名が、共有サーバー・アーキテクチャを使用してターゲット・データベースに接続されることが前提です。inst1 は、SERVICE\_NAMES 初期化パラメータの値です。

```
inst1_shs =
  (DESCRIPTION=
    (ADDRESS= (PROTOCOL=tcp) (HOST=inst1_host) (port1521))
    (CONNECT_DATA= (SERVICE_NAME=inst1) (SERVER=shared))
  )
```

**Recovery Manager を共有サーバーとともに使用するには、次のようにします。**

1. 非共有 SID に接続するネット・サービス名を tnsnames.ora ファイルに作成します。たとえば、次のように入力します。

```
inst1_ded =
  (DESCRIPTION=
    (ADDRESS= (PROTOCOL=tcp) (HOST=inst1_host) (port1521))
    (CONNECT_DATA= (SERVICE_NAME=inst1) (SERVER=dedicated))
  )
```

2. SQL\*Plus を起動し、共有サーバーと専用サーバーのサービス名を両方とも使用して接続し、各セッションのモードを確認します。たとえば、専用セッションに接続するには、次のように発行します。

```
SQL> CONNECT SYS/oracle@inst1_ded
Connected.
SQL> SELECT SERVER FROM V$SESSION WHERE SID = (SELECT DISTINCT SID FROM
V$MYSTAT);

SERVER
-----
DEDICATED
1 row selected.
```

共有サーバー・セッションに接続するには、次のコマンドを発行します。

```
SQL> CONNECT SYS/oracle@inst1_shs AS SYSDBA
Connected.
SQL> SELECT SERVER FROM V$SESSION WHERE SID = (SELECT DISTINCT SID FROM
V$MYSTAT);

SERVER
-----
SHARED
1 row selected.
```

3. 専用サービス名を使用して、ターゲット・データベースに（必要に応じてリカバリ・カタログにも）接続します。たとえば、次のように入力します。

```
% rman TARGET SYS/oracle@inst1_ded CATALOG rman/cat@catdb
```

**関連項目：** Oracle Net 接続文字列の構文について、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 ドキュメントおよび『Oracle9i Net Services リファレンス・ガイド』を参照してください。

## リカバリ・カタログの設定

Recovery Manager を使用する場合は、リカバリ・カタログは不要です。リカバリ・カタログを使用するには、リカバリ・カタログ・データベースにスキーマを作成する必要があります。カタログは、スキーマのデフォルト表領域に置かれます。ただし、SYS はリカバリ・カタログ所有者にはなれないので、注意してください。

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ホストの別のディスク上の別のデータベースに作成してください。そうしないと、データベースを消失してリストアが必要なときに、リカバリ・カタログを使用する利点がありません。

---

---

**注意：** リカバリ・カタログとターゲット・データベースが、同じディスク上に配置され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場合、データベースを消失した場合、リカバリ・カタログも失うことになります。

---

---

リカバリ・カタログを設定する基本手順は、次のとおりです。詳細は第 16 章「[Recovery Manager のリポジトリの管理](#)」に説明されています。

1. リカバリ・カタログ・スキーマを所有するユーザーをリカバリ・カタログ・データベースに作成します。たとえば、SQL\*Plus を起動して、次を実行します。

```
-- connect as SYS to recovery catalog database
CONNECT SYS/oracle@catdb AS SYSDBA
```

```
-- create user that will own catalog tables
CREATE USER rman IDENTIFIED by cat
  TEMPORARY TABLESPACE temp
  DEFAULT TABLESPACE tools QUOTA UNLIMITED ON tools;
```

2. リカバリ・カタログの所有者の権限と、他の必要な権限を付与します。たとえば、SQL\*Plus で次のコマンドを実行します。

```
GRANT RECOVERY_CATALOG_OWNER TO rman;
GRANT CONNECT, RESOURCE TO rman;
```

3. Recovery Manager を起動し、CREATE CATALOG コマンドを使用してリカバリ・カタログ・スキーマ自体を作成します。たとえば、次のように入力します。

```
% rman TARGET / CATALOG rman/cat@catdb
RMAN> CREATE CATALOG
```

4. REGISTER DATABASE コマンドを使用して、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。たとえば、次のように入力します。

```
RMAN> REGISTER DATABASE;
```

**関連項目：** 16-2 ページの「[リカバリ・カタログの作成](#)」および 16-6 ページの「[データベースのリカバリ・カタログへの登録](#)」を参照してください。



---

# 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 を使用したバックアップとコピー操作の管理方法について説明します。この章の項目は、次のとおりです。

- バックアップ・ジョブおよびコピー・ジョブで使用するチャネルの構成および割当て
- Recovery Manager を使用したデータベース・ファイルおよびアーカイブ・ログのバックアップ
- 二重化バックアップ
- Recovery Manager を使用した増分バックアップの作成
- Recovery Manager を使用したミラーの分割によるバックアップの作成
- Recovery Manager を使用したスタンバイ・データベース・サイト上のファイルのバックアップ
- Recovery Manager を使用したバックアップ・セットのバックアップ
- Recovery Manager によるバックアップの再起動および最適化
- Recovery Manager を使用したバックアップの妥当性チェックの実行
- Recovery Manager を使用したファイルのコピー
- 制御ファイルの自動バックアップ書式の上書き
- Recovery Manager のバックアップおよびコピーの例

## バックアップ・ジョブおよびコピー・ジョブで使用するチャネルの構成および割当て

バックアップおよびコピー・ジョブの実行用に、次のような相互の排他的オプションがあります。

- CONFIGURE コマンドを使用して自動チャネルを構成し、BACKUP および COPY コマンドを Recovery Manager プロンプトから、または RUN ブロック内で発行します。
- 手動でチャネルを割り当て、BACKUP および COPY コマンドを RUN ブロック内で発行します。

バックアップを作成する最も簡単な方法は、自動チャネルを構成することです。たとえば、sbt デバイス・タイプをすでに構成済みの場合にかぎり、デフォルトの sbt チャネルを次のように構成できます (PARMS 値はベンダー固有です。注意してください)。

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 1;
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE CHANNEL DEVICE TYPE sbt PARMS 'ENV=(NSR_SERVER=bksvr1)';
```

次のように、Recovery Manager プロンプトから、データベースをテープにバックアップできます。

```
BACKUP DATABASE;
```

Recovery Manager は DISK チャネルを事前構成します。これにより、自動チャネルを使用し、他の構成方法は一切実行することなく、ディスク上にバックアップを作成できます。

その他は、RUN ジョブ内で、チャネルを手動で割り当てる方法です。たとえば、このコマンドは複数のディスク・チャネルを割り当て、データベースとアーカイブ REDO ログをバックアップします。

```
RUN
{
  ALLOCATE CHANNEL ch1 DEVICE TYPE DISK;
  ALLOCATE CHANNEL ch2 DEVICE TYPE DISK;
  ALLOCATE CHANNEL ch3 DEVICE TYPE DISK;
  BACKUP DATABASE PLUS ARCHIVELOG;
}
```

この例では、sbt チャネルを手動で割り当て (ベンダー固有の PARMS 値を使用して)、データ・ファイルのコピーをバックアップします。

```
RUN
{
  ALLOCATE CHANNEL ch1 DEVICE TYPE sbt PARMS 'ENV=(NSR_SERVER=bksvr1)';
  BACKUP DATAFILECOPY '/tmp/system01.dbf';
}
```

この章にある手順の大部分は、自動チャネルがすでに構成されているものと想定しています。

## Recovery Manager を使用したデータベース・ファイルおよびアーカイブ・ログのバックアップ

この項では、次の項目について説明します。

- [Recovery Manager のバックアップ](#)
- [Recovery Manager を使用した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](#)
- [Recovery Manager を使用したデータベース全体のバックアップ](#)
- [Recovery Manager を使用した表領域のバックアップ](#)
- [Recovery Manager を使用したデータ・ファイルおよびデータ・ファイルのコピーのバックアップ](#)
- [Recovery Manager を使用した制御ファイルのバックアップ](#)
- [二重化バックアップ](#)
- [Recovery Manager を使用したアーカイブ REDO ログのバックアップ](#)

### 関連項目：

- [Recovery Manager のバックアップの概要は、5-13 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、『Oracle9i Recovery Manager リファレンス』を参照してください。](#)
- [Recovery Manager のバックアップの各タイプの説明は、5-41 ページの「バックアップ・タイプ」を参照してください。](#)

## Recovery Manager のバックアップ

データベースがマウントされている、またはオープンされているときのどちらでも、Recovery Manager の BACKUP コマンドを使用して、次のオブジェクトをバックアップします。

- プライマリあるいはスタンバイ・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（カレントまたはイメージ・コピー）
- アーカイブ REDO ログ
- 制御ファイル（カレントまたはイメージ・コピー）
- サーバー・パラメータ・ファイル（現在使用中のファイルのみ）
- バックアップ・セット

BACKUP コマンドは、データ・ファイル、アーカイブ REDO ログ・ファイル、制御ファイルおよびサーバー・パラメータ・ファイルに使用しますが、本番の操作においてはデータベースは、他のファイルにも依存します。オペレーティング・システム上にある、他の重要なファイルも、サード・パーティーのバックアップ製品を使用して、バックアップしておく必要があります。オペレーティング・システムの実行に必要なファイルも、ネットワーキング・ファイル、パスワード・ファイルおよび Oracle ホームにあるファイルなどと同様に、バックアップしておく必要があります。

BACKUP コマンドは、ディスクまたはテープ上の、1 つ以上のバックアップ・セットにデータベース・ファイルをバックアップします。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、および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。

Recovery Manager のバックアップは、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方法によって異なります。Recovery Manager のバックアップは、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。つまり、使用したブロックが、すべてバックアップされます。



## Oracle Enterprise Manager での Recovery Manager を使用したバックアップ

Oracle Enterprise Manager を使用している場合、コマンドライン・インタフェースのかわりにバックアップ・ウィザードを使用できます。バックアップ・ウィザードを介して、次の Recovery Manager タスクを実行できます。

- **事前定義済みの方針を使用したデータベースのバックアップ。** 少数の意思決定を行うのみでデータベース全体のバックアップができます。
- **カスタマイズされた方針によるデータベースのバックアップ。** 全体バックアップまたは増分バックアップ、オンライン・バックアップまたはオフライン・バックアップを選択できます。また、データベースに設定されているバックアップ方針と保存方針をオーバーライドするか、または処理中にバックアップとコピーを廃止にするかを選択できます。
- **表領域やデータ・ファイルの個別ファイルのバックアップ。** バックアップ後に廃止になったバックアップを削除するかどうか、全体バックアップと増分バックアップのどちらを実行するか、オンライン・モードとオフライン・モードのどちらで実行するか、データベースに設定されているバックアップ方針と保存方針をオーバーライドするかどうかなど、各種のオプションを選択できます。
- **アーカイブ・ログのバックアップ。** バックアップ対象範囲に含まれる最初と最後のアーカイブ・ログの日時と、バックアップ完了後に入力ログを（プライマリのアーカイブ先からのみ）自動的に削除するかどうかを指定します。
- **データ・ファイルのコピー。** 後でリカバリに使用します。
- **現行のバックアップ方針と保存方針の設定の表示**（ターゲット・データベース内）。

Oracle Enterprise Manager のバックアップ構成を作成するプロパティ・シートを使用すると、既存のバックアップ構成を編集したり、他のバックアップ構成を作成できます。バックアップ構成は、Enterprise Manager によってターゲット・データベースごとに自動的に作成されます。バックアップ構成では、次のオプションを指定できます。

- バックアップ・セットをディスクにバックアップする構成を作成します。
- バックアップ・セットをテープにバックアップする構成を作成します。
- イメージ・コピー形式でバックアップする構成を作成します。
- バックアップやコピーを行なう際の制限を設定します。
- Recovery Manager のかわりにメディア管理ソフトウェアを使用してバックアップやリカバリを行うための設定を指定します。
- Recovery Manager によってデータ・ファイルまたは ARCHIVELOG について計算されたデフォルト設定を上書きする記憶域パラメータを設定します。
- リカバリ・カタログにデータベースを登録します。Enterprise Manager のリポジトリをローカル・データベースに格納するように指定すると、リカバリ・カタログが作成されます。

**関連項目：**

- Recovery Manager のバックアップの概要は、5-13 ページの「[バックアップ・セット](#)」を参照してください。
- Recovery Manager のバックアップの各タイプの説明は、5-41 ページの「[バックアップ・タイプ](#)」を参照してください。
- BACKUP コマンドの構文について、完全な説明は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- Recovery Manager でのバックアップ作成方法の詳細は、『Oracle Enterprise Manager 管理者ガイド』を参照してください。

## Recovery Manager を使用した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

一貫性バックアップは、リカバリを実行することなくリストアできます。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状態である必要があります。

- マウントされているがオープン状態ではない。
- 前回オープンされてからインスタンス障害が発生していないか、SHUTDOWN ABORT でクローズされていない。

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。非一貫性バックアップは、リストア時にメディア・リカバリが必要となりますが、他の点では、一貫性バックアップと同様に有効です。

データベースの起動および停止に、SQL\*Plus あるいは Recovery Manager を使用できます。次の例では、ターゲット・データベースに接続を行い、これを正しく停止し、バックアップに備えてマウントします。

```
% rman TARGET /  
RMAN> SHUTDOWN IMMEDIATE # closes database consistently  
RMAN> STARTUP MOUNT # uses SPFILE
```

## Recovery Manager を使用したデータベース全体のバックアップ

プライマリ・データベースをクローズする余裕がある場合は、クローズしてデータベース全体の一貫性バックアップを作成します。データベースを停止できない場合は、データベースがオープン状態の間にバックアップを作成することのみが可能です。

**関連項目：** スタンバイ・データベースのバックアップの詳細は、『Oracle9i Data Guard 概要および管理』を参照してください。

### データベース全体のバックアップ作成

1. Recovery Manager の起動後、BACKUP DATABASE コマンドを Recovery Manager プロンプトから実行します。この例では、全データ・ファイルと制御ファイルおよびサバー・パラメータ・ファイル（使用している場合）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、Recovery Manager 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（UNIX の場合は \$ORACLE\_HOME/dbs）に格納します。

```
BACKUP DATABASE; # uses automatic channels to make backup
SQL 'ALTER SYSTEM ARCHIVE LOG CURRENT'; # switches logs and archives all 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。

```
BACKUP DATABASE FORMAT '/tmp/%U'; # %U generates a unique filename
```

必要に応じて、TAG パラメータを使用して、バックアップのタグを指定します。たとえば、次のように入力します。

```
BACKUP DATABASE TAG = 'weekly_backup'; # gives the backup a tag identifier
```

Recovery Manager がバックアップにデフォルト・タグを割り当てることに注意してください。デフォルトの書式については、『Oracle9i Recovery Manager リファレンス』の BACKUP ... TAG の説明を参照してください。

2. LIST BACKUP OF DATABASE コマンドを発行して、バックアップ・セットとピースのリストを表示します。

## Recovery Manager を使用した表領域のバックアップ

表領域のバックアップは、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。オープン状態のデータベースのバックアップは、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。オンライン表領域バックアップの作成前には、ALTER DATABASE BEGIN BACKUP を発行しないでください。

### 表領域のバックアップ

1. Recovery Manager の起動後、BACKUP TABLESPACE コマンドを Recovery Manager プロンプトから実行します。この例では、3 つの表領域をバックアップします。その際、FILESERSET パラメータを使用して、各バックアップ・セットに 3 つ以上のデータ・ファイルが組み込まれないように指定しています。

```
BACKUP FILESPERSET = 3 TABLESPACE system, users, tools;
```

2. LIST BACKUP OF TABLESPACE コマンドを発行して、バックアップ・セットおよびピースのリストを表示します。たとえば、次のように入力します。

```
LIST BACKUP OF system, users, tools;
```

## Recovery Manager を使用したデータ・ファイルおよびデータ・ファイルのコピーのバックアップ

データ・ファイルとデータ・ファイルのコピーのバックアップは、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。オープン状態のデータベースのバックアップは、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。

### データ・ファイルのバックアップ

BACKUP DATAFILE コマンドを使用して、個々のデータ・ファイルをバックアップします。データ・ファイルは、名前または番号で指定できます。

#### データ・ファイルのバックアップ

1. Recovery Manager の起動後、BACKUP DATAFILE コマンドを Recovery Manager プロンプトから実行します。この例では、データ・ファイル 1 ～ 4 とデータ・ファイルのイメージ・コピーをバックアップします。

```
BACKUP DATAFILE 1,2,3,4 FILESPERSET 3 DATAFILECOPY '/tmp/system01.dbf';
```

CONFIGURE CONTROLFILE TO AUTOBACKUP が OFF の場合は、現行の制御ファイルとサーバー・パラメータ・ファイルが、データ・ファイルのバックアップ・セットに自動的に含まれます。ON の場合、これらのファイルは別の自動バックアップ・ピースに書き込まれます。

2. LIST BACKUP OF DATAFILE コマンドを発行して、バックアップ・セットおよびピースのリストを表示します。たとえば、次のように入力します。

```
LIST BACKUP OF DATAFILE 1,2,3,4;
```

## データ・ファイルのコピーのバックアップ

BACKUP DATAFILECOPY コマンドを使用して、データ・ファイルのコピーをバックアップします。データ・ファイルのコピーは、ディスク上にのみ存在します。

### データ・ファイルのコピーのバックアップ

1. Recovery Manager の起動後、BACKUP DATAFILECOPY コマンドを Recovery Manager プロンプトから実行します。この例では、データ・ファイル /tmp/system01.dbf をテープにバックアップします。

```
BACKUP DATAFILECOPY '/tmp/system01.dbf';
```

2. LIST BACKUP OF DATAFILECOPY コマンドを発行して、バックアップ・セットとピースのリストを表示します。たとえば、次のように入力します。

```
LIST BACKUP OF DATAFILE 1;
```

## Recovery Manager を使用した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。Recovery Manager は、スナップショット制御ファイルを使用して、読み取り一貫性を確保します。

CONFIGURE CONTROLFILE AUTOBACKUP を ON にすると（デフォルトでは OFF）、次の場合に制御ファイルと現行のサーバー・パラメータ・ファイル（使用している場合）が自動的にバックアップされます。

- Recovery Manager プロンプトで BACKUP または COPY コマンドが発行された後。
- RUN ブロック内に BACKUP または COPY コマンドがあり、後に続くコマンドが BACKUP でも COPY でもない場合。
- ブロック内の最後のコマンドが BACKUP または COPY の場合の各 RUN ブロックの最後。
- 新しい表領域の追加、表領域またはデータ・ファイルの状態の変更（オンライン化など）、新しいオンライン REDO ログの追加、ファイル名の変更、新しい REDO スレッドの追加など、データベース構造に変更があった後。前述の場合に発生する自動バックアップとは異なり、このタイプの自動バックアップはディスクにのみ作成されます。CONFIGURE CONTROLFILE AUTOBACKUP FOR DEVICE TYPE DISK を実行すると、デフォルト以外のディスク位置を設定できます。自動バックアップによって、それに関連する構造変更が失敗することはありません。たとえば、データ・ファイルを追加したために自動バックアップが失敗しても、データ・ファイルの追加は正常終了します。

**注意：** Oracle Enterprise Manager を使用する場合は、メンテナンス・ウィザードを使用して自動バックアップ機能を構成できます。

バックアップまたはコピー・ジョブで最初に割り当てられたチャネルによって自動バックアップが作成され、専用バックアップ・セットに格納され、構造の事後自動バックアップ用にデフォルトのディスク・チャネルによってバックアップが作成されます。制御ファイルとサーバー・パラメータ・ファイルは、同じバックアップ・ピースに書き込まれます。制御ファイルの自動バックアップが完了すると、バックアップ・ピースのフルパスとデバイス・タイプを含むメッセージが Oracle によってアラート・ログに書き込まれます。

次の表に示すように、BACKUP コマンドにデータ・ファイル 1 が含まれている場合の Recovery Manager の動作は、CONFIGURE CONTROLFILE AUTOBACKUP が ON であるか OFF であるかによって異なります。

CONTROLFILE AUTOBACKUP	BACKUP コマンドの動作
ON	<p>バックアップにデータ・ファイル 1 が含まれている場合、現行の制御ファイルがデータ・ファイルのバックアップ・セットに自動的に組み込まれることはありません。かわりに、Recovery Manager は制御ファイルとサーバー・パラメータ・ファイルを別の自動バックアップ・ピースに書き込みます。</p> <p><b>注意：</b>自動バックアップは、BACKUP DATABASE INCLUDE CURRENT CONTROLFILE など、BACKUP または COPY コマンドで現行の制御ファイルが明示的に指定されているかどうかに関係なく発生します。</p>
OFF	<p>バックアップにデータ・ファイル 1 が含まれている場合は、現行の制御ファイルとサーバー・パラメータ・ファイルがデータ・ファイルのバックアップ・セットに自動的に組み込まれます。制御ファイルとサーバー・パラメータ・ファイルを含む別個の自動バックアップ・ピースは作成されません。</p>

したがって、ファイルまたはファイルのセットをバックアップすると、必ずその後に、Recovery Manager が自動的に制御ファイルをバックアップします。この制御ファイルのバックアップには、前回のバックアップのメタデータが含まれています。このメタデータは、障害時リカバリを行う際に重要になります。

自動バックアップ機能が設定されていない場合、次に示す方法の 1 つを使用して、手動で制御ファイルをバックアップする必要があります。

- 自動的に制御ファイルをバックアップに含むデータ・ファイル 1 をバックアップ
- BACKUP CURRENT CONTROLFILE を実行
- BACKUP コマンドの INCLUDE CURRENT CONTROLFILE オプションを使用して制御ファイルのバックアップをあらゆるバックアップ内に含める

---

**注意：** 制御ファイルの手動バックアップは、制御ファイルの自動バックアップと同じではありません。制御ファイルの手動バックアップでは、現行の Recovery Manager セッション内のバックアップ用メタデータのみが制御ファイルのバックアップに含まれます。また、制御ファイルのバックアップは自動的にリストアされません。

---

## 制御ファイルの手動バックアップ

この手順では、BACKUP CURRENT CONTROLFILE コマンドを使用して、制御ファイルのバックアップを作成します。

### 現行の制御ファイルの手動バックアップ

1. Recovery Manager の起動後、BACKUP CURRENT CONTROLFILE コマンドを Recovery Manager プロンプトから実行します。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、さらにタグを割り当てます。

```
BACKUP CURRENT CONTROLFILE TAG = mondaypmbackup;
```

自動バックアップ機能が使用可能な場合、この例では、Recovery Manager は制御ファイルのバックアップを 2 つ作成するようになっているので、注意してください。明示的な制御ファイルのバックアップ (BACKUP CURRENT CONTROLFILE) および後続する制御ファイルとサーバー・パラメータ・ファイルの自動バックアップです。

2. 必要に応じて、LIST コマンドを発行して、バックアップ・セットおよびピースを表示します。たとえば、次のように入力します。

```
LIST BACKUP OF CONTROLFILE;
```

## バックアップ・セットに制御ファイルを含める

この手順では、制御ファイルを他のオブジェクトのバックアップに含めることで、制御ファイルのバックアップを作成します。

### 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・オブジェクトの指定後、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 `users` 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。

```
BACKUP TABLESPACE users INCLUDE CURRENT CONTROLFILE;
```

自動バックアップ機能が使用可能な場合、Recovery Manager は BACKUP TABLESPACE コマンドの完了後、制御ファイルの自動バックアップも生成するので、注意してください。

## 制御ファイル・コピーのバックアップ

この手順では、BACKUP CONTROLFILECOPY コマンドを使用して、制御ファイルのバックアップを作成します。

### 制御ファイル・コピーのバックアップ

1. データベースがマウントされているか、オープン状態であることを確認してください。
2. Recovery Manager の起動後、BACKUP CONTROLFILECOPY コマンドを Recovery Manager プロンプトから実行します。この例では、制御ファイル・コピー `'/tmp/control01.ctl'` を作成し、バックアップします。

```
COPY CURRENT CONTROLFILE TO '/tmp/control01.ctl';  
BACKUP CONTROLFILECOPY '/tmp/control01.ctl';
```

3. 必要に応じて、LIST BACKUP OF CONTROLFILE コマンドを発行し、バックアップ・セットおよびピースのリストを表示します。たとえば、次のように入力します。

```
LIST BACKUP OF CONTROLFILE;
```



## Recovery Manager を使用したサーバー・パラメータ・ファイルのバックアップ

9-9 ページの「[Recovery Manager を使用した制御ファイルのバックアップ](#)」で説明したように、Recovery Manager では、特定の場合に現行のサーバー・パラメータ・ファイルが自動的にバックアップされます。また、BACKUP SPFILE コマンドを使用して、サーバー・パラメータ・ファイルを明示的にバックアップすることもできます。たとえば、次のように入力します。

```
BACKUP COPIES 2 DEVICE TYPE sbt SPFILE;
```

バックアップ対象は、インスタンスによって現在使用されているサーバー・パラメータ・ファイルのみに限定されるため注意してください。インスタンスがクライアント側の初期化パラメータ・ファイルを使用して起動されている場合、Recovery Manager では何もバックアップされません。

## Recovery Manager を使用したアーカイブ REDO ログのバックアップ

アーカイブ REDO ログは、正常なメディア・リカバリ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 ARCHIVELOG を発行して、ログをバックアップできます。また、データ・ファイルと制御ファイルをバックアップし、BACKUP ... PLUS ARCHIVELOG を指定することでも、ログをバックアップできます。

特定の条件を満たすとき、Recovery Manager はアーカイブ REDO ログをバックアップするときに、現行のオンライン REDO ログをオフに切り替え、アーカイブしようとするので注意してください。

**関連項目：** Recovery Manager が現行のオンライン・ログをオフに切り替える条件については、5-19 ページの「[アーカイブ・ログのバックアップ時に切り替える自動オンライン REDO ログ](#)」を参照してください。

## BACKUP ARCHIVELOG を使用したログのバックアップ

アーカイブ・ログをバックアップするには、必要なフィルタ・オプションを使用して、BACKUP ARCHIVELOG を実行します。複数の位置にアーカイブする場合、Recovery Manager は、同じログ順序番号の複数コピーを、同じバックアップ・セットに入れません。BACKUP ARCHIVELOG ALL コマンドにより、各ログ順序番号の 1 つのコピーが個別にバックアップされます。

BACKUP コマンドの DELETE INPUT オプションを指定できます。このオプションは、アーカイブ・ログをバックアップした後、削除します。したがって、アーカイブ・ログをテープにバックアップし、古いログを消去してディスク領域を解放することが、1 つの手順で実行できます。Recovery Manager はバックアップするアーカイブ REDO ログの特定のコピーを削除し、次に、このコピーのみを削除します。

DELETE ALL INPUT オプションを指定した場合、Recovery Manager は、各ログ順序番号のバックアップを作成しますが、すべての使用可能なアーカイブ先からログを削除します。たとえば、/arc\_dest1、/arc\_dest2 および /arc\_dest3 にアーカイブし、次のコマンドを実行するとします。

```
BACKUP ARCHIVELOG ALL DELETE ALL INPUT;
```

Recovery Manager は、これらのディレクトリで検出される、各ログ順序番号のコピーを 1 つバックアップし、これらのディレクトリで検出される、すべてのログのすべてのコピーを削除します。DELETE ALL INPUT ではなく DELETE INPUT を指定した場合、Recovery Manager は、Recovery Manager がバックアップした特定のディスク・コピー（たとえば、/arc\_dest1 にあるログのみ）のみを削除します。

BACKUP ARCHIVELOG ALL または BACKUP ARCHIVELOG LIKE '...' を発行する場合で、アーカイブ REDO ログがない場合、Recovery Manager はエラーを表示しないことに注意してください。

### 関連項目：

- 複数の宛先にあるログのバックアップ例は、9-48 ページの「[アーカイブ REDO ログの複数コピーのバックアップと削除：例](#)」を参照してください。
- Oracle Real Application Clusters 構成でのバックアップについては、9-16 ページの「[Recovery Manager を使用した Real Application Clusters データベース内のアーカイブ・ログバックアップ](#)」を参照してください。

## BACKUP ARCHIVELOG を使用したアーカイブ REDO ログのバックアップ

1. Recovery Manager の起動後、BACKUP ARCHIVELOG コマンドを Recovery Manager プロンプトから実行します。この例では、構成済みのチャンネルを使用して、各ログ順序番号のコピーを 1 つ、テープにバックアップし、ディスク上のコピーをすべて削除します。

```
BACKUP ARCHIVELOG ALL DELETE ALL INPUT;
```

条件によっては（5-19 ページの「[アーカイブ・ログのバックアップ時に切り替える自動オンライン REDO ログ](#)」を参照）、Recovery Manager は REDO ログの切替えを試行して、現行の REDO ログもアーカイブを行うので注意してください。

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。次の例では、30 日前から 7 日前までのリカバリに使用できるすべてのログをバックアップします。

```
BACKUP ARCHIVELOG FROM TIME 'SYSDATE-30' UNTIL TIME 'SYSDATE-7';
```

UNTIL TIME を指定したため、Recovery Manager は自動的に切替えを行って現行のオンライン REDO ログをアーカイブしません。注意してください。

2. LIST BACKUP OF ARCHIVELOG ALL コマンドを発行して、バックアップ・セットおよびピースのリストを表示します。たとえば、次のように入力します。

```
LIST BACKUP OF ARCHIVELOG ALL;
```

## BACKUP ... PLUS ARCHIVELOG を使用したログのバックアップ

アーカイブ・ログ以外のオブジェクトをバックアップするときには、BACKUP ... PLUS ARCHIVELOG を指定して、アーカイブ・ログをバックアップします。コマンドは、次のアクションを順次実行します。

1. ALTER SYSTEM ARCHIVE LOG CURRENT コマンドを実行します。
2. BACKUP ARCHIVELOG ALL を実行します。バックアップの最適化が可能な場合（9-31 ページの「[バックアップの最適化を使用したファイルのバックアップ](#)」を参照）、Recovery Manager は指定のデバイスにバックアップ済みのログをスキップします。
3. BACKUP コマンドで指定したファイルをバックアップします。
4. ALTER SYSTEM ARCHIVE LOG CURRENT コマンドを実行します。
5. バックアップ中に生成された、残りのアーカイブ・ログをバックアップします。

この場合、バックアップしたデータ・ファイルが一貫性のある状態にリカバリ可能なことを保証します。

## BACKUP ... PLUS ARCHIVELOG を使用したアーカイブ REDO ログのバックアップ

1. データベースがマウントされているか、オープン状態であることを確認してください。
2. Recovery Manager の起動後、BACKUP コマンドを Recovery Manager プロンプトから実行し、PLUS ARCHIVELOG を指定します。次の例では、データベースと、すべてのアーカイブ REDO ログをバックアップします。

```
BACKUP DATABASE PLUS ARCHIVELOG;
```

3. LIST BACKUP コマンドを発行して、バックアップ・セットおよびピースのリストを表示します。たとえば、次のように入力します。

```
LIST BACKUP OF DATABASE SUMMARY;  
LIST BACKUP OF ARCHIVELOG SUMMARY;
```

## Recovery Manager を使用した Real Application Clusters データベース内のアーカイブ・ログバックアップ

Oracle Real Application Clusters 環境でアーカイブ REDO ログをバックアップする場合には、特別な問題が発生します。説明のために、UNIX 上の NFS または Windows 上のマップされたドライブを使用しないでリモートのディレクトリにマウントし、クラスタを次のように構成してあると仮定します。

- ノード 1 は、ノード 1 のローカル・ファイル・システム /d1/arc\_dest にアーカイブする。
- ノード 2 は、ノード 2 のローカル・ファイル・システム /d2/arc\_dest にアーカイブする。
- ノード 3 は、ノード 3 のローカル・ファイル・システム /d3/arc\_dest にアーカイブする。

NFS またはマップされたドライブを使用しない場合、各ノードは、異なるパスのディレクトリにアーカイブする必要があります。これを行わない場合、ログのバックアップに失敗するので注意してください。

この例では、制御ファイルに、次に示すようなアーカイブ・ログのレコードがあります。

```
/d1/arc_dest/log100_1.arc  
/d1/arc_dest/log101_1.arc  
/d3/arc_dest/log143_3.arc  
/d3/arc_dest/log144_3.arc  
/d2/arc_dest/log55_2.arc  
/d2/arc_dest/log56_2.arc  
/d2/arc_dest/log57_2.arc  
/d3/arc_dest/log145_3.arc  
/d3/arc_dest/log146_3.arc  
/d2/arc_dest/log58_2.arc  
/d1/arc_dest/log102_1.arc  
/d2/arc_dest/log59_2.arc  
/d1/arc_dest/log103_1.arc  
/d1/arc_dest/log104_1.arc
```

Recovery Manager をノード 2 で起動し、次のコマンドを実行するとします。

```
RUN  
{  
  ALLOCATE CHANNEL node2 DEVICE TYPE sbt;  
  BACKUP ARCHIVELOG ALL;  
}
```

この場合、チャンネルは /d2/arc\_dest のログのみを参照し、/d1 または /d3 ログが検出できないために、ジョブは失敗します。

この状況を回避するための環境の構成方法は、『Oracle9i Real Application Clusters 管理』を参照してください。たとえば、クラスタ・ファイル・システム（CFS）を実装できるか、NFS またはマップされたドライブを使用してディレクトリをマウントできるので、Recovery Manager があらゆるノードへ接続するとき、すべてのノードによって生成されたログにアクセスできます。ただし、環境をリモート・アクセス用に構成していない場合は、どのチャンネルがどのログをバックアップする必要があるかを指定することで、ログのバックアップの問題を解決できます。

### アーカイブ REDO ログの Oracle Real Application Clusters 構成へのバックアップ

1. データベースがマウントされているか、オープン状態であることを確認してください。
2. チャンネルが Oracle Real Application Clusters データベースの各ノードに割り当てられていることを確認します。次の例では、構成済みの、3 つの sbt チャンネルを示しています。1 つのチャンネルが各ノードに割り当てられています。

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 3;
CONFIGURE DEFAULT DEVICE TYPE to sbt;
CONFIGURE CHANNEL 1 DEVICE TYPE sbt CONNECT 'SYS/oracle@node1'
    PARMS 'ENV=(NSR_SERVER=bksvr1)'; # channel 1 is for first node
CONFIGURE CHANNEL 2 DEVICE TYPE sbt CONNECT 'SYS/oracle@node2'
    PARMS 'ENV=(NSR_SERVER=bksvr2)'; # channel 2 is for second node
CONFIGURE CHANNEL 3 DEVICE TYPE sbt CONNECT 'SYS/oracle@node3'
    PARMS 'ENV=(NSR_SERVER=bksvr3)'; # channel 3 is for third node
```

3. ログをバックアップします。たとえば、次のように入力します。

```
BACKUP ARCHIVELOG ALL;
```

#### 関連項目：

- BACKUP コマンドの構文および LIKE パラメータの説明は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- Oracle Real Application Clusters 環境でのアーカイブ・ログのバックアップ例は、9-52 ページの「[Oracle Real Application Clusters 環境でのバックアップ: 例](#)」を参照してください。

## 二重化バックアップ

災害、メディア障害または人為的なミスに備えて、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。Recovery Manager では、同一内容のバックアップ・セットのコピーを、最高 4 つまで、同時に作成できます。バックアップ・セットのコピーは、バックアップ・セットの各バックアップ・ピースのコピーであり、各コピーには一意のコピー番号が付いています（たとえば、0tcm8u2s\_1\_1 および 0tcm8u2s\_1\_2）。

ほとんどの場合、バックアップを二重化する最も簡単な方法は、BACKUP COPIES または CONFIGURE ... BACKUP COPIES を使用することです。同じ物理メディアに複数のコピーを作成しても意味がないことに注意してください。DISK チャネルの場合は、FORMAT オプションで複数の値を指定して、複数のコピーを異なる物理ディスクに送ります。sbt チャネルの場合は、SBT API のバージョン 2 をサポートしているメディア・マネージャを使用すると、各コピーが別々のメディア（別々のテープなど）に自動的に書き込まれます。

---

---

**注意：** sbt デバイスへの二重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ります。これを行わない場合、エラーが表示されます。Recovery Manager は、必要なバックアップ・コピーの数に応じて、起動された処理を使用します。

---

---

## CONFIGURE BACKUP COPIES を使用した多重化バックアップ

指定したデバイス・タイプ上で作成する同一バックアップの数を指定するには、CONFIGURE ... BACKUP COPIES コマンドを使用します。このコマンドは、データ・ファイル（現行の制御ファイルとその自動バックアップを含む）とアーカイブ・ログにのみ適用されます。構成済みの自動チャネルが必要です。

### CONFIGURE BACKUP COPIES を使用してバックアップを多重化する

1. データ・ファイルとアーカイブ REDO ログに必要なデバイス・タイプについて、コピーの数を構成します。この例では、テープ上のデータ・ファイルとアーカイブ・ログの二重化と、ディスク上のデータ・ファイルの二重化（アーカイブ・ファイルは除外）を構成しています。

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 1;
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '/save1/%U', '/save2/%U';
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE sbt TO 2;
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE sbt TO 2;
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 2;
```

2. バックアップを二重化します。次のコマンドにより、データベースとアーカイブ・ログをテープにバックアップし、データ・ファイルとアーカイブ REDO ログについてそれぞれ 2 つのコピーを作成します。

```
BACKUP DATABASE PLUS ARCHIVELOG; # uses default sbt channel
```

次のコマンドでは、データベースをディスクにバックアップし、一方のコピーを /save1 ディレクトリに、他方を /save2 ディレクトリに格納します。

```
BACKUP DEVICE TYPE DISK DATABASE;
```

3. LIST BACKUP コマンドを発行して、バックアップ・セットおよびピースのリストを表示します (#Copies 列は多重化の程度を示します)。たとえば、次のように入力します。

```
LIST BACKUP SUMMARY;
```

## BACKUP COPIES を使用した多重化バックアップ

BACKUP コマンドの COPIES オプションは、他のすべての COPIES または DUPLEX 設定をオーバーライドします。

### BACKUP COPIES を使用してバックアップを多重化する

1. BACKUP コマンドの COPIES オプションを使用して、同一内容のコピーの数を指定します。たとえば、次のように実行して、各バックアップ・セットのコピーを 3 つ、デフォルトの DISK ディレクトリに作成します。

```
BACKUP COPIES 3 INCREMENTAL LEVEL = 0 DATABASE;
```

BACKUP コマンド自体で COPIES を指定しているため、Recovery Manager は CONFIGURE DATAFILE COPIES の設定に関係なく各データ・ファイルのコピーを 3 つ作成することに注意してください。

2. LIST BACKUP コマンドを発行して、バックアップ・セットおよびピースのリストを表示します (#Copies 列は多重化の程度を示します)。たとえば、次のように入力します。

```
LIST BACKUP SUMMARY;
```

## Recovery Manager を使用した増分バックアップの作成

一貫性または非一貫性の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。データベースを NOARCHIVELOG モードで運用している場合は、一貫性のある増分バックアップのみが作成できるため、データベースを正しくクローズする必要があります。ARCHIVELOG モードでは、データベースをオープンまたはクローズできます。

### 増分バックアップの作成

1. Recovery Manager の起動後、BACKUP INCREMENTAL コマンドを Recovery Manager プロンプトから実行します。この例では、レベル 0 のバックアップを作成します。

```
BACKUP INCREMENTAL LEVEL = 0 DATABASE;
```

この例では、差分レベル 1 で、SYSTEM 表領域とデータ・ファイル sales.f のバックアップを作成します。この場合は、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。

```
BACKUP INCREMENTAL LEVEL = 1
  TABLESPACE SYSTEM
  DATAFILE '?/oradata/trgt/tools01.dbf';
```

この例では、累積レベル 2 で、表領域 users のバックアップを作成します。この場合は、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。

```
BACKUP INCREMENTAL LEVEL = 2 CUMULATIVE TABLESPACE users;
```

2. 必要に応じて、LIST BACKUP コマンドを発行して、バックアップ・セットおよびピースのリストを表示します。たとえば、次のように入力します。

```
LIST BACKUP OF DATABASE;
```



## Recovery Manager を使用したミラーの分割によるバックアップの作成

多くのサイトでは、プライマリ・データベースの障害発生や、ユーザーの誤った DROP TABLE 処理などが原因で不完全リカバリが必要になる場合に備えて、ディスク上に格納されたデータベースのバックアップを保持しています。ディスク上にあるデータ・ファイルのバックアップにより、リカバリのリストア手順が簡素化され、より迅速で信頼性の高いリカバリを行うことができます。

---

**注意：** オンライン REDO ログのバックアップは、ミラーの分割または他の方法で作成しないでください。オンライン REDO ログのバックアップのリストアは、同じ順序番号で、内容の異なる 2 つのアーカイブ・ログを作成する可能性があります。また、制御ファイルのバックアップは、ミラーの分割ではなく、BACKUP CONTROLFILE コマンドで作成する方法が最善です。

---

ディスク上にデータ・ファイルのバックアップを作成する方法の 1 つは、ディスクのミラー化を使用する方法です。たとえば、オペレーティング・システムを使用して、データベースにある各ファイルの 3 つの同一コピーをメンテナンスできます。この構成では、データベースのミラー化されたコピーを分割し、バックアップとして使用できます。

Recovery Manager では、ミラーを自動的に分割しませんが、バックアップ操作およびリカバリ操作で、ミラーの分割を使用できます。たとえば、Recovery Manager では、データ・ファイルのミラーの分割をデータ・ファイルのコピーとして使用でき、このコピーをディスクまたはテープにバックアップすることもできます。

次の手順では、オプションの SUSPEND/RESUME 機能を使用した、ミラーの分割によるバックアップ方法を示します。ほとんどの場合、SUSPEND/RESUME 機能はミラーの分割によるバックアップには必要ありません。ただし、使用しているシステムで、ボリュームを分割する前に、データベースのキャッシュの使用済みバッファの解放を必要とする場合には必要です。

### SUSPEND/RESUME を使用した表領域のミラーの分割によるバックアップの作成

1. Recovery Manager を起動し、バックアップが必要な表領域を、ALTER TABLESPACE ... BEGIN BACKUP 文を使用してバックアップ・モードにします。たとえば、表領域 users をバックアップ・モードにするには、次のコマンドを実行します。

```
% rman TARGET SYS/oracle@trgt CATALOG rman/cat@catdb
RMAN> SQL 'ALTER TABLESPACE users BEGIN BACKUP';
```

2. ミラー化ソフトウェアあるいはハードウェアから要求があった場合は、I/O を一時停止します。たとえば、次の SQL 文を入力します。

```
RMAN> SQL 'ALTER SYSTEM SUSPEND';
```

3. これらの表領域に含まれている基礎となるデータ・ファイルについて、ミラーを分割します。ミラーの分割の詳細は、『Oracle9i ユーザー管理バックアップおよびリカバリ・ガイド』を参照してください。

4. データベースを一時停止状態から戻します。

```
RMAN> SQL 'ALTER SYSTEM RESUME';
```

5. 表領域をバックアップ・モードから外します。たとえば、次のように入力します。

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

6. Recovery Manager セッションを起動し、ユーザー管理のミラーのコピーを、CATALOG コマンドを使用して、データ・ファイルのコピーとしてカタログに追加します。たとえば、次のように入力します。

```
RMAN> CATALOG DATAFILECOPY '/dk2/oradata/trgt/users01.dbf'; # catalog split mirror
```

7. データ・ファイルのコピーをバックアップします。たとえば、自動チャネルが構成済みで、プロンプトから BACKUP DATAFILECOPY コマンドを実行すると想定します。

```
RMAN> BACKUP DATAFILECOPY '/dk2/oradata/trgt/users01.dbf';
```

8. 分割されたミラーの再構成を行う場合は、最初に CHANGE ... UNCATALOG コマンドを使用して、手順 6 でカタログに追加したデータ・ファイルのコピーをカタログから削除します。たとえば、次のように入力します。

```
RMAN> CHANGE DATAFILECOPY '/dk2/oradata/trgt/users01.dbf' UNCATALOG;
```

9. 影響を受けるデータ・ファイルの分割されたミラーの再構成を実行します。

**関連項目：** ALTER SYSTEM SUSPEND の構文は、『Oracle9i SQL リファレンス』を参照してください。

## Recovery Manager を使用したスタンバイ・データベース・サイト上のファイルのバックアップ

この項では、次の項目について説明します。

- [Recovery Manager](#) によるスタンバイ・データベース・データ・ファイルおよびアーカイブ・ログのバックアップ
- [Recovery Manager](#) スタンバイ・データベース・バックアップ作成時の制限事項および使用方法
- [RC\\_ARCHIVED\\_LOG](#) ビューの解析
- スタンバイ・データベースのアーカイブ REDO ログをバックアップする時期の判断
- [Recovery Manager](#) を使用したスタンバイ・データベースのバックアップ

**関連項目：** Recovery Manager とスタンバイ・データベースの相互作用の概要は、[第 13 章「Recovery Manager によるスタンバイ・データベースの作成」](#)を参照してください。

## Recovery Manager によるスタンバイ・データベース・データ・ファイルおよびアーカイブ・ログのバックアップ

Recovery Manager では、スタンバイ・データベースおよび対応付けられたアーカイブ REDO ログのバックアップが可能です。データ・ファイルおよびアーカイブ REDO ログのスタンバイ・バックアップは、プライマリ・データベースのバックアップで完全に置き換えることができます。つまり、RESTORE コマンドを実行して、スタンバイ・データ・ファイルのバックアップをプライマリ・データベースにリストアすることも、プライマリ・データ・ファイルのバックアップをスタンバイ・データベースにリストアすることも可能だということです。ただし、スタンバイ制御ファイルとプライマリ制御ファイルを置き換えることはできません。

スタンバイ・ファイルのバックアップは、多くの場合、本番ファイルのバックアップより適切です。その理由は、次のとおりです。

- スタンバイ・データベースは本番データベースではないので、スタンバイ・バックアップはトランザクションや本番データベースのバッチ・ジョブと競合が発生しません。したがって、本番システムに干渉せずに、スタンバイ・データベースをバックアップ・ホストとして使用できます。
- スタンバイとプライマリのデータベースが別のホスト上にある場合、スタンバイ・バックアップの操作は CPU サイクルを使用せず、メモリーも割り当てません。また、本番ホストの他のリソースも使用しません。

プライマリ・データベースとスタンバイ・データベースは、同じリカバリ・カタログを使用する必要があります。両者が同じ DBID を共有している場合も、**Recovery Manager** ではプライマリ・データベースとスタンバイ・データベースを区別できます。プライマリがすでに登録済みの場合、スタンバイ・データベースをカタログに登録する必要はなく、単にスタンバイ・データベースに接続して **BACKUP** コマンドを実行できることに注意してください。

**ALTER DATABASE ACTIVATE STANDBY DATABASE** を使用してスタンバイ・データベースをアクティブにする場合、スタンバイ・データベースは新規のプライマリ・データベースになります。**RESETLOGS** はスタンバイの活性化で実行される必要があるため、**Recovery Manager** が新しいプライマリ・データベース用に、新規のインカネーション・レコードを作成します。この新しい、プライマリ・データベース・インカネーションのバックアップは、**RESETLOGS** 操作の後のプライマリ・データベースのバックアップと同じになります。

**関連項目：** **RESETLOGS** 操作については、『**Oracle9i ユーザー管理バックアップおよびリカバリ・ガイド**』の「ユーザー管理のメディア・リカバリの実行」章の「ユーザー管理のメディア・リカバリ後のデータベースのオープン」を参照してください。

## Recovery Manager スタンバイ・データベース・バックアップ作成時の制限事項および使用方法

スタンバイ・データベースの作成時には、次の制限事項に注意してください。

- スタンバイ・データベースのバックアップをプライマリ・サイトでリストア・ジョブに使用するためには、スタンバイ・データベースのバックアップ時、またはバックアップの少し後にスタンバイ・データベースを再同期化するときに、リカバリ・カタログに接続する必要があります。この手順が必要になるのは、これらのバックアップ・レコードがリカバリ・カタログに格納されていない場合、プライマリ・データベースがスタンバイのバックアップについて認識する方法が他にないためです。
- スタンバイ・データベースが使用中のスタンバイ制御ファイルは、バックアップできません。
- イメージ・コピーやスタンバイ制御ファイルの **Recovery Manager** 以外のバックアップを作成し、これを使用してプライマリ・データベースをリストアすることはできません。
- スタンバイ・データベースに接続し、**DUPLICATE** コマンドを使用して別のスタンバイ・データベースを作成することはできません。別のスタンバイ・データベースを作成するには、プライマリ・データベースに接続し、**DUPLICATE** を実行します。
- スタンバイ・データベースを、リカバリ・カタログに登録しないでください。
- スタンバイ・データベースのバックアップ時には、**TARGET** キーワード (**AUXILIARY** キーワードではない) を使用して、スタンバイ・データベースに接続する必要があります。基本的に、バックアップを行っている間、スタンバイ・データベースはプライマリ・データベースの代替物です。

**関連項目：** スタンバイ・データベースのバックアップの詳細は、『Oracle9i Data Guard 概要および管理』を参照してください。

## RC\_ARCHIVED\_LOG ビューの解析

スタンバイ・サイト上でアーカイブ・ログのバックアップを作成する場合、障害のイベントに必要なアーカイブ・ログは、すべてプライマリ・サイトで使用可能なことを確認してください。アーカイブ・ログは、次の場所のどこにでも存在可能なので、状況は混乱します。

- プライマリ・サイト上のアーカイブ・ログの接続先ディレクトリ内
- プライマリ・サイトでバックアップし、プライマリ・サイト上のアーカイブ・ログの接続先ディレクトリから削除
- スタンバイ・サイトでバックアップし、プライマリ・サイト上のアーカイブ・ログの接続先ディレクトリから削除

リカバリ・カタログ・ビュー RC\_ARCHIVED\_LOG は、アーカイブ・ログがプライマリ・サイトに置かれている時期およびスタンバイ・サイトに置かれている時期を示します。RC\_ARCHIVED\_LOG にあるアーカイブ・ログ情報は重要ですが、これは、ログをバックアップする時期またはログをスタンバイ・サイトからプライマリ・サイトへコピーする時期を認識する必要があるためです。

たとえば、SQL\*Plus を起動し、リカバリ・カタログ・データベースにリカバリ・カタログのスキーマ所有者として接続して、次の問合せを実行すると仮定します。

```
SELECT SEQUENCE#, IS_STANDBY
FROM RC_ARCHIVED_LOG;
```

```
SEQUENCE# IS_
-----
113 YES
114 NO
115 NO
116 YES
116 NO
```

IS\_STANDBY 列は、ログがスタンバイ・サイトに置かれているか (YES)、プライマリ・サイトに置かれているか (NO) を示します。同じログ順序番号に IS\_STANDBY の、YES および NO の両方設定がある場合、ログはスタンバイ・サイトとプライマリ・サイトの両方に置かれます。たとえば、順序番号 116 の IS\_STANDBY に YES と NO の両方の値がある場合、このログはプライマリ・サイトとスタンバイ・サイトの両方にあります。

## スタンバイ・データベースのアーカイブ REDO ログをバックアップする時期の判断

すべてのバックアップをスタンバイ・サイトで作成する場合、プライマリ・データベースで生成したアーカイブ・ログすべてをバックアップしていることを確認する必要があります。Recovery Manager がリカバリに使用するために、スタンバイ・データベースのアーカイブ・ログのバックアップが必要かどうかを判断するには、2 つの方法があります。

- LIST コマンドの実行
- RC\_ARCHIVED\_LOG の問合せ

### LIST コマンドを使用したスタンバイ・ログのバックアップ時期の判断

LIST BACKUP OF ARCHIVELOG ALL コマンドを使用して、Recovery Manager でバックアップ済みのログを判断します。

#### LIST コマンドの使用にログのバックアップが必要かどうかの判断方法

1. リカバリ・カタログに問い合せて、アーカイブ REDO ログの位置を判断します。たとえば、次のように発行します。

```
SELECT SEQUENCE#, IS_STANDBY
FROM RC_ARCHIVED_LOG;
```

```
SEQUENCE# IS_
-----
113 YES
114 NO
115 NO
116 NO
```

この出力は、ログ順序 113 がスタンバイ・サイトにあり、プライマリ・サイトではないことを示します。また、アーカイブ・ログ 114 から 116 はプライマリ・サイトにあり、スタンバイ・サイトにはないことを示します。

2. リカバリ・カタログに接続し、Recovery Manager で LIST BACKUP コマンドを実行してバックアップされるログがどれかを判断します。たとえば、次のように入力します。

LIST BACKUP OF ARCHIVELOG ALL;

List of Backup Sets						
Key	Recid	Stamp	LV	Set Stamp	Set Count	Completion Time
-----	-----	-----	-----	-----	-----	-----
319	4	394624547	0	394624546	5	11-APR-00
List of Backup Pieces						
Key	Pc#	Cp#	Status	Completion Time	Piece Name	
-----	-----	-----	-----	-----	-----	
320	1	1	AVAILABLE	11-APR-00	/vobs/oracle/dbs/05boavh2_1_1	
List of Archived Logs Included						
Thrd	Seq	Low SCN	Next SCN	Low Time	Next Time	
-----	-----	-----	-----	-----	-----	
1	116	95153	95156	07-APR-00	07-APR-00	

この出力は、Recovery Manager によってアーカイブ・ログ 116 はバックアップされ、アーカイブ・ログ 113 はバックアップされていないことを示しています。ログ 113 はスタンバイ・サイトにしか存在しないため、このログをバックアップするか、プライマリ・サイトにコピーする必要があります。

RC\_ARCHIVED\_LOG に問い合わせたスタンバイ・ログのバックアップ時期の判断

リカバリ・カタログに問い合わせ、Recovery Manager がバックアップしたログがどれかを判断することができます。

カタログの問合せにログのバックアップが必要かどうかの判断方法

1. RC\_ARCHIVED\_LOG リカバリ・カタログ・ビューに問い合わせ、リカバリに必要なアーカイブ・ログが、すべてディスク上にあるかどうかを判断します。たとえば、次の問合せを発行します。ここでは first\_log\_needed\_for\_recovery が、リカバリを開始するログの順序番号であり、expected\_num\_of\_logs は、完全リカバリの間に適用される必要のあるログの番号です。

```
SELECT 1 FROM RC_ARCHIVED_LOG
WHERE SEQUENCE# >= first_log_needed_for_recovery
AND IS_STANDBY='NO'
AND STATUS='A'
HAVING COUNT(*) = expected_num_of_logs;
```

問合せが行を戻さない場合は、完全リカバリに必要なすべてのログが、ディスク上に揃っていません。問合せが行を戻す場合は、完全リカバリに必要なすべてのログが、ディスク上に揃っています。

2. RC\_BACKUP\_REDOLOG ビューに問い合せて、完全リカバリに必要なログのバックアップがあるかどうかを判断します。たとえば、次の問合せを発行します。ここでは `first_log_needed_for_recovery` が、リカバリを開始するログの順序番号であり、`expected_num_of_logs` は、完全リカバリの間に適用される必要のあるログの番号です。

```
SELECT 1 FROM RC_BACKUP_REDOLOG
WHERE SEQUENCE# >= first_log_needed_for_recovery
AND STATUS='A'
HAVING COUNT(DISTINCT SEQUENCE#) = expected_num_of_logs;
```

問合せが行を戻さない場合は、完全リカバリに必要なすべてのログのバックアップが、ディスク上に揃っていません。問合せが行を戻す場合は、完全リカバリに必要なすべてのログのバックアップが、ディスク上に揃っています。

## Recovery Manager を使用したスタンバイ・データベースのバックアップ

Recovery Manager の BACKUP コマンドを使用して、スタンバイ・データベースをバックアップします。スタンバイ・データベースのバックアップは、実際にはプライマリ・データベースのバックアップと同じです。ただし、バックアップはスタンバイ・サイトで行われます。プライマリ・データベースは、スタンバイ・データベースのバックアップに影響を与えません。バックアップ実行のためにスタンバイ・データベースに接続するときには、AUXILIARY キーワードではなく TARGET キーワードを使用して接続します。

次の表は、スタンバイ・データベースのバックアップが一貫性のあるものか、またはバックアップが作成されたときのスタンバイ・データベースの状態に依存した、非一貫性のものなのかを示しています。一貫性バックアップのみが、メディア・リカバリを実行することなく、リストアできます。

スタンバイ・データベースの状態	バックアップの状態
完全に停止し、マウントされている（ただし、リカバリ・モードに移行していない）	一貫性
インスタンス障害後または SHUTDOWN ABORT の実行後にマウントされている	非一貫性
手動リカバリ・モード	非一貫性
管理リカバリ・モード	非一貫性
読取り専用モード	非一貫性



## スタンバイ・データベース全体のバックアップ作成

1. スタンバイ・データベースの一貫性バックアップを作成するために、スタンバイ・データベースの最後の停止が完全で、その後リカバリ・モードに移行していないことを確認してください。確認後、制御ファイルをマウントします。次に例を示します。

```
sqlplus SYS/oracle@sbdb1 <<EOF
SHUTDOWN IMMEDIATE
STARTUP NOMMOUNT PFILE=initSTANDBY.ora
ALTER DATABASE MOUNT STANDBY DATABASE;
EOF
```

スタンバイ・データベースがその他のモードの場合は、バックアップが可能です。バックアップは一貫性のないものになります。

2. Recovery Manager を起動し、TARGET キーワード (AUXILIARY キーワードではない) とリカバリ・カタログ・データベースを使用して、スタンバイ・データベースに接続します。このとき、リカバリ・カタログに接続している必要があります。たとえば、次のように入力します。

```
% rman TARGET SYS/oracle@sbdb1 CATALOG rman/cat@catdb
```

3. 自動チャネルの構成がない場合は、手動で 1 つ以上の、DISK または sbt タイプのチャネルを割り当てます。スタンバイ・ホストに接続しているため、バックアップはスタンバイ・ホスト (プライマリではなく) 上で、サーバー・セッションによって作成されることに注意してください。

この例では、自動チャネルを使用して、制御ファイルおよびアーカイブ・ログに加えて、すべてのスタンバイ・データ・ファイルもバックアップします。

```
BACKUP DATABASE PLUS ARCHIVELOG;
```

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できます。たとえば、次のように入力します。

```
BACKUP DATABASE FORMAT '/tmp/standby_%U'; # %U generates a unique filename
```

TAG を指定して、バックアップにタグを付けることができます。たとえば、次のように入力します。

```
BACKUP DATABASE TAG = 'weekly_standby_backup'; # gives the standby backup a tag
```

Recovery Manager がバックアップにデフォルト・タグを割り当てます。デフォルトの書式については、『Oracle9i Recovery Manager リファレンス』の BACKUP ... TAG の説明を参照してください。

4. 必要に応じて、LIST コマンドを発行して、バックアップ・セットおよびピースのリストを表示します。

## Recovery Manager を使用したバックアップ・セットのバックアップ

BACKUP BACKUPSET コマンドを使用して、データベース・ファイルではなくバックアップ・セットをバックアップします。このコマンドは、次の使用例で特に便利です。

- すべてのバックアップがディスクとテープの両方に存在していることの確認
- バックアップをディスクからテープへ移動し、ディスク上の領域を割当て解除

---

**注意：** BACKUP BACKUPSET の実行中は、制御ファイルの自動バックアップを二重化できません。Recovery Manager では、指定したメディアに常にコピーが 1 つのみ作成されます。

---

### バックアップ・セットのディスクからテープへのバックアップ

1. 構成済みの自動 sbt チャンネルがあり、Recovery Manager プロンプトから BACKUP BACKUPSET コマンドを発行するとします。この例では、デフォルト・ディスク・チャンネルと構成済みの sbt チャンネルを割り当てて、すべてのバックアップ・セットをテープにバックアップします。

```
BACKUP DEVICE DEVICE TYPE sbt BACKUPSET ALL;
```

この例では、ディスクのバックアップ・セットすべてをテープにバックアップし、入力されたディスクのバックアップを削除します。

```
BACKUP DEVICE TYPE sbt BACKUPSET ALL DELETE INPUT;
```

2. LIST コマンドを発行して、バックアップ・セットとピースのリストを表示します。

## Recovery Manager によるバックアップの再起動および最適化

Recovery Manager は、次のバックアップが必要ファイルのみをバックアップする、2つの個別の機能をサポートしています。再起動可能なバックアップとバックアップの最適化です。

再起動可能なバックアップ機能を使用すると、Recovery Manager は指定日以降にバックアップされていないファイルのみをバックアップします。たとえば、NOT BACKED UP SINCE TIME 句を指定して、最新のバックアップの日と指定した日の間にバックアップされなかったファイルのみをバックアップするように、Recovery Manager に対して指示できます。

バックアップの最適化を使用すると、BACKUP コマンドでは、割り当てられたデバイス・タイプで同一ファイルがバックアップ済みの場合に、ファイルのバックアップをスキップします。この動作をオーバーライドして、変更されたかどうかに関係なくすべてのファイルをバックアップするには、BACKUP コマンドに FORCE オプションを指定します。バックアップの最適化を有効または無効にするには、CONFIGURE BACKUP OPTIMIZATION コマンドに ON または OFF オプションを指定します。

さらに、3 番目の機能として、アーカイブ・ログのバックアップのみでなく、アーカイブされていないオンライン・ログのアーカイブもできます (9-15 ページの「[BACKUP ... PLUS ARCHIVELOG を使用したログのバックアップ](#)」を参照してください)。

**関連項目：** 最適化の概要は 5-62 ページの「[バックアップの最適化](#)」を参照してください。再起動可能なバックアップの概要は 5-68 ページの「[再起動可能なバックアップ](#)」を参照してください。

## バックアップの最適化を使用したファイルのバックアップ

バックアップの最適化を有効にするには、CONFIGURE BACKUP OPTIMIZATION を ON にする必要があります。バックアップの最適化は、デフォルトでは OFF です。

### バックアップの最適化

1. バックアップの最適化を有効にしていない場合は、CONFIGURE OPTIMIZATION コマンドを実行して有効にします。たとえば、次のように入力します。

```
CONFIGURE OPTIMIZATION ON;
```

2. 必要なファイルをバックアップします。たとえば、次のコマンドは事前構成されたチャネルを使用して、データベースとアーカイブ・ログの 2 つのコピーをディスクにバックアップします。

```
BACKUP DEVICE TYPE DISK COPIES 2 DATABASE PLUS ARCHIVELOG;
```

次の例では、sbt デバイスにログをバックアップします。

```
BACKUP DEVICE TYPE sbt ARCHIVELOG ALL;
```

保存方針とバックアップの二重化設定にもよりますが、Recovery Manager は最後のバックアップ以降に変更されているファイルのみをバックアップします。Recovery Manager ではファイルをスキップしたときにエラーを通知しません。すべてのファイルをスキップした場合も同様です。

**関連項目：** 最適化とバックアップ保存方針の概要は、5-62 ページの「[バックアップの最適化](#)」を参照してください。

### 部分的に完了したバックアップの再起動

BACKUP コマンドの SINCE TIME パラメータは、その日付以降に新規バックアップが必要になる日付を指定するために使用します。SINCE パラメータを指定していない場合、Recovery Manager はこれまで 1 度もバックアップしていないファイルのみをバックアップします。

#### 指定された日付以降にバックアップされていないファイルのみをバックアップする方法

SINCE TIME パラメータに有効な日付を指定します。たとえば、次のコマンドは、デフォルトで構成されているチャンネルを使用して、最近 2 週間以内にバックアップされていないデータベース・ファイルとアーカイブ REDO ログをバックアップします。

```
BACKUP NOT BACKED UP SINCE TIME 'SYSDATE-14'  
DATABASE PLUS ARCHIVELOG;
```

## Recovery Manager を使用したバックアップの妥当性チェックの実行

BACKUP コマンドの VALIDATE キーワードを使用すると、次のことができます。

- 物理破損および論理破損のチェック
- データベース・ファイルが正しい場所に存在していることの確認

Recovery Manager は実際にはバックアップ・セットを作成するのではなく、指定されたファイルをスキャンしてバックアップが可能で破損していないかどうかを判断します。この意味から、BACKUP VALIDATE コマンドは、リストア・ジョブ用ではなくバックアップ用である点を除いて、RESTORE VALIDATE コマンドと同じです。バックアップの妥当性チェックで破損ブロックが検出されると、Recovery Manager は破損の説明を示す行で V\$DATABASE\_BLOCK\_CORRUPTION ビューを更新します。破損ブロックの修復後は、このブロックを識別する行がビューから削除されます。

たとえば、すべてのデータベース・ファイルおよびアーカイブ REDO ログのバックアップが可能であることを検査するために、次のようにコマンドを実行します。

```
BACKUP VALIDATE DATABASE ARCHIVELOG ALL;
```

Recovery Manager は、実際にファイルをバックアップした場合と同じ出力を表示します。Recovery Manager が 1 つ以上のファイルのバックアップを検査できない場合は、エラー・メッセージが表示されます。たとえば、次のような出力が表示されます。

```
RMAN-00571: =====
RMAN-00569:===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: backup コマンドが 08/29/2001 14:33:47 で失敗しました
ORA-19625: ファイル /oracle/oradata/trgt/arch/archive1_6.dbf の識別中にエラーが発生しました。
ORA-27037: ファイル・ステータスを取得できません。
SVR4 Error: 2: No such file or directory
Additional information: 3
```

VALIDATE オプションとともに MAXCORRUPT または PROXY パラメータを使用することはできません。

#### 関連項目：

- BACKUP コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- BACKUP ...VALIDATE によって検出された破損ブロックの修復方法は、10-42 ページの「[V\\$DATABASE\\_BLOCK\\_CORRUPTION](#) にリストされるブロックのリカバリ」を参照してください。

## Recovery Manager を使用したファイルのコピー

ほとんどの場合、バックアップの作成よりコピーの作成の方が適しています。コピーは Recovery Manager 固有の形式ではなく、追加処理を行わずに適切に使用できるためです。これに対して、バックアップ・セットは、使用可能とするために RESTORE コマンドで処理する必要があります。したがって、メディア・リカバリは、データ・ファイルのコピーに対しては実行できても、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・コピーの作成には、COPY コマンドを使用します。Recovery Manager は、出力ファイルを常にディスクに書き込みます。コピー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）

## データベース全体のコピー

BACKUP DATABASE コマンドに対応する COPY DATABASE コマンドはありません。データ・ファイルは個別にコピーする必要があります。データ・ファイルの名前を判断するには、REPORT SCHEMA コマンドを実行します。

### 全データベース・ファイルの一貫性コピーの作成

1. 一貫性バックアップを作成する場合は、正しく停止されたことを確認してからデータベースをマウントします（ただしオープンしません）。たとえば、次のように入力します。

```
SHUTDOWN IMMEDIATE
STARTUP MOUNT
```

データベースがオープンされていても、またマウントされている場合でも、前回オープンされたときに正しくクローズされていない場合は、非一貫性バックアップとなります。

2. データベースの現行のスキーマに関するレポートを生成します。

```
REPORT SCHEMA;
```

```
Report of database schema
File K-bytes      Tablespace      RB segs Datafile Name
-----
1          307200 SYSTEM                ***      /oracle/oradata/trgt/system01.dbf
2          20480 UNDOTBS                ***      /oracle/oradata/trgt/undotbs01.dbf
3          10240 CWMLITE                ***      /oracle/oradata/trgt/cwmlite01.dbf
4          10240 DRSYS                  ***      /oracle/oradata/trgt/drsys01.dbf
5          10240 EXAMPLE                ***      /oracle/oradata/trgt/example01.dbf
6          10240 INDX                    ***      /oracle/oradata/trgt/indx01.dbf
7          10240 TOOLS                    ***      /oracle/oradata/trgt/tools01.dbf
8          10240 USERS                    ***      /oracle/oradata/trgt/users01.dbf
```

3. 構成済みの自動チャネルがあり、**Recovery Manager** プロンプトから **COPY** コマンドを発行するとします。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含めます。たとえば、**Recovery Manager** プロンプトから次のように入力します。

```
COPY
DATAFILE 1 TO '/tmp/system01.dbf',
DATAFILE 2 TO '/tmp/undotbs01.dbf',
DATAFILE 3 TO '/tmp/cwmlite01.dbf',
DATAFILE 4 TO '/tmp/drsys01.dbf',
DATAFILE 5 TO '/tmp/example01.dbf',
DATAFILE 6 TO '/tmp/indx01.dbf',
DATAFILE 7 TO '/tmp/tools01.dbf',
DATAFILE 8 TO '/tmp/users01.dbf',
CURRENT CONTROLFILE TO '/tmp/control01.ctl';
```

4. **LIST COPY OF DATABASE** コマンドを発行して、イメージ・コピーのリストを表示します。たとえば、次のように入力します。

```
LIST COPY OF DATABASE;
```

## 任意のデータ・ファイルのコピーおよびアーカイブ REDO ログのコピー

データ・ファイルと制御ファイルの他に、その他のコピーおよびアーカイブ REDO ログ（さらにアーカイブ REDO ログのコピー）もコピーできます。

### 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. 一貫性バックアップを作成する場合は、正しく停止されたことを確認してからデータベースをマウントします（ただしオープンしません）。たとえば、次のように入力します。

```
SHUTDOWN IMMEDIATE
STARTUP MOUNT
```

データベースがオープンされていても、またマウントされている場合でも、前回オープンされたときに正しくクローズされていない場合は、非一貫性バックアップとなります。

2. 構成済みの自動チャネルがあり、Recovery Manager プロンプトから COPY コマンドを発行するとします。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します。たとえば、次のように入力します。

```
COPY
DATAFILECOPY '/tmp/system01.dbf' TO '/save/system01.dbf',
DATAFILECOPY TAG = 'weekly_df8_copy' TO '/tmp/users01.dbf',
# copy archived redo logs
ARCHIVELOG '?/oradata/trgt/arch/archive1_1.dbf' TO '/tmp/archive1_1.dbf',
ARCHIVELOG '?/oradata/trgt/arch/archive1_2.dbf' TO '/tmp/archive1_2.dbf',
# copy a control file copy
CONTROLFILECOPY '/tmp/control01.ctl' TO '/save/control01.ctl';
```

3. LIST COPY コマンドを発行して、イメージ・コピーのリストを表示します。



## 制御ファイルの自動バックアップ書式の上書き

5-52 ページの「[制御ファイルとサーバー・パラメータ・ファイルの自動バックアップ](#)」に説明されているように、自動バックアップ機能が使用可能な場合、Recovery Manager は BACKUP コマンドまたは COPY コマンドの後で制御ファイルを自動的にバックアップします。

8-20 ページの「[制御ファイルの自動バックアップ書式の構成](#)」には、制御ファイルの自動バックアップのデフォルトの書式が説明されています。CONFIGURE コマンドを使用して、デフォルトの自動バックアップ書式を新しい値に変更することもできます。

文字列に %F 書式を含める必要があるので注意してください。含まれていないとエラーが発生します。この書式の変更は、すべての Recovery Manager セッションに対して永続的です。たとえば、次のようにデフォルトを変更できます。

```
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE sbt TO 'c_%F';  
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '?:oradata/%F.bck';
```

制御ファイルのバックアップの構成場所は Recovery Manager セッションで上書きできます。これには、SET コマンドを使用して、ディレクトリと、必要に応じて接頭辞と接尾辞を指定します。たとえば、次のように設定できます。

```
SET CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE sbt TO 'controlfile_%F';  
RUN { SET CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '/tmp/%F.bck'; }
```

制御ファイルの自動バックアップの書式を指定するコマンドの優先順位は次のとおりです。

1. RUN 内の SET CONTROLFILE AUTOBACKUP FORMAT
2. Recovery Manager プロンプトでの SET CONTROLFILE AUTOBACKUP FORMAT
3. CONFIGURE CONTROLFILE AUTOBACKUP FORMAT

### 関連項目：

- スナップショット制御ファイルを使用した、Recovery Manager の再同期化の概要は、16-13 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。
- CONFIGURE SNAPSHOT CONTROLFILE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- エンキューのために失敗したバックアップの実例は、15-35 ページの「[制御ファイルのエンキューが原因でバックアップが失敗する場合：例](#)」を参照してください。

## Recovery Manager のバックアップおよびコピーの例

この項では、次の項目について説明します。

- BACKUP コマンドでのデバイス・タイプの指定 : 例
- データベースのバックアップ時の表領域のスキップ : 例
- バックアップの再起動 : 例
- 複数ディスク・ドライブへのバックアップの分散 : 例
- 複数ファイル・システムへの大規模データベースのバックアップ : 例
- バックアップ・セットのサイズ指定 : 例
- バックアップ・ピースのサイズ制限 : 例
- バックアップにおけるデータ・ファイルの多重化 : 例
- フェイルオーバーでのアーカイブ REDO ログのバックアップの使用例 : 例
- オンライン・バックアップのリカバリに必要なアーカイブ・ログのバックアップ : 例
- アーカイブ REDO ログの複数コピーのバックアップと削除 : 例
- 差分増分バックアップの実行 : 例
- 累積増分バックアップの実行 : 例
- チャンネルによるバックアップ作業負荷分散の決定 : 例
- NOARCHIVELOG モードでのバックアップ : 例
- Oracle Real Application Clusters 環境でのバックアップ : 例
- オペレーティング・システム・コピーのカatalog化 : 例
- 長期バックアップの保持 : 例
- バックアップの最適化 : 例
- バックアップおよびコピー時のエラー処理 : 例

## BACKUP コマンドでのデバイス・タイプの指定：例

自動 sbt チャンネルを次のように構成するとします。

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 1; # configure device
CONFIGURE CHANNEL DEVICE TYPE sbt PARMS='...'; # configure options for channels
CONFIGURE DEFAULT DEVICE TYPE to sbt; # set default device type
```

データベースをディスクにバックアップし、デフォルトで構成されている DISK チャンネルを使用すると想定します。次のようにすると、BACKUP コマンドで DISK チャンネルを使用するように指定できます。

```
BACKUP DEVICE TYPE DISK DATABASE;
```

データベースを sbt デバイスにバックアップするには、次のコマンドを実行します。

```
BACKUP DATABASE;
```

## データベースのバックアップ時の表領域のスキップ：例

次の例では、データベースが ARCHIVELOG モードで実行されていて、自動 sbt チャンネルが次のように構成されていると想定されています。

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 1;
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE CHANNEL DEVICE TYPE sbt PARMS='ENV=(NSR_DATA_VOLUME_POOL=BackupPool)';
```

オフライン表領域と読取り専用表領域をスキップしながらデータベースをバックアップするには、次のコマンドを実行します。

```
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なのは、読取り専用にした後の 1 回のみです。SKIP READONLY オプションを使用すると、読取り専用のデータ・ファイルをスキップできます。SKIP OFFLINE オプションを使用すると、BACKUP コマンドはオフライン・データ・ファイルへのアクセスを試行しません。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。

Recovery Manager セッション全体で永続的に表領域をスキップする別の方法としては、常にスキップする各表領域に対して CONFIGURE EXCLUDE コマンドを発行します。たとえば、読取り専用になっている example 表領域を常にスキップするとします。次のように発行できます。

```
CONFIGURE EXCLUDE FOR TABLESPACE example;
```

この後、BACKUP DATABASE を実行するたびに、Recovery Manager はこの表領域をスキップします。BACKUP コマンドに SKIP 句を指定する必要はありません。この動作は、次のようにオーバーライドして、old\_data 表領域を含めることができます。

```
BACKUP DATABASE NOEXCLUDE;
```

## バックアップの再起動 : 例

次のコマンドを実行し、データベースとアーカイブ・ログを毎晩テープにバックアップするとします。

```
BACKUP FILESPERSET 2 DATABASE PLUS ARCHIVELOG;
```

このコマンドは、各バックアップ・セットを 2 つのデータ・ファイルに制限するため、複数のバックアップ・セットが生成されます。メディア管理デバイスがバックアップの途中で障害を起こし、その後再起動されたとします。翌日、バックアップ・セットの半分のみが完了していることがわかります。この場合は、夕方に次のコマンドを実行します。

```
BACKUP
# Note that the NOT BACKED UP SINCE clause should be placed immediately after the BACKUP
# keyword or after each individual backupSpec clause
NOT BACKED UP SINCE TIME 'SYSDATE-1'
FILESPERSET 2
DATABASE PLUS ARCHIVELOG;
```

Recovery Manager では、過去 24 時間以内にバックアップされていないファイルのみをバックアップします。特定のファイルがバックアップ済みであることがわかると、Recovery Manager は次のような出力を表示します。

```
RMAN-06501: データ・ファイル 1 をスキップします。すでに 2001 年 5 月 2 日 18:10:00 でバックアップされています
RMAN-06501: データ・ファイル 2 をスキップします。すでに 2001 年 5 月 2 日 18:09:45 でバックアップされています
RMAN-06501: データ・ファイル 3 をスキップします。すでに 2001 年 5 月 2 日 18:09:45 でバックアップされています
```

## 複数ディスク・ドライブへのバックアップの分散 : 例

通常、テープへのバックアップを実行する場合、書式の指定は必要ありません。これは、デフォルトの %U 変数によって、すべてのテープ・バックアップに一意的なファイル名が生成されるためです。ただし、ディスクへのバックアップの際に、バックアップを複数のドライブに分散してパフォーマンスの向上を図る場合は、書式を指定できます。この場合は、各ディスク・ドライブに 1 つの DISK チャンネルを割り当て、ファイルが別々のディスク上に作成されるように、書式文字列を ALLOCATE CHANNEL コマンドに指定します。たとえば、次のように発行します。

```
RUN
{
  ALLOCATE CHANNEL disk1 DEVICE TYPE DISK FORMAT '/disk1/%d_backups/%U';
  ALLOCATE CHANNEL disk2 DEVICE TYPE DISK FORMAT '/disk2/%d_backups/%U';
  ALLOCATE CHANNEL disk3 DEVICE TYPE DISK FORMAT '/disk3/%d_backups/%U';
  BACKUP DATABASE;
}
```

自動チャンネルを次のように構成しても、同じ結果になります。

```
CONFIGURE DEVICE TYPE DISK PARALLELISM 3;
CONFIGURE DEFAULT DEVICE TYPE TO DISK;
CONFIGURE CHANNEL 1 DEVICE TYPE DISK FORMAT '/disk1/%d_backups/%U'; # configure 1st ch
CONFIGURE CHANNEL 2 DEVICE TYPE DISK FORMAT '/disk2/%d_backups/%U'; # configure 2nd ch
CONFIGURE CHANNEL 3 DEVICE TYPE DISK FORMAT '/disk3/%d_backups/%U'; # configure 3rd ch
BACKUP DATABASE;
```

存在しないディレクトリを指定すると、次のような出力が表示されます。

```
RMAN-00571: =====
RMAN-00569:===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03009: backup コマンド (ORA_DISK_1 チャンネル上) が 08/29/2001 14:36:04 で失敗しました
ORA-19504: ファイル "foo/0cd2momi_1_1" の作成に失敗しました。
ORA-27040: skgfrcre: 作成エラーが発生しました。ファイルを作成できません。
SVR4 Error: 2: No such file or directory
```

## 複数ファイル・システムへの大規模データベースのバックアップ: 例

この例では、40GB のデータベースをディスクにバックアップします。Recovery Manager は、RAW ディスク・デバイスにはバックアップ・ピースを 1 つしか書き込めないため、複数のファイル・システムにバックアップを分散することになります。そこで、各バックアップ・セットをほぼ同じサイズの 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なります。各バックアップ・セットが 5 つのバックアップ・ピースを含むようにするため、各バックアップ・ピースを 2GB 以下にします。

CONFIGURE 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に設定します。置換変数を使用して、各バックアップ・ピースに一意的な名前を付与します。たとえば、次のコマンドを使用すると、チャネルが 4 つのファイル・システム (/fs1, /fs2, /fs3, /fs4) に構成され、データ・ファイルがグループ化されます。これにより、各バックアップ・セットは、ほぼ同じサイズになります。

```
CONFIGURE DEVICE TYPE DISK PARALLELISM 4; # first, configure the device for parallelism 4
CONFIGURE CHANNEL 1 DEVICE TYPE DISK FORMAT='/fs1/%u.%p' MAXPIECESIZE 2G;
CONFIGURE CHANNEL 2 DEVICE TYPE DISK FORMAT='/fs2/%u.%p' MAXPIECESIZE 2G;
CONFIGURE CHANNEL 3 DEVICE TYPE DISK FORMAT='/fs3/%u.%p' MAXPIECESIZE 2G;
CONFIGURE CHANNEL 4 DEVICE TYPE DISK FORMAT='/fs4/%u.%p' MAXPIECESIZE 2G;
CONFIGURE DEFAULT DEVICE TYPE TO DISK;
```

次に、このコマンドを毎晩実行し、バックアップ・セットを 4 つ生成します。バックアップ・セットはそれぞれ別々のディレクトリに入れられ、ほぼ同じサイズになります。

```
BACKUP DATABASE;
```

ディスクからのバックアップ・セットを、テープ・プールの 4 つの異なるテープにバックアップすることもできます。これには、次の例のように、sbt デバイスに PARALLELISM=4 を設定（さらに sbt チャネルにベンダー固有の適切な PARMS を指定）します。

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 4;
CONFIGURE CHANNEL DEVICE TYPE sbt PARMS='...';
BACKUP DEVICE TYPE sbt BACKUPSET ALL DELETE INPUT;
```

## バックアップ・セットのサイズ指定：例

バックアップを作成する際、Recovery Manager 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。この計算結果をオーバーライドして、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ためには、FILESERSET パラメータと MAXSETSIZE パラメータを使用します。

### FILESERSET を使用したバックアップ・セットのサイズの制御

FILESERSET パラメータを指定すると、Recovery Manager は FILESERSET の値と、自動的に算出された値（割り当てられた各チャンネルに対するファイル数）を比較し、いずれか低い方を選択します。これにより、すべてのチャンネルがすべて確実に使用されます。たとえば、3 つのチャンネルで 12 個のデータ・ファイルをバックアップする場合に、FILESERSET=2 を設定したとします。この場合 Recovery Manager はデータ・ファイルを 4 つずつではなく 2 つずつ各バックアップに入れます。

指定されたファイル数または *backupSpec* 句で暗黙的に指定したファイル数が、FILESERSET より多い場合（たとえば、FILESERSET=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）、Recovery Manager は各バックアップ・セットに対して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。

FILESERSET パラメータを指定しない場合、Recovery Manager は算出された値（ファイル数を割り当てられたチャンネル数で除算）を、デフォルト値の 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。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、小さい値を指定するか、MAXSETSIZE パラメータを使用します。

Recovery Manager は常に、割り当てられたチャンネルがすべて作業状態となるようにバックアップ・セットを作成しようとします。例外が発生するのは、バックアップするファイル数よりチャンネル数が多い場合です。たとえば、3 つのチャンネルが割り当てられ、Recovery Manager が 1 つのデータ・ファイルをバックアップする場合、2 つのチャンネルがアイドル状態になります。

この例では、ディスクの平行化を 4 に構成します。

```
CONFIGURE DEVICE TYPE DISK PARALLELISM 4;
```

次のスクリプトは、1 つのバックアップ・セットに 4 個以上のデータ・ファイルおよび 17 個以上のアーカイブ・ログが含まれないように指定して、バックアップを実行します。

```
BACKUP
{
  DATABASE
    FILESERSET = 3
  ARCHIVELOG ALL
    FILESERSET = 16;
}
```

## MAXSETSIZE を使用したバックアップ・セットのサイズの制御

MAXSETSIZE パラメータは、バックアップ・セットの最大サイズを、バイト（デフォルト）、キロバイト（KB）、メガバイト（MB）、ギガバイト（GB）単位で指定します。したがって、バックアップ・セットを 305MB に制限するには、MAXSETSIZE=305M と指定します。

Recovery Manager は、すべてのセットをこのサイズに制限します。

MAXSETSIZE を使用すると、データベースが複数のバックアップ・セット間で分割されるようにバックアップ・セットのサイズを制限できます。そうしない場合、バックアップが途中で失敗した場合に、データベース・バックアップを最初からやり直す必要があります。複数のバックアップ・セットが生成されるように MAXSETSIZE を構成し、バックアップが途中で失敗した場合は、再起動可能バックアップ機能を使用して、最初のバックアップ試行でバックアップされなかったファイルのみをバックアップできます。

アーカイブ REDO ログのバックアップの作成は、FILESERSET パラメータより MAXSETSIZE パラメータを使用する方が簡単です。次の例では、テープ・デバイスを構成してから、アーカイブ REDO ログをテープにバックアップします。バックアップが途中で失敗しても再起動できるように、サイズは 100MB に制限します。

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 1;
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
BACKUP MAXSETSIZE = 100M ARCHIVELOG ALL;
```

次の例では、CONFIGURE MAXSETSIZE を指定して上と同じ結果を得ます。

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE MAXSETSIZE = 100M;
BACKUP ARCHIVELOG ALL;
```

MAXSETSIZE の設定値がバックアップ対象の最小ファイルよりも小さい場合は、次のようなエラー・スタックが表示されます。

```
RMAN-00571: =====
RMAN-00569:===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: backup コマンドが 08/29/2001 14:40:33 で失敗しました
RMAN-06182: アーカイブ・ログが SETSIZE を超えています。スレッド 1 seq 1
           /oracle/oradata/trgt/arch/archive1_1.dbf
```



## バックアップ・ピースのサイズ制限：例

バックアップ・ピースのサイズは、ファイル・システムまたはメディア管理ソフトウェアの最大ファイル・サイズを超える場合に、指定する必要があります。CONFIGURE CHANNEL または ALLOCATE CHANNEL コマンドの MAXPIECESIZE パラメータを使用して、バックアップ・ピースのサイズを制限します。

たとえば、バックアップ・ファイル・サイズを 2000MB 以下に制限するには、次のように自動 DISK チャンネルを構成してから、BACKUP DATABASE を実行します。

```
CONFIGURE CHANNEL DEVICE TYPE DISK MAXPIECESIZE 2000M; # max file size for backup pieces
BACKUP DATABASE;
```

Media Management API のバージョン 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できます。指定すると、Recovery Manager は自動的にその制限に従うため、注意してください。

## バックアップにおけるデータ・ファイルの多重化：例

trgt というデータベースをバックアップするとします。データベース環境には次のような条件があります。

- バックアップする際、テープ・ドライブを 3 台使用できること。
- データベースに比較的小さい 1000 のデータ・ファイルが含まれていること。

多数のバックアップ・セットが作成されないように、FILESERSET=64 に設定します。ただし、多重化するファイルは 4 つ以内にします。テープ・ドライブのストリームを一定に維持するには、この値で十分です。したがって、MAXOPENFILES=4 に設定します。

3 つの sbt チャンネルを次のように構成してあるとします（各メディア管理デバイスが必要とする PARMS 設定を使用）。

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 3; # parallelize to 3 tape drives
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE CHANNEL 1 DEVICE TYPE sbt MAXOPENFILES 4 PARMS='...';
CONFIGURE CHANNEL 2 DEVICE TYPE sbt MAXOPENFILES 4 PARMS='...';
CONFIGURE CHANNEL 3 DEVICE TYPE sbt MAXOPENFILES 4 PARMS='...';
```

この場合、次のバックアップ・スクリプトを作成できます。

```
CREATE SCRIPT TRGT_FULL
{
    BACKUP FILESPERSET 4
        DATABASE FORMAT 'TRGT.FULL.%d.%s.%p';
}
```

かわりに、チャンネルを手動で割り当てるスクリプトを作成することもできます。

```
CREATE SCRIPT TRGT_FULL
{
  ALLOCATE CHANNEL t1 DEVICE TYPE sbt MAXOPENFILES 4 PARMS='...';
  ALLOCATE CHANNEL t2 DEVICE TYPE sbt MAXOPENFILES 4 PARMS='...';
  ALLOCATE CHANNEL t3 DEVICE TYPE sbt MAXOPENFILES 4 PARMS='...';
  BACKUP FILESPERSET 64
    DATABASE FORMAT 'TRGT.FULL.%d.%s.%p';
}
```

次のように、スクリプトを実行できます。

```
RUN { EXECUTE SCRIPT TRGT_FULL; }
```

このバックアップ・スクリプトでは、すべてのデータ・ファイルと制御ファイルを含めて、データベース全体のバックアップが作成されます。バックアップ対象のファイルは 1001 個（データ・ファイルが 1000 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、16 のバックアップ・セットが作成されます。MAXOPENFILES が 4 であるため、各チャンネルは 4 つのデータ・ファイルを同時に読み込みます。バックアップ・ピースのファイル名は、次の書式になります。db\_name はデータベース名を、set\_num はバックアップ・セットの番号を、piece\_num はピースの番号をそれぞれ示します。

```
TRGT.FULL.db_name.set_num.piece_num
```

たとえば、バックアップ・ピースには次のファイル名を指定できます。

```
TRGT.FULL.trgt.3.1
```

このジョブの前にリカバリ・カタログに記録されたバックアップ・セットがない場合、set\_num の範囲は 1 ～ 7、piece\_num は 1 以上となります。

## フェイルオーバーでのアーカイブ REDO ログのバックアップの使用例：例

次に示すローカルの宛先にアーカイブするように初期化パラメータを設定したとします。

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

各ディレクトリには、ログ順序が 1 から始まり 400 で終わる同じログ・セットが含まれています。あるユーザーが誤って /disk1/arch のログ 300 から 400 までと /disk2/arch のログ 350 から 400 までを削除してしまった場合、次のバックアップ・コマンドを実行します。

```
BACKUP ARCHIVELOG
  FROM SEQUENCE 288 UNTIL SEQUENCE 388
  THREAD 1
  DELETE INPUT;
```

Recovery Manager はログ順序 288 からログのバックアップを開始します。/disk1/arch から削除されたログ 300 のコピーが、Recovery Manager がバックアップしようとしているログである場合、Recovery Manager はリポジトリをチェックして、このログ順序の他のコピーがあるかどうかを判断し、/disk2/arch または /disk3/arch にあるログをバックアップします。順序 288 ～ 388 の各ログのコピーは 3 つのディレクトリのうち 1 つ以上にあるため、Recovery Manager は指定されたログをすべてバックアップできます。

## オンライン・バックアップのリカバリに必要なアーカイブ・ログのバックアップ：例

データベース trgt が開いている間に、このデータベースをバックアップするとします。このオンライン・バックアップのリカバリに必要なアーカイブ REDO ログのみをバックアップします。バックアップ対象のログがどれであるかはどのように判断するのでしょうか。

### オンライン・バックアップのリカバリに必要なアーカイブ・ログの判断

1. SQL\*Plus を起動し、現行のログを含み、アーカイブされていないログをすべてアーカイブします。

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

2. 次の例（出力を含む）に示すように、V\$LOG を問い合せて、現行 REDO ログのログ順序番号を判断します。

```
SELECT SEQUENCE#
FROM V$LOG
WHERE STATUS = 'CURRENT';
```

```
SEQUENCE#
-----
          9100
```

3. **Recovery Manager** を起動し、データベースのオンライン・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。

```
BACKUP DATABASE;
```

4. 現行のログを含み、アーカイブされていないログをすべてアーカイブします。

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

5. **SQL\*Plus** で **v\$log** を問い合せて、現行 **REDO** ログのログ順序番号を判断します。

```
SELECT SEQUENCE#
FROM V$log
WHERE STATUS = 'CURRENT';
```

```
SEQUENCE#
-----
          9112
```

6. 問い合せた最初の順序番号のログからバックアップを開始し、最後の順序番号から 1 を引いた番号のログでバックアップを終了します。現行ログの前のログが最後にアーカイブされたログです。たとえば、最初の問合せで 9100 が返された場合は、9100 から開始します。2 番目の問合せで 9112 が返された場合は、9111 で終了します。

たとえば、必要なアーカイブ・ログをバックアップするには、次のように発行します。

```
BACKUP ARCHIVELOG FROM SEQUENCE 9100 UNTIL SEQUENCE 9111;
```

## アーカイブ REDO ログの複数コピーのバックアップと削除 : 例

この使用例では、初期化パラメータを設定して、`*/oradata/trgt/arch/dest_1` および `*/oradata/trgt/arch/dest_2` という 2 つのディレクトリに、**REDO** ログが自動的にアーカイブされるようにします。したがって、ログ順序番号 1 に対して、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、その後にオリジナルを削除します。

この場合の最も簡単な解決方法は、**DELETE ALL INPUT** オプションの使用です。これは、**ARCHIVELOG** 基準に一致するすべてのログを **Recovery Manager** が削除することを意味します。このため、**Recovery Manager** は `*/oradata/trgt/arch/dest_1` および `*/oradata/trgt/arch/dest_2` の両方からログをすべて削除できます。

たとえば、次のコマンドを実行し、10 日前の時点からのリカバリに使用できるログをすべてバックアップし、指定した時点からのログをすべてディスクから削除します。

```
BACKUP DEVICE TYPE sbt
ARCHIVELOG ALL FROM TIME 'SYSDATE-10'
DELETE ALL INPUT;
```

## 差分増分バックアップの実行 : 例

差分増分バックアップには、同レベル以下で最新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ています。最初の増分バックアップは、使用したブロックすべてを含んだレベル 0 のバックアップにしてください。次に示すのは、レベル 0 の基本バックアップです。

```
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含まれます。前回のレベル 1 のバックアップが使用できない場合、**Recovery Manager** はベースとなるレベル 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。次に示すのは、レベル 1 のデータベースのバックアップです。

```
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。作成しない場合、**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ため、表領域またはデータベースの増分バックアップに失敗します。次に示すのは、レベル 0 の単一の表領域のバックアップです。

```
BACKUP INCREMENTAL LEVEL 0 TABLESPACE users2;
```

増分バックアップは NOARCHIVELOG モードで実行できますが、バックアップには一貫性が必要です。このため、オンラインの増分バックアップはできません。

## 累積増分バックアップの実行 : 例

レベル  $n$  の累積増分バックアップには、レベル  $n-1$  以下の最新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大きい記憶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、リストア操作を重要視する場合は、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、使用したブロックすべてを含んだレベル 0 のバックアップとす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 の最新のバックアップ以降に変更されたブロックがすべて含まれます。前回のレベル 1 が使用不可能であった場合に限り、レベル 0 の基礎バックアップ以降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 のバックアップのいずれ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。

```
BACKUP INCREMENTAL LEVEL 2 CUMULATIVE DATABASE; # blocks changed since level 0 or level 1
```

## チャンネルによるバックアップ作業負荷分散の決定 : 例

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ときには、Recovery Manager が複数のバックアップ・セットの書込みもパラレルに行います。割り当てられたサーバー・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共有します。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。

Recovery Manager 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ると、*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ませることができます。

たとえば、この例では、Recovery Manager がディスクにバックアップするチャンネルを指定し、どれを sbt に指定するかも指定して、バックアップ処理をパラレル化しています。

```
RUN
{
  ALLOCATE CHANNEL ch1 DEVICE TYPE DISK FORMAT = '/backup/df/%U';
  ALLOCATE CHANNEL ch2 DEVICE TYPE DISK FORMAT = '/backup/cf/%U';
  ALLOCATE CHANNEL ch3 DEVICE TYPE sbt;
  BACKUP
    # channel ch1 backs up datafiles to /backup/df directory
    DATAFILE 1,2,3,4
    CHANNEL ch1
    # channel ch2 backs up control file copy to /backup/cf directory
    CONTROLFILECOPY '/tmp/control01.ctl'
    CHANNEL ch2
    # channel ch3 backs up archived redo logs to tape
    ARCHIVELOG FROM TIME 'SYSDATE-14'
    CHANNEL ch3;
}
```

自動チャンネルを使用して、DISK および sbt に同時にバックアップすることはできません。これらは手動で割り当てる必要があります。

## NOARCHIVELOG モードでのバックアップ: 例

次のスクリプトは、データベースを一貫性のデータベース全体のバックアップに適したモードに設定してから、バックアップします。このスクリプトは、停止、起動、停止、再起動を行った後で、バックアップの複数コピーを作成します。

```
# 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 suffered instance failure or was closed with SHUTDOWN
# ABORT before starting this script. The scripts performs 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;

# The database is cleanly closed and is now ready for a consistent backup. RMAN requires
# that the database be started and mounted to perform a backup.
STARTUP MOUNT;

# this example uses automatic channels to make the backup
BACKUP COPIES 2 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以降にオフラインや読取り専用になっていない表領域をスキップすると、バックアップからデータベースをリストアしたときに、その表領域が失われます。また、ディスクにバックアップするときは、宛先（ファイル・システムまたは RAW デバイス）に十分な空き領域があることを確認してください。

## Oracle Real Application Clusters 環境でのバックアップ : 例

/node1 は、Oracle Real Application Clusters 構成のノード 1 がアクセスできるローカル・ディレクトリで、/node2 はノード 2 がアクセスできるローカル・ディレクトリであるとし、ます。自動チャンネル構成は次のようになります。

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 2;
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
# the different CONNECT strings in the channels cause RMAN to enable
# the autolocation feature
CONFIGURE CHANNEL 1 DEVICE TYPE sbt CONNECT 'SYS/oracle@node_1';
CONFIGURE CHANNEL 2 DEVICE TYPE sbt CONNECT 'SYS/oracle@node_2';
```

この使用例では、ノード 1 の方がノード 2 より強力であるため、最大の表領域をノード 1 でバックアップします。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、2 つのノードに分散するものです。

```
BACKUP FILESPERSET 1
  (TABLESPACE system, tools, users, undotbs
   CHANNEL ORA_SBT_TAPE_1)
  (TABLESPACE cwm1ite, drsys, example, indx
   CHANNEL ORA_SBT_TAPE_2);
BACKUP FILESPERSET 20
  ARCHIVELOG ALL;
```

自動位置検索機能により、ノード 1 に接続しているチャンネルはノード 1 で読取り可能なアーカイブ・ログのみをバックアップし、ノード 2 に接続しているチャンネルはノード 2 で読取り可能なアーカイブ・ログのみをバックアップします。

**関連項目：** Oracle Real Application Clusters でのバックアップについて、詳細は『Oracle9i Real Application Clusters 管理』を参照してください。



## オペレーティング・システム・コピーのカタログ化 : 例

オペレーティング・システムの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てから、その内容をリカバリ・カタログでカタログ化できます。カタログ化できるのはコピーのみです。バックアップ・ピースは Oracle 独自の形式なので、オペレーティング・システムのユーティリティでは、Recovery Manager で読取り可能な形式でバックアップを書き込むことはできません。

データ・ファイルのコピーは、ユーザーが管理する方式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で、データ・ファイルがオンラインの場合は、ALTER TABLESPACE ... BEGIN BACKUP を発行します。たとえば、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 '/tmp/users01.dbf';
```

接続されているターゲット・データベース以外のデータベースからのデータ・ファイルのコピーをカタログしようとする、次のようなエラーが発生します。

```

RMAN-00571: =====
RMAN-00569:===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03009: catalog コマンド (デフォルト・チャネル上) が 08/29/2001 14:44:34 で失敗しました
ORA-19563: データ・ファイルのコピーのヘッダーの妥当性チェックでエラーが検出されました (ファイル
/tmp/tools01.dbf)。
```

## 長期バックアップの保持 : 例

保存方針を構成する場合は、特定のバックアップをこの方針から除外することがあります。たとえば、履歴レコードとして残すために毎年 1 回データベースの一貫性バックアップをアーカイブとして作成する場合があります。この長期バックアップは、リカバリを実行する対象のバックアップとしてではなく、特定の時点でのデータのアーカイブ・スナップショットとして機能します。

保存方針からバックアップを除外するには、BACKUP コマンドに KEEP オプションを指定します。また、LOGS または NOLOGS を指定すると、このバックアップをリカバリするときのために、Recovery Manager でアーカイブ・ログを保存するかどうかを指定することもできます。NOLOGS を指定する場合は、一貫性バックアップにする必要があります。

次の例は、データベースのバックアップを期限なしで保存しますが、このバックアップのリカバリに必要なアーカイブ・ログは保存しません。

```

SHUTDOWN IMMEDIATE;
STARTUP MOUNT; # put database in consistent state
BACKUP DATABASE KEEP FOREVER NOLOGS TAG 'db_archive_1'; # make long-term consistent backup

# mark backup as unavailable in the repository so that RMAN does not attempt to restore it
# unless explicitly specified on the RESTORE command
CHANGE BACKUP TAG 'db_archive_1' UNAVAILABLE;
SQL 'ALTER DATABASE OPEN';
```

## バックアップの最適化 : 例

バックアップの最適化を有効にするには、CONFIGURE BACKUP OPTIMIZATION コマンドを実行します。特定の条件に一致した (5-62 ページの「バックアップの最適化のアルゴリズム」を参照) 場合、Recovery Manager ではすでにバックアップ済みのファイルと同一のファイルのバックアップをスキップします。

最適化と保存方針を次のように構成したとします。

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;  
CONFIGURE BACKUP OPTIMIZATION ON;  
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 4 DAYS;
```

### データベース・バックアップの最適化 : 例

次のコマンドを毎晩実行し、データベースをテープにバックアップします。

```
BACKUP DATABASE;
```

バックアップの最適化が構成されているため、Recovery Manager がオフライン・データ・ファイルおよび読取り専用データ・ファイルのバックアップをスキップするのは、リカバリ・ウィンドウの最初の時点以降に最新のバックアップが作成されている場合のみです。Recovery Manager は、最新のバックアップがリカバリ・ウィンドウより古い場合、バックアップをスキップしません。たとえば、最適化によって、読取り専用データ・ファイル ?/oradata/trgt/history01.dbf を含むバックアップ・セットがリカバリ・ウィンドウ内に 1 つでも存在するかぎり、このファイルの新規バックアップが毎晩作成されることはなくなります。

たとえば、データ・ファイルの最新のバックアップが日曜日で、リカバリ可能ポイント（つまり、リカバリ・ウィンドウの最初の日付）が土曜日の場合、水曜日のバックアップを実行すると、Recovery Manager はこのデータ・ファイルをスキップします。金曜日になると、リカバリ可能ポイントは月曜日になるため、日曜日のバックアップは期間の範囲外になります。このため、金曜日のバックアップではデータ・ファイルはスキップされません。

### 単一テープへの日次アーカイブ・ログ・バックアップの最適化 : 例

すべてのアーカイブ・ログを毎晩バックアップするとします。ただし、各ログ順序番号の複数のコピーは不要です。そこで、バックアップの最適化を ON に構成し、スクリプト内の次のコマンドを毎晩午前 1 時に実行します。

```
BACKUP DEVICE TYPE sbt ARCHIVELOG ALL;
```

Recovery Manager は過去 24 時間以内に生成されたログを除き、すべてのログをスキップします。このようにして、各アーカイブ・ログのコピーが 1 つのみテープに保存されます。

## 複数テープへの日次アーカイブ・ログ・バックアップの最適化 : 例

この例では、テープにはまだ保存されていないログを 1 つのテープ・プールにバックアップし、同じログを 2 つ目のテープ・プールにバックアップします。さらに、古いログを削除します。

最初の手順では、ワнтаイム構成を実行します。

```
# configure backup optimization
CONFIGURE BACKUP OPTIMIZATION ON;
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
```

次に、毎晩同じ時刻に次のスクリプトを実行して、前日に生成されたログを 2 つのテープ・プールにバックアップします。

```
# The following command will back up just the archived logs that are not on tape. The
# first copies are saved to the tapes from the pool "archivelog_pool_1"
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt
  PARMS='NSR_DATA_VOLUME_POOL=ARCHIVELOG_POOL_1';
  BACKUP CHANNEL 'ORA_SBT_TAPE_1' ARCHIVELOG ALL;
}
# Make one more copy of the archived logs and save them to tapes from a different pool
RUN
{
  ALLOCATE CHANNEL c2 DEVICE TYPE sbt
  PARMS='NSR_DATA_VOLUME_POOL=ARCHIVELOG_POOL_2';
  BACKUP CHANNEL 'ORA_SBT_TAPE_2' ARCHIVELOG
  FROM TIME 'SYSDATE-1'
  UNTIL TIME 'SYSDATE'; # specify UNTIL so that RMAN does not archive current log
}
# Delete old logs - for example, delete logs created within the last week.
DELETE ARCHIVELOG ALL COMPLETED AFTER 'SYSDATE-7';
```

## アーカイブ・ログの週次セカンダリ・バックアップの作成 : 例

より洗練された使用例として、アーカイブ・ログを毎日テープにバックアップする必要があります。ただし、テープの障害の発生に備えて、週に1度ディスクからログを削除する前に、各ログ順序番号のコピーが別のテープに保存されるようにします。

まず、ワンタイム構成を実行します。

```
# configure backup optimization
CONFIGURE BACKUP OPTIMIZATION ON;
CONFIGURE DEVICE TYPE sbt PARALLELISM 1;
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
# configure a default channel that sends the backups to the tape pool called "first_copy"
CONFIGURE CHANNEL DEVICE TYPE sbt PARMS='ENV=(NSR_DATA_VOLUME_POOL=first_copy);'
```

最適化を使用可能にしているため、次のコマンドを毎晩実行し、まだバックアップされていない **first\_copy** プールに全アーカイブ・ログをバックアップできます。

```
BACKUP ARCHIVELOG ALL;
```

毎週金曜日の夜間に、すべてのアーカイブ・ログの追加バックアップを別のテープ・プールに作成します。また、バックアップの終了時に、テープ上に2つ以上のコピーがあるアーカイブ・ログをすべて削除します。したがって、次のスクリプトを実行します。

```
BACKUP ARCHIVELOG ALL; # backs up logs not already on tape to pool "first_copy"

RUN
{
  # manually allocate a channel so that you bypass backup optimization for this job.
  # specify that the backup run by this channel should go to the pool "second_copy."
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt
    PARMS='ENV=(NSR_DATA_VOLUME_POOL=second_copy)';
  BACKUP ARCHIVELOG UNTIL TIME 'SYSDATE' # use UNTIL clause so RMAN does not switch logs
    NOT BACKED UP 2 TIMES                # back up only logs without 2 backups on tape
    TAG SECOND_COPY;                     # specify TAG for convenience
}

# now delete from disk all logs that have been backed up to tape at least twice
DELETE ARCHIVELOG ALL
  BACKED UP 2 TIMES TO DEVICE TYPE sbt;
```

金曜日に実行するスクリプトでは、すべてのアーカイブ・ログの第2コピーが **second\_copy** テープ・プールに作成されます。バックアップ後は、テープをプール **second\_copy** から保管庫に送るか、ロックできるディスクに保管できます。このテープ・バックアップを使用する必要があるのは、プール **first\_copy** からのプライマリ・テープが破損した場合のみです。セカンダリ・テープは安全な場所に保管されているため、**Recovery Manager** でリカバリに使用されないように、バックアップに使用不可能マークを設定できます。

```
CHANGE BACKUP OF ARCHIVELOG TAG SECOND_COPY UNAVAILABLE;
```

## バックアップおよびコピー時のエラー処理 : 例

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算出され、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、チェックサムは算出されません。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は、そのチェックサムが妥当性チェック後、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けられます。

SET MAXCORRUPT FOR DATAFILE コマンドは、BACKUP または COPY で許容されるデータ・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、このコマンドは終了します。CHECK LOGICAL オプションを指定すると、Recovery Manager は論理的な破損と物理的な破損を検出します。

デフォルトでは、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に設定されています。次の表に示すように、終了を防ぐためのパラメータを指定できます。

指定するオプション	Recovery Manager がスキップするファイル
SKIP INACCESSIBLE	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、アクセス不可と判断されます。一部のオフライン・データ・ファイルは、ディスク上に残っているために読取りが可能です。他のデータ・ファイルは削除または移動されたためにアクセス不可となり、読取り不可となります。
SKIP OFFLINE	オフラインのデータ・ファイル。
SKIP READONLY	読取り専用表領域内のデータ・ファイル。

次の例では、自動チャネルを使用してデータベースをバックアップし、SYSTEM 表領域のデータ・ファイルに破損レベルを設定します。

```
RUN
{
  SET MAXCORRUPT FOR DATAFILE 1 TO 0;
  BACKUP DATABASE
    SKIP INACCESSIBLE
    SKIP READONLY
    SKIP OFFLINE;
}
```



---

## Recovery Manager によるリストアと リカバリ

この章では、Recovery Manager を使用したリストアおよびリカバリの実行方法について説明します。この章の項目は、次のとおりです。

- Recovery Manager を使用したリストアおよびリカバリ : 概要
- 基本的な Recovery Manager メディア・リカバリの実行
- サーバー・パラメータ・ファイルのリストア
- バックアップ制御ファイルを使用したリカバリの実行
- 新しい場所へのファイルのリストア
- 新しいホストへのデータベースのリストア
- Oracle Real Application Clusters データベースのリカバリ
- Recovery Manager を使用した RESETLOGS 操作によるリカバリ
- 障害時リカバリの実行
- Recovery Manager を使用したデータ・ブロックのリカバリ
- バックアップとコピーのリストアの妥当性チェック
- Recovery Manager のリストアおよびリカバリ例

**関連項目：** リストアおよびリカバリ操作の監視方法は、15-20 ページの「Recovery Manager ジョブの進行状況の監視」を参照してください。

## Recovery Manager を使用したリストアおよびリカバリ : 概要

一般に、データベースやデータベースのサブセットに対するリストアおよびリカバリは次の場合に行います。

- メディア障害によって制御ファイルまたはデータ・ファイルの一部またはすべてを破損した場合。
- 表の削除などのユーザー・エラーが発生した前の時点までデータベースをリカバリする場合。

データベースのリストアをテストする場合には、RESTORE ではなくまず DUPLICATE コマンドを実行してください。

**関連項目：** データベースを複製する方法は、[第 12 章「Recovery Manager を使用したデータベースの複製」](#)を参照してください。

## メディア・リカバリの一般的な手順

リストアおよびリカバリを Recovery Manager で実行するための基本手順は、次のとおりです。

1. リカバリを必要とするデータベース・ファイルを判断します。
2. データベースを、実行するリカバリのタイプに適した状態にします。たとえば、すべてのデータ・ファイルをリカバリする場合には、データベースをマウントします。単一の表領域またはデータ・ファイルをリカバリする場合には、データベースをオープンして表領域またはデータ・ファイルをオフラインにします。
3. RESTORE コマンドを使用して、必要なファイルをリストアします。
4. RECOVER コマンドを使用して、リストアしたファイルをリカバリします。
5. データベースを通常の状態にします。たとえば、クローズされているデータベースをオープンします。また、リカバリしたファイルがオフラインになっている場合には、それをすべてオンラインにします。

リストアおよびリカバリには様々な使用例があるため、実際のリカバリ手順は場合により異なります。

Oracle Enterprise Manager を使用している場合は、RESTORE および RECOVER コマンドを実行するかわりに、Recovery Manager コマンドライン・インタフェースを介してリカバリ・ウィザードを使用できます。リカバリ・ウィザードを介して、次の Recovery Manager リストアおよびリカバリ・タスクを実行できます。

- 現時点またはそれ以外の時点までの**データベース全体のリストアおよびリカバリ**
- 最新時点まで、または最新かそれ以前のバックアップからの、**表領域またはデータ・ファイルのリストアおよびリカバリ**



- データベースが NOMOUNT 状態で、リカバリ・カタログを使用するバックアップ構成がない場合の、自動バックアップからの**制御ファイルのリストア**
- 時刻、SCN またはログ順序を指定した**アーカイブ・ログのリストア**
- 破損リスト、データ・ファイルまたは表領域を使用した**データ・ブロックのリカバリ**

**関連項目：** Recovery Manager のリストアおよびリカバリの詳細は、『Oracle Enterprise Manager 管理者ガイド』を参照してください。

## リストアとリカバリの使用例の相違

リカバリを実行する前に、どのような条件の下でリカバリを実行するかを確認します。リカバリ手順は、次に示す条件により異なります。

- 現行の制御ファイルは使用可能か
- リカバリ・カタログを使用しているか
- リストア・ホストと元のターゲット・ホストは同一か
- リストアしたデータベース・ファイルに付けられる名前はターゲット・データベース・ファイルと同名か
- ターゲット・データベースは Oracle Real Application Clusters 構成内で実行しているか
- 現時点へのリカバリが必要なのか、現時点以外へのリカバリが必要なのか
- データベース全体をリカバリするのか
- 保持しているバックアップは、最後の RESETLOGS 以後に作成したのか
- 限定した数の破損データ・ブロックではなく、データ・ファイル全体をリカバリするのか

10-5 ページの「[基本的な Recovery Manager メディア・リカバリの実行](#)」では、一般的なリカバリの使用例を説明しています。この使用例は、メディア・リカバリの一般的なテンプレートを提供する意味では典型的なものです。この一般的なテンプレートは、リストアの使用例の一部にのみ対応しています。この章の、10-5 ページの「[基本的な Recovery Manager メディア・リカバリの実行](#)」以外の項では、様々なリストアの使用例を説明します。[表 10-1](#)では、一般的な手順とは異なるリカバリを行う場合に参照する項を、判断できます。

表 10-1 リカバリの使用例の判別

質問	YES の場合	NO の場合
現行の制御ファイルは使用可能ですか？	10-5 ページ「基本的な Recovery Manager メディア・リカバリの実行」	10-15 ページ「バックアップ制御ファイルを使用したリカバリの実行」
現行のサーバー・パラメータ・ファイルは使用可能ですか？	10-5 ページ「基本的な Recovery Manager メディア・リカバリの実行」	10-15 ページ「サーバー・パラメータ・ファイルのリストア」
リカバリ・カタログを使用していますか？	10-5 ページ「基本的な Recovery Manager メディア・リカバリの実行」	10-5 ページ「基本的な Recovery Manager メディア・リカバリの実行」を参照してください。バックアップ制御ファイルを使用していて、カタログを使用していない場合には、10-18 ページの「バックアップ制御ファイルを使用してリカバリ・カタログを使用しないリカバリの実行」を参照してください。
リストア・ホストはターゲット・ホストと同一ですか？	10-5 ページ「基本的な Recovery Manager メディア・リカバリの実行」	10-25 ページ「新しいホストへのデータベースのリストア」
リストアしたデータベース・ファイルは、元のデータベース・ファイルと同じ名前になりますか？	10-5 ページ「基本的な Recovery Manager メディア・リカバリの実行」	10-21 ページ「新しい場所へのファイルのリストア」
ターゲット・データベースは Oracle Real Application Clusters 内にありますか？	10-27 ページ「Oracle Real Application Clusters データベースのリカバリ」	10-5 ページ「基本的な Recovery Manager メディア・リカバリの実行」
完全リカバリを実行しますか？	10-9 ページ「完全リストアおよびリカバリの実行」	データベース全体のリカバリについては、10-11 ページの「不完全リストアおよびリカバリの実行」を参照してください。個別の表領域に対する Point-in-Time リカバリについては、第 11 章「Recovery Manager による表領域の Point-in-Time リカバリの実行」を参照してください。

表 10-1 リカバリの使用例の判別（続き）

質問	YES の場合	NO の場合
データベース全体をリカバリしますか？	10-9 ページ「デフォルトの場所でのデータベース全体のリストアおよびリカバリ」	データベースのサブセットに対する完全リカバリを実行する場合は、10-10 ページの「データベースのサブセットのリストアおよびリカバリ」を参照してください。データベースのサブセットに対する Point-in-Time リカバリを実行する場合は、第 11 章「Recovery Manager による表領域の Point-in-Time リカバリの実行」を参照してください。
最新のデータベースの RESETLOGS の後に、必要なバックアップが行われていますか？	10-5 ページ「基本的な Recovery Manager メディア・リカバリの実行」	10-28 ページ「Recovery Manager を使用した RESETLOGS 操作によるリカバリ」
少数の破損ブロックではなく、データ・ファイル全体のリカバリが必要ですか？	10-5 ページ「基本的な Recovery Manager メディア・リカバリの実行」	10-40 ページ「Recovery Manager を使用したデータ・ブロックのリカバリ」

## 基本的な Recovery Manager メディア・リカバリの実行

この項では、次の項目について説明します。

- 基本的なメディア・リカバリ
- メディア・リカバリの準備
- 完全リストアおよびリカバリの実行
- 不完全リストアおよびリカバリの実行

## 基本的なメディア・リカバリ

この項のすべての手順は、次の項目を前提としています。

- 現行の制御ファイルが使用可能であり、**Recovery Manager** リポジトリとしてリカバリ・カタログまたは現行の制御ファイルを使用していること。
- リストア・ノードとターゲット・ノードが同じノードであり、リストアしたデータ・ファイルは元のターゲット・データ・ファイルと同じファイル名になること。
- ターゲット・データベースが **Oracle Real Application Clusters** 構成内では実行していないこと。
- 完全リカバリまたは不完全リカバ리를、データ・ファイル全体（個々のデータ・ブロックではない）に対して実行すること。不完全リカバ리를実行する場合には、データベース全体のリカバリを行うこと。完全リカバ리를実行する場合には、データベース全体またはデータベースのサブセットにリカバリを行うこと。

**RESTORE** および **RECOVER** コマンドを使用して、リカバリを実行します。**RESTORE** コマンドは、ディスクまたはメディア・マネージャからバックアップをリストアします。ただしイメージ・コピーのリストアはディスクからのみ行います。

---

---

**注意：** **Oracle9i** では、以前の **Recovery Manager** リリースと異なり **RESTORE** コマンドが最適化されています。この最適化では、データ・ファイルのヘッダーに予想された情報が含まれていないファイルのみがリストアされます。つまり、リストアの必要がないファイルに対して、**Recovery Manager** はリストアを実行しません。この動作は、**FORCE** オプションを指定することで変更可能です。

---

---

自動チャンネルの構成が行われている場合、**Recovery Manager** はパラレル化の設定に応じて、使用可能なデバイス・タイプに対して構成されたすべてのチャンネルを割り当てます。たとえば、2 つの **sbt** チャンネルを構成し、並列性を 1 に設定して、**Disk** チャンネルのパラレル化を 3 に設定すると、**Recovery Manager** は自動的に 1 つの **sbt** チャンネルと 3 つの **Disk** チャンネルを割り当てます。リストアを行うために、**Recovery Manager** では構成したすべてのチャンネルを割り当てますが、**DEVICE TYPE** オプションにより **Recovery Manager** がリストアを行うデバイス・タイプを制限することもできます。

手動でチャンネルの割り当てを行う場合には、ファイルのリストア時に適切な **Disk** チャンネルや **sbt** チャンネルを割り当ててください。適切なデバイス・タイプが割り当てられていないと、**Recovery Manager** はリストアを行うバックアップ・セットやコピーの検索ができません。このため、**RESTORE** コマンドは失敗します。

## メディア・リカバリの準備

いつ、どのような方法でリカバリを行うかは、データベースの状態と、そのデータ・ファイルの場所によって異なります。できるかぎり、固定ビューを問い合わせる必要となる情報を取得してください。

### メディア・リカバリの必要性を判断する方法

1. SQL\*Plus を起動し、ターゲット・データベースに接続します。たとえば、次のコマンドを発行して trgt に接続します。

```
% sqlplus 'SYS/oracle@trgt AS SYSDBA'
```

2. 次の SQL 文を実行し、データベースのステータスを判断します。

```
SELECT STATUS FROM V$INSTANCE;
```

```
STATUS
-----
OPEN
```

ステータスが OPEN ならば、そのデータベースはオープンしていますが、表領域の一部またはそのデータ・ファイルには、リストアまたはリカバリが必要である可能性が残されています。

3. V\$DATAFILE\_HEADER ビューの RECOVER 列と ERROR 列をチェックします。これらの列は、データ・ファイルのステータスを示します。次の SQL スクリプトを実行して、データ・ファイルのヘッダーをチェックし、この例の次にある表に従って対応してください。

```
COL FILE# FORMAT 999
COL STATUS FORMAT A7
COL ERROR FORMAT A10
COL TABLESPACE_NAME FORMAT A10
COL NAME FORMAT A30

SELECT FILE#, STATUS, ERROR, RECOVER, TABLESPACE_NAME, NAME
FROM V$DATAFILE_HEADER
/
```

ERROR 列	RECOVER 列	解決策
NULL	NO	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かぎり、データ・ファイルをリストアするか、またはそのデータ・ファイルのコピーに切り替えてください。
NULL	YES	データ・ファイルをリカバリしてください。RECOVER コマンドでは、最初に適切な増分バックアップが適用され、次に REDO ログが適用されます。Recovery Manager は、必要に応じて増分バックアップとアーカイブ REDO ログをリストアします。
NOT NULL		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かぎり、データ・ファイルをリストアするか、またはそのデータ・ファイルのコピーに切り替えてください。

**注意：** V\$DATAFILE\_HEADER では、各データ・ファイルのヘッダー・ブロックのみが読み取られます。このため、データ・ファイルのリストアが必要となるような問題のすべてが検出されるわけではありません。たとえば、データ・ファイルに破損データ・ブロックが含まれていても、そのデータ・ファイルのヘッダー・ブロックに異常がないと、Oracle ではエラーがレポートされません。

4. 必要に応じて、V\$DATAFILE および V\$TABLESPACE の問合せを行い、ステータスおよびエラー情報に加えて、リカバリが必要なデータ・ファイルの表領域名を取得します。たとえば、次のように SQL\*Plus スクリプトを実行します。

```
COL DF# FORMAT 999
COL DF_NAME FORMAT A30
COL TBSP_NAME FORMAT A7
COL STATUS FORMAT A7
COL ERROR FORMAT A10
COL CHANGE# FORMAT 99999999
SELECT r.FILE# AS df#, d.NAME AS df_name, t.NAME AS tbsp_name,
       d.STATUS, r.ERROR, r.CHANGE#, r.TIME
FROM V$RECOVER_FILE r, V$DATAFILE d, V$TABLESPACE t
WHERE t.TS# = d.TS#
AND d.FILE# = r.FILE#
/
```

このスクリプトは、次のような出力を生成します。

DF#	DF_NAME	TBSP_NA	STATUS	ERROR	CHANGE#	TIME
7	/oracle/oradata/trgt/tools01.dbf	TOOLS	OFFLINE	OFFLINE	0	
				NORMAL		

メディア・リカバリが必要な場合には、ERROR 列に問題が示され、リカバリが必要となります。

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

## 完全リストアおよびリカバリの実行

リカバリが必要なデータ・ファイルを決定した後で、データベース内のすべてのデータ・ファイルまたはデータ・ファイルのサブセットのみをリストアできます。

この項では、次の項目について説明します。

- デフォルトの場所でのデータベース全体のリストアおよびリカバリ
- データベースのサブセットのリストアおよびリカバリ

### デフォルトの場所でのデータベース全体のリストアおよびリカバリ

この使用例では、現行の制御ファイルがあり、すべてのデータ・ファイルは破損または消失しています。データベース全体をリストアして、リカバリする必要があります。

#### 現行の制御ファイルが使用可能な場合の、データベースのリストアおよびリカバリ手順

1. ターゲット・データベースに（必要があれば、リカバリ・カタログ・データベースにも）接続した後、データベースがマウントされていることを確認します。

```
STARTUP MOUNT
```

2. 次の操作を行います。
  - a. 自動チャネルの構成が行われていない場合には、手動で 1 つ以上のチャネルを割り当てます。SHOW ALL を実行して、現在の構成を確認します。
  - b. RESTORE コマンドを使用して、データベースをリストアします。次に、RECOVER コマンドを使用して、データベースをリカバリします。

この例では、リカバリの実行に自動チャネルを使用して、読取り専用の history 表領域はスキップします。

```
RESTORE DATABASE;
RECOVER DATABASE
  # optionally, delete logs restored for recovery and limit disk space used
  DELETE ARCHIVELOG MAXSIZE 1M
  SKIP TABLESPACE history; # optionally, skip the recovery of some tablespaces
```

3. 正常にリカバリしたかどうかを出力で調べます。リカバリが正常に行われた場合には、データベースをオープンします。

```
ALTER DATABASE OPEN;
```

## データベースのサブセットのリストアおよびリカバリ

この使用例では、データ・ファイルの一部のみが破損しています。操作対象となる表領域の決定は、10-7 ページの「メディア・リカバリの準備」の手順に従い行われているものとします。

次に示す手順では、データベースがオープンしているため、リカバリする表領域をオフラインにする必要があります。

### 表領域をデフォルトの場所にリカバリする方法

1. ターゲット・データベースに（必要があれば、リカバリ・カタログ・データベースにも）接続した後、データベースがマウントまたはオープンされていることを確認します。たとえば、次のように実行します。

```
STARTUP MOUNT
```

2. 対象となるデータベースを、リストアした後にリカバリします。次の操作を行います。
  - a. 自動チャネルの構成を行っていない場合、手動で1つ以上のチャネルを割り当てます。SHOW ALL を実行して、現在の構成を確認します。
  - b. リカバリを必要とする表領域がオフラインでない場合には、これをオフラインにするために ALTER TABLESPACE ... OFFLINE IMMEDIATE を使用します。
  - c. 表領域またはデータ・ファイルをリストアするために RESTORE コマンドを使用します。続けてリカバリを行うために RECOVER コマンドを使用します。

この例では、users 表領域のリストアおよびリカバリを行います。

```
SQL 'ALTER TABLESPACE users OFFLINE IMMEDIATE';  
RESTORE TABLESPACE users;  
RECOVER TABLESPACE users;
```

3. リカバリが正常に終了した場合には、表領域をオンラインにします。

```
SQL 'ALTER TABLESPACE users ONLINE';
```



## 不完全リストアおよびリカバリの実行

この項では、次の項目について説明します。

- 不完全リカバリ
- 現行の制御ファイルによる不完全リカバリの実行

### 不完全リカバリ

Recovery Manager では、指定した時刻、SCN またはログ順序番号までのデータベース全体のリカバリを実行できます。このタイプのリカバリは、使用可能なすべての REDO ログが完全には使用されないため、**不完全リカバリ**と呼ばれます。データベース全体の不完全リカバリは、**データベースの Point-in-Time リカバリ (DBPITR)** とも呼びます。

データベースの不完全リカバリでは、RESETLOGS オプションによるデータベースのオープンが必要になります。このオプションを使用すると、オンライン REDO ログに新しいタイム・スタンプと SCN が記録されます。その結果、不要なアーカイブ REDO ログの適用によるデータ・ファイル破損の危険性がなくなります。すべてのデータ・ファイルをリカバリする必要があることに注意してください。データ・ファイルには、RESETLOGS の前にリカバリできないものと RESETLOGS の後にリカバリできないものがあります。事実、Oracle ではデータ・ファイルがオフラインのときに、ログをリセットすることはできません。唯一の例外は、データ・ファイルが NORMAL モードでオフラインにされているか、または読取り専用の場合です。RESETLOGS の後には、ファイルを読取り専用にするか、NORMAL モードでオフラインにされた表領域をオンラインにできます。これは、これらのファイルや表領域に REDO を適用する必要がないためです。

DBPITR を実行する最も簡単な方法は、SET UNTIL コマンドを使用することです (RESTORE および RECOVER コマンドで個別に UNTIL 句を指定するのではなく)。これにより、以降のどの RESTORE、SWITCH および RECOVER コマンドにも、同一の RUN ジョブ内で必要な時刻が設定されます。SET UNTIL コマンドを RESTORE の後、RECOVER の前に指定すると、データベースを必要な時刻までリカバリできない場合があります。これは、リストアされたファイルが設定時刻より後のタイム・スタンプを持つ可能性があるためです。したがって、SET UNTIL コマンドを、RESTORE コマンドの前に指定することをお勧めします。

### 現行の制御ファイルによる不完全リカバリの実行

データベースの Point-in-Time リカバリを実行するには、データベースをクローズしておく必要があります。ある時刻までのリカバリを行う場合は、Recovery Manager を起動する前に、時刻書式の環境変数を設定する必要があります (8-28 ページの「[Recovery Manager のグローバリゼーション・サポートの環境変数の設定](#)」を参照)。次に示すのは、グローバリゼーション・サポートの設定のサンプルです。

```
NLS_LANG = american_america.us7ascii  
NLS_DATE_FORMAT="Mon DD YYYY HH24:MI:SS"
```

指定した時間、SCN またはログ順序番号までのデータベースのリカバリ手順

- 1. ターゲット・データベースに（必要があれば、リカバリ・カタログ・データベースにも）接続した後で、データベースがマウントされていることを確認します。データベースがオープン状態の場合は、停止してからマウントします。

```
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
```

- 2. リカバリを終了する時刻、SCN またはログ順序番号を判断します。たとえば、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は、削除が発生した直前の午前 9 時までのリカバリができます。その時刻以降にデータベースに対して行った変更は消失します。

また、alert.log を調べてイベントの SCN を検索することで、前の SCN までのリカバリを行うこともできます。別な方法としては、ログ順序番号に含まれているリカバリ終了の SCN を判断して、そのログによるリカバリを行うことができます。たとえば、V\$LOG\_HISTORY を問い合せて、アーカイブしてあるログを表示します。

RECID	STAMP	THREAD#	SEQUENCE#	FIRST_CHAN	FIRST_TIM	NEXT_CHANG
1	344890611	1	1	20037	24-SEP-01	20043
2	344890615	1	2	20043	24-SEP-01	20045
3	344890618	1	3	20045	24-SEP-01	20046

- 3. RUN コマンド内で、次の操作を実行します。
  - a. リカバリの終了時刻、SCN またはログ順序番号を設定します。時刻を指定する場合には、NLS\_LANG および NLS\_DATE\_FORMAT 環境変数で指定された日付書式を使用します。
  - b. 自動チャンネルの構成を行っていない場合、手動で 1 つ以上のチャンネルを割り当てます。
  - c. データベースをリストアしてリカバリします。

次の例では、11 月 15 日午前 9 時までの不完全リカバリを実行します。

```
RUN
{
  SET UNTIL TIME 'Nov 15 2001 09:00:00';
  # SET UNTIL SCN 1000;          # alternatively, you can specify SCN
  # SET UNTIL SEQUENCE 9923;    # alternatively, you can specify log sequence
  number
  RESTORE DATABASE;
  RECOVER DATABASE;
}
```

- リカバリが正常に終了した場合には、データベースをオープンした後、オンライン・ログをリセットします。

```
ALTER DATABASE OPEN RESETLOGS;
```

- ただちに、データベースのバックアップを行うことをお勧めします。このときデータベースは、できるかぎりマウントした状態にします（オープン状態のデータベースでのデータ消失を回避するため）。データベースは新しいインカネーションであるため、RESETLOGS 以前に作成したバックアップの使用は難しくなります。たとえば、次に示すようにデータベースのバックアップを実行します。

```
SHUTDOWN IMMEDIATE
STARTUP MOUNT
BACKUP DATABASE;
ALTER DATABASE OPEN;
```

## サーバー・パラメータ・ファイルのリストア

Recovery Manager では、サーバー・パラメータ・ファイルをデフォルト位置またはそれ以外の位置にリストアできます。また、サーバー・パラメータ・ファイルとして、またはクライアント側の初期化パラメータ・ファイルとしてリストアできます。

サーバー・パラメータ・ファイルのリストアを行う際には、次の制限と使用方法に注意してください。

- インスタンスがすでにサーバー・パラメータ・ファイルを使用して起動されている場合は、既存のサーバー・パラメータ・ファイルを上書きできません。
- インスタンスがクライアント側の初期化パラメータ・ファイルを使用して起動されている場合に、TO 句を使用しなければ、Recovery Manager ではサーバー・パラメータ・ファイルがデフォルトの位置にリストアされます。デフォルトの位置はプラットフォーム固有です（Solaris の場合は ?/dbs/spfile@.ora など）。

### サーバー・パラメータ・ファイルのリストア

- ターゲット・データベースと、必要に応じてリカバリ・カタログ・データベースに接続します。たとえば、次のように実行します。

```
% rman TARGET / CATALOG rman/cat@catdb
```

- カタログに接続しており、ターゲット・データベースの DB\_NAME がカタログ内で一意の場合は、この手順をスキップします。それ以外の場合は、ターゲット・データベースの DBID を設定します。たとえば、次のようにします。

```
SET DBID 676549873;
```

3. インスタンスをシャットダウンし、マウントしないで再起動します。サーバー・パラメータ・ファイルが失われるため、**Recovery Manager** は仮パラメータ・ファイルを使用してインスタンスを起動します。たとえば、次のようにします。

```
STARTUP FORCE NOMOUNT;
```

4. サーバー・パラメータ・ファイルのリストアします。デフォルトの位置にリストアする場合は、単に次のように実行します。

```
RESTORE SPFILE; # if you are using a catalog
RESTORE SPFILE FROM AUTOBACKUP; # if in NOCATALOG mode
```

デフォルト以外の位置にリストアする場合は、次の例のようなコマンドを実行できます。

```
RESTORE SPFILE TO '/tmp/spfileTEMP.ora'; # if you are using a catalog
RESTORE SPFILE TO '/tmp/spfileTEMP.ora' FROM AUTOBACKUP; # if in NOCATALOG mode
```

次の例のように `TO PFILE 'filename'` 句を使用すると、サーバー・パラメータ・ファイルをクライアント側の初期化パラメータ・ファイルとしてリストアできます。

```
RESTORE SPFILE TO PFILE '/tmp/initTEMP.ora';
```

5. リストアしたファイルを使用してインスタンスを再起動します。デフォルト以外の位置にあるサーバー・パラメータ・ファイルを使用して再起動する場合は、`SPFILE=new_location` を使用して新規のクライアント側初期化パラメータ・ファイルを作成します。この場合、`new_location` はリストアされたサーバー・パラメータ・ファイルのパス名です。次に、クライアント側の初期化パラメータ・ファイルを使用してインスタンスを再起動します。たとえば、次のようにします。

```
HOST 'echo "SPFILE=/tmp/spfileTEMP.ora" > /tmp/init.ora';
STARTUP FORCE PFILE=/tmp/init.ora; # starts instance with /tmp/spfileTEMP.ora
```

サーバー・パラメータ・ファイルをクライアント側の初期化パラメータ・ファイルとしてリストアした場合は、単にこのリストアしたファイルのパス名を指定します。たとえば、次のようにします。

```
STARTUP FORCE PFILE=/tmp/pfileTEMP.ora; # starts instance with
/tmp/pfileTEMP.ora
```

## バックアップ制御ファイルを使用したりカバリの実行

現行の制御ファイルのコピーのすべてが消失または破損した場合には、バックアップ制御ファイルのリストアとマウントを行った後で、リカバリを実行します。この手順は、カタログを使用している場合と使用していない場合で異なります。どちらの場合にも適用される使用方法と制限事項を次に示します。

- バックアップ制御ファイルのリストアした後は、リストアされたデータ・ファイルが存在しない場合でも、必ず RECOVER コマンドを実行します。
- バックアップ制御ファイルのリストアすると、ローカル管理の一時表領域から一時ファイルのエントリが削除されます。したがって、RESETLOGS オプションを使用してオープンした後に、これらの表領域用に新規一時ファイルを追加する必要があります。追加しないと、Oracle ではソート時に「ORA-25153: 一時表領域が空です。」というエラーが表示される場合があります。
- バックアップ制御ファイルを使用した完全または不完全リカバリを実行した後に、RESETLOGS オプションでデータベースをオープンする必要があります。
- オンライン REDO ログがアクセス不可能な場合、オンライン REDO ログ内で最初の SCN の前の SCN まで不完全リカバリを実行する必要があります。この制限事項が必要になるのは、Recovery Manager ではオンライン・ログのバックアップが行われなためです。
- Oracle9i で Recovery Manager を起動すると、特定の場所のオンラインおよびアーカイブ REDO ログをリカバリ中に自動的に検索します。これらは、Recovery Manager リポジトリにも、Recovery Manager が検索するどのカタログにも記録されていません。Recovery Manager は、現行のログの書式で、現行のアーカイブ先のすべてに対して有効なアーカイブ・ログの検索を試行します。現行の書式は、インスタンス（または Real Application Clusters のすべてのインスタンス）の起動に使用された初期化パラメータ・ファイルで指定されます。同様に、Recovery Manager は、制御ファイルで指定されるファイル名を使用して、オンライン REDO ログの検索を実行します。

Recovery Manager では、リカバリ中にアーカイブ先や書式を変更したことにより、必要とされるオンラインまたはアーカイブ・ログを自動的にカタログに追加できない場合や、制御ファイルのバックアップ後に新規のオンライン・ログ・メンバーを追加した場合には、次に示すようなエラーがレポートされます。

```

RMAN-00571: =====
RMAN-00569:===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: recover コマンドが 08/29/2001 14:23:09 で失敗しました
RMAN-06054: メディア・リカバリが不明なログを要求しています: スレッド 1 scn 86945

```

この場合、CATALOG コマンドを使用して、手動で必要なログをリポジトリに追加する必要があります。これにより、リカバリを進めることができます。カタログ化の手順の詳細は、18-18 ページの「[アーカイブ・ログおよびユーザー管理コピーのカタログ化](#)」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。

- [バックアップ制御ファイルおよびリカバリ・カタログを使用したりカバリの実行](#)
- [バックアップ制御ファイルを使用してリカバリ・カタログを使用しないリカバリの実行](#)

## バックアップ制御ファイルおよびリカバリ・カタログを使用したりカバリの実行

リカバリ・カタログを使用していて使用可能なバックアップ制御ファイルも存在する場合の手順は、10-9 ページの「[完全リストアおよびリカバリの実行](#)」と実質上同じです。この項の手順では、制御ファイルをデフォルトの場所にリストアすることを前提とします。制御ファイルを新しい場所にリストアする必要がある場合は、10-22 ページの「[新しい場所への制御ファイルのリストア](#)」の方法を参照してください。

バックアップ制御ファイルとカタログを使用してリストア操作を実行する場合、**Recovery Manager** は、リストアしたバックアップの構造が制御ファイルに反映されるように、自動的に調整します。

次に示す手順では、2 つ以上のターゲット・データベースを同じ名前でカタログに登録していないことが前提です。複数のターゲット・データベースが同じ名前で登録されている場合には、**SET DBID** コマンドで **DBID** を指定する必要があります。これにより、**Recovery Manager** にリストアする制御ファイルを認識させます。**DBID** は、データベースの一意の数値 ID です。

**関連項目：** **DBID** の設定方法は、10-18 ページの「[バックアップ制御ファイルを使用してリカバリ・カタログを使用しないリカバリの実行](#)」を参照してください。

### バックアップ制御ファイルとリカバリ・カタログを使用したデータベースのリカバリ

1. 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後、データベースをマウントしないでインスタンスを起動します。

STARTUP NOMOUNT

2. バックアップ制御ファイルをリストアした後、データベースのリストアおよびリカバリを行います。次の操作を行います。
  - a. **RESTORE CONTROLFILE** コマンドを実行して、**CONTROL\_FILES** 初期化パラメータで指定した、すべてのデフォルトの場所に制御ファイルをリストアします。より古いバックアップから制御ファイルをリストアするためには、**SET UNTIL** を実行するか、**RESTORE CONTROLFILE** コマンドで **UNTIL** 句を指定します。
  - b. リストアした制御ファイルを使用してデータベースをマウントします。
  - c. 必要に応じて、**SET UNTIL** コマンドを実行して不完全リカバリを行います。**RESTORE** および **RECOVER** コマンドに **UNTIL** 句を指定することもできます。

- d. 10-5 ページの「[基本的な Recovery Manager メディア・リカバリの実行](#)」の説明のように、データベースのリストアおよびリカバリを行います。

この例では、制御ファイルをデフォルトの場所にリストアした後、データベースをリストアして、完全にリカバリします。

```
RESTORE CONTROLFILE;  
ALTER DATABASE MOUNT;  
RESTORE DATABASE;  
RECOVER DATABASE;
```

3. リカバリが正常に終了した場合には、データベースをオープンした後、オンライン・ログをリセットします。

```
ALTER DATABASE OPEN RESETLOGS;
```

4. データベースでローカル管理の一時表領域が使用されている場合は、これらの表領域に新規の一時ファイルを追加します。たとえば、次のようにします。

```
SQL "ALTER TABLESPACE temp ADD TEMPFILE '"/oradata/trgt/temp01.dbf"' REUSE";
```

5. ただちに、データベースのバックアップを行うことをお勧めします。このときデータベースは、できるかぎりマウントした状態にします（オープン状態のデータベースでのデータ消失を回避するため）。データベースは新しいインカネーションであるため、RESETLOGS より前のバックアップの使用は難しくなります。たとえば、次に示すようにターゲット・データベースのバックアップを実行します。

```
SHUTDOWN IMMEDIATE  
STARTUP MOUNT  
BACKUP DATABASE;  
ALTER DATABASE OPEN;
```

## バックアップ制御ファイルを使用してリカバリ・カタログを使用しないリカバリの実行

この項では、Recovery Manager の制御ファイルのバックアップがあり、リカバリ・カタログを使用していないことを前提とします。制御ファイルの自動バックアップ機能をターゲット・データベースに対して有効にしてあると仮定すると、制御ファイルの自動バックアップのリストアが可能です。自動バックアップではデフォルトの書式が使用されるため、Recovery Manager では、使用可能なバックアップがリストされた使用可能なリポジトリを保持していない場合でもリストアが可能です。自動バックアップをデフォルトまたは新しい場所にリストアします。Recovery Manager では、制御ファイルを自動的に CONTROL\_FILES で特定したすべての場所に複製します。

---

**注意：** バックアップ・ピースの名前がわかっている場合（たとえば、メディア・マネージャ、またはディスク上にあるピース）には、RESTORE CONTROLFILE FROM '*filename*' コマンドを使用して、ピースの名前を指定できます。サーバーでは、各自動バックアップの位置がアラート・ログに記録されます。

---

リカバリ・カタログに接続していないために、すべてのバックアップに必要なレコードは、制御ファイルに保持されている必要があります。バックアップが制御ファイルにリストされていないと、Recovery Manager ではそれらをリストアできません。ディスク上で場所を特定されているデータ・ファイルのコピーが、制御ファイル内では場所を特定されていない場合には、これを制御ファイルのリポジトリに追加するために CATALOG コマンドを使用します。このカタログ化の手順の詳細は、18-18 ページの「[アーカイブ・ログおよびユーザー管理コピーのカタログ化](#)」を参照してください。

制御ファイルのリストア時にはリポジトリを使用できないため、SET DBID コマンドを実行してターゲット・データベースを識別します。SET DBID コマンドを実行する必要があるのは、次の特殊な場合のみです。

- リカバリ・カタログに接続しておらず、制御ファイルまたはサーバー・パラメータ・ファイルをリストアする必要がある場合
- リカバリ・カタログに接続しており、制御ファイルをリストアする必要があるが、データベース名がリカバリ・カタログ内で一意でない場合
- サーバー・パラメータ・ファイルが消失し、それをリストアする必要がある場合



### 制御ファイルの自動バックアップを使用してリカバリ・カタログを使用しないデータベースのリカバリ

1. **Recovery Manager** を起動し、ターゲット・データベースに接続します。たとえば、次のように実行します。

```
CONNECT TARGET /
```

2. データベースをマウントせずにターゲット・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. ターゲット・データベースのデータベース識別子を **SET DBID** で設定します。ターゲットに接続すると、**Recovery Manager** が **DBID** を表示します。また、**LIST** を実行するか、カタログを問い合わせるか、制御ファイルの自動バックアップのファイル名を調べて取得する方法もあります（10-46 ページの「[カタログ内の複数データベースが同じ名前を共有する場合のリストア:例](#)」を参照してください）。たとえば、次のように実行します。

```
SET DBID 676549873;
```

4. 自動バックアップ制御ファイルをリストアした後、リカバリを実行します。次の操作を行います。
  - a. 必要に応じて、最新のバックアップのタイム・スタンプを指定します。**Recovery Manager** は、このタイム・スタンプを使用してリストアする制御ファイルの自動バックアップを検索します。
  - b. 非デフォルトの書式を制御ファイルの作成に使用していた場合には、非デフォルトの書式を指定して制御ファイルのリストアを行います。
  - c. 制御ファイルの自動バックアップを作成したチャンネルがデバイス・タイプ **sbt** の場合には、1 つ以上の **sbt** チャンネルを割り当てる必要があります。使用可能なリポジトリが存在しないと、自動チャンネルは使用できないからです。ただし、自動バックアップがディスク・チャンネルに作成されていた場合には、手動でチャンネルを割り当てる必要はありません。
  - d. 制御ファイルの自動バックアップをリストアします。必要に応じて、**Recovery Manager** が検索でさかのぼる最大の日数（366 まで）と、最初の日を検索するために使用する初期順序番号を設定します。
  - e. データベースをマウントします。この時点でリポジトリが使用可能になるため、すべての構成済みの自動チャンネルも使用可能になります。

- f. オンライン・ログがアクセス不可能な場合は、10-11 ページの「[不完全リストアおよびリカバリの実行](#)」の説明に従い、データベースのリストアおよびリカバリを行います。UNTIL 句にオンライン REDO ログの前の時刻、ログ順序または SCN を設定して、リカバリを終了させる必要があります。オンライン・ログが使用可能な場合は、10-11 ページの「[完全リストアおよびリカバリの実行](#)」の説明に従い、データベースのリストアおよびリカバリを行います。

この例では、オンライン REDO ログは消失しています。例では制御ファイルの自動バックアップのリストアを制限して、データベースのリカバリを最新のアーカイブ・ログであるログ順序 13243 まで実行します。

```
RUN
{
  # Optionally, set upper limit for eligible time stamps of control file backups
  # SET UNTIL TIME '09/10/2000 13:45:00';
  # Specify a nondefault autobackup format only if required
  # SET CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO
  '?/oradata/%F.bck';
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt; # manually allocate one or more channels
  RESTORE CONTROLFILE FROM AUTOBACKUP
    MAXSEQ 100          # start at sequence 100 and count down
    MAXDAYS 180;        # start at UNTIL TIME and search back 6 months
  ALTER DATABASE MOUNT DATABASE;
}
# uses automatic channels configured in restored control file
RESTORE DATABASE UNTIL SEQUENCE 13243;
RECOVER DATABASE UNTIL SEQUENCE 13243; # recovers to most recent archived log
```

5. リカバリが正常に終了した場合には、データベースをオープンした後、オンライン・ログをリセットします。

```
ALTER DATABASE OPEN RESETLOGS;
```

6. ただちに、データベースのバックアップを行うことをお勧めします。このときデータベースは、できるかぎりマウントした状態にします（オープン状態のデータベースでのデータ消失を回避するため）。データベースは新しいインカネーションであるため、RESETLOGS 以前に作成したバックアップの使用は難しくなります。たとえば、次のように入力します。

```
SHUTDOWN IMMEDIATE
STARTUP MOUNT
BACKUP DATABASE;
ALTER DATABASE OPEN;
```

## 新しい場所へのファイルのリストア

リカバリの使用例の中で、データベース・ファイルやアーカイブ・ログの一部またはすべてを元の場所にリストアできない場合があります。たとえば、2 台のディスク・ドライブを使用しているマシンで 2 番目のドライブが障害を起こすと、そのドライブのデータ・ファイルは 1 番目のドライブにリストアされます。

この項では、次の項目について説明します。

- [新しい場所へのデータ・ファイルのリストア](#)
- [新しい場所への制御ファイルのリストア](#)
- [アーカイブ REDO ログの新しい場所へのリストア](#)

## 新しい場所へのデータ・ファイルのリストア

データ・ファイルをデフォルトの場所にリストアできない場合には、10-10 ページの「[データベースのサブセットのリストアおよびリカバリ](#)」にある一般的な手順に従います。ただし、リストアを実行する前に `SET NEWNAME` コマンドを実行して各データ・ファイルを改名します。Recovery Manager は、各データ・ファイルを元の場所ではなく、`NEWNAME` の場所にリストアします。

データ・ファイルをリストアした後で、リカバリを行う前に `SWITCH` コマンドを実行すると `SET NEWNAME` コマンドで改名したデータ・ファイルのファイル名の変更が失われません。`SWITCH` コマンドは、SQL 文の `ALTER DATABASE RENAME FILE` と同じです。`SWITCH DATAFILE ALL` を実行すると、このジョブ内で `SET NEWNAME` が発行された全データ・ファイルは、それぞれ新しい名前に切替えられます。

**関連項目：** `SWITCH` コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

### 表領域を新しい場所にリストアし、リカバリする手順

1. ターゲット・データベースに（必要があれば、リカバリ・カタログ・データベースにも）接続した後の手順は、次のとおりです。
  - a. リカバリが必要な表領域をオフラインにします。
  - b. オフライン表領域のデータ・ファイルに新しいファイル名を指定します。
  - c. 新しい場所にデータ・ファイルをリストアします。
  - d. 制御ファイルが、リストアしたデータ・ファイルを指すようにします。

この例では、`users` および `tools` 表領域にあるデータ・ファイルを新しい場所にリストアした後、リカバリを実行します。

```
RUN
{
  SQL 'ALTER TABLESPACE users OFFLINE IMMEDIATE';
  SQL 'ALTER TABLESPACE tools OFFLINE IMMEDIATE';
  # restore the datafile to a new location
  SET NEWNAME FOR DATAFILE '?/oradata/trgt/users01.dbf' TO '/tmp/users01.dbf';
  SET NEWNAME FOR DATAFILE '?/oradata/trgt/tools01.dbf' TO '/tmp/tools01.dbf';
  RESTORE TABLESPACE users, tools;
  SWITCH DATAFILE ALL;  # point control file to new filenames
  RECOVER TABLESPACE users, tools;
}
```

2. リカバリが正常に終了した場合には、表領域をオンラインにします。

```
SQL 'ALTER TABLESPACE users ONLINE';
SQL 'ALTER TABLESPACE tools ONLINE';
```

## 新しい場所への制御ファイルのリストア

メディア障害で制御ファイルが破損し、多重コピーがない場合は、バックアップのリストアが必要です。制御ファイルを新しい場所にリストアする場合、`RESTORE CONTROLFILE TO 'filename'` で新しい名前を指定します。デフォルトの場所は、`CONTROL_FILES` 初期化パラメータで指定した最初の場所です。既存のファイルがある場合は、上書きされます。

制御ファイルを新しい場所にリストアした後、`RESTORE CONTROLFILE FROM 'filename'` コマンドを実行し、制御ファイルをすべての `CONTROL_FILES` の宛先にコピーします。`RESTORE CONTROLFILE FROM 'filename'` コマンドは、`COPY CONTROLFILE` コマンドの複数実行と同じです。メディア障害により `CONTROL_FILES` が特定する場所の一部が永続的に破損した場合には、インスタンスを起動する前にサーバー・パラメータ・ファイルを編集して、新しく `CONTROL_FILES` が特定する場所を指定できます。

### 制御ファイルを新しい場所にリストアする手順

1. ターゲット・データベースに（必要に応じて、リカバリ・カタログにも）接続した後、データベースをマウントしないで起動します。

```
STARTUP NOMOUNT
```

2. 制御ファイルをリストアしてマウントします。次の操作を行います。
  - a. 必要に応じて `SET UNTIL` コマンドを実行し、指定日より前に作成された制御ファイルをリストアします。
  - b. バックアップ制御ファイルを一時的な場所にリストアします。
  - c. リストアした場所にある制御ファイルを、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リストアします。

- d. 新しくリストアした制御ファイルを使用して、データベースをマウントします。

```

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';
  RESTORE CONTROLFILE TO '/tmp/control01.ctl'; # restore to new location
  # replicate the control file manually to CONTROL_FILES locations
  RESTORE CONTROLFILE FROM '/tmp/control01.ctl';
  STARTUP MOUNT;
}

```

3. 10-15 ページの「バックアップ制御ファイルを使用したリカバリの実行」の説明のように、メディア・リカバリを実行します。

**関連項目：** RESTORE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## アーカイブ REDO ログの新しい場所へのリストア

Recovery Manager 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\_1 パラメータを組み合わせた名前で、アーカイブ REDO ログをリストアします。リストアされたアーカイブ・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。

デフォルトの名前は、SET ARCHIVELOG DESTINATION コマンドを使用して変更できます。このコマンドは、データベースのリストア実行中に、手動でアーカイブ・ログを異なる場所に移動します。リカバリ中、Recovery Manager は新しくリストアされたアーカイブ・ログの場所を認識しています。つまり、これらのログは初期化パラメータ・ファイルで指定した場所に存在する必要はありません。

### アーカイブ REDO ログをリストアする方法

1. ターゲット・データベースに（必要があれば、リカバリ・カタログ・データベースにも）接続した後、データベースの起動、マウントまたはオープンを確認します。たとえば、次のように実行します。

```
STARTUP MOUNT
```

2. RUN コマンド内で、次の操作を実行します。
  - a. SET ARCHIVELOG DESTINATION コマンドを使用して、リストアしたアーカイブ REDO ログに対して新しい場所を指定します。
  - b. アーカイブ REDO ログをリストアします。

この例では、アーカイブ・ログのバックアップすべてを新しい場所にリストアします。

```
RUN
{
  SET ARCHIVELOG DESTINATION TO '/oracle/temp_restore';
  RESTORE ARCHIVELOG ALL;
  # restore and recover datafiles as needed
  .
  .
  .
}
```

アーカイブ・ログに対して複数のリストアの宛先も指定できます。ただし、複数の宛先を同時に指定することはできません。たとえば、次のように実行できます。

```
RUN
{
  # Set a new location for logs 1 through 10.
  SET ARCHIVELOG DESTINATION TO '/tmp';
  RESTORE ARCHIVELOG FROM SEQUENCE 1 UNTIL SEQUENCE 10;
  # Set a new location for logs 11 through 20.
  SET ARCHIVELOG DESTINATION TO '?/oradata';
  RESTORE ARCHIVELOG FROM SEQUENCE 11 UNTIL SEQUENCE 20;
  # Set a new location for logs 21 through 30.
  SET ARCHIVELOG DESTINATION TO '?/dbs';
  RESTORE ARCHIVELOG FROM SEQUENCE 21 UNTIL SEQUENCE 30;
  # restore and recover datafiles as needed
  .
  .
  .
}
```

アーカイブ REDO ログを複数の場所にリストアする場合、必要な RECOVER コマンドの発行は1回のみです。Recovery Manager は、リストアされたアーカイブ REDO ログを自動的に検索し、その内容をデータ・ファイルに適用します。

## 新しいホストへのデータベースのリストア

この項では、次の項目について説明します。

- [新しいホストへのデータベースのリストア](#)
- [新しいホストにリストアする場合のファイル名の指定](#)

## 新しいホストへのデータベースのリストア

データベースの新しいホストへのリストアについては、様々な使用例が考えられます。たとえば、次のような場合に必要になります。

- 本番データベースを元のホストに維持したまま、テストなどの用途で本番データベースの複製バージョンを作成する場合
- 本番データベースを元のホストに維持したまま、新しいホストに本番データベースをリストアするテストを行う場合
- 本番データベースの場所を新しいホストに移動する場合（元のホストが障害を起こした場合など）

元のデータベースを維持しながら、テスト用の複製データベースを作成するには、RESTORE コマンドではなく、DUPLICATE コマンドを使用します（[第 12 章「Recovery Manager を使用したデータベースの複製」](#)を参照してください）。Recovery Manager は、複製データベース用に一意のデータベース識別子を作成します。この章では、RESTORE コマンドの使用のみを説明しています。

新しいホストへのデータベースのリストアまたは新しいホストへのデータベースの移動をテストするために RESTORE コマンドを実行します。テストでのみリストアを実行する場合には、リカバリ・カタログ内のターゲット・レコードを上書きしないように、次に示す手順が必要になります。

- データ・ファイルをリストアするときに、Recovery Manager をデフォルトの NOCATALOG モードで実行します。
- 制御ファイルのサイズが不足してリストアに必要なバックアップのすべてを格納できないために、リカバリ・カタログを必要とする場合は、カタログをエクスポートして、それを別のスキーマまたはデータベースにインポートします。リストアのテストにはコピーしたリカバリ・カタログを使用します。これを行わないと、カタログはリストアしたデータベースを現行のターゲット・データベースとみなします。

[表 10-2](#) では、新しいホストへのリストアおよび複製を行った場合に、Recovery Manager リポジトリが受ける影響を説明しています。

表 10-2 新しいホストへのリストアおよび複製

コマンド	カタログの使用	リポジトリへの影響
RESTORE	使用する	リストアの後に SWITCH コマンドを実行すると、Recovery Manager はリストアされたデータベースをターゲット・データベースとみなします。SWITCH コマンドを実行しない場合、Recovery Manager はリストアされたデータ・ファイルをイメージ・コピーとみなし、将来のリストア・ジョブの候補とします。
RESTORE	使用しない	リストアの後に SWITCH コマンドを実行すると、Recovery Manager はリストアされたデータベースをターゲット・データベースとみなします。SWITCH コマンドを実行しない場合、リポジトリはリストア操作による影響を受けません。
DUPLICATE	使用する	複製データベース用の新しい DBID が生成されます。この DBID は手動でカタログに登録する必要があります。登録後、リポジトリは、ターゲットおよび複製の異なる 2 つのデータベースのレコードを保持します。
DUPLICATE	使用しない	複製データベース用の新しい DBID が生成されます。ターゲットの制御ファイル内にあるリポジトリは影響されません。

新しいホストにリストアする場合のファイル名の指定

不完全リカバリを新しいホストで実行する基本手順は、元のホストで不完全リカバリを行う場合と大きな違いはありません。主な問題は、新しいホスト上でのデータベース・ファイルのパス名が、プライマリ・ホスト上のファイルのパス名と同じものになるかどうかです。

次の表では、状況に応じて必要となるリストア手順を示しています。

リストアしたファイルのパス名	手順
ターゲット・データベースと同じパス名	10-11 ページ「 <a href="#">不完全リストアおよびリカバリの実行</a> 」
ターゲット・データベースと異なるパス名	10-21 ページ「 <a href="#">新しい場所へのファイルのリストア</a> 」

新しいホストにリストアする場合には、次の点に注意してください。

- 新しいホスト上でターゲット初期化パラメータ・ファイルにアクセス可能にするために、このファイルをオペレーティング・システムのユーティリティを使用して元のホストから新しいホストにコピーします。
- リストアに使用するバックアップが、リストア・ホストでアクセスできるようにします。たとえば、バックアップをメディア・マネージャで作成していた場合には、テープ・デバイスが新しいホストに接続していることを確認します。



- 別のホストで作成したディスク・バックアップやイメージ・コピーを、**Recovery Manager** を使用して新しいホストにリストアすることはできません。ただし、オペレーティング・システムのユーティリティを使用することで、ファイルの転送が可能です。ファイルが新しいホストの同じ場所にある場合は、そのファイルを再度カタログする必要はありません。ファイルを新しい場所に転送する場合は、CATALOG コマンドを使用して、**Recovery Manager** リポジトリを新しいファイル名で更新し、CHANGE ... UNCATALOG コマンドを使用して、古いファイル名をカタログから削除します。
- リストアし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ため、各スレッドにある最新のアーカイブ・ログの最小 SCN まで不完全リカバリを実行してから、RESETLOGS オプションでデータベースをオープンします。各スレッドの最新アーカイブ・ログの中で最小 SCN を検索して、リカバリ終了の SCN を取得します。

必要な SCN を決めるには、SQL\*Plus を起動して次の問合せをします。

```
SELECT MIN(SCN)
FROM (SELECT MAX(NEXT_CHANGE#) SCN
      FROM V$ARCHIVED_LOG
      GROUP BY THREAD#);
```

## Oracle Real Application Clusters データベースのリカバリ

Oracle Real Application Clusters データベースに対するリカバリの基本手順は、10-5 ページの「[基本的な Recovery Manager メディア・リカバリの実行](#)」で説明した Oracle Real Application Clusters 以外のデータベースのリカバリと大きな違いはありません。主な違いは、クラスタの各ノードに接続するチャンネルを、構成して割り当てるか、手動で割り当てる必要があるかということです。

**Recovery Manager** ではバックアップ、データ・ファイルのコピーおよび制御ファイルのコピーをリストアするときに、**自動位置検索**が実行されるため、どのチャンネルで各ノードのファイルをリストアする必要があるかが認識されます。たとえば、データ・ファイルのコピー df1.copy をノード 2 で作成すると、ノード 2 に割り当てられているチャンネルのみがこのファイルのリストアを試みます。割り当てられているチャンネルの PARMS または CONNECT 設定が異なる場合は、必ず自動位置検索機能が使用可能になります。

### データベースの Oracle Real Application Clusters 構成内へのリストア

1. ターゲット・データベースに（必要があれば、リカバリ・カタログ・データベースにも）接続した後、データベースがマウントされていることを確認します。データベースがオープン状態の場合は、停止してからマウントします。

```
SHUTDOWN IMMEDIATE
STARTUP MOUNT
```

### 2. RUN コマンド内で、次の操作を実行します。

- a. 自動チャンネルが構成されていない場合には、手動で各ノードにチャンネルの割当てを行います。
- b. データベースをリストアしてリカバリします。

この例では、自動チャンネルは次のように構成されています。

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 3;  
CONFIGURE DEFAULT DEVICE TYPE TO sbt;  
CONFIGURE CHANNEL 1 DEVICE TYPE sbt CONNECT 'SYS/oracle@node_1';  
CONFIGURE CHANNEL 2 DEVICE TYPE sbt CONNECT 'SYS/oracle@node_2';  
CONFIGURE CHANNEL 3 DEVICE TYPE sbt CONNECT 'SYS/oracle@node_3';
```

次のコマンドでは、完全リストアおよびリカバリが実行されます。

```
RUN  
{  
    RESTORE DATABASE;  
    RECOVER DATABASE;  
}
```

**関連項目：** RESTORE コマンドの詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。Real Application Clusters 環境でのリカバリの詳細は、『Oracle9i Real Application Clusters 管理』を参照してください。

## Recovery Manager を使用した RESETLOGS 操作によるリカバリ

この項では、次の項目について説明します。

- [Recovery Manager を使用した RESETLOGS 操作によるリカバリ](#)
- [RESETLOGS 操作によるメディア・リカバリの要件](#)
- [RESETLOGS 操作によるリカバリ: 基本手順](#)
- [RESETLOGS 操作によるリカバリ: 例](#)

## Recovery Manager を使用した RESETLOGS 操作によるリカバリ

バックアップ制御ファイルを使用したリカバリを実行した後や、不完全リカバリを実行した後は、オンライン REDO ログをリセットする必要があります。データベースを RESETLOGS オプションでオープンした後では、一般的に RESETLOGS の実行前に作成されたデータベースのバックアップは使用できません。オラクル社では、オンライン REDO ログのリセット後には、データベース全体のバックアップを行うことをお勧めします。

RESETLOGS の後、バックアップを実行する前にメディア障害が発生した場合、最後に行ったデータベースの RESETLOGS の前に取ったバックアップでメディア・リカバリを実行することを強制されます。通常のリカバリ計画に従った RESETLOGS によるメディア・リカバリの実行手順を利用することはできません。さらに、RESETLOGS 以後の使用可能なバックアップが存在しないなどの、最悪のケースの使用例でのみリストアを行うことになります。

## RESETLOGS 操作によるメディア・リカバリの要件

メディア・リカバリの実行に RESETLOGS を使用する必要があると判断した場合、次のものがようになります。

- RESETLOGS 以前に作成した制御ファイルのバックアップ。
- 現行の制御ファイルまたは RESETLOGS の後で作成されたバックアップ制御ファイル。
- **RESETLOGS SCN**。RESETLOGS SCN とは、データベースを RESETLOGS オプションでオープンした時点での SCN のことです。次のいずれかの方法で、SCN を取得します。
  - RESETLOGS の時間から `alert_SID.log` を調べます。
  - RESETLOGS 以後に作成した制御ファイルをマウントし、`V$DATABASE` 制御ファイル・ビューを問い合わせる `RESETLOGS_CHANGE#` を得ます。この値から 1 を引きます。
  - `LIST INCARNATION` を実行して、Reset SCN の値から 1 を引きます。
- RESETLOGS の前からのリカバリに必要なすべてのデータベースのバックアップ。
- RESETLOGS SCN までのアーカイブ REDO ログ。
- RESETLOGS 以後、メディア・リカバリの終了ポイントまでのアーカイブ・ログ。

RESETLOGS 以後の使用可能な制御ファイルのバックアップまたはコピーが存在しない場合でも、RESETLOGS によるメディア・リカバリの実行は可能です。

## RESETLOGS 操作によるリカバリ : 基本手順

RESETLOGS によるリカバリは、この項で説明する 2 つの連続するフェーズで発生します。

1. [RESETLOGS 以前のリカバリ : 基本手順](#)
2. [RESETLOGS 以後のリカバリ : 基本手順](#)

### RESETLOGS 以前のリカバリ : 基本手順

最初のフェーズでは、データベースをリストアして、RESETLOGS コマンドの直前の状態にまでデータベースをリカバリします。

1. 次に示すいずれかの方法で、RESETLOGS SCN を取得します。
  - LIST INCARNATION を実行して、Reset SCN 列の値から 1 を引きます。また、Inc Key 列にリストされるすべてのインカネーションの、データベース・インカネーション・キーもメモしておきます。
  - RESETLOGS の時間によって alert\_SID.log を調べて、ワード RESETLOGS を検索します。次のような行を探します。RESETLOGS after incomplete recovery UNTIL CHANGE 1234. この値をそのまま使用します。
  - RESETLOGS 以後の制御ファイル（現行の制御ファイルまたは RESETLOGS でオープンした後で作成したバックアップ）を使用して、次の問合せを実行します。

```
SELECT (RESETLOGS_CHANGE#)-1 FROM V$DATABASE;
```

2. ABORT オプションでデータベースをシャットダウンします。
3. 現行の制御ファイルが消失していない場合には、これを違う場所にコピーします。これらのファイルに対して、以降の手順を繰り返します。コピーの作成には Recovery Manager ではなく、データベースをシャットダウンした後で、オペレーティング・システムのコマンドを使用します。
4. Recovery Manager を起動して、ターゲット・データベースとリカバリ・カタログに接続します。
5. データベースをマウントせずにターゲット・インスタンスを起動します。
6. LIST INCARNATION コマンドを実行して、前のインカネーションのインカネーション・キーを取得します（まだ取得していない場合）。RESET DATABASE コマンドで RESETLOGS した後に作成したデータベース・インカネーションが、すでに登録している場合には、RESET DATABASE TO INCARNATION inc\_key コマンドを実行します。これにより、Recovery Manager は RESETLOGS 以前に取ったバックアップをリストアできるようにします。ここに示す inc\_key は、前のインカネーションのキーです。

7. 次のサブコマンドを持つ RUN コマンドを実行します。
  - a. SET UNTIL SCN `resetlogs_scn` コマンドを発行します。`resetlogs_scn`は、最初の手順で取得した SCN です。
  - b. 必要なチャンネルのすべてを割り当てた後で（自動チャンネルの構成を行っていない場合）、RESETLOGS 以前に作成した制御ファイルのバックアップをリストアし、これをマウントします。
  - c. RESETLOGS 以前に作成したバックアップからリカバリを行う必要があるデータ・ファイルのバックアップをリストアします。リストアを行うのはリカバリが必要なデータ・ファイルのみです。
  - d. リカバリする必要のないデータ・ファイルがある場合や、これらのデータ・ファイルについて RESETLOGS 後からのバックアップがある場合は、SQL 文 ALTER DATABASE DATAFILE ... OFFLINE を実行してオフラインにします。
  - e. SQL\*Plus を起動して V\$DATAFILE の問合せを行います。これにより、すべてのデータ・ファイルが有効な場所を示していることと、リカバリを必要とするファイルのみがオンラインであることを確認します。
  - f. データベースの不完全メディア・リカバリを実行します。これには、RECOVER DATABASE を使用します。
8. IMMEDIATE オプションでデータベースをシャットダウンします。

## RESETLOGS 後のリカバリ：基本手順

前述のフェーズが完了した後で、ここで示す手順を実行します。

1. 次のいずれかのリストアを行います。
  - 10-30 ページの「[RESETLOGS 以前のリカバリ：基本手順](#)」にある手順 3 で保存した場所からの現行の制御ファイル
  - 現行の制御ファイルのコピーをすべて消失している場合には、RESETLOGS 後のバックアップ制御ファイルのコピー（ここで使用するバックアップ制御ファイルは、リカバリ完了後に別の OPEN RESETLOGS を必要とします）
2. データベース・インスタンスを起動します。ただしデータベースはマウントしません。
3. データベースを最新のインカーネーションにリセットします。インカーネーション・キーを LIST INCARNATION の出力から取得します。
4. データベースをマウントします。
5. データベースを必要な時点までリカバリするために、RECOVER コマンドを使用します。
6. リカバリ前にデータ・ファイルをオフラインにした場合は、オンラインに戻します。

- 7. メディア・リカバリ完了後にデータベースをオープンします。このとき、バックアップ制御ファイルを使用した場合や不完全リカバリを実行した場合のみ、RESETLOGS を指定します。RESETLOGS 操作は、オンライン REDO ログからすべての変更を消去します。このため、RESETLOGS をオープンすると、このリカバリ手順の再開および同一ポイントへのリカバリが不可能になることに注意してください。
- 8. データベースをバックアップすることで、今後この手順を実行するような事態を回避します。

## RESETLOGS 操作によるリカバリ : 例

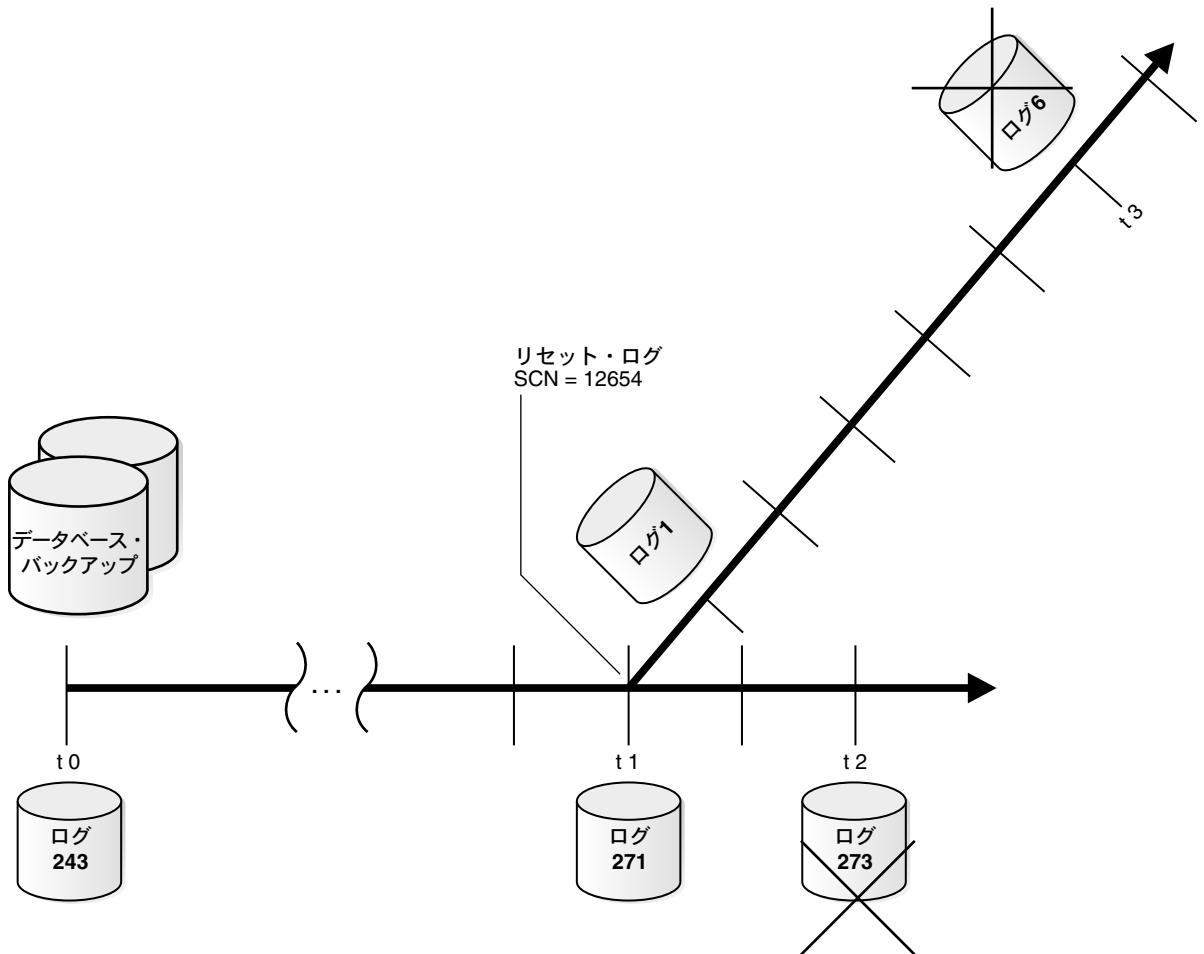
この例では、データベース trgt に対して RESETLOGS 操作によるリカバリが必要であると仮定します。trgt には、表 10-3 に示されるデータベース・ファイルが含まれていると仮定します。

表 10-3 trgt 内のデータベース・ファイル

ファイル	名前	表領域
制御ファイル 1	?/oradata/trgt/control01.ctl	
制御ファイル 2	?/oradata/trgt/control02.ctl	
制御ファイル 3	?/oradata/trgt/control03.ctl	
データ・ファイル 1	?/oradata/trgt/system01.dbf	SYSTEM
データ・ファイル 2	?/oradata/trgt/undotbs01.dbf	undotbs
データ・ファイル 3	?/oradata/trgt/cwmlite01.dbf	cwmlite
データ・ファイル 4	?/oradata/trgt/drsys01.dbf	drsys
データ・ファイル 5	?/oradata/trgt/example01.dbf	example
データ・ファイル 6	?/oradata/trgt/indx01.dbf	indx
データ・ファイル 7	?/oradata/trgt/tools01.dbf	tools
データ・ファイル 8	?/oradata/trgt/users01.dbf	users
オンライン REDO ログ 1	?/oradata/trgt/redo01.log	
オンライン REDO ログ 2	?/oradata/trgt/redo02.log	
オンライン REDO ログ 3	?/oradata/trgt/redo03.log	

この事例では、図 10-1 に示す使用例を想定しています。

図 10-1 RESETLOGS を使用したリカバリ



次に示す順序でイベントが発生していると仮定します。

1. 時刻 t0 に trgt をバックアップ。
2. 時刻 t2 にメディア障害が発生。その結果、アクティブなログ・ファイル・メンバーの一部が破損。
3. t0 のバックアップをリストアして、時刻 t1 までの不完全リカバリを実行。
4. RESETLOGS オプションを使用して trgt データベースをオープン。

5. RESETLOGS をオープンした後、ログ順序は 1 にリセットされ、REDO は時刻 t3 まで生成された。時刻 t3 では、別のメディア障害が発生した結果、データ・ファイル system01.dbf が消失した。

trgt に対する RESETLOGS 後の alert.log は、次のようになります。

```
Starting ORACLE instance
alter database mount
Successful mount of redo thread 1.
Tue Nov 7 15:39:41 2001
Completed: alter database mount
Tue Nov 7 15:39:43 2001
ALTER DATABASE RECOVER database until time 'Nov 07 2001 15:37:54' using backup controlfile
Media Recovery Start
Media Recovery Log
ORA-279 signaled during: ALTER DATABASE RECOVER database until time 'Nov...
Tue Nov 7 15:39:43 2000
ALTER DATABASE RECOVER CONTINUE DEFAULT
Media Recovery Log /oracle/oradata/trgt/arch/archive1_271.dbf
Incomplete recovery done UNTIL CHANGE 12654
Media Recovery Complete
Completed: ALTER DATABASE RECOVER CONTINUE DEFAULT
Tue Nov 7 15:39:44 2000
alter database open resetlogs
RESETLOGS after incomplete recovery UNTIL CHANGE 12654
```

### RESETLOGS 以前の trgt のリカバリ

最初のフェーズでは、制御ファイルおよび trgt のデータ・ファイル system01.dbf をリストアして、RESETLOGS オプションでオープンした直前の状態にまでデータベースをリカバリします。

1. 現行の制御ファイルはアクセス可能なため、V\$DATABASE ビューを問い合わせ、RESETLOGS SCN を取得します。たとえば、次のように入力します。

```
SQL> SELECT RESETLOGS_CHANGE# FROM V$DATABASE;
```

別の方法として、RESETLOGS 発生以降の alert.log がある場合には、次のような行を探します。

```
RESETLOGS after incomplete recovery UNTIL CHANGE 12654.
```

2. 次の文でターゲット・インスタンスを強制終了します。

```
SQL> SHUTDOWN ABORT
```



3. 現行の制御ファイルは維持されていてアクセス可能なため、これを新しい場所にコピーします。たとえば、次のように入力します。

```
% cp $ORACLE_HOME/oradata/trgt/control01.ctl /tmp/control01.ctl
% cp $ORACLE_HOME/oradata/trgt/control02.ctl /tmp/control02.ctl
% cp $ORACLE_HOME/oradata/trgt/control03.ctl /tmp/control03.ctl
```

4. Recovery Manager を起動し、ターゲット・データベースに接続します。例のように、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。

```
% rman TARGET / CATALOG rman/cat@catdb
```

5. 例のように、ターゲット・データベース・インスタンスをマウントしないで起動します。

```
STARTUP NOMOUNT
```

6. RESETLOGS 後のインカーネーションは Recovery Manager に登録されているため、LIST INCARNATION コマンドの実行により前のインカーネーションの主キーを取得して、Recovery Manager をこのインカーネーションにリセットします。たとえば、次に示すコマンドを実行します。ここに示す *inc\_key* は、LIST の出力から取得したキーです。

```
LIST INCARNATION OF DATABASE trgt;
RESET DATABASE TO INCARNATION inc_key;
```

7. 次のサブコマンドを持つ RUN コマンドを実行します。
  - a. 最初の手順で取得した値を使用して、リカバリ終了の SCN を設定します。
  - b. 自動チャンネルが構成されていない場合は、少なくとも 1 つのチャンネルを手動で割り当てます。
  - c. RESETLOGS 以前に作成した制御ファイルをリストアして、この制御ファイルをマウントします。
  - d. RESETLOGS 以前に作成したバックアップからリカバリを行う必要があるデータ・ファイルのバックアップをリストアします。リカバリを必要とするデータ・ファイルは system01.dbf のみのため、このファイルのみをリストアします。
  - e. リカバリ不要なすべてのデータ・ファイルをオフラインにするために、SQL 'ALTER DATABASE DATAFILE ... OFFLINE' を実行します。
  - f. SQL\*Plus を起動して v\$DATAFILE の問合せを行います。これにより、すべてのデータ・ファイルが有効な場所を示していることと、リカバリを必要とするファイルのみがオンラインであることを確認します。
  - g. データベースをリカバリします。
  - h. IMMEDIATE オプションでデータベースをシャットダウンします。

たとえば、次のコマンドを実行します。

```
RUN
{
  SET UNTIL SCN 12654;
  RESTORE CONTROLFILE;
  ALTER DATABASE MOUNT;
  RESTORE DATAFILE system01.dbf;
  SQL 'ALTER DATABASE DATAFILE ?/oradata/trgt/undotbs01.dbf,
      ?/oradata/trgt/cwmlite01.dbf, ?/oradata/trgt/drsys01.dbf,
      ?/oradata/trgt/example01.dbf, ?/oradata/trgt/indx01.dbf,
      ?/oradata/trgt/tools01.dbf, ?/oradata/trgt/users01.dbf OFFLINE';
  HOST; # check V$DATAFILE to make sure everything is OK
  RECOVER DATABASE;
  SHUTDOWN IMMEDIATE;
}
```

## RESETLOGS 以後の trgt のリカバリ

1. 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、前述したフェーズの手順 3 でセーブした制御ファイルをリストアします。たとえば、次のように入力します。

```
% cp /tmp/control01.ct1 $ORACLE_HOME/oradata/trgt/control01.ct1
% cp /tmp/control02.ct1 $ORACLE_HOME/oradata/trgt/control02.ct1
% cp /tmp/control03.ct1 $ORACLE_HOME/oradata/trgt/control03.ct1
```

2. ターゲット・データベースに（必要があれば、リカバリ・カタログ・データベースにも）接続した後、データベースを起動します。たとえば、次のように入力します。

```
STARTUP NOMOUNT
```

3. データベースを現行のインカーネーション・キーにリセットします（キーを取得するためには、LIST INCARNATION コマンドを実行します）。たとえば、次のように入力します。curr\_inc\_key は、現行のインカーネーション・キーです。

```
RESET DATABASE TO INCARNATION curr_inc_key;
```

4. データベースをマウントします。たとえば、次のように入力します。

```
ALTER DATABASE MOUNT;
```

5. 次のサブコマンドを持つ RUN コマンドを実行します。

- a. 不完全リカバリを実行する場合には、必要に応じて、終了時刻、SCN またはログ順序番号を設定します。
- b. 自動チャンネルを構成していない場合には、少なくとも 1 つのチャンネルを手動で割り当てます。
- c. データベースをリカバリします。

たとえば、完全リカバリを実行するには、次のコマンドを実行します。

```
RECOVER DATAFILE system01.dbf;
```

- リカバリする前にオフラインにした表領域をすべて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 DATABASE DATAFILE ?/oradata/trgt/undotbs01.dbf,  
    ?/oradata/trgt/cwmlite01.dbf, ?/oradata/trgt/drsys01.dbf,  
    ?/oradata/trgt/example01.dbf, ?/oradata/trgt/indx01.dbf,  
    ?/oradata/trgt/tools01.dbf, ?/oradata/trgt/users01.dbf ONLINE';
```

- ALTER DATABASE OPEN または ALTER DATABASE OPEN RESETLOGS を使用してデータベースをオープンします。これは、完全リカバリを行うか、不完全リカバリを行うかにより決まります。たとえば、次のように入力します。

```
ALTER DATABASE OPEN;
```

- ただちに、データベースのバックアップを行うことをお勧めします。このときデータベースは、できるかぎりマウントした状態にします（オープン状態のデータベースでのデータ消失を回避するため）。データベースは新しいインカネーションであるため、RESETLOGS 以前に作成したバックアップの使用は難しくなります。たとえば、データベース・バックアップを新規に作成するには、次のように入力します。

```
SHUTDOWN IMMEDIATE  
STARTUP MOUNT  
BACKUP DATABASE;  
ALTER DATABASE OPEN;
```

## 障害時リカバリの実行

障害時リカバリの使用例では、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、すべての制御ファイル、オンライン REDO ログおよびパラメータ・ファイルを消失していることが考えられます。最低でも、一部のデータ・ファイルのバックアップ、バックアップ時以後に生成された一部のアーカイブ REDO ログ、および少なくとも 1 つの制御ファイルの自動バックアップは保持している必要があります。

**関連項目：** 5-52 ページ「[制御ファイルとサーバー・パラメータ・ファイルの自動バックアップ](#)」

障害時リカバリの基本手順は、10-15 ページの「[バックアップ制御ファイルを使用したリカバリの実行](#)」のとおりですが、最初の手順としてサーバー・パラメータ・ファイルの自動バックアップをリストアします。インスタンスの起動後に、制御ファイルの自動バックアップをリストアし、マウントしてから、データ・ファイルのリストアとリカバリを行うことができます。新しいホストにリストアを行うことになるため、10-25 ページの「[新しいホストへのデータベースのリストア](#)」にある説明を考慮に入れ再検討する必要があります。

次の例では、使用可能な最新のアーカイブ・ログ（この例では、スレッド 1 内のログ 1124）にまでデータベースをリストアしてリカバリします。この使用例では次のように想定します。

- データベースを新しいホストに同じディレクトリ構造でリストアします。
- すべてのデータ・ファイルのバックアップ、ログ 1124 までのアーカイブ REDO ログ、および制御ファイルとサーバー・パラメータ・ファイルの自動バックアップを含むテープ・ドライブを 1 つ保持しています。
- リカバリ・カタログを使用しません。

この使用例では、次の手順を実行します。

1. できるかぎり、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、`tnsnames.ora` や `listener.ora` などの関連ネットワーク・ファイルをすべてリストアしてください。
2. **Recovery Manager** を起動し、ターゲット・データベースに接続します。Oracle Net ファイルがない場合は、オペレーティング・システム認証を介して接続します。
3. 10-18 ページの「バックアップ制御ファイルを使用してリカバリ・カタログを使用しないリカバリの実行」の説明に従い、SET DBID コマンドを使用してターゲット・データベースの DBID を指定します。
4. STARTUP NOMOUNT コマンドを実行します。**Recovery Manager** はダミーのサーバー・パラメータ・ファイルを使用してインスタンスを起動します。
5. メディア・マネージャにチャンネルを割り当ててから、RESTORE SPFILE FROM AUTOBACKUP コマンドを実行します。
6. リストアされたサーバー・パラメータ・ファイルを使用してインスタンスが再起動されるように、STARTUP FORCE NOMOUNT モードを実行します。
7. メディア・マネージャにチャンネルを割り当ててから、制御ファイルの自動バックアップをリストアします（10-18 ページの「バックアップ制御ファイルを使用してリカバリ・カタログを使用しないリカバリの実行」を参照してください）。
8. リストアした制御ファイルをマウントします。
9. リポジトリに記録されていないすべてのアーカイブ・ログを、CATALOG コマンドでカタログに追加します（18-18 ページの「アーカイブ・ログおよびユーザー管理コピーのカタログ化」を参照してください）。
10. 元の場所にデータ・ファイルをリストアします。ボリューム名を変更した場合には、リストアの前に SET NEWNAME コマンドを実行して、リストアの後に切替えを実行します。これにより、制御ファイルをデータ・ファイルの新しい場所で更新します（10-21 ページの「新しい場所へのファイルのリストア」を参照してください）。
11. データ・ファイルを、リカバリします。**Recovery Manager** は、指定したログ順序番号に到達するとリカバ리를停止します。

12. RESETLOGS モードでデータベースをオープンします。この最後の手順は、適用されるアーカイブ・ログが他に存在しないことが確実な場合にのみ完了します。

---

**注意：** オラクル社では、RESETLOGS 操作の後には、データベースをバックアップすることをお勧めします（例には含まれていません）。

---

```
# Start RMAN and connect to the target database
% rman TARGET SYS/oracle@trgt

# set the DBID for the target database
SET DBID 676549873;
STARTUP FORCE NOMOUNT; # rman starts instance with dummy parameter file
RUN
{
  ALLOCATE CHANNEL t1 DEVICE TYPE sbt;
  RESTORE SPFILE FROM AUTOBACKUP;
}
STARTUP FORCE NOMOUNT;

RUN
{
  # manually allocate a channel to the media manager
  ALLOCATE CHANNEL t1 DEVICE TYPE sbt;
  # Restore an autobackup of the control file. This example assumes that you have
  # accepted the default format for the autobackup name.
  RESTORE CONTROLFILE FROM AUTOBACKUP;
  # 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 RMAN restores
  # the database to the same structure that the database had at the specified time.
  SET UNTIL SEQUENCE 1124 THREAD 1;
  ALTER DATABASE MOUNT;
  RESTORE DATABASE;
  RECOVER DATABASE;
}
ALTER DATABASE OPEN RESETLOGS; # Reset the online logs after recovery completes
```

次に示す RUN コマンドの例は同じ使用例ですが、リストアしたデータ・ファイルが新しいファイル名になります。

```
RUN
{
  # If you need to restore the files to new locations, tell Recovery Manager
  # to do this using SET NEWNAME commands:
  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';
  ALLOCATE CHANNEL t1 DEVICE TYPE sbt;
  RESTORE CONTROLFILE FROM AUTOBACKUP;
  SET UNTIL SEQUENCE 124 THREAD 1;
  ALTER DATABASE MOUNT;
  RESTORE DATABASE;
  SWITCH DATAFILE ALL;  # Update the control file with new location of the datafiles.
  RECOVER DATABASE;
}
ALTER DATABASE OPEN RESETLOGS;
```

## Recovery Manager を使用したデータ・ブロックのリカバリ

BLOCKRECOVER コマンドでは、データ・ファイル内の個々のデータ・ブロックをリストアおよびリカバリできます。この手順が役立つのは、トレース・ファイルや標準出力の中で、データ・ファイル内の少数のブロックが破損していると示された場合です。ブロック・メディア・リカバリは、データ消失または破損の範囲が不明な場合の使用には適していません。この場合は、かわりにデータ・ファイルのリカバリを使用します。

この項では、次の項目について説明します。

- 使用可能なすべてのバックアップによるデータ・ブロックのリカバリ
- 任意のバックアップを使用したデータ・ブロックのリカバリ
- V\$DATABASE\_BLOCK\_CORRUPTION にリストされるブロックのリカバリ

### 関連項目：

- ブロック・メディア・リカバリの概要は、6-11 ページの「[Recovery Manager を使用したブロック・メディア・リカバリ](#)」を参照してください。
- BLOCKRECOVER の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- V\$DATABASE\_BLOCK\_CORRUPTION の構文は、『Oracle9i データベース・リファレンス』を参照してください。

## 使用可能なすべてのバックアップによるデータ・ブロックのリカバリ

この使用例では、リカバリが必要なブロックを識別した後に、それらのブロックのリストアおよびリカバリを実行するために、使用可能なあらゆるバックアップを使用します。

使用可能なすべてのバックアップを使用してデータ・ブロックのリカバリを行います。

1. 破損ブロックのデータ・ファイル番号およびブロック番号を取得します。一般的に、これを取得するためには、標準出力からの出力、alert.log、トレース・ファイルまたはメディア管理インタフェースを使用します。たとえば、トレース・ファイルでは次のような表示を確認できます。

```
ORA-01578: Oracle データ・ブロックに障害が発生しました (ファイル番号 8、ブロック番号 13)
ORA-01110: データ・ファイル 8: '/oracle/oradata/trgt/users01.dbf'
ORA-01578: Oracle データ・ブロックに障害が発生しました (ファイル番号 2、ブロック番号 19)
ORA-01110: データ・ファイル 2: '/oracle/oradata/trgt/undotbs01.dbf'
```

2. 事前割当て済みの自動チャネルがある場合、破損ブロックのファイルおよびブロック番号を指定して Recovery Manager プロンプトから BLOCKRECOVER コマンドを実行するには次のようにします。

```
BLOCKRECOVER DATAFILE 8 BLOCK 13 DATAFILE 2 BLOCK 19;
```

## 任意のバックアップを使用したデータ・ブロックのリカバリ

この使用例では、リカバリが必要なブロックを識別した後に、そのブロックのリストアおよびリカバリを実行するために選択したバックアップのみを使用します。

データ・ブロックのリカバリを、限られたバックアップのタイプで行います。

1. 破損ブロックのデータ・ファイル番号およびブロック番号を取得します。一般的に、これを取得するためには、標準出力からの出力、alert.log、トレース・ファイルまたはメディア管理インタフェースを使用します。たとえば、トレース・ファイルでは次のような表示を確認できます。

```
ORA-01578: Oracle データ・ブロックに障害が発生しました (ファイル番号 8、ブロック番号 13)
ORA-01110: データ・ファイル 8: '/oracle/oradata/trgt/users01.dbf'
ORA-01578: Oracle データ・ブロックに障害が発生しました (ファイル番号 2、ブロック番号 19)
ORA-01110: データ・ファイル 2: '/oracle/oradata/trgt/undotbs01.dbf'
```

2. 事前割当て済みの自動チャネルがある場合、Recovery Manager プロンプトから BLOCKRECOVER コマンドを発行します。このとき、破損ブロックのファイル番号とブロック番号の指定を行い、使用可能なオプションによりバックアップの候補を制限します。たとえば、ブロックのリストアに使用するバックアップのタイプは、次のように指定します。

```
# restore from backupset
BLOCKRECOVER DATAFILE 8 BLOCK 13 DATAFILE 2 BLOCK 19 FROM BACKUPSET;
# restore from datafile image copy
```

```
BLOCKRECOVER DATAFILE 8 BLOCK 13 DATAFILE 2 BLOCK 19 FROM DATAFILECOPY;
```

タグで指定して、バックアップを指示することができます。

```
# restore from backupset with tag "mondayam"
BLOCKRECOVER DATAFILE 8 BLOCK 13 DATAFILE 2 BLOCK 199 FROM TAG = mondayam;
```

バックアップの候補を、特定の時点の前に作成した候補に制限することができます。

```
# restore using backups made before one week ago
BLOCKRECOVER DATAFILE 8 BLOCK 13 DATAFILE 2 BLOCK 19 RESTORE UNTIL 'SYSDATE-7';
# restore using backups made before SCN 100
BLOCKRECOVER DATAFILE 8 BLOCK 13 DATAFILE 2 BLOCK 19 RESTORE UNTIL SCN 100;
# restore using backups made before log sequence 7024
BLOCKRECOVER DATAFILE 8 BLOCK 13 DATAFILE 2 BLOCK 19 RESTORE UNTIL SEQUENCE
7024;
```

UNTIL 句を使用してデータ・ブロックのリストアを制限した場合、Recovery Manager はブロックに対してより多くのリカバリを実行する必要があります。さらに、リカバリ・フェーズでは、指定したブロックの変更のために、すべてのログをスキャンする必要があります。よって、特に必要がない場合はリカバリの制限は行わないでください。

## V\$DATABASE\_BLOCK\_CORRUPTION にリストされるブロックのリカバリ

V\$DATABASE\_BLOCK\_CORRUPTION ビューには最後に BACKUP、BACKUP ... VALIDATE または COPY コマンドが実行された後に破損マークが設定されたデータ・ファイル内のブロックが表示されます。破損ブロックの修復後は、このブロックを識別する行がビューから削除されます。

データベース内の論理的な破損のチェックは、BACKUP を（VALIDATE オプションの有無にかかわらず）実行するか、COPY コマンドで CHECK LOGICAL コマンドを使用して行います。Recovery Manager が破損ブロックを検出して、そのブロック数が MAXCORRUPT の設定以下の場合、V\$DATABASE\_BLOCK\_CORRUPTION に移入します。Recovery Manager のバックアップとコピーにあるブロック破損の履歴レコードは、V\$BACKUP\_CORRUPTION および V\$COPY\_CORRUPTION に保持されます。

この使用例では、リカバリが必要なブロックを識別するために V\$DATABASE\_BLOCK\_CORRUPTION を問い合せて、CORRUPTION LIST キーワードを使用して、このビューにリストされたすべてのブロックをリカバリするように Recovery Manager に指示します。

データ・ブロックのリカバリを、限られたバックアップのタイプで行います。

1. V\$BACKUP\_CORRUPTION を問い合せて、最新のデータ・ファイルのバックアップとコピーに破損ブロックが存在するかどうかを判断します。

```
SQL> SELECT * FROM V$DATABASE_BLOCK_CORRUPTION;
```



2. 事前割当て済みの自動チャネルがある場合は、V\$DATABASE\_BLOCK\_CORRUPTION で破損のマークが示されたすべてのブロックを、BLOCKRECOVER CORRUPTION LIST コマンドを実行してリカバリします。たとえばこのコマンドでは、11 日以上前に作成したバックアップ・セットからのブロックをリストアします。

```
BLOCKRECOVER CORRUPTION LIST
FROM BACKUPSET
RESTORE UNTIL TIME 'SYSDATE-10';
```

## バックアップとコピーのリストアの妥当性チェック

リストアの妥当性チェックでは、実際にはファイルのリストアを行わずに、リストアのテストを実行します。リストアのテストは、データベース全体、または個々の表領域、データ・ファイルまたは制御ファイルに対しても行うことができます。RESTORE ... VALIDATE および VALIDATE BACKUPSET コマンドでは、バックアップまたはコピーをリストアできるかどうかをテストします。次の方法が選択できます。

- RESTORE ... VALIDATE では、バックアップまたはコピーから指定したオブジェクトを Recovery Manager でリストアできるかどうかをテストします。使用するバックアップまたはコピーは、Recovery Manager により選択されます。
- VALIDATE BACKUPSET では、指定したバックアップ・セットの妥当性をテストします。

**関連項目：** RESTORE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。VALIDATE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

### 妥当性チェックの対象となるバックアップ・セットまたはコピーを Recovery Manager に自動的に選択させる方法

妥当性チェックを実行するために、データベースをマウントまたはオープンします。妥当性チェックを行うために、データ・ファイルをオフラインにする必要はありません。

1. バックアップ・セットとコピーのリストアの妥当性チェックを行います。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・ログのリストアについて、妥当性チェックを行います。

```
RESTORE CONTROLFILE VALIDATE;
RESTORE TABLESPACE SYSTEM VALIDATE;
RESTORE ARCHIVELOG ALL VALIDATE;
```

2. 出力をチェックします。次のようなエラー・メッセージ・スタックが表示される場合には、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: 見つからないターゲットがあります - リストアを中止します
```

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リストアに問題があることを示しています。

```
RMAN-03009: restore コマンド (c1 チャンネル上) が 12-DEC-01 23:22:30 で失敗しました
ORA-19505: ファイル "oracle/dbs/1fafv9gl_1_1" の識別に失敗しました。
ORA-27037: ファイル・ステータスを取得できません。
SVR4 Error: 2: No such file or directory
Additional information: 3
```

エラー・スタックがない場合は、**Recovery Manager** でファイルの妥当性チェック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. 妥当性チェックをデータベース全体に対して行う必要がない場合は、主キーを明示して LIST コマンドを実行し、妥当性チェックを行うバックアップ・セットとコピーを検索します。

```
LIST BACKUPSET;
LIST COPY;
```

2. バックアップ・セットのリストアの妥当性チェックを行います。この例では、バックアップ・セット 1121 のリストアの妥当性チェックを行います。

```
VALIDATE BACKUPSET 1121;
```

3. 出力をチェックします。validation complete というメッセージが確認できた場合は、**Recovery Manager** では指定したバックアップ・セットのリストアの妥当性チェックを正常に完了しています。次に例を示します。

```
using channel ORA_DISK_1
channel ORA_DISK_1: starting validation of archive log backupset
channel ORA_DISK_1: restored backup piece 1
piece handle=/oracle/dbs/0mdg9v8l_1_1 tag=TAG20020208T155604 params=NULL
channel ORA_DISK_1: validation complete
```

## Recovery Manager のリストアおよびリカバリ例

この項では、次の項目について説明します。

- [新しいホストへのデータ・ファイルのコピーのリストア : 例](#)
- [カタログ内の複数データベースが同じ名前を共有する場合のリストア : 例](#)
- [RESETLOGS 操作以前に作成されたバックアップのリカバリ : 例](#)
- [NOARCHIVELOG モードでのデータベースのリカバリ : 例](#)
- [バックアップがないデータ・ファイルを消失した場合のリカバリ : 例](#)
- [Recovery Manager を使用した異なるデータベースへの表領域バックアップの移動 : 例](#)

### 新しいホストへのデータ・ファイルのコピーのリストア : 例

データ・ファイルのコピーを使用してデータベースを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・データベースとリカバリ・カタログに接続してから、LIST コマンドを実行して、次の例のようにデータ・ファイルのコピーおよびその主キーのリストを表示します。

```
LIST COPY;
```

2. 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・ファイルのコピーを新しいホストにコピーします。たとえば、UNIX では次のように入力します。

```
% cp -r /tmp/*dbf /net/new_host/oracle/oradata/trgt
```

3. Recovery Manager を起動し、旧ホストのデータ・ファイルのコピーをカタログから削除します。たとえば、次のように入力します。

```
CHANGE COPY OF DATAFILE 1,2,3,4,5,6,7,8 UNCATALOG;
```

4. 新しいファイル名を使用してデータ・ファイルのコピーをカタログ化します。たとえば、次のように実行します。

```
CATALOG DATAFILECOPY
  '?/oradata/trgt/system01.dbf', '?/oradata/trgt/undotbs01.dbf',
  '?/oradata/trgt/cwmlite01.dbf', '?/oradata/trgt/drsys01.dbf',
  '?/oradata/trgt/example01.dbf', '?/oradata/trgt/indx01.dbf',
  '?/oradata/trgt/tools01.dbf', '?/oradata/trgt/users01.dbf';
```

5. 10-25 ページの「[新しいホストへのデータベースのリストア](#)」に説明されているリストア操作とリカバリ操作を実行します。

## カタログ内の複数データベースが同じ名前を共有する場合のリストア : 例

『Oracle9i Recovery Manager リファレンス』にある SET DBID の説明のように、SET DBID コマンドを実行して制御ファイルのリストアを行う必要があるのは、同じ名前を共有する複数のデータベースがリカバリ・カタログに登録されており、ターゲット・データベースがマウントされていない場合です。この使用例では、次の手順を実行します。

1. Recovery Manager を起動し、ターゲット・データベースに接続します。
2. STARTUP FORCE NOMOUNT コマンドを実行します。
3. SET DBID コマンドを実行して、この接続先ターゲット・データベースを、同じ名前を持つ他のターゲット・データベースと区別します。
4. RESTORE CONTROLFILE コマンドを実行します。制御ファイルのリストアが完了した後は、データベースをマウントして、データベースの残りの部分をリストアできます。

この手順により、制御ファイルのリストアを試行したときに RMAN-20005 のメッセージを回避します。このメッセージが発生するのは、2 つ以上のターゲット・データベースが同じ名前を持っているため、Recovery Manager が個別のデータベースを識別する一意の DBID を要求する場合です。

### リストアが必要なデータベースの DBID の取得

Recovery Manager の出力を保存している場合は、この情報を参照してデータベース識別子を確認してください。データベースに接続する際、Recovery Manager はこの情報を自動的に表示します。

```
% rman TARGET /
```

```
Recovery Manager: Release 9.2.0.0.0
```

```
connected to target database: RMAN (DBID=1231209694)
```

Recovery Manager の出力を保存していないときに、リストア操作のためにデータベースの DBID の値が必要になった場合は、RC\_DATABASE または RC\_DATABASE\_INCARNATION リカバリ・カタログ・ビューを問い合わせ、DBID の値を取得してください。

この例では、リカバリ・カタログに登録されているデータベースの名前は一意ではないと想定しているため、他の一意の情報を使用して、正しい DBID を判断する必要があります。リストアするデータベースに関連したデータ・ファイルまたはオンライン REDO ログ・ファイル名がわかっていて、このファイル名がリカバリ・カタログに登録されているすべてのデータベース間で一意の場合は、次に示す問合せにある *filename\_of\_log\_or\_df* を完全なファイル名に置き換えます。次の問合せのいずれかを実行して、DBID を確認します。

```
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リストア

DBID を設定するために、Recovery Manager をターゲット・データベースに接続し、次に示すように SET コマンドを実行します。ここに示す *target\_dbid* は、前述の手順で取得した値です。

```
SET DBID = target_dbid;
```

制御ファイルをデフォルトの場所にリストアし、データベースをマウントするには、次のように実行します。

```
RESTORE CONTROLFILE;
ALTER DATABASE MOUNT;
```

データベースをリストアしてリカバリするには、次のように実行します。

```
RESTORE DATABASE;
RECOVER DATABASE
  # optionally, delete logs restored for recovery and limit disk space used
  DELETE ARCHIVELOG MAXSIZE 2M;
```

## RESETLOGS 操作以前に作成されたバックアップのリカバリ : 例

次の状況を想定します。

- リカバリ・カタログを使用して Recovery Manager を実行します。
- trgt のバックアップを 2001 年 1 月 2 日に作成しました。
- このデータベースに不完全リカバリを行い、2001 年 1 月 10 日に RESETLOGS オプションでオープンしました。これにより新しいデータベース・インカネーションが作成されました。

必要な重大データが 2001 年 1 月 8 日午前 8 時にデータベースから削除されたことを 1 月 25 日に発見しました。trgt を前のインカネーションにリセットし、1 月 2 日のバックアップをリストアしてから、1 月 8 日午前 7 時 55 分までリカバリすることになりました。

**注意：** 現行のデータベースが別のインカネーションにある場合は、以前のインカネーションのデータ・ファイルを 1 つのみリストアすることはできません。データベース全体をリストアする必要があります。

### 以前のインカネーションのバックアップを使用したデータベースのリストア

1. LIST コマンドを実行して、以前のインカネーションの主キーを取得します。

```
# obtain primary key of old incarnation
LIST INCARNATION OF DATABASE trgt;
```

List of Database Incarnations

DB Key	Inc Key	DB Name	DB ID	CUR	Reset SCN	Reset Time
1	2	TRGT	1224038686	NO	1	02-JAN-01
1	582	TRGT	1224038686	YES	59727	10-JAN-01

2. データベースが起動されていて、マウントされていないことを確認します。

```
SHUTDOWN ABORT
STARTUP NOMOUNT
```

3. インカネーションを取得した主キーにリセットします。

```
# reset database to old incarnation
RESET DATABASE TO INCARNATION 2;
```

4. RUN コマンド内で次の操作を実行して、データベースをリカバリします。
  - リカバリの終了時刻を、データが消失する直前の時刻に設定します。
  - 自動チャンネルが構成されていない場合には、手動で1つ以上のチャンネルを割り当てます。
  - 制御ファイルをリストアしてマウントします。
  - データベースをリストアしてリカバリします。

たとえば、次のコマンドを実行します。

```
RUN
{
  SET UNTIL TIME 'Jan 8 2001 07:55:00'; # set time to just before data was lost
  RESTORE CONTROLFILE;
  ALTER DATABASE MOUNT; # mount database after restoring control file
  RESTORE DATABASE;
  RECOVER DATABASE;
}
```

5. リカバリが正常に終了した場合には、オンライン REDO ログをリセットします。

```
ALTER DATABASE OPEN RESETLOGS; # this command automatically resets the database
                                # so that this incarnation is the new incarnation
```

## NOARCHIVELOG モードでのデータベースのリカバリ : 例

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をリカバリすることができます。増分バックアップに一貫性がない場合、データベースがオープンしているときには、そのデータベースのバックアップは作成できません。

次の例を想定してみます。

- データベース `trgt` を NOARCHIVELOG モードで実行しています。
- リカバリ・カタログを使用しています。
- データベースを一貫してシャットダウンし、日曜の午後にデータベース `trgt` のレベル 0 のバックアップをテープに作成します。
- データベースを一貫してシャットダウンし、水曜日と金曜日の午前 3 時に、レベル 1 の差分増分バックアップをテープに作成します。
- 土曜日に、データベースにメディア障害が発生し、データ・ファイルの半分とオンライン REDO ログを破棄しました。

この場合、金曜日が最新の増分バックアップの日付なので、金曜日までの不完全メディア・リカバリを実行する必要があります。水曜および金曜のレベル 1 のバックアップと同様に、Recovery Manager は日曜日のレベル 0 のバックアップも使用します。

オンライン REDO ログが失われたため、RECOVER コマンドで NOREDO オプションを指定する必要があります。また、オンライン・ログが使用可能でも、増分に REDO を適用できない場合は、NOREDO も指定する必要があります。NOREDO オプションの指定がない場合、Recovery Manager は金曜日の増分バックアップを適用した後、REDO ログを探し、見つからない場合は、エラー・メッセージを発行します。オンライン・ログが使用可能な場合は、NOREDO を指定せずに RECOVER DATABASE を実行できます。

trgt 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リカバリします。

```
STARTUP FORCE MOUNT;
RESTORE CONTROLFILE; # restore control file from consistent backup
ALTER DATABASE MOUNT;
RESTORE DATABASE; # restore datafiles from consistent backup
RECOVER DATABASE NOREDO; # specify NOREDO because online redo logs are lost
ALTER DATABASE OPEN RESETLOGS;
```

この使用例では、増分バックアップを行った金曜日から障害が発生した土曜日の間に生成されたすべての変更は適用されません。

## バックアップがないデータ・ファイルを消失した場合のリカバリ : 例

この例では、次の状況を想定しています。

1. ARCHIVELOG モードのデータベースで、データベース全体のバックアップを作成します。
2. 単一のデータ・ファイル ?/oradata/trgt/history01.dbf を含む表領域を作成します。
3. 新規に作成したデータ・ファイルにデータを移入します。
4. すべてのアクティブなオンライン REDO ログをアーカイブします。
5. ?/oradata/trgt/history01.dbf のバックアップを作成する前に、あるユーザーが誤ってこのデータ・ファイルをオペレーティング・システムから削除しました。

消失したデータ・ファイルのバックアップがなくても、データはリカバリできます。データをリカバリするには、消失したデータ・ファイルと同一のファイル名で新規にデータ・ファイルを作成し、RECOVER コマンドを実行して、このファイルの REDO を適用します。

たとえば、Recovery Manager を起動してターゲット・データベースに接続した後、Recovery Manager プロンプトで次の文を実行します。

```
# take the missing datafile offline
# note that SQL statement is bounded by double quotes, but the datafile name has two
# individual single quotes both before and after it
SQL "ALTER DATABASE DATAFILE
    ' ?/oradata/trgt/history01.dbf ' OFFLINE";
# create a new datafile with the same name as the missing datafile
SQL "ALTER DATABASE CREATE DATAFILE
```



```

    '' ?/oradata/trgt/history01.dbf '' ";
# recover the newly created datafile
RECOVER DATAFILE '?/oradata/trgt/history01.dbf';
# bring the recovered datafile back online
SQL "ALTER DATABASE DATAFILE
    '' ?/oradata/trgt/history01.dbf '' ONLINE";

```

## Recovery Manager を使用した異なるデータベースへの表領域バックアップの移動 : 例

トランスポート表領域機能を使用して、あるデータベースから別のデータベースに表領域をコピーできます。表領域を移動する基本手順では、Recovery Manager は使用しません。詳細は、『Oracle9i データベース管理者ガイド』を参照してください。ただし、Recovery Manager を使用すると、ターゲット・データベースをバックアップする場合に、表領域のバックアップをデータベース間で移動できます。

後述の手順では、次のことを想定します。

- 表領域 users のバックアップを、コンピュータ hosta にあるデータベース trgt\_a から、コンピュータ hostb にあるデータベース trgt\_b に移動します。
- 1 つのデータ・ファイルを持つ表領域 users のバックアップを移動します。
- users データ・ファイルのリストア後のバックアップの名前を、データベース trgt\_b では /net/hostb/oracle/oradata/trgt\_b/users01.dbf にします。
- 次のファイルには、データベース trgt\_a からのリカバリ可能バックアップがあります。
  - 移動する表領域（この例では users 表領域）内のすべてのデータ・ファイル
  - SYSTEM 表領域内のデータ・ファイル
  - ロールバック・セグメントまたは UNDO セグメントを持つデータ・ファイル
  - 前述のデータ・ファイルのバックアップ用メタデータを含む制御ファイル
- trgt\_a のバックアップには、hostb からテープ・デバイスを介してアクセス可能です。

**関連項目：** 表領域を移動する方法は、『Oracle9i データベース管理者ガイド』を参照してください。

### 別のデータベースへの表領域の移動

1. 12-10 ページの「複製用の補助インスタンスの準備 : 基本手順」の指示に従って、hostb 上で補助インスタンスを作成します。
2. Recovery Manager を、新規ターゲット・インスタンスの場合と同様に補助インスタンスに接続します。たとえば、次のように入力します。

```
rman TARGET SYS/oracle@auxdb CATALOG rman/rman@catdb
```

3. 制御ファイルを一時的な場所にリストアしてから、マウントしてセッションを終了します。たとえば、次のように入力します。

```
RESTORE CONTROLFILE TO '/net/hostb/tmp/cf.f';
STARTUP FORCE MOUNT;
EXIT
```

4. Recovery Manager を同じ補助インスタンスに NOCATALOG モードで再接続し、補助データベースをリストアしてリカバリします。次の手順を実行します。
  - a. 表領域をリカバリする現時点以外の時点、SCN またはアーカイブ・ログを指定します。表領域は現時点までリカバリできません。指定した UNTIL 時刻を使用して、表領域のどのバックアップをリストアするかを指定します。
  - b. リストアする制御ファイルに構成済みチャネルが含まれていない場合は、バックアップを含むデバイスにチャネルを手動で割り当てます。
  - c. SET NEWNAME を実行して、SYSTEM データ・ファイルとロールバック・セグメントまたは UNDO セグメントを含むデータ・ファイルの一時ファイル名を指定します。
  - d. SET NEWNAME を実行して、移動された表領域内のデータ・ファイルに使用される、trgtb データベース内のファイル名を指定します。
  - e. 表領域をリストアしてリカバリします。

たとえば、次のコマンドを実行します。

```
% rman TARGET SYS/oracle@auxdb NOCATALOG

RUN
{
  SET UNTIL ARCHIVELOG 1243 THREAD 1; # set the end recovery log
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt; # allocate a channel if not configured
  # specify temporary name for SYSTEM datafile
  SET NEWNAME FOR DATAFILE 1 TO '/net/hostb/tmp/df1.dbf';
  # specify temporary names for datafiles with undo or rollback segments
  SET NEWNAME FOR DATAFILE 2 TO '/net/hostb/tmp/df2.dbf';
  # specify names for datafiles to be plugged into trgtb database
  SET NEWNAME FOR DATAFILE 8 TO '/net/hostb/oracle/oradata/trgtb/users01.dbf';
  # restore and recover the datafiles
  RESTORE DATAFILE 1, 2, 8;
  RECOVER DATAFILE 1, 2, 8;
}
```

5. 前述の手順でリカバリした表領域を除き、すべての補助表領域をオフラインにします。たとえば、次のように入力します。

```
SQL 'ALTER TABLESPACE cwnlite, drsys, example, indx, tools OFFLINE IMMEDIATE';
```

6. RESETLOGS オプションを使用して補助データベースをオープンします。たとえば、次のように入力します。

```
ALTER DATABASE OPEN RESETLOGS;
```

7. 移動する表領域を読取り専用にします。たとえば、次のように入力します。

```
SQL 'ALTER TABLESPACE users READ ONLY';
```

8. 移動された表領域からメタデータをエクスポートします。手順は、『Oracle9i データベース管理者ガイド』の「手順 2: トランスポータブル表領域セットの生成」を参照してください。たとえば、次のように入力します。

```
exp TRANSPORT_TABLESPACE=y TABLESPACES=(users)
    TRIGGERS=y CONSTRAINTS=n GRANTS=n FILE=expdat.dmp
```

9. 補助インスタンスをシャットダウンし、移動された表領域のデータ・ファイルを除き、すべての補助ファイルを削除します。たとえば、次のように入力します。

```
sqlplus SYS/oracle@auxdb <<EOF
SHUTDOWN ABORT
EXIT
EOF
rm /net/hostb/tmp/*
```

10. 移動された表領域からメタデータをインポートします。手順は、『Oracle9i データベース管理者ガイド』の「手順 4: 表領域セットのプラグイン」を参照してください。たとえば、次のように入力します。

```
imp TRANSPORT_TABLESPACE=y FILE=expdat.dmp
    DATAFILES=('/net/hostb/oracle/oradata/trgtb/users01.dbf')
    TABLESPACES=(users) TTS_OWNERS=(usera)
    FROMUSER=(usera) TOUSER=(userb)
```



---

## Recovery Manager による表領域の Point-in-Time リカバリの実行

この章では、Recovery Manager を使用した表領域の Point-in-Time リカバリ（TSPITR）の実行方法を説明します。この章の項目は、次のとおりです。

- [TSPITR の概要](#)
- [TSPITR の計画](#)
- [補助セットとリカバリ・セットに含まれるファイルのネーミング・メソッドの選択](#)
- [TSPITR の実行に必要な補助インスタンスの準備](#)
- [TSPITR の実行](#)
- [TSPITR の実行後に使用するターゲット・データベースの準備](#)
- [TSPITR が失敗した場合の対応](#)

## TSPITR の概要

Recovery Manager (RMAN) の自動化された表領域の Point-in-Time リカバリ (TSPITR) 機能によって、データベースの残りの表領域とは別の時点に、1 つ以上の表領域を迅速にリカバリできます。

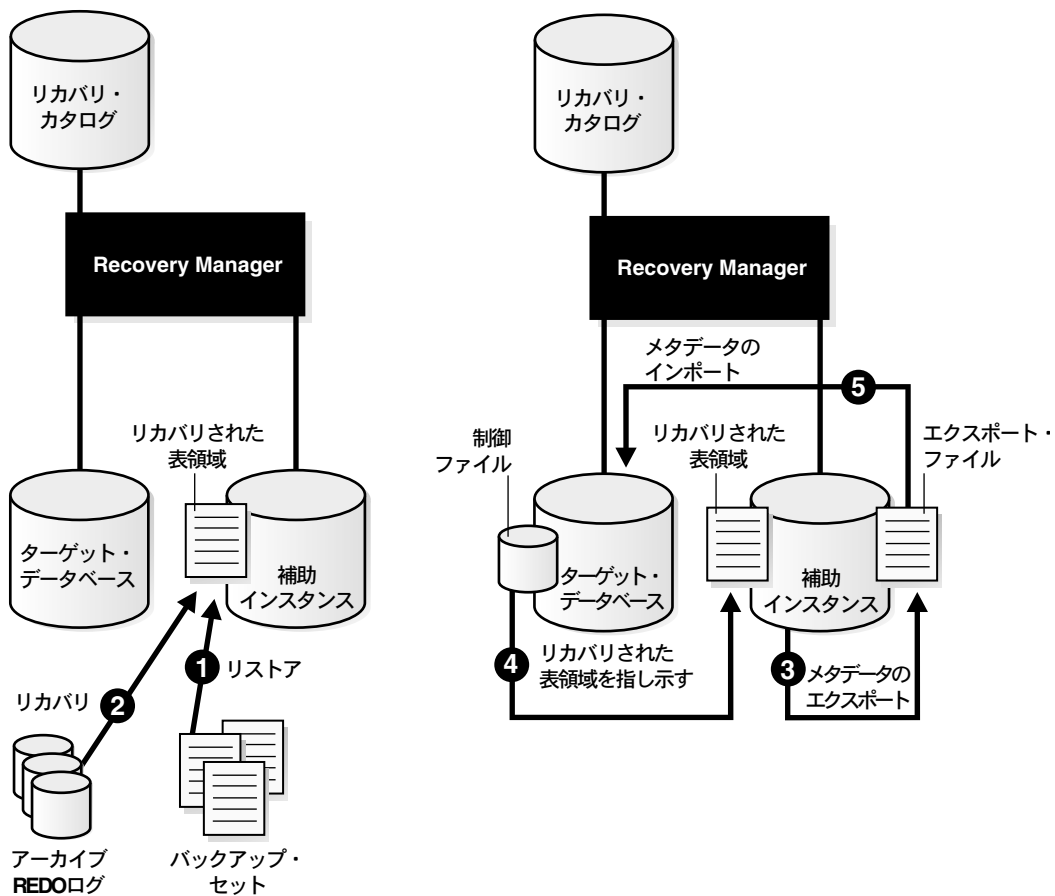
Recovery Manager の TSPITR 機能は、次のリカバリに最適です。

- DROP TABLE または TRUNCATE TABLE 文の誤り
- 論理的に破損した表
- 不適切なバッチ・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
- 1 つの物理データベースの個別の表領域に複数のスキーマが存在している場合、その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スキーマ

表のエクスポートと同様に、Recovery Manager の TSPITR によって、一貫性のあるデータ・セットをリカバリできます。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です。[図 11-1](#) で示すように、Recovery Manager は次の作業を行います。

1. 指定した表領域のバックアップをリストアします。
2. 指定した表領域をリカバリします。
3. 補助インスタンスからメタデータをエクスポートします。
4. ターゲット・データベースの制御ファイルが、新しくリカバリされたデータ・ファイルを指すようにします。
5. ターゲット・データベースにメタデータをインポートします。

図 11-1 Recovery Manager 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 (Tablespace Point-in-Time Recovery)

表領域の Point-in-Time リカバリ

### 補助インスタンス

バックアップ表領域のリカバリに使用される補助的なインスタンス。TSPITR が作成したデータベースは、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。

### リカバリ・セット

TSPITR の実行が必要な、ターゲット・データベース内の表領域。たとえば、表領域 `users` を現時点以外の時刻までリカバリする必要がある場合、`users` はリカバリ・セット内の表領域です。

### 補助セット

TSPITR に必要な他のファイル。次の内容が含まれます。

- バックアップ制御ファイル。
- SYSTEM 表領域。
- ロールバック・セグメントまたは UNDO セグメントを含むデータ・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に必要な小さな領域。
- 補助データベースのオンライン REDO ログ。

**関連項目：** ソート領域の詳細は、11-20 ページの「[TSPITR が失敗した場合の対応](#)」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について説明します。

- [リカバリ・カタログを使用しない TSPITR の実行](#)
- [一般的な制限事項の理解](#)
- [データの関連の管理](#)

---

**注意：** この章で説明する制限事項や計画手順の多くは、『Oracle9i ユーザー管理バックアップおよびリカバリ・ガイド』にも記載されていますが、異なる点もあります。その異なる点を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

Recovery Manager の TSPITR は、リカバリ・カタログを使用しても、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は、次の制限事項に注意してください。

- ロールバック・セグメントまたは UNDO セグメントは、TSPITR の実行時に補助セット内に存在することが必要です。これは、Recovery Manager では UNDO の履歴レコードを制御ファイル内に持たないため、現行のロールバック・セグメントまたは UNDO セグメントはリカバリ実行時に存在していたセグメントと同一であると仮定するためです。ロールバック・セグメントまたは UNDO セグメントがリカバリの実行後に変更された場合には、存在しないファイルに書込みを行おうとして TSPITR は失敗します。
- Oracle が必要なコピーとバックアップの制御ファイル・レコードを再利用している場合は、TSPITR がリカバリする時点が古すぎると、TSPITR が失敗する可能性があります。
- 表領域に対して TSPITR を実行し、時刻  $t$  に表領域をオンラインにするとします。時刻  $t$  以前に作成された表領域のバックアップは、現行の制御ファイルでのリカバリには使用できなくなります。したがって、この表領域に対して TSPITR を再実行して時刻  $t$  以前の時点までリカバリすることも、現行の制御ファイルを使用してデータベースを時刻  $t$  以前の時点までリカバリすることもできません。

NOCATALOG モードが追加の制限事項を伴うのは、現行の制御ファイルにはリカバリした表領域の古いインカネーションのレコードが含まれていないためです。そのため、この表領域に関連する現行の制御ファイルでのリカバリには、時刻  $t$  以前に作成されたバックアップを使用できなくなります。ただし、時刻  $t$  以前にバックアップされた制御ファイルをリストアすると、時刻  $t$  以前の時点までデータベース全体の不完全リカバリを実行できます。

## 一般的な制限事項の理解

Recovery Manager 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コンピュータ上で実行すること。ただし、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するクラスタ構成内に作成できます。
- 削除された表領域をリカバリすること。
- 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リカバリすること。
- 表領域に追加されたデータ・ファイルを削除すること。そのファイルが、Recovery Manager の TSPITR のリカバリ対象となる時点より後に追加された場合は、Recovery Manager 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。
- 補助インスタンスで DML 文を発行すること。補助インスタンスはリカバリ専用の一時インスタンスです。

- リカバリ・カタログに接続しているときに、表領域の TSPITR を実行し、時刻  $t$  に表領域をオンラインに戻すとします。時刻  $t$  以前に作成されたバックアップを使用して、時刻  $t$  以前の時点まで表領域またはデータベース全体をバックアップできます。したがって、TSPITR の実行後は、即時に表領域をバックアップする必要があります。
- 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リカバリすること。統計情報は TSPITR の実行後に再度算出してください。
- 次のオブジェクトのいずれかをリカバリ・セット内に入れること。
  - レプリケートされたマスター表。
  - 表の一部。たとえば、Recovery Manager の TSPITR をパーティション表に対して実行し、複数の表領域にパーティションを分散すると、Recovery Manager はエラー・メッセージをエクスポート・フェーズ時に戻します。
  - 制約を持たない表または表を持たない制約。
  - 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・ログとスナップショット表。
  - UNDO セグメントまたはロールバック・セグメントを含む表領域。
  - SYS が所有するオブジェクト（ロールバック・セグメントを含む）。

## データの関連の管理

TSPITR は、リカバリ・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、または関連オブジェクトをリカバリ・セットに含めるか、いずれかの方法でこのデータの関連を管理しないかぎり、TSPITR は正常に実行できません。

**関連項目：** . 依存性の問題の調査および解決方法は、『Oracle9i ユーザー管理バックアップおよびリカバリ・ガイド』を参照してください。

## オブジェクトが消失するかどうかの判断

Recovery Manager の TSPITR を表領域で実行すると、リカバリ時点以後に作成されたオブジェクトは消失します。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・データベースで TS\_PITR\_OBJECTS\_TO\_BE\_DROPPED を問い合わせます。表 11-1 に、このビューの内容を示します。

表 11-1 TS\_PITR\_OBJECTS\_TO\_BE\_DROPPED ビュー

列名	意味
OWNER	削除されるオブジェクトの所有者
NAME	TSPITR の結果として消失されるオブジェクトの名前
CREATION_TIME	オブジェクトの作成時のタイムスタンプ
TABLESPACE_NAME	オブジェクトを含む表領域の名前

このビューを問い合わせるときに、データ・フィールドのすべての要素を指定しないと、デフォルト設定が使用されます。また、TO\_CHAR および TO\_DATE ファンクションを使用してください。たとえば、リカバリ・セットが users および tools で構成されており、リカバリ時点が '2001-06-02:07:03:11' の場合は、次の文を発行します。

```
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 ('USERS','TOOLS')
AND CREATION_TIME > TO_DATE('01-JUN-02:07:03:11','YY-MON-DD:HH24:MI:SS')
ORDER BY TABLESPACE_NAME, CREATION_TIME;
```

**関連項目：** TS\_PITR\_OBJECTS\_TO\_BE\_DROPPED ビューの詳細は、『Oracle9i データベース・リファレンス』を参照してください。

## プライマリ・データベースへの依存性の調査と解決

TS\_PITR\_CHECK ビューを問い合わせ、リカバリ・セットの境界とオーバーラップするオブジェクト間の関連を識別します。このビューの問合せ時に行が戻される場合は、問題を調査して解決してください。リカバリ・セットにない表領域について TS\_PITR\_CHECK ビューで行が戻されなくなるまで、TSPITR は実行しないでください。TSPITR の完了後にこれらの関連を再度トレースできるように、この手順で実行したアクションをすべて記録してください。

リカバリ・セットの表領域を記述する 4 行の述語を指定して、TS\_PITR\_CHECK ビューを問い合わせます。たとえば、リカバリ・セットが tools および users で構成される場合、TS\_PITR\_CHECK に対する SELECT 文は次のようになります。

```
SELECT *
FROM SYS.TS_PITR_CHECK
WHERE (
    TS1_NAME IN ('USERS','TOOLS')
    AND TS2_NAME NOT IN ('USERS','TOOLS')
)
OR (
    TS1_NAME NOT IN ('USERS','TOOLS')
    AND TS2_NAME IN ('USERS','TOOLS')
);
```

データベース内の（リカバリ・セット内の表領域のみでなく）すべての表領域に対して完全 TSPITR チェックを実行するには、次の問合せを実行します。

```
SELECT *
FROM SYS.TS_PITR_CHECK
WHERE (
    'SYSTEM' IN (TS1_NAME, TS2_NAME)
    AND TS1_NAME <> TS2_NAME
    AND TS2_NAME <> '-1'
)
OR (
    TS1_NAME <> 'SYSTEM'
    AND TS2_NAME = '-1'
);
```

TS\_PITR\_CHECK ビューの列の数と幅により、問合せの実行時には列を次のようにフォーマットすることが必要な場合があります。

```
SET LINESIZE 120
COLUMN OBJ1_OWNER HEADING "own1"
COLUMN OBJ1_OWNER FORMAT a6
COLUMN OBJ1_NAME HEADING "name1"
COLUMN OBJ1_NAME FORMAT a5
COLUMN OBJ1_SUBNAME HEADING "subname1"
COLUMN OBJ1_SUBNAME FORMAT a8
COLUMN OBJ1_TYPE HEADING "obj1type"
COLUMN OBJ1_TYPE FORMAT a8 word_wrapped
COLUMN TS1_NAME HEADING "ts1_name"
COLUMN TS1_NAME FORMAT a6
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 a6
COLUMN TS2_NAME HEADING "ts2_name"
COLUMN TS2_NAME FORMAT a6
COLUMN CONSTRAINT_NAME HEADING "cname"
COLUMN CONSTRAINT_NAME FORMAT a5
COLUMN REASON HEADING "reason"
COLUMN REASON FORMAT a25 word_wrapped
```

パーティション表 tp の 2 つのパーティション p1 および p2 が、それぞれ表領域 users および tools に存在するとします。また、パーティション索引 tpind が tp に定義されており、この索引に 2 つのパーティション id1 および id2 が（それぞれ表領域 id1 および id2 に）存在するとします。この場合は、表領域 users および tools について TS\_PITR\_CHECK を問い合わせると、次の出力が得られます（適切な書式設定の場合）。

own1	name1	subname1	obj1type	ts1_name	name2	subname2	obj2type	own2	ts2_name	cname	reason
---	-----	-----	-----	-----	-----	-----	-----	---	-----	-----	-----
SYSTEM	TP	P1	TABLE	USER	TPIND	IP1	INDEX	PARTITION	PARTITION	SYS	
ID1	Partitioned Objects not fully contained in the recovery set										
SYSTEM	TP	P2	TABLE	TOOLS	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、この索引は 2 つのパーティション、つまり表領域 id1 内の ip1 と表領域 id2 内の ip2 で構成されています。tpind を削除するか、id1 および id2 リカバリ・セットに含めてください。

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

## 補助セットとリカバリ・セットに含まれるファイルのネーミング・メソッドの選択

この項では、次の項目について説明します。

- [Recovery Manager TSPITR でのデータ・ファイルのネーミングの概要](#)
- [SET NEWNAME を使用した補助セット内とリカバリ・セット内のファイルの名前付け](#)
- [補助セットとリカバリ・セットでのデータ・ファイルのコピーの使用](#)
- [初期化パラメータを使用した補助セットのファイルの名前付け](#)

### Recovery Manager TSPITR でのデータ・ファイルのネーミングの概要

TSPITR を実行するために補助インスタンスのセットアップを開始する前に、補助セットに含まれるデータ・ファイルのネーミング・メソッドを決定する必要があります。補助セット内のファイル名がターゲット・データベースで使用するファイル名と同一の場合、Recovery Manager は TSPITR の実行中にエラーを通知して終了します。

たとえば、/tmp ディレクトリに補助データベースを作成する場合は、補助データ・ファイルの絶対パス名に接頭辞 /tmp を付ける必要があります。これらの補助ファイルは、TSPITR の実行専用であるため、TSPITR の実行直後に削除してください。ファイル名が補助セット内で変換されていない場合、Recovery Manager は TSPITR 時にエラーを通知して終了します。

必要の場合は、ターゲット・データベース上で（補助データベース上ではなく）、リカバリ・セットの表領域にあるデータ・ファイルの名前を変更するように選択することもできます。たとえば、users 表領域内のデータ・ファイル ?/oradata/trgt/users01.dbf で TSPITR を実行する場合に、このデータ・ファイル名を /disk2/datafiles/users01.dbf に変更するように決定できます。この操作は、ターゲット・データベース内で指定したファイルに対して ALTER DATABASE RENAME FILE を実行するのと同じことです。

表 11-1 に、TSPITR 時に補助セットとリカバリ・セット内のデータ・ファイルに名前を付けるために使用するコマンドとパラメータを示します。次の表では、優先順位が上から下へ順序付けされています。したがって、SET NEWNAME は、CONFIGURE AUXNAME や DB\_FILE\_NAME\_CONVERT より優先されます。

表 11-2 データ・ファイルのネーミング・メソッド

優先順位	コマンド / パラメータ	補助セット内のファイルの名前を指定可能？	リカバリ・セット内のファイルの名前を変更可能？
1	SET NEWNAME	○	○
2	CONFIGURE AUXNAME	○	○
3	DB_FILE_NAME_CONVERT	○	×

SET NEWNAME を使用した補助セット内とリカバリ・セット内のファイルの名前付け

Recovery Manager コマンド SET NEWNAME を使用して、補助セット内のデータ・ファイルに新しい名前を指定できます。Recovery Manager は、この新しいファイル名を、そのデータ・ファイルをリストアしてリカバリするための一時的な位置として利用します。また、この新しいファイル名は、初期化パラメータ・ファイル内で DB\_FILE\_NAME\_CONVERT パラメータが設定されている場合に、その設定をオーバーライドします。

SET NEWNAME を使用して、リカバリ・セットの表領域内のデータ・ファイルの名前を変更することもできます。新しい名前を指定すると、ターゲット制御ファイルにある元のファイル名が、新しいファイル名に置換されます。新しいファイル名を設定する場合、Recovery Manager 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ているかどうかを確認しません。競合があった場合は、TSPITR 時に Recovery Manager エラーが発生します。

SET NEWNAME を使用したファイルの名前付け : 例

たとえば、補助セットに次のデータ・ファイルが含まれているとします。

- SYSTEM 表領域の ?/oradata/trgt/system01.dbf
- undotbs 表領域の ?/oradata/trgt/undotbs01.dbf

リカバリ・セットには、次のデータ・ファイルが含まれているとします。

- users 表領域の ?/oradata/trgt/users01.dbf
- tools 表領域の ?/oradata/trgt/tools01.dbf

補助データベースを /tmp ディレクトリに作成します。また、tools 内のデータ・ファイルの名前を /private1/tools01.dbf に変更し、このデータ・ファイルを users 表領域には元の名前で残すとします。

この場合は、次のコマンドを実行して `tools` および `users` で TSPITR を実行し、ターゲット・データベース上で名前 `?/oradata/trgt/tools01.dbf` を `?/dbs/tools01.dbf` に変更できます。

```
RUN
{
  # set newnames for auxiliary set datafiles
  SET NEWNAME FOR DATAFILE '?/oradata/trgt/system01.dbf' TO '/tmp/system01.dbf';
  SET NEWNAME FOR DATAFILE '?/oradata/trgt/undotbs01.dbf' TO '/tmp/undotbs01.dbf';
  # rename one recovery set datafile
  SET NEWNAME FOR DATAFILE '?/oradata/trgt/tools01.dbf' TO '/disk1/tools01.dbf';
  RECOVER TABLESPACE tools, users UNTIL SEQUENCE 1034 THREAD 1;
}
```

## 補助セットとリカバリ・セットでのデータ・ファイルのコピーの使用

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・ファイルをリストアするよりはるかに高速です。したがって、データ・ファイルをリストアしてリカバリするかわりに、リカバリ・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 の TSPITR はデータ・ファイルのコピーを使用します。

1. データ・ファイルのコピーの名前は、対応するデータ・ファイルの補助名として、リカバリ・カタログ内に次のコマンドで登録されていること (`filename` はデータ・ファイルの名前または番号、`aux_datafile_name` はデータ・ファイルの補助名)。

```
CONFIGURE AUXNAME FOR DATAFILE FILENAME TO auxiliary_datafile_name;
```

2. データ・ファイルのコピーは、次の Recovery Manager コマンドを使用して、UNTIL 句で指定した時刻より前に作成されていること (`'filename'` はデータ・ファイルのファイル名)。

```
COPY DATAFILE 'filename' TO AUXNAME;
```

Recovery Manager がデータ・ファイルのコピーを使用して TSPITR を正常に完了した場合、`auxiliary_datafile_name` がリカバリ・カタログから削除され、制御ファイル内のステータスが DELETED に更新されます。Recovery Manager 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・ファイルのコピーで置き換えられます。

---

**注意：** 同じデータ・ファイルに対して SET NEWNAME を使用した場合、Recovery Manager はそのデータ・ファイルのコピーを使用しません。

---

## CONFIGURE AUXNAME を使用したファイルの名前付け : 例

たとえば、補助セットに次のデータ・ファイルが含まれているとします。

- SYSTEM 表領域の ?/oradata/trgt/system01.dbf
- undotbs 表領域の ?/oradata/trgt/undotbs01.dbf

リカバリ・セットには、次のデータ・ファイルが含まれているとします。

- users 表領域の ?/oradata/trgt/users01.dbf
- tools 表領域の ?/oradata/trgt/tools01.dbf

また、必要な TSPITR の終了時点より前の時点で次のコマンドを実行したとします。

```
CONFIGURE AUXNAME FOR DATAFILE '?/oradata/trgt/system01.dbf' TO '/copy/system01.dbf';
CONFIGURE AUXNAME FOR DATAFILE '?/oradata/trgt/undotbs01.dbf' TO '/copy/undotbs01.dbf';
CONFIGURE AUXNAME FOR DATAFILE '?/oradata/trgt/tools01.dbf' TO '?/dbs/tools01.dbf';
```

```
COPY DATAFILE '?/oradata/trgt/system01.dbf' TO AUXNAME;
COPY DATAFILE '?/oradata/trgt/undotbs01.dbf' TO AUXNAME;
COPY DATAFILE '?/oradata/trgt/tools01.dbf' TO AUXNAME;
```

この場合は、次のコマンドを実行して tools および users で TSPITR を実行し、ターゲット・データベース上で名前 ?/oradata/trgt/tools01.dbf を ?/dbs/tools01.dbf に変更できます。

```
RECOVER TABLESPACE tools, users UNTIL SEQUENCE 1034 THREAD 1;
```

## 初期化パラメータを使用した補助セットのファイルの名前付け

補助初期化パラメータ・ファイルの DB\_FILE\_NAME\_CONVERT パラメータを使用すると、リカバリ・セットではなく補助セットのデータ・ファイル名を設定できます。補助セットの表領域内のデータ・ファイルに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に指定されている変換済みファイル名を使用して、リストアおよびリカバリを実行できます。Recovery Manager は、補助データベースとターゲット・データベースの間で競合しているデータ・ファイル名をチェックします。競合があると、エラーが戻ります。

リカバリ・セットの表領域内にあるデータ・ファイルに、新規名も補助名も設定されていない場合や、補助名でファイルが使用できない場合、Recovery Manager は元のデータ・ファイルの位置を使用します。

**関連項目：** 補助初期化パラメータ・ファイルの例は、11-14 ページの「[タスク 2: 補助インスタンス用パラメータ・ファイルの作成](#)」を参照してください。



**初期化パラメータを使用したファイルの名前付け: 例** たとえば、補助セットに次のファイルが含まれているとします。

- SYSTEM 表領域の ?/oradata/trgt/system01.dbf
- undotbs 表領域の ?/oradata/trgt/undotbs01.dbf

リカバリ・セットには、次のファイルが含まれているとします。

- users 表領域の ?/oradata/trgt/users01.dbf
- tools 表領域の ?/oradata/trgt/tools01.dbf

補助データベースを /tmp ディレクトリに作成します。また、tools 内のデータ・ファイルの名前を /private1/tools01.dbf に変更し、このデータ・ファイルを users 表領域には元の名前で残すとします。

この場合は、補助初期化パラメータ・ファイル内で次のパラメータを設定して、補助セットのファイルの名前を指定できます。

```
DB_FILE_NAME_CONVERT=('/oradata/trgt','/tmp') # captures all auxiliary set files
```

この場合は、次のコマンドを実行して tools および users で TSPITR を実行し、ターゲット・データベース上で名前 ?/oradata/trgt/tools01.dbf を ?/dbs/tools01.dbf に変更できます。

```
RUN
{
  # rename one recovery set datafile
  SET NEWNAME FOR DATAFILE '?/oradata/trgt/tools01.dbf' TO '?/dbs/tools01.dbf';
  RECOVER TABLESPACE tools, users UNTIL SEQUENCE 1034 THREAD 1;
}
```

## TSPITR の実行に必要な補助インスタンスの準備

この項で説明する次の要件を満たした上で、Recovery Manager の TSPITR を実行します。

- **タスク 1: 補助インスタンスの Oracle パスワード・ファイルの作成**
- **タスク 2: 補助インスタンス用パラメータ・ファイルの作成**
- **タスク 3: 補助インスタンスへの Oracle Net 接続性の確認**
- **タスク 4: 補助インスタンスの起動**
- **タスク 5: Recovery Manager コマンドライン・インタフェースの起動**

## タスク 1: 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、『Oracle9i データベース管理者ガイド』を参照してください。

## タスク 2: 補助インスタンス用パラメータ・ファイルの作成

補助インスタンス用の初期化パラメータ・ファイルを作成し、[表 11-3](#) の説明に従ってパラメータを設定します。

表 11-3 補助インスタンスの初期化パラメータ

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	同一 Oracle ホーム内のどのデータベースとも異なる値。 わかりやすいように、 <code>_dbname</code> と指定してください。 たとえば、ターゲット・データベース名が <code>trgt</code> の場合は、 <code>_trgt</code> を指定します。
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すると、 <code>SET NEWNAME</code> または <code>CONFIGURE AUXNAME</code> を実行して名前付けしていないファイルにファイル名を生成できます。 <code>V\$DATAFILE.NAME</code> を問い合せてデータ・ファイルのファイル名を取得し、変換パターンがビューに表示されるファイル名の書式と一致していることを確認します。  <b>注意：</b> 一部のプラットフォームでは、末尾がスラッシュまたは円記号（/ または ¥）のパターンはサポートされません。  <b>関連項目：</b> <a href="#">11-9 ページ「補助セットとリカバリ・セットに含まれるファイルのネーミング・メソッドの選択」</a>
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。 <code>V\$LOGFILE.MEMBER</code> を問い合せてオンライン REDO ログのファイル名を取得し、変換パターンがビューに表示されるファイル名の書式と一致していることを確認します。  <b>注意：</b> 一部のプラットフォームでは、末尾がスラッシュまたは円記号（/ または ¥）のパターンはサポートされません。

表 11-3 補助インスタンスの初期化パラメータ（続き）

パラメータ	指定対象
CONTROL_FILES	ターゲット・パラメータ・ファイル内の CONTROL_FILES パラメータとは異なる値。
REMOTE_LOGIN_PASSWORDFILE	パスワード・ファイルを使用して補助インスタンスに接続する場合は、EXCLUSIVE に設定します。
COMPATIBLE	ターゲット・データベース内のパラメータと同じ値。
DB_BLOCK_SIZE	この初期化パラメータがターゲット・データベース内で設定されている場合は、補助インスタンス内で同じ値に設定する必要があります。

SYSDBA として Oracle Net への接続を可能にするパラメータを含め、その他のパラメータも必要に応じて設定します。

次に示すのは、補助インスタンス用の初期化パラメータの設定例です。

```
DB_NAME=trgt
LOCK_NAME_SPACE=_trgt
CONTROL_FILES=/tmp/control01.ctl
DB_FILE_NAME_CONVERT=('/oracle/oradata/trgt/', '/tmp/')
LOG_FILE_NAME_CONVERT=('/oracle/oradata/trgt/redo', '/tmp/redo')
REMOTE_LOGIN_PASSWORDFILE=exclusive
COMPATIBLE = 9.0.1
DB_BLOCK_SIZE=8192
```

---

**注意：** これらのパラメータを設定した後に、本番ファイルに関する初期化設定値をターゲット・データベースで上書きしないでください。

---

**関連項目：** DB\_FILE\_NAME\_CONVERT の詳細は、11-14 ページの「[タスク 2: 補助インスタンス用パラメータ・ファイルの作成](#)」を参照してください。Oracle Net の詳細は、『Oracle9i Net Services 管理者ガイド』を参照してください。

## タスク 3: 補助インスタンスへの Oracle Net 接続性の確認

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。

## タスク 4: 補助インスタンスの起動

Recovery Manager の TSPITR を実行する前に、SQL\*Plus を使用して補助インスタンスに接続し、NOMOUNT モードで（必要に応じてパラメータ・ファイルを指定して）起動します。たとえば、次のように入力します。

```
SQL> CONNECT SYS/oracle@aux AS SYSDBA
SQL> STARTUP NOMOUNT PFILE='/tmp/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 モードでのみ起動できます。制御ファイルは作成しないでください。また、TSPITR の補助インスタンスのマウントやオープンは行わないでください。

## タスク 5: Recovery Manager コマンドライン・インタフェースの起動

この項で説明する、次のいずれかの方法で、Recovery Manager のコマンドライン・インタフェースを起動してください。

- [オペレーティング・システムのコマンドラインからの接続](#)
- [Recovery Manager プロンプトからの接続](#)

### オペレーティング・システムのコマンドラインからの接続

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/oracle@trgt CATALOG rman/cat@catdb AUXILIARY SYS/oracle@aux
```

次の表に、この Recovery Manager 接続に使用する変数を示します。

変数	意味
SYS	SYSDBA 権限を持つユーザー
oracle	ターゲット・データベースの orapwd ファイルで指定される、SYSDBA で接続するためのパスワード
trgt	ターゲット・データベースのネット・サービス名
rman	リカバリ・カタログの所有者
cat	リカバリ・カタログの orapwd ファイル内に指定された、ユーザーの Recovery Manager のパスワード
catdb	リカバリ・カタログ・データベースのネット・サービス名
aux	補助データベースのネット・サービス名

## Recovery Manager プロンプトからの接続

Recovery Manager コマンドライン・インタフェースを補助インスタンスへの接続を行わずに起動し、起動後に Recovery Manager プロンプトから CONNECT コマンドを使用できます。例では、デフォルトの NOCATALOG モードで接続します。

```
% rman
RMAN> CONNECT AUXILIARY SYS/oracle@aux
RMAN> CONNECT TARGET SYS/oracle@trgt
```

CONNECT CATALOG コマンドを実行して、カタログに接続します。

```
RMAN> CONNECT CATALOG rman/cat@catdb
```

## TSPITR の実行

すべての計画要件を満たした後、リカバリのエンド・ポイントを指定してリカバリ・セットの表領域をリカバリします。**Recovery Manager** で自動的に実行されるため、最初に表領域をオフラインにする必要はありません。次のコマンドでは、ログ順序 13 まで **users** 表領域に対して **TSPITR** が実行されます。

```
RECOVER TABLESPACE users UNTIL SEQUENCE 13 THREAD 1;
```

補助デバイス構成が指定されていない場合や、**Recovery Manager** が自動的に割り当てた補助チャンネルを必要とする場合、**Recovery Manager** はターゲット・データベースのデバイス構成を使用します。ターゲット・チャンネルとは異なるパラメータを必要としないかぎり、補助チャンネルを構成する必要はありません。

次の例では、自動チャンネルの構成が行われていないため、手動で補助チャンネルの割当てが必要と想定しています。

```
# manually allocate at least one auxiliary channel
RUN
{
  ALLOCATE AUXILIARY CHANNEL c1 DEVICE TYPE DISK;
  RECOVER TABLESPACE users UNTIL SEQUENCE 13 THREAD 1;
}
```

---

---

**注意：** UNTIL TIME を指定する場合は、時刻に NLS\_DATE\_FORMAT パラメータと同じ書式を使用する必要があります。

---

---

**Recovery Manager** は、**TSPITR** の実行中に次の作業を自動的に実行します。

1. リカバリ対象となる表領域をオフラインにします。
2. 補助インスタンスにデータ・ファイルをリストアします。
3. リストアされたデータ・ファイルを指定された時間までリカバリします。
4. RESETLOGS オプションを使用して補助データベースをオープンします。
5. リカバリされた表領域内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 とリカバリ済データ・ファイルにある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します。
6. 補助インスタンスをシャットダウンします。
7. ターゲット制御ファイルが、補助データベースでリカバリされたばかりのリカバリ・セット内にあるデータベース・ファイルを指し示すようにするために、SWITCH コマンドを発行します。
8. 補助インスタン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、リカバリされたオブジェクトをアクセス可能にします。

---

---

**注意：** Recovery Manager はリカバリ中のデータ・ファイルをリストアするかわりに、データ・ファイルのコピーを検索します。データ・ファイルのコピーが見つからない場合には、Recovery Manager はリストア操作を実行し、切替えは実行しません。データ・ファイルの名前が CONFIGURE AUXNAME コマンドですでに構成しており、適切なデータ・ファイルのコピーが AUXNAME の位置に存在している場合、Recovery Manager は最適化のためリストアを中断して、リカバリした AUXNAME のデータ・ファイル・コピーへの切替えを実行します。

---

---

## TSPITR の実行後に使用するターゲット・データベースの準備

リカバリ・セット内の表領域は、Recovery Manager の TSPITR が正常に完了するまでオフラインのままです。

### TSPITR 後に再利用するためのターゲット・データベースの準備

1. Recovery Manager を起動し、ターゲット・データベースに接続します。たとえば、次のように実行します。

```
% rman TARGET SYS/oracle@trgt
```

2. リカバリ・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。たとえば、表領域 `users` の新規バックアップを作成します。

```
BACKUP TABLESPACE users;
```

---

---

**注意：** NOCATALOG モードで実行している場合、表領域で TSPITR を実行した後、同じ表領域に対して TSPITR の時点以前まで TSPITR を再実行することはできません。ただし、TSPITR の完了後に表領域の新規バックアップを作成すると、バックアップ後の任意の時点まで表領域に対して TSPITR を実行できます。

---

---

3. リカバリ・セットの表領域をオンライン化します。たとえば、次のように入力します。

```
SQL "ALTER TABLESPACE users ONLINE";
```

4. SQL\*Plus を使用して補助インスタンスに接続し、補助インスタンスをシャットダウンします。たとえば、次のように入力します。

```
sqlplus 'SYS/oracle@aux AS SYSDBA'<<EOF
SHUTDOWN ABORT;
EXIT;
EOF
```

5. 次のファイルをオペレーティング・システムから削除します。
  - TSPITR の実行時に一時的な位置にリストアされた補助セット・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

## TSPITR が失敗した場合の対応

TSPITR は様々な問題によって失敗す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は、補助インスタンスを停止して変換されたデータ・ファイル名を修正し、STARTUP NOMOUNT を発行して、Recovery Manager の TSPITR を再実行する必要があります。

また、エクスポート・ユーティリティのためのソート領域が不足しているために失敗することもあります。この場合は、recover.bsq ファイル（UNIX の場合は \$ORACLE\_HOME/rdbms/admin にあります）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。

```
#
# tspitr_7: do the incomplete recovery and resetlogs. This member is used once.
#
define tspitr_7
<<<
# make the control file point at the restored datafiles, then recover them
recover clone database tablespace &l&;
alter clone 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_tspitr_tmp
#          datafile '/tmp/aux_tspitr_tmp.dbf' size 500K";
#
}>>>
```

コメント行の最後の 2 行目から '#' 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 が失敗した場合は、次の手順に従ってください。



### 失敗した TSPITR への対応

1. Recovery Manager 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. 11-18 ページの「[TSPITR の実行](#)」で示される指示に従い、TSPITR を再実行します。



---

# Recovery Manager を使用したデータベースの複製

この章では、DUPLICATE コマンドを使用して、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。この章の項目は、次のとおりです。

- [複製データベースの作成 : 概要](#)
- [複製データベース用のファイルの作成](#)
- [複製用の補助インスタンスの準備 : 基本手順](#)
- [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](#)
- [データベースの複製の例](#)

**関連項目：** DUPLICATE コマンドを使用したスタンバイ・データベースの作成方法は、『Oracle9i Data Guard 概要および管理』を参照してください。

## 複製データベースの作成：概要

Recovery Manager の DUPLICATE コマンドを使用すると、元のターゲット・データベースを保持した状態で、そのターゲット・データベースのバックアップから**複製データベース**を作成できます。複製データベースは、元のデータベースと同一の場合と、そのサブセットのみを含む場合があります。複製データベースはターゲット・データベースのコピーで、様々な目的のために独立して実行できます。たとえば、次の項目に使用できます。

- バックアップおよびリカバリ手順のテスト
- 本番データベースから誤って削除された表などのデータのエクスポートおよび本番データベースへのインポート

たとえば、host1 上の本番データベースを host2 に複製してから、host2 上の複製データベースを使用してリストアおよびリカバリを試行することができます。その間、host1 の本番データベースは通常どおり使用できます。

複製データベースはスタンバイ・データベースとは異なります。ただし、どちらも DUPLICATE コマンドを使用して作成します。スタンバイ・データベースはプライマリ・データベースのコピーで、プライマリ・データベースからのアーカイブ・ログを使用して連続的または定期的に更新できます。プライマリ・データベースが損傷または破損した場合、スタンバイ・データベースへのフェイルオーバーを実行し、これを新しいプライマリ・データベースとして使用できます。一方、複製データベースはこの用途には使用できません。複製データベースはフェイルオーバーでの使用を意図したものではなく、スタンバイの様々なリカバリおよびフェイルオーバー・オプションはサポートしていません。

**関連項目：** DUPLICATE コマンドを使用したスタンバイ・データベースの作成方法は、[第 13 章「Recovery Manager によるスタンバイ・データベースの作成」](#)を参照してください。

## Recovery Manager によるデータベースの複製方法

データベースの複製を行う準備として、12-10 ページの「[複製用の補助インスタンスの準備：基本手順](#)」で説明されている**補助インスタンス**を最初に作成する必要があります。複製を行うには、ターゲット（プライマリ）データベースと NOMOUNT モードで起動された補助インスタンスの両方に Recovery Manager を接続する必要があります。

補助インスタンスには補助チャネルを少なくとも 1 つ割り当てる必要があります。複製のための基本作業は補助チャネルにより実行されます。補助チャネルが複製ホスト上のサーバー・セッションを起動します。次に、このチャネルはプライマリ・データベースの必要なバックアップをリストアし、これを使用して複製データベースを作成し、リカバリを開始します。

Recovery Manager がプライマリ・インスタンスと複製インスタンスに接続可能な間は、Recovery Manager クライアントはどのマシンでも実行できます。ただし、複製データベースの作成およびリカバリに使用するバックアップ、データ・ファイルのコピーおよびアーカイブ・ログは、すべて複製ホスト上のサーバー・セッションでアクセスできるようにする必要があります。複製ホストがプライマリ・ホストと異なる場合は、プライマリ・ホストのディスク上のバックアップとコピーを、プライマリ・データベース内のフル・パス名と同じ名前でもリモート・ノードから使用できるようにする必要があります。これを行うには、次の 2 つの方法があります。

- オペレーティング・システム・ユーティリティを使用して、バックアップ・ピース、イメージ・コピーおよびアーカイブ・ログをプライマリ・ホストからリモート・ホストの同一パスに移動します。
- NFS または共有ディスクを使用して、同一パスがリモート・ホストからアクセスできるようにします。

テープ・バックアップの場合は、バックアップを含むテープをリモート・ノードでアクセスできるようにする必要があります。これには、テープを物理的にリモート・ホストに移動するか、ネットワーク・テープ・サーバーを使用します。

複製操作の一環として、Recovery Manager は次の処理を行います。

- ターゲット・データ・ファイルを複製データベースにリストアし、使用可能な増分バックアップとアーカイブ・ログすべてを使用して、不完全リカバリを実行します。
- 補助データベースをシャットダウンしてから再起動します（クライアント側およびサーバー側初期化パラメータ・ファイルに関する問題は、12-12 ページの「[タスク 4: 補助インスタンスの起動](#)」を参照してください）。
- 不完全リカバリの実行後、RESETLOGS オプションを指定して複製データベースをオープンし、オンライン REDO ログを作成します（ただし、DUPLICATE ... FOR STANDBY を実行すると、データベースはオープンされません）。
- 複製データベース用に新しく一意の DBID を生成します（ただし、DUPLICATE ... FOR STANDBY を使用してスタンバイ・データベースを作成する場合、Recovery Manager では一意の DBID は作成されません）。

複製中は、ターゲット・データベースのオンライン REDO ログがバックアップされず、複製データベースに適用できないため、Recovery Manager は不完全リカバリを実行する必要があります。複製データベースのリカバリで Recovery Manager が最終的に使用できる項目は、ターゲット・データベースによりアーカイブされた最新の REDO ログです。

**関連項目：** Recovery Manager を使用したスタンバイ・データベースの作成方法は、第 13 章「[Recovery Manager によるスタンバイ・データベースの作成](#)」を参照してください。

## データベース複製オプション

データベースの複製では、次のことが可能です。

- リカバリ・カタログありまたはなしで `DUPLICATE` コマンドを実行できます。
- `SKIP READONLY` 句を使用して、読取り専用表領域をスキップできます。読取り専用表領域は、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できます。
- `SKIP TABLESPACE` 句を使用して、複製データベースから表領域を除外できます。`SYSTEM` 表領域や、ロールバック・セグメントまたは `UNDO` セグメントを含む表領域は、スキップできません。
- 複製データベースを新しいホストに作成できます。新しいホストでディレクトリ構造が同じ場合は、`NOFILENAMECHECK` オプションを指定して、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利用することができます。
- 複製データベースの作成時に `SET UNTIL` コマンドまたは `UNTIL` 句を指定した `DUPLICATE` コマンドを使用し、複製データベースを指定した時刻にリカバリします。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・ログに含まれた、一貫性のある最新の状態にまでリカバリするように設定されています。
- 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・カタログに登録できます。このオプションが可能なのは、複製時に、**Recovery Manager** によって複製データベースに新しい `DBID` が与えられるためです。

---

**注意：** 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ターゲット・データベースをコピーした場合、コピーしたデータベースの `DBID` は元のデータベースと同一のままです。コピー・データベースを元のデータベースと同じリカバリ・カタログに登録するには、`DBNEWID` ユーティリティを使用して `DBID` を変更する必要があります (『Oracle9i データベース・ユーティリティ』を参照してください)。

---

- 場合によっては、複製データベースの `DB_NAME` をターゲット・データベースの `DB_NAME` とは別のものに設定します。複製データベースがターゲット・データベースと同じ Oracle ホームにある場合は、`DB_NAME` 初期化パラメータを別のものにする必要があります。複製データベースが (同一マシンまたは別マシン上の) 別の Oracle ホームにある場合は、複製データベースの `DB_NAME` を複製ホスト上の同一 Oracle ホーム内にある他のデータベース名とは別の名前にします。

## 前提条件および制限事項

Recovery Manager での複製には、前提条件、制限事項および注意事項が多数あります。複製を行う前に、DUPLICATE 構文に関する制限事項を参照してください。『Oracle9i Recovery Manager リファレンス』

**関連項目：** DUPLICATE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## 複製データベース用のファイルの作成

データベースの複製時に、Recovery Manager は必要なデータベース・ファイルを作成します。この項では、ファイル作成の段階について説明します。

- [複製制御ファイルの作成](#)
- [複製オンライン REDO ログの作成](#)
- [複製データ・ファイルの命名](#)

## 複製制御ファイルの作成

DUPLICATE コマンドは、補助インスタンスの初期化パラメータ・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、初期化パラメータの設定を正しく設定して、ターゲット・データベースの本番ファイルを上書きしないようにしてください。

## 複製オンライン REDO ログの作成

複製オンライン REDO ログの名前を作成する際のオプションを、表 12-1 に示します。オプションは優先順位の高い順に並べられています。

表 12-1 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 句を指定します。	指定のとおり REDO ログを作成します。
2	LOG_FILE_NAME_CONVERT 初期化パラメータを設定します。	たとえば、log_* から duplog_* に、ターゲット・ファイル名を変換します。変換するペアは複数指定できます。  このパラメータによって、REDO ログはそのサイズが一致しているかぎり、存在し続けます。これは、このパラメータがログ作成時に REUSE パラメータを使用しているためです。
3	前述の手順のいずれも実行しません。	複製ファイル名をターゲット・ファイル名と同名にします。この方法を使用するときには、NOFILENAMECHECK オプションを指定する必要があります。また、複製データベースは別のホストに存在する必要があります。

Recovery Manager がどのようにオンライン REDO ログを改名するかは、優先順序によって決まります。たとえば、管理者が LOGFILE 句と LOG\_FILE\_NAME\_CONVERT パラメータの両方を指定した場合、Recovery Manager は LOGFILE 句を使用します。最初の 2 つのオプションのいずれも指定しない場合、Recovery Manager は複製データベース・ファイルに対して元のターゲット REDO ログ・ファイル名を使用します。

**注意：** ターゲット・データベースと複製データベースが同一ホスト内に存在する場合は、ターゲット・データベースが使用しているオンライン REDO ログの名前を使用しないでください。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しているオンライン・ログの名前は使用しないでください。



## 複製データ・ファイルの命名

複製データ・ファイルに異なる名前が必要な場合は、パラメータまたはコマンドを使用して指定する必要があります。データ・ファイルの改名に関するオプションを[表 12-2](#)に示します。オプションは優先順位の高い順に並べられています。

**表 12-2 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 コマンドを発行します。	新しいデータ・ファイルのファイル名を作成します。このコマンドは、ファイル名を変更するたびに、再発行する必要があります。
2	CONFIGURE AUXNAME コマンドを発行します。	新しいデータ・ファイルのファイル名を作成します。この設定は、CONFIGURE AUXNAME ... CLEAR 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します。	ターゲット・ファイル名を変換します。たとえば、/oracle/ から /dup/oracle/ などです。変換するペアは複数指定できます。このパラメータは、SET NEWNAME および CONFIGURE AUXNAME で改名されないファイルに対して使用できます。
4	前述の手順のいずれも実行しません。	ターゲット・ファイル名を再使用します。この方法を使用するときには、NOFILENAMECHECK オプションを指定する必要があります。また、複製データベースは別のホストに存在する必要があります。

Recovery Manager がどのようにファイルに命名するかは、優先順序によって決まります。たとえば、すべてのコマンドと初期化パラメータを指定した場合、Recovery Manager は SET NEWNAME を使用します。CONFIGURE AUXNAME および DB\_FILE\_NAME\_CONVERT を実行した場合、Recovery Manager は CONFIGURE 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、Recovery Manager 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 を指定すると、Recovery Manager は読取り専用表領域のデータ・ファイルを複製しません。複製が完了した後、[表 12-3](#) および [表 12-4](#) に説明されている複製データベースのビューを問い合わせると、データ・ファイルが読取り専用かどうかを判断できます。複製データ・ファイルの現在の状態を判断するには、STATUS 列および ENABLED 列が重要です。

表 12-3 複製データベースの V\$DATAFILE ビューに表示される読取り専用表領域

列	値
STATUS	OFFLINE
ENABLED	READ ONLY
NAME	MISSINGxxx

表 12-4 複製データベースのデータ・ディクショナリ・ビューに表示される読取り専用表領域

ビュー	列	値
DBA_DATA_FILES	STATUS	AVAILABLE
DBA_TABLESPACES	STATUS	READ ONLY

OFFLINE NORMAL 表領域のスキップ

表領域が OFFLINE NORMAL オプションでオフラインになると、Recovery Manager はこれらの表領域のデータ・ファイルを複製しません。複製の後で、手動でこれらの表領域を追加または削除することができます。データ・ファイルがオフラインかどうかを判断するには、表 12-5 および表 12-6 に説明されている複製データベースのビューを問い合わせます。データ・ファイルの現在の状態を判断するには、STATUS 列および ENABLED 列が重要です。

表 12-5 複製データベースの V\$ ビューに表示されるオフライン表領域

列	値
STATUS	OFFLINE
ENABLED	DISABLED
NAME	MISSINGxxx

表 12-6 複製データベースのデータ・ディクショナリ・ビューに表示されるオフライン表領域

ビュー	列	値
DBA_DATA_FILES	STATUS	AVAILABLE
DBA_TABLESPACES	STATUS	OFFLINE

IMMEDIATE オプションで表領域をオフラインにすると、この表領域にリカバリが必要になるため、Recovery Manager は表領域をスキップしないで複製します。オンライン表領域の場合と同じく、Recovery Manager では複製に対する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

CONFIGURE AUXNAME、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できます。この場合は NOFILENAMECHECK を指定して、エラー・メッセージを回避します。たとえば、ホスト A のデータベースに 2 つのファイルがあるとします。データ・ファイル 1 の名前は /oracle/data/file1.f、データ・ファイル 2 の名前は /oracle/data/file2.f です。ホスト B に複製するときには、次のように構成済みのチャンネルを使用して、複製を行います。

```
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
  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のため、DUPLICATE コマンドは正常に実行されません。ターゲットのデータ・ファイル 1 は /oracle/data/file2.f を使用しておらず、ターゲットのデータ・ファイル 2 も /oracle/data/file1.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、NOFILENAMECHECK を指定して、エラーを回避する必要があります。

---

---

**注意：** DB\_FILE\_NAME\_CONVERT パラメータによりファイル名がすべて変換される場合は、このパラメータは単独で使用します。つまり、SET NEWNAME または CONFIGURE AUXNAME は使用しないでください。Oracle9i では、この初期化パラメータで、変換するペアを複数指定できます。たとえば、/dsk1/dbs を /sby1/dbs に変換し、/dsk2/df を /sby2/dbs に変換するように、このパラメータを設定できます。

---

---

## 複製用の補助インスタンスの準備：基本手順

Recovery Manager の複製を実行する前に、次のタスクを実行してください。

- [タスク 1: 補助インスタンス用の Oracle パスワード・ファイルの作成](#)
- [タスク 2: 補助インスタンスへの Oracle Net 接続性の確認](#)
- [タスク 3: 補助インスタンス用の初期化パラメータ・ファイルの作成](#)
- [タスク 4: 補助インスタンスの起動](#)
- [タスク 5: ターゲット・データベースのマウントまたはオープン](#)
- [タスク 6: 必要なバックアップとアーカイブ REDO ログの確認](#)
- [タスク 7: 自動チャネルが構成されない場合の補助チャネルの割当て](#)

### タスク 1: 補助インスタンス用の 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、『Oracle9i データベース管理者ガイド』を参照してください。

### タスク 2: 補助インスタンスへの Oracle Net 接続性の確認

補助インスタンスは、Oracle Net 経由でアクセス可能である必要があります。先に進む前に、SQL\*Plus を起動して、補助インスタンスへの接続を確立できることを確認します。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。

## タスク 3: 補助インスタンス用の初期化パラメータ・ファイルの作成

補助インスタンス用に、クライアント側の初期化パラメータ・ファイルを作成し、少なくとも、次の表で説明するパラメータを設定します。

パラメータ	指定が必要な項目
DB_NAME	DUPLICATE コマンドで使用する名前と同じ名前。複製パラメータ・ファイルの DB_NAME は、DUPLICATE コマンドに指定したデータベース名と同じ名前に設定する必要があります。複製データベースがターゲット・データベースと同じ Oracle ホームにある場合は、複製とターゲットに同じデータベース名を使用することはできません。複製データベースがターゲットとは別の Oracle ホームにある場合は、複製データベースの DB_NAME を同一 Oracle ホーム内にある他のデータベース名と別の名前にしてください。
CONTROL_FILES	12-5 ページの「複製制御ファイルの作成」を参照してください。

必要に応じて、次の表で説明するパラメータを設定してください。

パラメータ	指定が必要な項目
DB_FILE_NAME_CONVERT	12-7 ページの「複製データ・ファイルの命名」を参照してください。
LOG_FILE_NAME_CONVERT	12-6 ページの「複製オンライン REDO ログの作成」を参照してください。

必要に応じて、Oracle Net を介して SYSDBA として接続することを可能にするパラメータを含む、その他のパラメータも設定します。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。データベースの複製先ホスト上ですべてのパスがアクセス可能であることを確認します。

次に示すのは、複製データベース用の初期化パラメータの例です。

```
DB_NAME=newdb
CONTROL_FILES=(/dup/oracle/oradata/trgt/control01.ctl,
               /dup/oracle/oradata/trgt/control02.ctl)
DB_FILE_NAME_CONVERT=(/oracle/oradata/trgt/,/dup/oracle/oradata/trgt/)
LOG_FILE_NAME_CONVERT=(/oracle/oradata/trgt/redo,/dup/oracle/oradata/trgt/redo)
```

クライアント側の初期化パラメータ・ファイルを作成した後は、SQL\*Plus から CREATE SPFILE コマンドを実行して、サーバー側初期化パラメータ・ファイルを作成できます。このコマンドは、インスタンス起動の前でも後でも実行できます。たとえば、次のようにサーバー側パラメータ・ファイルをデフォルトの場所に作成できます。クライアント側初期化パラメータ・ファイルのファイル名は FROM 句に指定します。

```
CREATE SPFILE FROM PFILE='/tmp/initDUPDB.ora';
```

サーバー側パラメータ・ファイルがデフォルトの場所にあると、DUPLICATE コマンドに PFILE パラメータを指定する必要があるため、データベースの複製時に便利です。Recovery Manager では複製プロセスの一部として補助インスタンスをシャットダウンしてから再起動するため、クライアント側パラメータ・ファイルを使用する場合は、どのクライアント側ファイルを使用するかを Recovery Manager に通知する必要があります。データベースの複製には、サーバー側パラメータ・ファイルの作成をお勧めします。

## タスク 4: 補助インスタンスの起動

Recovery Manager 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（必要に応じて、クライアント側のパラメータ・ファイルを指定）で起動します。この例では、oracle は SYSDBA レベルのユーザーのパスワードであり、aux は補助インスタンスのネット・サービス名です。

```
CONNECT SYS/oracle@aux AS SYSDBA
-- start instance with the server parameter file
STARTUP FORCE NOMOUNT
```

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 モードでのみ起動できます。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は実行しないでください。

Recovery Manager は、複製プロセスの一部として補助インスタンスを停止してから再起動します。このため、補助インスタンス用のサーバー側初期化パラメータ・ファイルをデフォルトの場所に作成しておくとう便利です。

補助インスタンス用のサーバー側初期化パラメータ・ファイルがデフォルトの場所でない場合は、DUPLICATE コマンドに PFILE パラメータを指定して、クライアント側初期化パラメータ・ファイルを指定する必要があります。補助インスタンス用のクライアント側パラメータ・ファイルは、複製の実行に使用される Recovery Manager 実行可能ファイルと同じホスト上に配置する必要があります。

## タスク 5: ターゲット・データベースのマウントまたはオープン

Recovery Manager で複製を開始する前に、ターゲット・データベースがマウントまたはオープン状態でない場合は、SQL\*Plus をターゲット・データベースに接続し（必要に応じて、クライアント側のパラメータ・ファイルを指定して）ターゲット・データベースをマウントまたはオープンします。たとえば、次のように入力します。

```
-- connect to target database
CONNECT SYS/oracle@trgt
-- mount or open target database
STARTUP
```

## タスク 6: 必要なバックアップとアーカイブ REDO ログの確認

複製ホスト上で、すべてのターゲット・データ・ファイルのバックアップが使用可能な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、複製操作は失敗します。データベースのバックアップは、データベース全体のバックアップである必要はありません。個々のデータ・ファイルの全体バックアップと増分バックアップを混合して使用できます。

次のいずれかの条件が満たされていることを確認してください。

- 必要な時刻、SCN またはログ順序番号までのリカバリに必要なすべてのアーカイブ REDO ログのバックアップが存在していること。これらのバックアップは、複製データベースの作成先のノードからアクセスできるようにする必要があります。たとえば、メディア・マネージャを使用してバックアップする場合、ターゲット・ホスト上にバックアップされたオブジェクトを複製ホストでリストアできるようにする必要があります。
- 複製データベースを作成するノードで、アーカイブ REDO ログがアクセス可能になっていること。ログは複製ホストに手動でコピーすることも、NFS を使用してターゲット・ホストにログ・ディレクトリをマウントすることもできます。

## タスク 7: 自動チャネルが構成されない場合の補助チャネルの割当て

Recovery Manager を起動して、ターゲット・データベース、複製データベースおよびリカバリ・カタログ・データベース（使用している場合）に接続します。すべてのインスタンスに接続可能な間は、Recovery Manager 実行可能ファイルをどのホストでも起動できます。複製インスタンスにクライアント側の初期化パラメータ・ファイルが必要な場合は、このファイルが Recovery Manager 実行可能ファイルを実行するのと同じホスト上に存在する必要があります。そのため注意してください。

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 TARGET SYS/oracle@trgt CATALOG rman/cat@catdb AUXILIARY SYS/oracle@aux
```

自動チャンネルを構成していない場合は、DUPLICATE コマンドを発行する前に、同じ RUN コマンド内で補助チャンネルを少なくとも 1 つ手動で割り当てます。チャンネル・タイプ (DISK または sbt) 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している必要があります。バックアップがディスク上にある場合は、割り当てるチャンネルが多いほど、短時間で実行できます。テープ・バックアップの場合、チャンネル数は操作に使用できるデバイスの数に制限してください。

```
RUN
{
  # to manually allocate a channel of type sbt issue:
  ALLOCATE AUXILIARY CHANNEL ch1 DEVICE TYPE sbt;

  # to manually allocate three auxiliary channels for disk issue (specifying whatever
  # channel id that you want):
  ALLOCATE AUXILIARY CHANNEL aux1 DEVICE TYPE DISK;
  ALLOCATE AUXILIARY CHANNEL aux2 DEVICE TYPE DISK;
  ALLOCATE AUXILIARY CHANNEL aux3 DEVICE TYPE DISK;
  .
  .
  .
  DUPLICATE ...
}
```

---

---

**注意：** 自動チャンネルを構成してある場合は、AUXILIARY オプションが指定されていなくても、**Recovery Manager** ではその構成済みチャンネルを複製に使用できます。ただし、補助チャンネルに特殊なパラメータ（たとえば、別のメディア管理サブシステムを指すなど）が必要な場合は、AUXILIARY オプションを指定して自動チャンネルを構成できます。

---

---



## 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手順は、構成によって異なります。この項では、次の項目について説明します。

- 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を持つリモート・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

最も簡単なケースは、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。この場合は、初期化パラメータ・ファイルを変更したり、複製データ・ファイルに新しいファイル名を設定する必要はありません。

1. 12-10 ページの「複製用の補助インスタンスの準備: 基本手順」の手順に従ってください。
2. DUPLICATE コマンドを実行します。その際、次の処理を必ず実行してください。
  - 自動チャンネルが構成されていない場合は、補助チャンネルを少なくとも 1 つ割り当てます。
  - DUPLICATE コマンドの NOFILENAMECHECK パラメータを指定します。
  - クライアント側パラメータ・ファイルを使用して補助インスタンスを起動する場合は、PFILE パラメータを指定します。クライアント側パラメータ・ファイルは、複製の実行に使用される Recovery Manager 実行可能ファイルと同じホスト上に配置する必要があります。

次の例では、Recovery Manager 実行可能ファイルを複製ホスト上で実行するとします。自動チャンネルを使用してデータベースを複製し、クライアント側の初期化パラメータ・ファイルを指定し、NOFILENAMECHECK オプションを指定します。

```
DUPLICATE TARGET DATABASE TO dupdb
# specify client-side parameter file (on same host as RMAN executable) for
# auxiliary instance if necessary
PFILE = /dup/oracle/dbs/initDUPDB.ora
NOFILENAMECHECK;
```

Recovery Manager は、構成済みチャンネルを自動的に割り当て、すべての増分バックアップ、アーカイブ REDO ログのバックアップおよびアーカイブ REDO ログを使用して、不完全リカバリを実行します。最後に、Recovery Manager で RESETLOGS オプションを使用してデータベースをオープンし、オンライン REDO ログを作成します。

## 異なる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は、いくつかの初期化パラメータを変更したり、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生成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 CONFIGURE AUXNAME コマンドを使用します。

この項では、次の項目について説明します。

- 初期化パラメータを使用した複製
- 初期化パラメータおよび LOGFILE 句を使用した複製
- SET NEWNAME を使用した複製
- CONFIGURE AUXNAME を使用した複製

**関連項目：** データ・ファイルのファイル名の変換の詳細は、12-7 ページの表 12-2 を参照してください。

### 初期化パラメータを使用した複製

この手順では、初期化パラメータを設定し、複製データ・ファイル名およびログ・ファイル名を変換することを想定します。

1. 12-10 ページの「複製用の補助インスタンスの準備 : 基本手順」の手順に従い、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、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位置から、複製ホストのディレクトリ構造内の同じ位置に、確実にコピーします。確実に次の設定を行ってください。
  - \_DEST または \_PATH で終了する初期化パラメータをすべて設定し、パス名を指定します。
  - DB\_FILE\_NAME\_CONVERT を設定します。このパラメータはすべてのターゲット・データ・ファイルを取得し、たとえば、/oracle/oradata/ から /dup/oracle/oradata/ のように、適切に名前を変換します。
  - LOG\_FILE\_NAME\_CONVERT を設定します。このパラメータはすべてのオンライン REDO ログを取得し、たとえば、/oracle/oradata/redo から /dup/oracle/oradata/redo のように、適切に名前を変換します。

---

**注意：** DB\_FILE\_NAME\_CONVERT および LOG\_FILE\_NAME\_CONVERT には、変換するペアを複数指定できます。たとえば、DB\_FILE\_NAME\_CONVERT で /fs1 を /dsk1 に変換し、/fs2 を /dsk2 に変換するように指定できます。

---

## 2. 複製の実行中に次の操作を実行します。

- 自動チャンネルが構成されていない場合は、補助チャンネルを少なくとも1つ割り当てます。
- 必要に応じて、ターゲット・データベースで使用したものと同一 REDO ログ・メンバーおよびグループの番号を指定します。
- クライアント側パラメータ・ファイルを使用して補助インスタンスを起動する場合は、PFILE パラメータを指定します。

次の例では、複製ホストがプライマリ・データベース・ホストと同じメディア・マネージャにアクセスできることを前提とします。この例では、自動 sbt チャンネルを使用してデータベースを複製し、複製ホスト上にあるサーバー側パラメータ・ファイルを使用して補助インスタンスを再起動します。

```
DUPLICATE
  TARGET DATABASE TO dupdb
  DEVICE TYPE sbt # restores from tape backups;
# Note that DUPLICATE DEVICE TYPE sbt works only if the sbt device is configured
# with CONFIGURE CHANNEL, CONFIGURE DEVICE TYPE, or CONFIGURE DEFAULT DEVICE.
```

Recovery Manager は、すべての増分バックアップ、アーカイブ REDO ログ・バックアップおよびアーカイブ REDO ログを使用して、不完全リカバリを実行します。

Recovery Manager は RESETLOGS オプションを使用して、データベースを停止、起動、オープンし、オンライン REDO ログを作成します。

## 初期化パラメータおよび LOGFILE 句を使用した複製

この手順では、初期化パラメータを設定して複製データ・ファイルの名前を変更し、LOGFILE 句を使用してオンライン REDO ログの名前とサイズを変更することが前提です。

1. 12-16 ページの「[初期化パラメータを使用した複製](#)」に説明されている手順を実行しますが、LOG\_FILE\_NAME\_CONVERT パラメータは設定しません。
2. 複製の実行中に次の操作を実行します。
  - 自動補助チャンネルが構成されていない場合は、補助チャンネルを少なくとも1つ割り当てます。
  - 複製データベースの REDO ログの名前とサイズを LOGFILE 句に指定します。
  - 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します。
  - クライアント側パラメータ・ファイルを使用して補助インスタンスを起動する場合は、PFILE パラメータを指定します。

次の例では、自動チャネルを使用してデータベースを複製し、初期化パラメータ・ファイルと LOGFILE 句を指定します。

```
DUPLICATE TARGET DATABASE TO dupdb
# specify client-side parameter file for auxiliary instance if necessary
PFILE = /dup/oracle/dbs/initDUPDB.ora
LOGFILE
'/dup/oracle/oradata/trgt/redo01.log' SIZE 200K,
'/dup/oracle/oradata/trgt/redo02.log' SIZE 200K,
'/dup/oracle/oradata/trgt/redo03.log' SIZE 200K;
```

### SET NEWNAME を使用した複製

この手順では、SET NEWNAME コマンドを使用して、複製データベース名を変更すると想定しています。

- 12-10 ページの「複製用の補助インスタンスの準備: 基本手順」の手順に従い、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、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位置から、複製ホストの同じ位置に、確実にコピーします。\_DEST または \_PATH で終了する初期化パラメータをすべて設定し、パス名を指定します。
- 複製の実行中に次の操作を実行します。
  - 自動補助チャネルが構成されていない場合は、補助チャネルを少なくとも 1 つ割り当てます。
  - 必要に応じて、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します。
  - 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します。
  - クライアント側パラメータ・ファイルを使用して補助インスタンスを起動する場合は、PFILE パラメータを指定します。

次の例では、自動チャネルとデフォルトのサーバー側初期化パラメータ・ファイルをデータベースの複製に使用し、LOGFILE 句を使用してオンライン REDO ログの名前とサイズを指定します。

```
RUN
{
# set new filenames for the datafiles
SET NEWNAME FOR DATAFILE 1 TO '/dup/oracle/oradata/trgt/system01.dbf';
SET NEWNAME FOR DATAFILE 2 TO '/dup/oracle/oradata/trgt/undotbs01.dbf';
. . .
# issue the duplicate command
DUPLICATE TARGET DATABASE TO dupdb
# create at least two online redo log groups
LOGFILE
GROUP1
```

```
(
  '/dup/oracle/oradata/trgt/redo01a.log',
  '/dup/oracle/oradata/trgt/redo01b.log',
  '/dup/oracle/oradata/trgt/redo01c.log';
) SIZE 200K,
GROUP2
(
  '/dup/oracle/oradata/trgt/redo02a.log',
  '/dup/oracle/oradata/trgt/redo02b.log',
  '/dup/oracle/oradata/trgt/redo02c.log';
) SIZE 200K,
GROUP3
(
  '/dup/oracle/oradata/trgt/redo03a.log',
  '/dup/oracle/oradata/trgt/redo03b.log',
  '/dup/oracle/oradata/trgt/redo03c.log';
) SIZE 200K;
```

Recovery Manager は、すべての増分バックアップ、アーカイブ REDO ログ・バックアップおよびアーカイブ REDO ログを使用して、不完全リカバリを実行します。Recovery Manager は RESETLOGS オプションを使用して、データベースを停止、起動、オープンし、オンライン・ログを作成します。

## CONFIGURE AUXNAME を使用した複製

この手順では、CONFIGURE AUXNAME コマンドを使用して、複製データベース名を変更すると想定しています。

- 12-10 ページの「複製用の補助インスタンスの準備: 基本手順」の手順に従い、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、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位置から、複製ホストのディレクトリ構造内の同じ位置に、確実にコピーします。\_DEST または \_PATH で終了する初期化パラメータをすべて設定し、パス名を指定します。
- 複製中に、次の操作を実行します。
  - 自動補助チャンネルが割り当てられていない場合は、補助チャンネルを少なくとも 1 つ割り当てます。
  - 必要に応じて、ターゲット・データベースで使用了のと同じ REDO ログ・メンバーおよびグループの番号を指定します。
  - クライアント側パラメータ・ファイルを使用して補助インスタンスを起動する場合は、PFILE パラメータを指定します。クライアント側パラメータ・ファイルは、複製の実行に使用される Recovery Manager 実行可能ファイルと同じホスト上に配置する必要があります。

次の例では、自動チャネルとクライアント側初期化パラメータ・ファイルをデータベースの複製に使用し、LOGFILE 句を使用してオンライン REDO ログの名前とサイズを指定します。

```
# run the DUPLICATE command
DUPLICATE TARGET DATABASE TO dupdb
# specify client-side parameter file for auxiliary instance if necessary
PFILE = /dup/oracle/dbs/initDUPDB.ora
.
.
.
# create at least two online redo log groups
LOGFILE
GROUP1
(
  '/dup/oracle/oradata/trgt/redo01a.log',
  '/dup/oracle/oradata/trgt/redo01b.log',
  '/dup/oracle/oradata/trgt/redo01c.log';
) SIZE 200K,
GROUP2
(
  '/dup/oracle/oradata/trgt/redo02a.log',
  '/dup/oracle/oradata/trgt/redo02b.log',
  '/dup/oracle/oradata/trgt/redo02c.log';
) SIZE 200K,
GROUP3
(
  '/dup/oracle/oradata/trgt/redo03a.log',
  '/dup/oracle/oradata/trgt/redo03b.log',
  '/dup/oracle/oradata/trgt/redo03c.log';
) SIZE 200K;
```

Recovery Manager は、すべての増分バックアップ、アーカイブ REDO ログ・バックアップおよびアーカイブ REDO ログを使用して不完全リカバリを実行し、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 ログを作成します。

3. データ・ファイルが誤って上書きされないように、データ・ファイルの補助名を消去します。たとえば、次のように入力します。

```
# un-specify auxiliary names for the datafiles
CONFIGURE AUXNAME FOR DATAFILE 1 CLEAR;
CONFIGURE AUXNAME FOR DATAFILE 2 CLEAR;
.
.
.
CONFIGURE AUXNAME FOR DATAFILE n CLEAR;
```

## 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、12-16 ページの「異なる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」に説明されている、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に従います。

ターゲット・データベースと同じ Oracle ホームにデータベースを複製できますが、別のホスト上で変換を行う場合と同じ方法で、ファイル名を変換する必要があることに注意してください。

---

---

**注意：** プライマリ・データベースと同じ Oracle ホームにデータベースを複製するときは、NOFILENAMECHECK オプションを使用しないでください。使用すると、DUPLICATE コマンドがエラーになる場合があります。

---

---

## データベースの複製の例

この項では、次の項目について説明します。

- 新しいファイル名の手動による設定 : 例
- 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 : 例
- 過去の複製データベースの作成 : 例
- クライアント側のパラメータ・ファイルを使用した複製 : 例

### 新しいファイル名の手動による設定 : 例

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 catdb を使用しています。
- ターゲット・データベース trgt は host1 にあり、8 つのデータ・ファイルが含まれています。
- このターゲットをリモート・ホスト host2 上のデータベース dupdb に複製する必要があります。
- host1 と host2 は、異なるファイル・システムを使用しています。
- host2 の ?/oradata/dup/ サブディレクトリにデータ・ファイルを格納する必要があります。
- 表領域 tools を複製データベースから除外しますが、他の表領域はすべて残す必要があります。
- 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、初期化パラメータ・ファイルを host1 から host2 の適切なファイルにコピーしてあります。

- `_DEST` または `_PATH` で終了し、パス名を指定する、すべての初期化パラメータをリセットしました。
- ターゲット・データベース内のすべてのデータ・ファイルとアーカイブ REDO ログには、ディスク・コピーまたはバックアップ・セットがあります。これらはディスクに格納されています。また、これら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、手動で `host2` に移動してあります。
- オンライン REDO ログ・グループが 2 つ必要です。各グループには、サイズ 200KB のメンバーが 2 つあります。
- デフォルト・デバイスを `sbt` に構成しました。メディア管理デバイスは `host2` からアクセスできます。
- 補助インスタンスは、デフォルトの場所にあるサーバー側初期化パラメータ・ファイルを使用します（したがって、`DUPLICATE` コマンドに `PFILE` パラメータは不要です）。

```
CONNECT TARGET;
CONNECT CATALOG rman/cat@catdb;
CONNECT AUXILIARY SYS/oracle@dupdb;

# note that a RUN command is necessary because you can only execute SET NEWNAME
# within a RUN command
RUN
{
  # the DUPLICATE command uses an automatic sbt channel
  SET NEWNAME FOR DATAFILE 1 TO '?/oradata/dup/system01.dbf';
  SET NEWNAME FOR DATAFILE 2 TO '?/oradata/dup/undotbs01.dbf';
  SET NEWNAME FOR DATAFILE 3 TO '?/oradata/dup/cwmlite01.dbf';
  SET NEWNAME FOR DATAFILE 4 TO '?/oradata/dup/drsys01';
  SET NEWNAME FOR DATAFILE 5 TO '?/oradata/dup/example01.dbf';
  SET NEWNAME FOR DATAFILE 6 TO '?/oradata/dup/indx01.dbf';
  # do not set a newname for datafile 7, because it is in the tools tablespace,
  # and you are excluding tools from the duplicate database
  SET NEWNAME FOR DATAFILE 8 TO '?/oradata/dup/users01.dbf';
  DUPLICATE TARGET DATABASE TO dupdb
    SKIP TABLESPACE tools
    LOGFILE
      GROUP 1 ('?/oradata/dup/redo01a.log',
               '?/oradata/dup/redo01b.log') SIZE 200K REUSE,
      GROUP 2 ('?/oradata/dup/redo02a.log',
               '?/oradata/dup/redo02b.log') 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 : 例

この例では、12-21 ページの「[新しいファイル名の手動による設定: 例](#)」と同じ内容を想定しています。さらに、複製データベースを、常にターゲット・データベースの現行の状態にしておくために、複製データベースを毎日更新する必要がある、と想定しています。

```
# start RMAN and then connect to the databases
CONNECT TARGET;
CONNECT CATALOG rman/cat@catdb;
CONNECT AUXILIARY SYS/oracle@dupdb;

# configure auxiliary names for the datafiles only once
CONFIGURE AUXNAME FOR DATAFILE 1 TO '?/oradata/dup/system01.dbf';
CONFIGURE AUXNAME FOR DATAFILE 2 TO '?/oradata/dup/undotbs01.dbf';
CONFIGURE AUXNAME FOR DATAFILE 3 TO '?/oradata/dup/cwmlite01.dbf';
CONFIGURE AUXNAME FOR DATAFILE 4 TO '?/oradata/dup/drsys01';
CONFIGURE AUXNAME FOR DATAFILE 5 TO '?/oradata/dup/example01.dbf';
CONFIGURE AUXNAME FOR DATAFILE 6 TO '?/oradata/dup/indx01.dbf';
# do not configure an auxiliary name for datafile 7, because it is in the tools
# tablespace, and you are excluding tools from the duplicate database
CONFIGURE AUXNAME FOR DATAFILE 8 TO '?/oradata/dup/users01.dbf';

# Create the duplicate database. Issue the same command daily
# to re-create the database, thereby keeping the duplicate
# in sync with the target.
DUPLICATE TARGET DATABASE TO dupdb
SKIP TABLESPACE tools
LOGFILE
  GROUP 1 ('?/oradata/dup/redo01a.log',
           '?/oradata/dup/redo01b.log') SIZE 200K REUSE,
  GROUP 2 ('?/oradata/dup/redo02a.log',
           '?/oradata/dup/redo02b.log') SIZE 200K REUSE;
```

## 過去の複製データベースの作成 : 例

この複製の例は、次の状況を想定しています。

- ターゲット・データベース `trgt` と複製データベース `dupdb` は、別々のホストにありますが、ディレクトリ構造はまったく同じです。
- 複製データベースのファイルに、ターゲットのファイルと同じ名前を付けます。
- リカバリ・カタログは使用しません。
- 自動チャンネルは構成されていません。
- `prod1` に 1 週間前にあったデータを表示できるように、複製データベースを 1 週間前の状態にリカバリする必要があります。

```
CONNECT TARGET SYS/oracle@trgt
CONNECT AUXILIARY SYS/oracle@dupdb
RUN
{
  SET UNTIL TIME 'SYSDATE-7';
  ALLOCATE AUXILIARY CHANNEL dupdb1 DEVICE TYPE sbt;
  DUPLICATE TARGET DATABASE TO dupdb
  NOFILENAMECHECK;
}
```

## クライアント側のパラメータ・ファイルを使用した複製 : 例

クライアント側初期化パラメータ・ファイルを使用して補助インスタンスを起動する場合は、複製の実行に使用される **Recovery Manager** 実行可能ファイルと同じホスト上にパラメータ・ファイルを配置する必要があります。次の例を想定してみます。

- ターゲット・ホストは `host_tar` で、複製ホストは `host_dup` です。
- `host_dup` 上のクライアント側初期化パラメータ・ファイルは、`?/dbs/initTEST.ora` です。
- ホスト `host_dup` と `host_tar` はネットワークでリンクされています。

この例では、**Recovery Manager** 実行可能ファイルは `host_tar` または `host_dup` のいずれでも実行できます（つまり、**Recovery Manager** セッションで `DUPLICATE` コマンドを実行できます）。

## host\_dup からの Recovery Manager の実行

host\_dup で実行可能ファイルを実行する場合は、次のようにしてデータベースを複製できます。

```
DUPLICATE
  TARGET DATABASE TO dupdb
  DEVICE TYPE sbt
  PFILE='?/dbs/initTEST.ora';
```

補助インスタンスにより使用される初期化パラメータ・ファイルが Recovery Manager 実行可能ファイルと同じノード上にあるため、パラメータ・ファイルのローカル・ファイル名を参照できます。

## host\_tar からの Recovery Manager の実行

この使用例では、Recovery Manager を、複製データベースがあるホスト上ではなくターゲット・データベースと同じホスト上で実行します。そのため、DUPLICATE コマンドに必要なクライアント側初期化パラメータ・ファイルは、Recovery Manager 実行可能ファイルと同じノード上にはありません。次のいずれかの操作を実行する必要があります。

- パラメータ・ファイルを host\_dup から host\_tar に送信します。
- NFS を使用して、パラメータ・ファイルを含む host\_dup ディレクトリを host\_tar ファイル・システムにマウントします。

**host\_dup から host\_tar へのパラメータ・ファイルのコピー** この使用例では、次のように、あるホストから別のホストにファイルを手動でコピーします。

```
% cp /net/host_dup/oracle/dbs/initTEST.ora /net/host_tar/tmp
```

host\_tar 上で Recovery Manager を起動し、次のシェル・スクリプトを使用して複製を実行できます。

```
#!/usr/bin/sh
rman TARGET SYS/oracle@trgt AUXILIARY SYS/oracle@dupdb <<EOF
DUPLICATE
  TARGET DATABASE TO dupdb
  DEVICE TYPE sbt
  PFILE='/tmp/initTEST.ora';
EOF
```

**host\_dup ファイル・システムの host\_tar へのマウント** この使用例では、/tmp をマウント・ポイントとして使用し、host\_dup ファイル・システムを host\_tar ファイル・システムにマウントします。host\_tar 上の /tmp/initTEST.ora ファイル名は、host\_dup 上にある ?/dbs/initTEST.ora ファイルを指します。次のように、複製を実行できます。

```
DUPLICATE
  TARGET DATABASE TO dupdb
  DEVICE TYPE sbt
  PFILE='/tmp/initTEST.ora';
```

---

## Recovery Manager によるスタンバイ・データベースの作成

この章では、Recovery Manager を使用したスタンバイ・データベースの作成方法を説明します。この章の項目は、次のとおりです。

- [Recovery Manager を使用したスタンバイ・データベースの準備](#)
- [Recovery Manager を使用したスタンバイ・データベースの作成 : 概要](#)
- [Recovery Manager およびスタンバイ・インスタンスの起動](#)
- [同じディレクトリ構造を持つリモート・ホストへのスタンバイ・データベースの作成](#)
- [異なるディレクトリ構造を持つリモート・ホストへのスタンバイ・データベースの作成](#)
- [ローカル・ホストへのスタンバイ・データベースの作成](#)
- [イメージ・コピーを使用したスタンバイ・データベースの作成](#)

**関連項目：** 9-23 ページ「[Recovery Manager を使用したスタンバイ・データベース・サイト上のファイルのバックアップ](#)」

## Recovery Manager を使用したスタンバイ・データベースの準備

Recovery Manager でスタンバイ・データベースを準備する手順は、複製データベースを準備する手順と基本的には同じです。ただし、スタンバイ・データベース固有の課題が解決されるように、12-15 ページの「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」に説明されている複製手順を修正する必要があります。

スタンバイ・データベースの準備とメンテナンスに関する説明は、『Oracle9i Data Guard 概要および管理』にあります。この章に説明されている Recovery Manager での作成手順を実行する前に、スタンバイ・データベースの概念とその作成方法を習得してください。

この項では、次の項目について説明します。

- Recovery Manager を使用したスタンバイ・データベースの準備
- Recovery Manager を使用したスタンバイ制御ファイルの作成
- スタンバイ制御ファイルのイメージ・コピーの作成
- Recovery Manager 使用時のスタンバイ・データベースのデータ・ファイルの命名
- Recovery Manager 使用時のスタンバイ・データベースのオンライン REDO ログの命名

## Recovery Manager を使用したスタンバイ・データベースの準備

プライマリ・データベースのバックアップからスタンバイ・データベースを作成するには、手動による方法を使用するか、Recovery Manager の DUPLICATE コマンドを使用します。作成手順を実行する前に、スタンバイ・インスタンスを準備する必要があります。表 13-1 に説明されている準備作業は、Recovery Manager を使用して実行できます。

表 13-1 Recovery Manager を使用したスタンバイ・データベースの準備

タスク	手順
スタンバイ・データベースの作成に使用するプライマリ・データベースのバックアップを作成します。	『Oracle9i Recovery Manager ユーザーズ・ガイド』に説明されているプライマリ・データベースの通常のバックアップ手順を使用します。
スタンバイ制御ファイルとして使用可能なプライマリ制御ファイルのバックアップを作成します（バックアップがない場合）。	13-3 ページ「Recovery Manager を使用したスタンバイ制御ファイルの作成」
スタンバイ・データ・ファイルのファイル名を選択します。	13-7 ページ「Recovery Manager 使用時のスタンバイ・データベースのデータ・ファイルの命名」
スタンバイ・データベースのオンライン REDO ログのファイル名を選択します。	13-8 ページ「Recovery Manager 使用時のスタンバイ・データベースのオンライン REDO ログの命名」

スタンバイ・データベースに必要な準備作業をすべて Recovery Manager が行うわけではありません。次のタスクは、手動で行う必要があります。

- 必要なすべての初期化パラメータをプライマリ初期化パラメータ・ファイルに設定します。
- スタンバイ・データベース用の初期化パラメータ・ファイルを作成し、必要なパラメータをすべて構成します。
- スタンバイ・インスタンスへの接続に必要な Oracle Net の設定および構成を実行します。
- スタンバイ・インスタンスを、制御ファイルをマウントせずに起動します。

**関連項目：** スタンバイ・データベースの準備の詳細（初期化パラメータの設定を含む）は、『Oracle9i Data Guard 概要および管理』を参照してください。Recovery Manager で正常にスタンバイ・データベース・ファイルを作成し、スタンバイ・データベースをマウントできるようにするには、このドキュメントに説明されている必須の準備作業をすべて実行しておく必要があります。

## Recovery Manager を使用したスタンバイ制御ファイルの作成

リリース 8.1.7 より前は、SQL の ALTER DATABASE 文を使用してスタンバイ制御ファイルを作成する必要がありました。現在は、Recovery Manager を使用して、スタンバイ・データベース制御ファイルとして使用可能なプライマリ・データベース制御ファイルの特殊なバックアップを作成できます。

Recovery Manager の使用により、次の項で説明されている方法のいずれかでスタンバイ制御ファイルを作成できます。

- [BACKUP コマンドを使用したスタンバイ制御ファイルの作成](#)
- [COPY コマンドを使用したスタンバイ制御ファイルの作成](#)
- [SQL で生成された制御ファイルのカatalog化によるスタンバイ制御ファイルの作成](#)

### BACKUP コマンドを使用したスタンバイ制御ファイルの作成

BACKUP コマンドを使用したスタンバイ制御ファイルの作成には、次のオプションがあります。

- プライマリ・データベースに接続し、FOR STANDBY オプションを指定して BACKUP CURRENT CONTROLFILE コマンドを実行して、スタンバイ制御ファイルを作成します。
- プライマリ・データベースに接続し、FOR STANDBY オプションを指定して BACKUP ... INCLUDE CURRENT CONTROLFILE コマンドを実行して、スタンバイ制御ファイルを作成します。

### スタンバイ制御ファイルのみを含むバックアップ・セットの作成

1. プライマリ・データベースに接続します。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 SYS/oracle@trgt CATALOG rman/cat@catdb
```

2. プライマリ・データベースをマウントまたはオープンします。たとえば、次のように入力します。

```
STARTUP MOUNT
```

3. BACKUP CURRENT CONTROLFILE FOR STANDBY コマンドを使用して、スタンバイ制御ファイルを作成します。次の例では、構成済みのチャンネルを使用してスタンバイ制御ファイルを作成し、アーカイブされていないログをすべてアーカイブし、バックアップされていないログを少なくとも 1 度はすべてバックアップします。

```
BACKUP CURRENT CONTROLFILE FOR STANDBY;  
SQL 'ALTER SYSTEM ARCHIVE LOG CURRENT'; # so backup is consistent and  
recoverable  
BACKUP ARCHIVELOG ALL NOT BACKED UP 1 TIMES;
```

スタンバイ制御ファイルにはタグを指定できません。

4. 必要に応じて、LIST コマンドを発行し、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します。

### スタンバイ制御ファイルの別のバックアップへの組み込み

1. プライマリ・データベースに接続します。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 SYS/oracle@trgt CATALOG rman/cat@catdb
```

2. プライマリ・データベースをバックアップし、プライマリ・データベースのバックアップにスタンバイ制御ファイルを含めます。次の例では、プライマリ・データ・ファイルをすべてディスクにバックアップし、スタンバイ制御ファイルも作成します。

```
BACKUP DATABASE  
INCLUDE CURRENT CONTROLFILE FOR STANDBY  
PLUS ARCHIVELOG;
```

PLUS ARCHIVELOG が指定されているため、Recovery Manager は BACKUP ARCHIVELOG ALL の実行の前後に、アーカイブされていないログをすべてアーカイブします。PLUS ARCHIVELOG を指定しない場合は、データベースのバックアップの後に次のコマンドを実行する必要があります。

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

3. LIST コマンドを発行して、バックアップ・セットとピースのリストを表示します。



## COPY コマンドを使用したスタンバイ制御ファイルの作成

スタンバイ制御ファイルを作成するには、FOR STANDBY オプションを指定した COPY CURRENT CONTROLFILE コマンドを使用できます。

### スタンバイ制御ファイルとして使用可能な制御ファイルのコピーの作成

1. プライマリ・データベースに接続します。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 SYS/oracle@trgt CATALOG rman/cat@catdb
```

2. 現行のプライマリ制御ファイルをコピーします。スタンバイ制御ファイルとして使用可能な現行の制御ファイルのコピーを作成するには、COPY CURRENT CONTROLFILE コマンドに FOR STANDBY オプションを指定します。たとえば、次のように入力します。

```
COPY CURRENT CONTROLFILE FOR STANDBY TO '/tmp/sby_control01.ctl';
```

3. LIST COPY コマンドを発行して、イメージ・コピーのリストを表示します。

## SQL で生成された制御ファイルのカタログ化によるスタンバイ制御ファイルの作成

リポジトリを更新して、ALTER DATABASE 文で生成されたスタンバイ制御ファイルを組み込むことができます。

### ALTER DATABASE で生成されたスタンバイ制御ファイルのカタログ

1. SQL の ALTER DATABASE 文を使用して、スタンバイ制御ファイルを作成します（まだ作成していない場合）。次の例では、SQL\*Plus を使用してスタンバイ制御ファイルを作成します。

```
SQL> ALTER DATABASE CREATE STANDBY CONTROLFILE AS '/tmp/sby_control01.ctl';
```

2. プライマリ・データベースに接続します。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 SYS/oracle@trgt CATALOG rman/cat@catdb
```

3. CATALOG コマンドを実行して、スタンバイ制御ファイルに関するメタデータをリカバリ・カタログに追加します。たとえば、次のように入力します。

```
CATALOG CONTROLFILECOPY '/tmp/sby_control01.ctl';
```

Recovery Manager では、ALTER DATABASE 文を使用して生成された制御ファイルを制御ファイルのコピーとみなします。

## スタンバイ制御ファイルのイメージ・コピーの作成

Recovery Manager では、次のいずれかの方法により生成された制御ファイル・コピーのイメージ・コピーを作成できます。

- Recovery Manager の COPY コマンド
- SQL の ALTER DATABASE 文

Recovery Manager で生成された制御ファイル・コピーまたは SQL で生成された制御ファイルのコピー

1. プライマリ・データベースに接続します。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 SYS/oracle@trgt CATALOG rman/cat@catdb
```

2. COPY CONTROLFILECOPY コマンドを使用して、スタンバイ制御ファイルをコピーします。たとえば、次のコマンドを実行します。

```
COPY CONTROLFILECOPY '/tmp/sby_control01.ctl' TO '/backup/sby_control01.ctl';
```

---

---

**注意：** Recovery Manager では、SQL で生成された制御ファイルと Recovery Manager で生成されたスタンバイ制御ファイルのコピーをまったく同じものとみなします。制御ファイルのヘッダーを検査し、スタンバイ制御ファイルか通常の制御ファイルかを判断します。

---

---

3. LIST COPY コマンドを発行して、イメージ・コピーのリストを表示します。

## Recovery Manager 使用時のスタンバイ・データベースのデータ・ファイルの命名

スタンバイ・データベースは、プライマリ・データベースと同じホスト上または別のホスト上で常駐できます。次の表は、2つのホスト上のディレクトリ構造が同じか異なるかにより、スタンバイ・データベースのデータ・ファイル名の変更が必要かどうかを示します。

スタンバイ・データベースのホスト	ディレクトリ構造	改名
プライマリと同じホスト	プライマリのホストと異なる	必要
プライマリと同じホスト	プライマリのホストと同じ	無効。スタンバイ・データベースは、プライマリ・データベースと同じホスト上の同じディレクトリには存在できません。
プライマリと異なるホスト	プライマリのホストと同じ	不要
プライマリと異なるホスト	プライマリのホストと異なる	必要

プライマリ・ホストとスタンバイ・ホストのディレクトリ構造が異なる場合は、スタンバイ・データ・ファイルの命名に関して次のオプションがあります。

- スタンバイ・データベースの初期化パラメータ `DB_FILE_NAME_CONVERT` を構成します。
- スタンバイ・データベースの作成時に、Recovery Manager で `CONFIGURE AUXNAME` コマンドまたは `SET NEWNAME` コマンドを使用します。

プライマリ・ホストとスタンバイ・ホストのディレクトリ構造が同じ場合は、スタンバイ・データ・ファイルの命名に関して次のオプションがあります。

- スタンバイ・ファイル名をプライマリ・ファイル名と同名にして（つまり、`DB_FILE_NAME_CONVERT` の設定や `CONFIGURE AUXNAME` または `SET NEWNAME` コマンドの発行を行わずに）、`DUPLICATE` コマンドの `NOFILENAMECHECK` オプションを指定します。
- `DB_FILE_NAME_CONVERT` パラメータを使用するか、`CONFIGURE AUXNAME` コマンドまたは `SET NEWNAME` コマンドを使用して、スタンバイ・データ・ファイル名を変更します。

`DB_FILE_NAME_CONVERT` を使用する場合、書式は次のようになります。

```
DB_FILE_NAME_CONVERT = ('oldstring1', 'newstring1', 'oldstring2', 'newstring2', ...)
```

たとえば、`DB_FILE_NAME_CONVERT` 初期化パラメータは次のように指定できます。

```
DB_FILE_NAME_CONVERT = ('/dbs/t1/', '/dbs/t1/s_', '/dbs/t2/', '/dbs/t2/s_')
```

スタンバイ制御ファイル内のデータ・ファイル名は様々な方法で指定できるため、設定に優先順位を付ける方法があります。スタンバイ・データベースのデータ・ファイルの命名に関する階層を表 13-2 に示します。

表 13-2 スタンバイ・データベースでのデータ・ファイルの命名の優先順位

	スタンバイ・データ・ファイルの命名方法	要件
1	CONFIGURE AUXNAME コマンドの発行。	リカバリ・カタログへの接続が必要です。また、NULL ではない AUXNAME をデータ・ファイルのカタログに格納する必要があります。
2	SET NEWNAME コマンドの発行。	このコマンドは、スタンバイ・データベースを作成する RUN ブロックで発行する必要があります。
3	スタンバイ制御ファイルに現在指定されているデータ・ファイルのファイル名。スタンバイ・ファイル名はプライマリ・ファイル名とまったく同じか、DB_FILE_NAME_CONVERT パラメータを使用して命名されます。	ファイル名が異なる場合は、スタンバイ初期化パラメータ・ファイルに DB_FILE_NAME_CONVERT パラメータを設定する必要があります。ファイル名が同じ場合は、DUPLICATE コマンドの NOFILENAMECHECK 句を指定する必要があります。

**関連項目：** DB\_FILE\_NAME\_CONVERT を使用したスタンバイ・ファイルの命名方法の詳細は、『Oracle9i Data Guard 概要および管理』を参照してください。

## Recovery Manager 使用時のスタンバイ・データベースのオンライン REDO ログの命名

オンライン REDO ログは、スタンバイ・データベースの作成時には作成されず、スタンバイ・データベース上にも作成されません。ただし、スタンバイ・データベースに対して実行する他のアクションによって、オンライン・ログを作成できます。詳細は、『Oracle9i Data Guard 概要および管理』を参照してください。作成されたオンライン・ログは、オンライン REDO ログに関する通常のルールに従ってメンテナンスおよびアーカイブされます。

スタンバイ・データベースでのオンライン REDO ログの命名の唯一のオプションは、スタンバイ制御ファイルに指定されるログ・ファイル名です。スタンバイ・データベースのオンライン・ログ・ファイル名をプライマリ・データベースのオンライン・ログ・ファイル名とは別のものにすることがある場合、1 つの方法は、スタンバイ・データベースの初期化パラメータ・ファイルに LOG\_FILE\_NAME\_CONVERT を設定して、オンライン REDO ログのファイル名を指定することです。

スタンバイ・データベースのオンライン REDO ログ・ファイル名を指定するときは、次の制限事項に注意してください。

- プライマリ・データベースとスタンバイ・データベースでのログ命名規則が異なる場合は、LOG\_FILE\_NAME\_CONVERT パラメータを使用してオンライン REDO ログに名前を付ける必要があります。
- オンライン REDO ログの名前変更に、SET NEWNAME コマンドまたは CONFIGURE AUXNAME コマンドを使用することはできません。
- オンライン REDO ログ・ファイル名の指定に、DUPLICATE コマンドの LOGFILE 句を使用することはできません。
- スタンバイのオンライン・ログ・ファイル名をプライマリのオンライン・ログ・ファイル名と同じにする場合は、DUPLICATE コマンドの NOFILENAMECHECK 句を指定する必要があります。そうでないと、スタンバイ・データベースが別ホストに作成される場合でもエラーが発生します。

**関連項目：** スタンバイ・データベースのオンライン REDO ログの命名の  
詳細は、『Oracle9i Data Guard 概要および管理』を参照してください。

## Recovery Manager を使用したスタンバイ・データベースの作成: 概要

スタンバイ・データベースの作成プロセスは、スタンバイ・データベースとプライマリ・データベースを同じホスト上に置くか別のホスト上に置くかにより異なります。この章で説明する手順では、『Oracle9i Data Guard 概要および管理』に説明されているスタンバイの設定および準備がすでに終了していることが前提です。必要な初期化パラメータの設定とネットワークの構成をすべて終了するまで、この手順は実行しないでください。

スタンバイ・インスタンスの準備に必要な手順を実行した後、Recovery Manager の DUPLICATE ...FOR STANDBY コマンドを実行し、プライマリ・データベースのバックアップからスタンバイ・データベースを作成します。FOR STANDBY OPTION を指定しない DUPLICATE コマンドで作成される複製データベースとは異なり、スタンバイ・データベースには新しい DBID は付けられません。このため、プライマリ・データベースのリポジトリにはスタンバイ・データベースを登録しないでください。

スタンバイ・データベースの作成手順は、スタンバイ・データベースの作成後にそのデータベースをリカバリするように Recovery Manager に対して指定するかどうかにより異なります。

**関連項目：** DUPLICATE コマンドを使用してスタンバイ・データベースではなく複製データベースを作成する方法は、[第 12 章「Recovery Manager を使用したデータベースの複製」](#)を参照してください。

## リカバリなしの Recovery Manager スタンバイの作成

デフォルトでは、Recovery Manager はスタンバイ・データベースの作成後にこのデータベースのリカバリは行いません。DUPLICATE コマンドの DORECOVER オプションを指定しない場合、Recovery Manager は次に示すスタンバイの作成手順を複製中に自動化します。

1. Recovery Manager がプライマリ・データベースとスタンバイ・データベース、およびリカバリ・カタログ（使用する場合）への接続を確立します。
2. Recovery Manager はリポジトリ（プライマリ制御ファイルまたはリカバリ・カタログのいずれか）に問い合せて、プライマリ・データベースのデータ・ファイルのバックアップとスタンバイ制御ファイルを識別します。
3. メディア・マネージャを使用する場合は、Recovery Manager がスタンバイ・ホスト上のメディア・マネージャに通信して、バックアップ・データを要求します。
4. Recovery Manager はスタンバイ制御ファイルをスタンバイ・ホストにリストアします。これでスタンバイ制御ファイルが作成されます。
5. Recovery Manager はプライマリ・データ・ファイルのバックアップおよびコピーをスタンバイ・ホストにリストアします。これでスタンバイ・データベースのデータ・ファイルが作成されます。
6. Recovery Manager はスタンバイ・データベースをマウントした状態にします。このデータベースを手動モードまたは管理リカバリ・モードにはしません。Recovery Manager は接続を切断し、スタンバイ・データベースのメディア・リカバリは動作しません。スタンバイ・データベースをリカバリ・カタログに登録しないよう注意してください。

## リカバリありの Recovery Manager スタンバイの作成

DUPLICATE コマンドの DORECOVER オプションを指定する場合は、13-10 ページの「[リカバリなしの Recovery Manager スタンバイの作成](#)」にある手順 1 から 5 までを実行します。手順 6 のかわりに、次の手順を実行します。

1. Recovery Manager はデータがすべてリストアされると、メディア・リカバリを開始します。リカバリにアーカイブ REDO ログが必要になり、リポジトリ内でのログの状態が AVAILABLE である場合、Recovery Manager はこれらのログを使用します。AVAILABLE とマークされているログがディスク上に実際に存在しない場合は、エラーを返します。AVAILABLE とマークされていないログが必要になった場合は、Recovery Manager はそのログのリストアを試みます。
2. 前述のいずれも指定されていない場合、Recovery Manager がスタンバイ・データベースを特定の時点、システム変更番号 (SCN)、ログ順序番号または生成済みの最新アーカイブ REDO ログまでリカバリします。

3. Recovery Manager は、メディア・リカバリの完了後、スタンバイ・データベースをマウントした状態にします。このデータベースを手動モードまたは管理リカバリ・モードにはしません。スタンバイ・データベースをリカバリ・カタログに登録しないよう注意してください。

---

**注意：** Recovery Manager がスタンバイ・データベースを作成した後、このデータベースを手動モードまたは管理リカバリ・モードにする前または読取り専用モードでオープンする前に、ギャップ・シーケンスがあれば解決しておく必要があります。ギャップ・シーケンスの解決方法の詳細は、『Oracle9i Data Guard 概要および管理』に説明されています。

---

スタンバイ・データベースを作成した後、Recovery Manager でスタンバイ・データベースのリカバリを行うには、スタンバイ制御ファイルがリカバリで使用する必要があります。したがって、次の条件を満たす必要があります。

- スタンバイ・データベースのリカバリ終了時刻は、スタンバイ制御ファイルのチェックポイント SCN より大きいか等しくする必要があります。
- スタンバイ制御ファイルのチェックポイント SCN を含むアーカイブ REDO ログを、スタンバイ・サイトでリカバリ用に使用できるようにする必要があります。

これらの条件が満たされていることを確認する 1 つの方法としては、スタンバイ制御ファイルの作成後に ALTER SYSTEM ARCHIVE LOG CURRENT 文を発行するという方法があります。この文は、プライマリ・データベースのオンライン・ログをアーカイブします。次に、Recovery Manager を使用して最新のアーカイブ・ログをバックアップするか、アーカイブ・ログをスタンバイ・サイトに移動します。

リカバリを実行するかどうかにかかわらず、Recovery Manager はスタンバイ・データベースの作成後、このデータベースをアクティブ化しません。スタンバイ・データベースをアクティブ化する唯一の方法は、ALTER DATABASE ACTIVATE STANDBY DATABASE 文を発行することです。スタンバイ・データベースがアクティブ化され REDO ログがリセットされると、古いプライマリ・データベースのバックアップとアーカイブ・ログはすべて、データベースの新規インカネーション用としては無効になります。

---

**注意：** この章の手順では、Recovery Manager のバックアップを使用してスタンバイ・データベースが作成されていることが前提です。Recovery Manager のイメージ・コピーを使用している場合は、13-23 ページの「[イメージ・コピーを使用したスタンバイ・データベースの作成](#)」を参照してください。

---

**関連項目：**

- スタンバイ・データベースの管理方法の詳細は、『Oracle9i Data Guard 概要および管理』を参照してください。
- Recovery Manager でスタンバイ・データベースを作成するときの DUPLICATE に関する制限事項のリストは、『Oracle9i Recovery Manager リファレンス』を参照してください。

## Recovery Manager およびスタンバイ・インスタンスの起動

スタンバイの作成にどの方法を使用するかにかかわらず、最初にスタンバイ・インスタンスを起動してから、Recovery Manager をそのインスタンスに接続します。詳しい手順は、スタンバイ・サイトとプライマリ・サイトのディレクトリ構造が異なるかどうかにより異なります。

### スタンバイ・インスタンスの起動

1. 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、初期化パラメータ・ファイルをターゲット・ホストからスタンバイ・ホストにコピーします。『Oracle9i Data Guard 概要および管理』に説明されているように、スタンバイ・データベースの初期化パラメータ・ファイルに必要なパラメータをすべて設定します。たとえば、スタンバイ・データベースを別ホストに異なるディレクトリ構造で作成する場合は、次を編集します。
  - `_DEST` あるいは `_PATH` で終了する初期化パラメータ。パス名を指定します。
  - `DB_FILE_NAME_CONVERT`。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 `sbytbs_*` へ適切に変換が行われるようにします。
  - `LOG_FILE_NAME_CONVERT`。すべてのオンライン REDO ログを獲得し、たとえば、`log_*` から `sbylog_*` へ適切に変換が行われるようにします。

スタンバイ・データベースの初期化パラメータ・ファイルのパラメータ設定の例を、次に示します。

```
STANDBY_ARCHIVE_DEST = /fs3/arc_dest/  
LOG_ARCHIVE_FORMAT = log%t_%s.arc  
DB_FILE_NAME_CONVERT = ('/oracle', '/fs3/oracle', '/dbf', '/fs3/oracle')  
LOG_FILE_NAME_CONVERT = ('/oracle', '/fs3/oracle')
```

2. SQL\*Plus を使用して、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、`sbdb1` に `SYS` (`SYSDBA` 権限を持つユーザー) として接続し、データベースを起動するには、次のように入力します。

```
SQL> CONNECT SYS/sys_pwd@sbdb1 AS SYSDBA  
SQL> STARTUP NOMOUNT PFILE=initSBDB1.ora
```



3. プライマリ・データベースがマウントまたはオープン状態でない場合は、SQL\*Plus を使用してマウントまたはオープンします。たとえば、prod1 に SYS で接続し、データベースをオープンするには、次のように入力します。

```
SQL> CONNECT SYS/sys_pwd@prod1 AS SYSDBA
SQL> STARTUP PFILE=initPROD1.ora
```

リカバリ・カタログを使用する場合は、カタログ・データベースがオープンしていることを確認してください。たとえば、catdb に SYS で接続し、リカバリ・カタログ・データベースをオープンするには、次のように入力します。

```
SQL> CONNECT SYS/oracle@catdb AS SYSDBA
SQL> STARTUP PFILE=initCATDB.ora
```

4. スタンバイ・インスタンスは、Oracle Net 経由でアクセスできる必要があります。先に進む前に、SQL\*Plus を使用して、スタンバイ・インスタンスへの接続が確立できることを確認します。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。注意してください。
5. ターゲット・データベース、スタンバイ・インスタンスおよびリカバリ・カタログ・データベース（使用している場合）に接続します。プライマリ・データベースは TARGET キーワードを使用して、スタンバイ・インスタンスは AUXILIARY キーワードを使用して指定します。

次の例では、オペレーティング・システム認証を使用して、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@sbdb1
```

## 同じディレクトリ構造を持つリモート・ホストへのスタンバイ・データベースの作成

最も簡単なのは、スタンバイ・データベースを別のホスト上に作成し、同じディレクトリ構造を使用する場合です。この場合は、スタンバイ初期化パラメータ・ファイルに DB\_FILE\_NAME\_CONVERT または LOG\_FILE\_NAME\_CONVERT パラメータを設定する、またはスタンバイ・データ・ファイルの新規ファイル名を設定する必要はありません。プライマリおよびスタンバイのデータ・ファイルとログはすべて同じファイル名です。

## リカバリを実行しないスタンバイ・データベースの作成

リカバリを実行しないでスタンバイ・データベースを作成する場合は、DUPLICATE コマンドに DORECOVER オプションを指定しないでください。デフォルトでは、スタンバイ・データベースはマウントした状態にあり、このデータベースのリカバリは行われません。

### リカバリを実行しないスタンバイ・データベースの作成

1. 13-12 ページの「[Recovery Manager およびスタンバイ・インスタンスの起動](#)」の手順に従ってください。スタンバイ初期化パラメータ・ファイルに必要なすべてのパラメータが設定されていることを確認します。
2. スタンバイ・データ・ファイルを作成するがリカバリしない場合は、複製中に次の手順に従います。
  - a. 自動チャネルを構成していない場合は、少なくとも 1 つの補助チャネルを手動で割り当てます。このチャネルで複製操作を実行します。
  - b. DUPLICATE コマンドに NOFILENAMECHECK を指定します。スタンバイおよびプライマリのデータ・ファイルとログが同じファイル名の場合は、NOFILENAMECHECK オプションが必要です。その他の場合、Recovery Manager はエラーを発行します。

たとえば、スタンバイ・データベースを作成するには、次のコマンドを実行します。

```
DUPLICATE TARGET DATABASE FOR STANDBY  
NOFILENAMECHECK;
```

## スタンバイ・データベースの作成とリカバリの実行

スタンバイ・データベースを作成し、リカバリを実行する場合は、DUPLICATE コマンドに DORECOVER オプションを指定します。

### スタンバイ・データベースの作成とリカバリの実行

1. 13-12 ページの「[Recovery Manager およびスタンバイ・インスタンスの起動](#)」の手順に従ってください。スタンバイ初期化パラメータ・ファイルに必要なすべてのパラメータが設定されていることを確認します。
2. 次の手順に従って、スタンバイ・データ・ファイルをリストアおよびリカバリします。
  - a. スタンバイ・データベースのリカバリ終了時刻が、スタンバイ制御ファイルのチェックポイント SCN より大きいと等しいことと、チェックポイント SCN を含むログがリカバリで使えることを確認します。
  - b. 必要の場合は SET コマンドを発行して、不完全リカバリの終了時刻、SCN またはログ順序番号を指定します。
  - c. 自動チャネルが構成されていない場合は、補助チャネルを少なくとも 1 つ手動で割り当てます。
  - d. DUPLICATE コマンドに NOFILENAMECHECK パラメータを指定し、DORECOVER オプションを使用します。

たとえば、スタンバイ・データベースの作成に構成済みチャンネルを使用するには、**Recovery Manager** プロンプトで次のように入力します。

```
# If desired, issue a LIST command to determine the SCN of the standby control
file.
# The SCN to which you recover must be greater than or equal to the standby
control
# file SCN.
LIST BACKUP OF CONTROLFILE;
LIST COPY OF CONTROLFILE;

RUN
{
  # If desired, issue a SET command to terminate recovery at a specified point.
  # SET UNTIL SCN 143508;
  DUPLICATE TARGET DATABASE FOR STANDBY
    NOFILENAMECHECK
    DORECOVER;
}
```

**Recovery Manager** は、すべての増分バックアップ、アーカイブ・ログ・バックアップおよびアーカイブ・ログを使用して、不完全リカバリを実行します。スタンバイ・データベースはマウントされた状態です。

## 異なるディレクトリ構造を持つリモート・ホストへのスタンバイ・データベースの作成

異なるディレクトリ構造を持つホストにスタンバイ・データベースを作成する場合は、スタンバイ・データベースのデータ・ファイルとオンライン REDO ログに新規ファイル名を指定する必要があります。次のようにします。

- スタンバイ初期化パラメータ・ファイルに `LOG_FILE_NAME_CONVERT` を設定し、スタンバイ・データベースのオンライン REDO ログに名前を付けます。`LOG_FILE_NAME_CONVERT` を設定しない場合は、`DUPLICATE` コマンドに `NOFILENAMECHECK` オプションを指定する必要があります。
- スタンバイ初期化パラメータ・ファイルに `DB_FILE_NAME_CONVERT` を設定し、スタンバイ・データ・ファイルに名前を付けます。
- **Recovery Manager** の `DUPLICATE` コマンドを使用してデータ・ファイルに名前を付ける場合は、`SET NEWNAME` コマンドまたは `CONFIGURE AUXNAME` コマンドを発行します。

異なるディレクトリ構造を持つホストにスタンバイ・データベースを作成するときは、次の項に説明されている手順のいずれかを使用します。

- [DB\\_FILE\\_NAME\\_CONVERT](#) を使用したスタンバイ・データベース・ファイルの命名
- [SET NEWNAME](#) コマンドを使用したスタンバイ・データベース・ファイルの命名
- [CONFIGURE AUXNAME](#) コマンドを使用したスタンバイ・データベース・ファイルの命名

### 関連項目：

- [SET NEWNAME](#) と [CONFIGURE AUXNAME](#) の違いの詳細は、12-7 ページの「[複製データ・ファイルの命名](#)」を参照してください。
- スタンバイ・データベースの準備と作成の詳細は、『Oracle9i Data Guard 概要および管理』を参照してください。

## DB\_FILE\_NAME\_CONVERT を使用したスタンバイ・データベース・ファイルの命名

この手順では、スタンバイ・データ・ファイルの命名に [DB\\_FILE\\_NAME\\_CONVERT](#) を使用し、スタンバイ・オンライン REDO ログの命名に [LOG\\_FILE\\_NAME\\_CONVERT](#) を使用します。

**関連項目：** [DB\\_FILE\\_NAME\\_CONVERT](#) および [LOG\\_FILE\\_NAME\\_CONVERT](#) パラメータを使用したスタンバイ・データベース・ファイルの命名方法の例は、『Oracle9i Data Guard 概要および管理』を参照してください。

## リカバリを実行しないスタンバイ・データベースの作成

リカバリを実行しないでスタンバイ・データベースを作成する場合は、[DUPLICATE](#) コマンドに [DORECOVER](#) オプションを指定しないでください。デフォルトでは、スタンバイ・データベースはマウントした状態にあり、このデータベースのリカバリは行われません。

### パラメータを使用してリカバリなしのスタンバイ・ファイルを命名する手順

1. 13-12 ページの「[Recovery Manager およびスタンバイ・インスタンスの起動](#)」の手順に従ってください。スタンバイ初期化パラメータ・ファイルに必要なすべてのパラメータが設定されていることを確認します。
2. [DUPLICATE](#) コマンドを実行します。たとえば、次のように実行します。

```
DUPLICATE TARGET DATABASE FOR STANDBY;
```

バックアップがリストアされた後、スタンバイ・データベースはマウントされた状態です。

## スタンバイ・データベースの作成とリカバリの実行

DB\_FILE\_NAME\_CONVERT を使用してスタンバイ・データ・ファイルを命名し、LOG\_FILE\_NAME\_CONVERT を使用してスタンバイ・オンライン REDO ログを命名した後、スタンバイ・データベースを作成してリカバリを実行するには、DUPLICATE コマンドに DORECOVER オプションを指定します。この手順の手順は、13-14 ページの「[スタンバイ・データベースの作成とリカバリの実行](#)」と同じです。

## SET NEWNAME コマンドを使用したスタンバイ・データベース・ファイルの命名

この手順では、SET NEWNAME コマンドを使用して、スタンバイ・データ・ファイルに命名します。

### リカバリを実行しないスタンバイ・データベースの作成

リカバリを実行しないでスタンバイ・データベースを作成する場合は、DUPLICATE コマンドに DORECOVER オプションを指定しないでください。デフォルトでは、スタンバイ・データベースはマウントした状態にあり、このデータベースのリカバリは行われません。

### SET NEWNAME コマンドを使用してリカバリなしのスタンバイ・データベース・ファイルを命名する手順

1. 13-12 ページの「[Recovery Manager およびスタンバイ・インスタンスの起動](#)」の手順に従ってください。スタンバイ初期化パラメータ・ファイルに必要なすべてのパラメータが設定されていることを確認します。
2. DUPLICATE コマンドを実行します。次の操作を実行します。
  - a. 自動チャネルが構成されていない場合は、補助チャネルを少なくとも 1 つ手動で割り当てます。
  - b. SET NEWNAME コマンドを使用して、スタンバイ・データベースのデータ・ファイルの新規ファイル名を指定します。
  - c. DUPLICATE コマンドを発行します。

次の例では、構成済みのチャネルを使用してスタンバイ・データベースを作成します。

```
RUN
{
  # set new filenames for the datafiles
  SET NEWNAME FOR DATAFILE 1 TO '?/dbs/standby_data_01.f';
  SET NEWNAME FOR DATAFILE 2 TO '?/dbs/standby_data_02.f';
  .
  .
  .
  # run the DUPLICATE command
```

```
        DUPLICATE TARGET DATABASE FOR STANDBY;  
    }
```

## スタンバイ・データベースの作成とリカバリの実行

スタンバイ・データベースを作成し、リカバリを実行する場合は、DUPLICATE コマンドに DORECOVER オプションを指定します。

### SET NEWNAME コマンドを使用してスタンバイ・データベースのファイルを命名しリカバリを実行する手順

1. 13-12 ページの「[Recovery Manager およびスタンバイ・インスタンスの起動](#)」の手順に従ってください。スタンバイ初期化パラメータ・ファイルに必要なすべてのパラメータが設定されていることを確認します。
2. DUPLICATE コマンドを実行します。次の手順に従います。
  - a. リカバリの終了時刻が、スタンバイ制御ファイルのチェックポイント SCN より大きいと等しいことと、チェックポイント SCN を含むログがリカバリで利用できることを確認します（13-10 ページの「[リカバリありの Recovery Manager スタンバイの作成](#)」を参照してください）。
  - b. 必要な場合は SET コマンドを発行して、不完全リカバリの終了時刻、SCN またはログ順序番号を指定します。
  - c. 自動チャンネルが構成されていない場合は、補助チャンネルを少なくとも 1 つ手動で割り当てます。
  - d. スタンバイ・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します。
  - e. DUPLICATE コマンドを、DORECOVER オプションで発行します。

たとえば、スタンバイ・データベースの作成に構成済みチャンネルを使用するには、Recovery Manager プロンプトで次のように入力します。

```
# If desired, issue a LIST command to determine the SCN of the standby control  
file.  
# The SCN to which you recover must be greater than or equal to the control file  
SCN.  
  
LIST BACKUP OF CONTROLFILE;  
LIST COPY OF CONTROLFILE;  
RUN  
{  
    # If desired, issue a SET command to terminate recovery at a specified point.  
    # SET UNTIL TIME 'SYSDATE-7';  
  
    # Set new filenames for the datafiles  
    SET NEWNAME FOR DATAFILE 1 TO '?:/dbs/standby_data_01.f';  
    SET NEWNAME FOR DATAFILE 2 TO '?:/dbs/standby_data_02.f';  
    .  
}
```

```

.
.
DUPLICATE TARGET DATABASE FOR STANDBY
DORECOVER;
}

```

Recovery Manager は、すべての増分バックアップ、アーカイブ REDO ログ・バックアップおよびアーカイブ REDO ログを使用して、不完全リカバリを実行します。スタンバイ・データベースはマウントされた状態です。

## CONFIGURE AUXNAME コマンドを使用したスタンバイ・データベース・ファイルの命名

この手順では、CONFIGURE AUXNAME コマンドを使用して、スタンバイ・データ・ファイルに命名します。

**関連項目：** 12-19 ページ「[CONFIGURE AUXNAME を使用した複製](#)」

### リカバリを実行しないスタンバイ・データベースの作成

リカバリを実行しないでスタンバイ・データベースを作成する場合は、DUPLICATE コマンドに DORECOVER オプションを指定しないでください。デフォルトでは、スタンバイ・データベースはマウントした状態にあり、このデータベースのリカバリは行われません。

### CONFIGURE AUXNAME を使用してリカバリなしのスタンバイ・データベース・ファイルを命名する手順

1. 13-12 ページの「[Recovery Manager およびスタンバイ・インスタンスの起動](#)」の手順に従ってください。スタンバイ初期化パラメータ・ファイルに必要なすべてのパラメータが設定されていることを確認します。
2. データ・ファイルの補助名を構成します。たとえば、次のように入力します。

```

# set auxiliary names for the datafiles
CONFIGURE AUXNAME FOR DATAFILE 1 TO '/oracle/auxfiles/aux_1.f';
CONFIGURE AUXNAME FOR DATAFILE 2 TO '/oracle/auxfiles/aux_2.f';
.
.
.
CONFIGURE AUXNAME FOR DATAFILE n TO '/oracle/auxfiles/aux_n.f';

```

3. DUPLICATE コマンドを実行します。自動チャネルが構成されていない場合は、次の例のように、DUPLICATE コマンドを発行する前に補助チャネルを少なくとも 1 つ手動で割り当てます。

```

RUN
{
  # allocate at least one auxiliary channel of type DISK or sbt

```

```
ALLOCATE AUXILIARY CHANNEL standby1 DEVICE TYPE sbt;  
.  
.  
.  
# issue the DUPLICATE command  
DUPLICATE TARGET DATABASE FOR STANDBY;  
}
```

4. データ・ファイルが誤って上書きされないように、データ・ファイルの補助名を未指定にします。たとえば、**Recovery Manager** プロンプトから次のように入力します。

```
# un-specify auxiliary names for the datafiles  
CONFIGURE AUXNAME FOR DATAFILE 1 CLEAR;  
CONFIGURE AUXNAME FOR DATAFILE 2 CLEAR;  
.  
.  
.  
CONFIGURE AUXNAME FOR DATAFILE n CLEAR;
```

## スタンバイ・データベースの作成とリカバリの実行

スタンバイ・データベースを作成し、リカバリを実行する場合は、**DUPLICATE** コマンドに **DORECOVER** オプションを指定します。

### **CONFIGURE AUXNAME** を使用してスタンバイ・ファイルを命名しリカバリを実行する手順

1. 13-12 ページの「**Recovery Manager およびスタンバイ・インスタンスの起動**」の手順に従ってください。スタンバイ初期化パラメータ・ファイルに必要なすべてのパラメータが設定されていることを確認します。
2. データ・ファイルの補助名を設定します。たとえば、次のように入力します。

```
# set auxiliary names for the datafiles  
CONFIGURE AUXNAME FOR DATAFILE 1 TO '/oracle/auxfiles/aux_1.f';  
CONFIGURE AUXNAME FOR DATAFILE 2 TO '/oracle/auxfiles/aux_2.f';  
.  
.  
.  
CONFIGURE AUXNAME FOR DATAFILE n TO '/oracle/auxfiles/aux_n.f';
```

3. **DUPLICATE** コマンドを実行します。次の手順に従います。
  - リカバリの終了時刻が、スタンバイ制御ファイルのチェックポイント SCN より大きいと等しいことと、チェックポイント SCN を含むログがリカバリで使用できることを確認します (13-10 ページの「**リカバリありの Recovery Manager スタンバイの作成**」を参照)。
  - 必要の場合は **SET** コマンドを発行して、不完全リカバリの終了時刻、SCN またはログ順序番号を指定します。



- 自動チャンネルが構成されていない場合は、補助チャンネルを少なくとも1つ手動で割り当てます。
- `DUPLICATE TARGET DATABASE FOR STANDBY` コマンドを発行します。

たとえば、スタンバイ・データベースの作成に構成済みチャンネルを使用するには、**Recovery Manager** プロンプトで次のように入力します。

```
# If desired, issue a LIST command to determine the SCN of the standby control
file.
# The SCN to which you recover must be greater than or equal to the control file
SCN.
LIST BACKUP OF CONTROLFILE;
LIST COPY OF CONTROLFILE;

DUPLICATE TARGET DATABASE FOR STANDBY
DORECOVER;
```

**Recovery Manager** は、すべての増分バックアップ、アーカイブ REDO ログ・バックアップおよびアーカイブ REDO ログを使用して、不完全リカバリを実行します。スタンバイ・データベースはマウントされた状態です。

4. データ・ファイルが誤って上書きされないように、データ・ファイルに設定された補助名を消去します。たとえば、**Recovery Manager** プロンプトから次のように入力します。

```
# un-specify auxiliary names for the datafiles
CONFIGURE AUXNAME FOR DATAFILE 1 CLEAR;
CONFIGURE AUXNAME FOR DATAFILE 2 CLEAR;
.
.
.
CONFIGURE AUXNAME FOR DATAFILE n CLEAR;
```

## ローカル・ホストへのスタンバイ・データベースの作成

プライマリ・データベースと同じホストにスタンバイ・データベースを作成するときは、13-15 ページの「異なるディレクトリ構造を持つリモート・ホストへのスタンバイ・データベースの作成」に説明されている、異なるディレクトリ構造を持つリモート・ホストへの複製と同じ手順に従います。

スタンバイ・データベースをプライマリ・データベースと同じホスト上に作成する場合は、次の制限事項に注意してください。

- ターゲット・データベースと同じ Oracle ホームにスタンバイ・データベースを作成できますが、別のホスト上で変換を行うときに使用するのと同じ方法でファイル名を変換する必要があります。つまり、同一 Oracle ホーム内のスタンバイ・データベースを、あたかも異なるディレクトリ構造を持つ別ホスト上のデータベースであるかのように扱う必要があります。プライマリ・データベースとスタンバイ・データベースが同一マシン上にある場合は、プライマリとスタンバイのデータベース・ファイルに同じ名前を使用しないでください。
- 両方のデータベースが同一ホストの同一 Oracle ホームにある場合は、LOCK\_NAME\_SPACE 初期化パラメータを設定する必要があります。

---

---

**注意：** スタンバイ・データベースをプライマリ・データベースと同じ Oracle ホームに作成するときは、NOFILENAMECHECK オプションを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・データベースが上書きされるか、DUPLICATE コマンドにエラーが発生し失敗します。

---

---

## イメージ・コピーを使用したスタンバイ・データベースの作成

この項では、次の項目について説明します。

- [イメージ・コピーを使用したスタンバイ・データベースの作成:概要](#)
- [コピーとデータ・ファイルが同じ名前を使用する場合のスタンバイ・データベースの作成](#)
- [コピーとデータ・ファイルが異なる名前を使用する場合のスタンバイ・データベースの作成](#)
- [バックアップとイメージ・コピーを使用したスタンバイ・データベースの作成:例](#)

### イメージ・コピーを使用したスタンバイ・データベースの作成:概要

Recovery Manager のイメージ・コピーを使用してスタンバイ・データ・ファイルを作成するときの主な制限事項は、データ・ファイルとアーカイブ・ログのイメージ・コピーのファイル名をプライマリ・ホストとスタンバイ・ホストで同じにする必要があるということです。たとえば、データ・ファイル 1 の名前がプライマリ・ホストでは `/oracle/dbs/df1.f` であるとして、Recovery Manager の COPY コマンドを使用してこのデータ・ファイルを `/data/df1.f` にコピーする場合は、このイメージ・コピーがスタンバイ・ホスト上に `/data/df1.f` という同じファイル名で存在する必要があります。存在しない場合、Recovery Manager はスタンバイ・イメージ・コピーのメタデータをリポジトリで見つけられません。

イメージ・コピーでスタンバイ・ホストに移入する方法としては、2つの方法があります。

- `ftp` またはその他のユーティリティを使用して手動でイメージ・コピーを転送する方法
- ネットワーク・ファイル・システム (NFS) を使用してスタンバイ・ディレクトリ構造をプライマリ・ホストにマウントする方法

NFS の方法を使用する場合は、スタンバイ・ホストのディレクトリに対応するディレクトリをプライマリ・ホストに作成します。この方法を使用する場合は、両方のマシン上の NFS マウント・ポイントが同一のディレクトリ名である必要があります。たとえば、プライマリ・ホスト上の `/data` をスタンバイ・ホスト上の `/data` にマップすることはできますが、プライマリ・ホストの `/data` をスタンバイ・ホストの `/dir` にマップすることはできません (ただし、これは、UNIX のシンボリック・リンクや Windows NT の論理ドライブなど機能を使用しない場合に限りです)。

スタンバイ・ホスト上のイメージ・コピーのファイル名は、プライマリ・ホスト上のイメージ・コピーのファイル名と同じにする必要があります。ただし、`SET NEWNAME` コマンドまたは `DB_FILE_NAME_CONVERT` 初期化パラメータを使用して、スタンバイ・データ・ファイルに異なるパス名を指定できます。

たとえば、スタンバイ・ホストのデータ・ファイル 1 のイメージ・コピーの名前が /data/df1.f の場合に、初期化パラメータまたは Recovery Manager のコマンドを使用して、スタンバイ制御ファイルにパス名 /oracle/sb/df1.f を指定できます。物理イメージ・コピーの名前を手動で変更しないように注意してください。DUPLICATE コマンドを実行すると、Recovery Manager はイメージ・コピー /data/df1.f をリストアし、初期化パラメータまたは Recovery Manager コマンドの情報に基づいて、スタンバイ・データ・ファイル 1 を /oracle/sb/df1.f として作成します。

表 13-3 は、NFS を使用してデータ・ファイルを 1 つ含むスタンバイ・データベースを作成する 2 つの方法を示します。

表 13-3 イメージ・コピーを使用したスタンバイ・データベースの作成：例

NFS マウント・ポイント	プライマリのデータ・ファイル名	イメージ・コピー・ファイル名	スタンバイ・データ・ファイル名	手順
/data (両方のホストで同じ)	/oracle/dbs/df1.f	/data/df1.f	/data/df1.f (イメージ・コピーと同じパス名)	13-25 ページ「コピーとデータ・ファイルが同じ名前を使用する場合のスタンバイ・データベースの作成」
/data (両方のホストで同じ)	/oracle/dbs/df1.f	/data/df1.f	/oracle/sb/df1.f (イメージ・コピーとは違うパス名)	13-26 ページ「コピーとデータ・ファイルが異なる名前を使用する場合のスタンバイ・データベースの作成」

表 13-3 では、スタンバイのディレクトリ構造がプライマリ・ホストにマウントされ、両方のホストでマウント・ポイントが /data であることが前提です。プライマリ・ホストにはスタンバイ・ホストのディレクトリ構造がマウントされるため、プライマリ・ホスト上にイメージ・コピー /data/df1.f を作成するとき、実際にはイメージ・コピー /data/df1.f をスタンバイ・ホストに作成していることになります。

最初の例では、スタンバイ・データ・ファイルにイメージ・コピーと同じ名前を付けます。スタンバイ・データベースの作成に Recovery Manager を使用する必要がないため、これが最も簡単な方法です。最初に、スタンバイの初期化パラメータ・ファイルに DB\_FILE\_NAME\_CONVERT パラメータを設定し、プライマリのデータ・ファイル名 /oracle/dbs/df1.f をスタンバイのファイル名 /data/df1.f に変換します。次に、ファイルをスタンバイ・ホストにコピーし、スタンバイ・データベースをマウントします。

2 番目の例では、スタンバイ・データ・ファイルとイメージ・コピーに異なる名前を使用します。このスタンバイ・データベースを作成するには、DUPLICATE コマンドを実行します。DUPLICATE コマンドは、データ・ファイル 1 のイメージ・コピーをリストアし、SET NEWNAME コマンドまたは DB\_FILE\_NAME\_CONVERT 初期化パラメータに基づいてこの名前を変更します。

## コピーとデータ・ファイルが同じ名前を使用する場合のスタンバイ・データベースの作成

この手順では、スタンバイ・データ・ファイルとプライマリ・データ・ファイルのイメージ・コピーに同じファイル名を使用することが前提です。

### コピーとスタンバイ・データ・ファイルが同じファイル名を共有する場合のスタンバイ・データベースの作成

1. プライマリ・データベースおよび必要な場合はリカバリ・カタログ・データベースに接続した後、プライマリ・データベースをマウントします（ただしオープンしません）。マウントする前にデータベースが正常にクローズされていることを確認します。たとえば、次のように入力します。

```
RMAN> STARTUP MOUNT PFILE=init.ora;
```

2. スタンバイ・データ・ファイル名がプライマリ・データ・ファイル名から変換されるように、スタンバイ初期化パラメータ・ファイルに DB\_FILE\_NAME\_CONVERT を設定しておきます。たとえば、次のように入力します。

```
DB_FILE_NAME_CONVERT = ('/oracle/dbs', '/dsk2/oracle')
```

3. すべてのデータ・ファイルおよびスタンバイ制御ファイルをコピーします。たとえば、次のように入力します。

```
COPY
DATAFILE 1 TO '/dsk2/oracle/df_1.f',
DATAFILE 2 TO '/dsk2/oracle/df_2.f',
DATAFILE 3 TO '/dsk2/oracle/df_3.f',
DATAFILE 4 to '/dsk2/oracle/df_4.f',
DATAFILE 5 TO '/dsk2/oracle/df_5.f',
DATAFILE 6 TO '/dsk2/oracle/df_6.f',
DATAFILE 7 TO '/dsk2/oracle/df_7.f',
DATAFILE 8 to '/dsk2/oracle/df_8.f',
DATAFILE 9 TO '/dsk2/oracle/df_9.f',
DATAFILE 10 TO '/dsk2/oracle/df_10.f',
DATAFILE 11 TO '/dsk2/oracle/df_11.f',
DATAFILE 12 to '/dsk2/oracle/df_12.f',
CURRENT CONTROLFILE FOR STANDBY TO '/dsk2/oracle/cf.f';
```

4. 補助インスタンスを起動し、スタンバイ制御ファイルをマウントします。たとえば、SQL\*Plus を起動して、次のように入力します。

```
SQL> STARTUP NOMOUNT PFILE=/dsk2/oracle/dbs/initSTANDBY1.ora
SQL> ALTER DATABASE MOUNT STANDBY DATABASE;
```

## コピーとデータ・ファイルが異なる名前を使用する場合のスタンバイ・データベースの作成

この手順では、スタンバイ・データ・ファイルとプライマリ・データ・ファイルのイメージ・コピーに異なるファイル名を使用することが前提です。

### リカバリを実行しないスタンバイ・データベースの作成

スタンバイ・データベースを作成し、リカバリを実行しない場合は、DUPLICATE コマンドを実行する必要はありません。デフォルトでは、Recovery Manager はスタンバイ・データベースをマウントした状態にして、このデータベースのリカバリは行いません。

#### コピーとスタンバイ・データ・ファイルのファイル名が異なり、リカバリを実行しない場合のスタンバイ・データベースの作成手順

1. プライマリ・データベースおよびスタンバイ・インスタンスに接続します。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 sys/sys_pwd@prod1 AUXILIARY sys/sys_pwd@sbdb1 CATALOG
rman/cat@catdb
```

2. プライマリ・データベースをマウントします。ただしオープンはしません。マウントする前にデータベースが正常にクローズされていることを確認します。たとえば、次のように入力します。

```
STARTUP MOUNT PFILE=initPROD1.ora
```

3. スタンバイ・データ・ファイル名がプライマリ・データ・ファイル名から変換されるように、スタンバイの初期化パラメータ・ファイルに DB\_FILE\_NAME\_CONVERT を設定するか、SET NEWNAME コマンドを発行します。たとえば、DB\_FILE\_NAME\_CONVERT パラメータを次のように設定します。

```
DB_FILE_NAME_CONVERT = ('/oracle/dbs', '/dsk2/oracle')
```

4. COPY コマンドを使用して、すべてのデータ・ファイルおよびスタンバイ制御ファイルをコピーします。たとえば、次のコマンドを実行します。

```
COPY
  DATAFILE 1 TO '/dsk2/oracle/df_1.f',
  DATAFILE 2 TO '/dsk2/oracle/df_2.f',
  DATAFILE 3 TO '/dsk2/oracle/df_3.f',
  DATAFILE 4 TO '/dsk2/oracle/df_4.f',
  DATAFILE 5 TO '/dsk2/oracle/df_5.f',
  DATAFILE 6 TO '/dsk2/oracle/df_6.f',
  DATAFILE 7 TO '/dsk2/oracle/df_7.f',
  DATAFILE 8 TO '/dsk2/oracle/df_8.f',
  DATAFILE 9 TO '/dsk2/oracle/df_9.f',
  DATAFILE 10 TO '/dsk2/oracle/df_10.f',
```

```

DATAFILE 11 TO '/dsk2/oracle/df_11.f',
DATAFILE 12 TO '/dsk2/oracle/df_12.f',
CURRENT CONTROLFILE FOR STANDBY TO '/dsk2/oracle/cf.f';
# To ensure that the control file checkpoint is archived, archive the current
# redo log
SQL 'ALTER SYSTEM ARCHIVE LOG CURRENT';

```

5. 補助インスタンスを起動し、スタンバイ制御ファイルをマウントします。たとえば、SQL\*Plus を起動して、次のように入力します。

```

SQL> STARTUP NOMOUNT PFILE=/dsk2/oracle/dbs/initSTANDBY1.ora
SQL> ALTER DATABASE MOUNT STANDBY DATABASE;

```

## スタンバイ・データベースの作成とリカバリの実行

スタンバイ・データベースを作成し、リカバリを実行する場合は、DUPLICATE コマンドに DORECOVER オプションを指定します。

**コピーとスタンバイ・データ・ファイルのファイル名が異なり、リカバリを実行する場合のスタンバイ・データベースの作成手順**

1. プライマリ・データベースおよびスタンバイ・インスタンスに接続します。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 sys/sys_pwd@prod1 AUXILIARY sys/sys_pwd@sbdb1 CATALOG
rman/cat@catdb

```

2. プライマリ・データベースをマウントします。ただしオープンはしません。マウントする前にデータベースが正常にクローズされていることを確認します。たとえば、次のように入力します。

```

STARTUP MOUNT PFILE=initPROD1.ora

```

3. スタンバイ・データ・ファイル名がプライマリ・データ・ファイル名から変換されるように、スタンバイの初期化パラメータ・ファイルに DB\_FILE\_NAME\_CONVERT を設定するか、SET NEWNAME コマンドを発行します。たとえば、DB\_FILE\_NAME\_CONVERT パラメータを次のように設定します。

```

DB_FILE_NAME_CONVERT = ('/oracle/dbs', '/dsk2/oracle')

```

### 4. DUPLICATE コマンドを実行します。次の手順に従います。

- a. リカバリの終了時刻が、スタンバイ制御ファイルのチェックポイント SCN より大きいと等しいことと、チェックポイント SCN を含むログがリカバリで使用できることを確認します (13-10 ページの「[リカバリありの Recovery Manager スタンバイの作成](#)」を参照)。
- b. 必要な場合は SET コマンドを発行して、リカバリの終了時刻、SCN またはログ順序番号を指定します。
- c. 自動チャンネルが構成されていない場合は、複製用の補助チャンネルを少なくとも 1 つ手動で割り当てます。
- d. すべてのデータ・ファイルおよびスタンバイ制御ファイルをコピーします。
- e. 現行の REDO ログをアーカイブします。
- f. DUPLICATE コマンドを、DORECOVER オプションで発行します。

たとえば、次のように入力します。

COPY

```
DATAFILE 1 TO '/dsk2/oracle/df_1.f',
DATAFILE 2 TO '/dsk2/oracle/df_2.f',
DATAFILE 3 TO '/dsk2/oracle/df_3.f',
DATAFILE 4 TO '/dsk2/oracle/df_4.f',
DATAFILE 5 TO '/dsk2/oracle/df_5.f',
DATAFILE 6 TO '/dsk2/oracle/df_6.f',
DATAFILE 7 TO '/dsk2/oracle/df_7.f',
DATAFILE 8 TO '/dsk2/oracle/df_8.f',
DATAFILE 9 TO '/dsk2/oracle/df_9.f',
DATAFILE 10 TO '/dsk2/oracle/df_10.f',
DATAFILE 11 TO '/dsk2/oracle/df_11.f',
DATAFILE 12 TO '/dsk2/oracle/df_12.f',
CURRENT CONTROLFILE FOR STANDBY TO '/dsk2/oracle/cf.f';
SQL 'ALTER SYSTEM ARCHIVE LOG CURRENT';
DUPLICATE TARGET DATABASE FOR STANDBY
DORECOVER;
```

Recovery Manager は、すべての増分バックアップ、アーカイブ REDO ログ・バックアップおよびアーカイブ REDO ログを使用して、不完全リカバリを実行します。スタンバイ・データベースはマウントされた状態です。



## バックアップとイメージ・コピーを使用したスタンバイ・データベースの作成 : 例

この例では、プライマリ・データ・ファイルのバックアップとイメージ・コピーの両方を使用する複製を実行します。Recovery Manager がデータ・ファイルのバックアップとコピーを使用してスタンバイ・ファイルを作成する方法と、増分バックアップとアーカイブ・ログの両方を使用してスタンバイ・データベースをリカバリする方法を示します。

スタンバイ・データベース環境の前提は次のとおりです。

- プライマリ・データベースは host1 にあり、スタンバイ・データベースは host2 にあります。
- データベース prod1 には 30 ファイルのデータ・ファイルが含まれています。データ・ファイル 1 からデータ・ファイル 25 は、/dev/rdsk### (### は 001 から 025) という名前の RAW ディスク上にあり、データ・ファイル 26 からデータ・ファイル 30 は /primary/datafile ディレクトリにあります。

1 週間の間に、次のアクションを実行します。

1. 月曜日に、次に示すレベル 0 の増分データベース・バックアップを実行します。

```
BACKUP DEVICE TYPE sbt INCREMENTAL LEVEL 0 DATABASE PLUS ARCHIVELOG;
```

2. 火曜日には、データ・ファイル 1 から 5 までを host1 上の /standby/datafile ディレクトリにコピーしてから、BACKUP ARCHIVELOG ALL を実行します。
3. 水曜日には、データ・ファイル 6 から 9 までを host1 上の /standby/datafile ディレクトリにコピーしてから、BACKUP ARCHIVELOG ALL を実行します。
4. 木曜日には、次に示すレベル 1 の増分データベース・バックアップを実行します。

```
BACKUP DEVICE TYPE sbt INCREMENTAL LEVEL 1 DATABASE PLUS ARCHIVELOG;
```

5. 金曜日には、データ・ファイル 10 から 15 までを host1 上の /standby/datafile ディレクトリにコピーしてから、BACKUP ARCHIVELOG ALL を実行します。
6. 土曜日の朝、次の Recovery Manager コマンドを実行します。

```
COPY CURRENT CONTROLFILE FOR STANDBY TO '/standby/datafile/cf.f';  
SQL 'ALTER SYSTEM ARCHIVELOG CURRENT';  
BACKUP DEVICE TYPE sbt ARCHIVELOG ALL;
```

7. 土曜日の夜は、host1 上の /standby/datafile にある全イメージ・コピーを host2 上の /standby/datafile に ftp し、host1 上のすべてのログも host2 に ftp します。さらに、prod1 のデータ・バックアップを host2 でアクセスできるようにします。

日曜日に、スタンバイ・データベースを作成し、土曜日のバックアップ時点までリカバリすることに決めました。スタンバイ・データ・ファイルはすべて host2 の /standby/datafile ディレクトリに格納することになります。

スタンバイ・データ・ファイルに名前を付ける方法を選択する必要があります。DB\_FILE\_NAME\_CONVERT を使用して、RAW ディスクのデータ・ファイル名をそれぞれ変更する方法がありますが、これにはパラメータ内に 25 の値のペアが必要です（変更対象の各 RAW ディスク・ファイル名に 1 つのペア）。かわりに、RAW ディスク上の 25 ファイルのデータ・ファイルに対しては SET NEWNAME コマンドを使用し、DB\_FILE\_NAME\_CONVERT は /primary/datafile 内の 5 個のデータ・ファイルの名前を /standby/datafile に変換するためにのみ使用することに決めました。

イメージ・コピーは host2 の /standby/datafile にありますが、コピーを作成したのはデータ・ファイル 1 から 15 のみです。ただし、全データ・ファイルの増分バックアップがあるので、これは問題にはなりません。Recovery Manager では常にバックアップではなくイメージ・コピーからのリストアを優先しますが、イメージ・コピーが使用できない場合は、バックアップからリストアします。したがって、次のスクリプトを実行します。

```
RUN
{
  # run SET NEWNAME commands for datafiles 1-25
  SET NEWNAME FOR DATAFILE 1 TO '/standby/datafile/df1.f';
  SET NEWNAME FOR DATAFILE 2 TO '/standby/datafile/df2.f';
  .
  .
  .
  SET NEWNAME FOR DATAFILE 25 TO '/standby/datafile/df25.f';
  DUPLICATE TARGET DATABASE FOR STANDBY DORECOVER;
}
```

Recovery Manager は、複製中に次のアクションを実行します。

- データ・ファイル 1 から 15 のイメージ・コピーを使用します。
- データ・ファイル 16 から 30 のバックアップをリストアします（これらのデータ・ファイルにはイメージ・コピーがないため）。
- 増分バックアップを使用して、データ・ファイルの 1 から 9 と 16 から 30 をリカバリします。データ・ファイル 10 から 15 はバックアップからはリカバリしません。これらのデータ・ファイルのコピーは、木曜日のレベル 1 の増分バックアップの後、金曜日に作成されたものであるためです。
- 必要に応じて、データ・ファイル 1 から 30 に対して、バックアップされた最新のアーカイブ・ログまでをリストアし適用します。
- ディスク上のアーカイブ・ログを最新のアーカイブ・ログまで適用します。

---

## Recovery Manager のチューニング

Recovery Manager のチューニングの最大の目標は、ディスクと記憶デバイス間に十分なデータの流れを作成することです。Recovery Manager のバックアップおよびリストア操作のチューニングには、この章で説明する次のようなタスクが含まれます。

- [チューニング概要](#)
- [チューニングの概念](#)
- [バックアップ・パフォーマンスの向上](#)

## チューニング概要

Recovery Manager のバックアップおよびリストア操作は、次の要素で構成されます。

- 入力データの読み込みまたは書き込み
- ブロックの検証と入力バッファから出力バッファへのブロックのコピーによるデータ処理

これらの操作で最も速度の遅いものが、**ボトルネック**と呼ばれます。Recovery Manager のチューニングとは、ボトルネックを識別し、Recovery Manager コマンド、初期化パラメータ設定または物理媒体の調整によりボトルネックを改善する作業をいいます。Recovery Manager のチューニングで重要になるのは、I/O の理解です。

Recovery Manager のバックアップおよびリストアのジョブでは、2 種類の I/O バッファを使用します。DISK および 3 次記憶装置（通常はテープ）です。バックアップを実行するとき、Recovery Manager はディスク・バッファを使用して入力ファイルを読み取り、ディスク・バッファまたはテープ・バッファのいずれかを使用して出力バックアップ・ファイルを書き込みます。リストアを実行するときは、この操作が逆になります。

I/O は DISK と sbt に分類されるのみでなく、**同期**と**非同期**にも分類されます。同期デバイスは、1 度に 1 つの I/O タスクのみを実行します。したがって、バックアップ・ジョブにどのくらいの時間が必要かを容易に判断できます。同期 I/O とは対照的に、非同期 I/O では 1 度に複数のタスクを実行します。

Recovery Manager を効果的にチューニングするには、同期 I/O と非同期 I/O、ディスク・バッファとテープ・バッファおよびチャネル・アーキテクチャなどの概念を十分理解する必要があります。このような概念を理解して初めて、固定ビューを使用したボトルネックの監視方法を習得したり、14-11 ページの「[バックアップ・パフォーマンスの向上](#)」に説明されている技法を用いて問題を解決することができます。

## チューニングの概念

この項では、次の項目について説明します。

- [ディスク・バッファの割当て](#)
- [テープ・バッファの割当て](#)
- [同期 I/O と非同期 I/O](#)
- [テープへのバックアップ速度に影響する要因](#)
- [チャネルのチューニング・オプション](#)
- [バックアップのチューニング・オプション](#)

## ディスク・バッファの割当て

Recovery Manager の I/O では、ディスクとテープという 2 種類のバッファを使用します。これらのバッファは、通常、サイズが異なります。Recovery Manager がディスク・バッファを割り当てる方法を理解するには、Recovery Manager がどのように作業を多重化するか (5-20 ページの「[多重バックアップ・セット](#)」を参照してください) を理解する必要があります。次に進む前に、この項を参照してください。

Recovery Manager の**多重化**とは、同時に読み取られ同一バックアップ・ピースに書き込まれるバックアップのファイル数です。多重化の程度は、BACKUP コマンドの FILESPERSET パラメータと、CONFIGURE CHANNEL または ALLOCATE CHANNEL コマンドの MAXOPENFILES パラメータに依存します。

たとえば、1 つのチャネルを使用して、2 つのデータ・ファイルをバックアップするとします。FILESPERSET を 3 に設定し、MAXOPENFILES を 8 に設定します。この場合、各バックアップ・セット内のファイル数は 2 (FILESPERSET と各チャネルで読み込まれるファイル数の小さい方) であるため、多重化の程度は 2 (MAXOPENFILES と各バックアップ・セット内のファイル数の小さい方) になります。

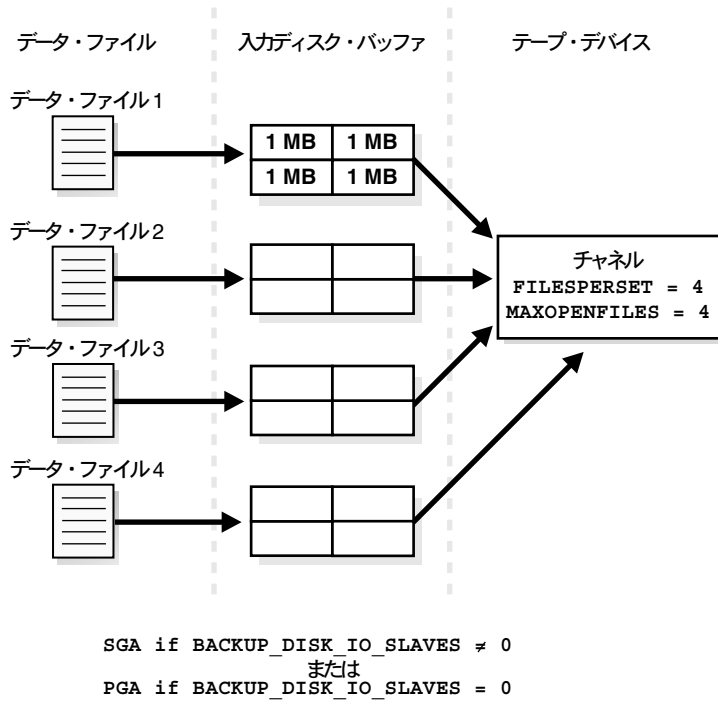
Recovery Manager がディスクからバックアップするとき、Recovery Manager は[表 14-1](#)に説明されているアルゴリズムを使用して、割り当てるバッファ数とバッファ・サイズを決定します。

**表 14-1 ディスク・バッファの割当てのアルゴリズム**

多重化のレベル..	処理..
4 より小さいか等しい	Recovery Manager は、全入力ファイル用の合計バッファ・サイズが 16MB になるように、1MB のバッファを割り当てます。たとえば、FILESPERSET=1 の場合、Recovery Manager はバックアップ・セットに入る 1 つのファイルに対して 16 個のバッファを割り当てます。FILESPERSET=4 の場合、Recovery Manager はバックアップ・セットに入る 4 つの入力ファイルのそれぞれに対して 4 個ずつバッファを割り当てます。
4 より大きく、8 より小さいか等しい	Recovery Manager は、すべてのファイル用合計バッファ・サイズが 16MB 未満になるように、512KB のディスク・バッファを割り当てます。
8 より大きい	Recovery Manager は、各ファイルの合計バッファ・サイズが 512KB になるように、各ファイルに 128KB の固定ディスク・バッファを 4 つずつ割り当てます。

[図 14-1 「ディスク・バッファの割当て」](#)に示されている例では、強固なストライプ・ディスク構成で、1 つのチャネルが 4 つのデータ・ファイルをバックアップします。MAXOPENFILES を 4 に、FILESPERSET を 4 に設定します。このため、多重化レベルは 4 で、各データ・ファイル用のバッファの合計サイズは 4MB です。

図 14-1 ディスク・バッファの割当て



バックアップ・セットに割り当てられるバッファの合計サイズを計算するには、各データ・ファイルの合計バイト数にチャンネルが同時にアクセスするデータ・ファイル数を掛け、その結果の数値とチャンネル数を掛け合えます。

1つのチャンネルで4つのデータ・ファイルをバックアップし、[図 14-1](#)に示されている設定を使用するとします。この場合、バックアップに割り当てるバッファの合計サイズを取得するには、次のように計算します。

$$4 \text{ MB/データ・ファイル} \times 1 \text{ チャンネル} \times 4 \text{ データ・ファイル/チャンネル} = 16 \text{ MB}$$

MAXOPENFILES パラメータを指定して、同時に読み込まれるファイル数が、出力デバイスをフルに活用できる数になるようにします。この配慮は、出力デバイスがテープの場合には特に重要です。

## テープ・バッファの割当て

sbt デバイスにバックアップする場合、Oracle ではテープ書込み（リストアの場合はテープ読み込み）用のチャンネルのそれぞれに対してバッファを 4 つ割り当てます。Oracle では、チャンネルが sbt チャンネルである場合にのみ、これらのバッファを割り当てます。通常、テープ・バッファはそれぞれ 256KB です。バックアップまたはリストア中に使用されるバッファの合計サイズを計算するには、このバッファ・サイズに 4 を掛け、その結果の数値とチャンネル数を掛け合えます。

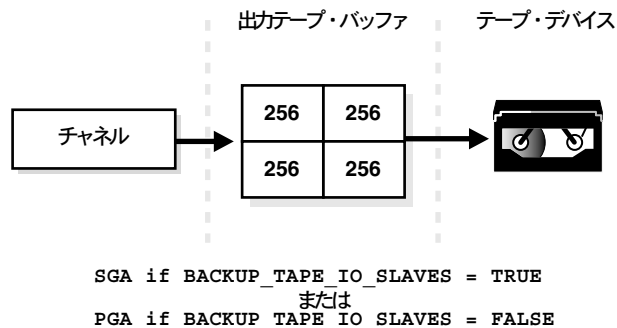
図 14-2 に示されているように、テープ・チャンネルを 1 つ使用し、各バッファ・サイズは 256KB であるとします。この場合、バックアップ中に使用されるバッファの合計サイズは、次のようになります。

256 KB/ バッファ x 4 バッファ / チャンネル x 1 チャンネル = 1024 KB

Recovery Manager では、I/O スレーブが使用されるかどうかにより、テープ・バッファを SGA または PGA に割り当てます。初期化パラメータ `BACKUP_TAPE_IO_SLAVES` を `true` に設定した場合、Recovery Manager は SGA からテープ・バッファを割り当てます。`LARGE_POOL_SIZE` 初期化パラメータが設定されている場合はラージ・プールから割り当てます。このパラメータを `false` に設定した場合、Recovery Manager はバッファを PGA から割り当てます。

I/O スレーブを使用する場合は、`LARGE_POOL_SIZE` 初期化パラメータを設定し、この大きなメモリー割当てを保持する専用の SGA メモリーを確保しておきます。このようにすれば、Recovery Manager の I/O バッファが SGA メモリーのライブラリ・キャッシュと競合しません。

図 14-2 テープ・バッファの割当て



## 同期 I/O と非同期 I/O

Recovery Manager がデータの読み込みまたは書き込みを行うとき、I/O は**同期**か**非同期**のいずれかです。I/O が同期の場合、サーバー・プロセスは 1 度に 1 つのタスクのみ実行できます。I/O が非同期の場合、サーバー・プロセスは I/O を開始し、この I/O の完了を待つ間に他の処理を実行できます。最初の I/O が完了する前に、複数の I/O 操作を開始することもできます。

I/O のタイプを決定する初期化パラメータを設定できます。BACKUP\_TAPE\_IO\_SLAVES を true に設定すると、テープ I/O は非同期になります。それ以外の場合、I/O は同期です。BACKUP\_TAPE\_IO\_SLAVES は常に true に設定しておくことをお勧めします。

オペレーティング・システムの中にはネイティブな非同期 I/O をサポートするものがあり、Oracle ではこの機能が使用可能な場合は使用します。ネイティブな非同期 I/O をサポートしないオペレーティング・システムの場合、Oracle では、別プロセスのかわりに I/O を実行する専用の特殊な I/O スレーブ・プロセスを使用して、この機能をシミュレートします。ディスク I/O スレーブを制御するには、DBWR\_IO\_SLAVES パラメータをゼロ以外の値に設定します。Oracle では、DBWR\_IO\_SLAVES がゼロ以外の値に設定されている場合、バックアップ・ディスク I/O スレーブを 4 つ割り当てます。

図 14-3 は、テープへのバックアップでの同期 I/O を示します。次の手順が発生します。

1. サーバー・プロセスがテープ・バッファにブロックを書き込みます。
2. テープ・プロセスがテープにデータを書き込みます。メディア・マネージャが Oracle バッファからメディア・マネージャの内部バッファにデータをコピーする間、サーバー・プロセスはアイドル状態になっている必要があります。
3. テープ・プロセスが、書き込みを完了したことを通知するメッセージをサーバー・プロセスに返します。
4. サーバー・プロセスが新規タスクを処理できます。



図 14-3 同期 I/O

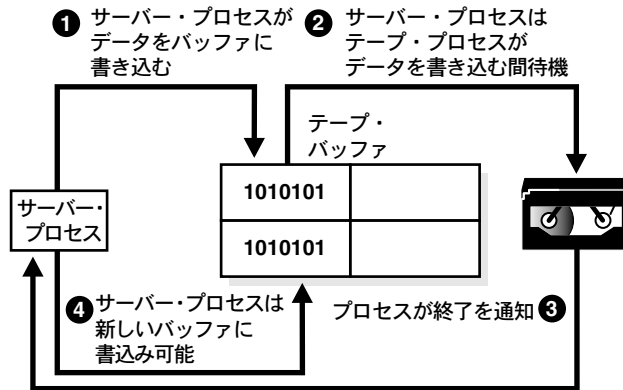
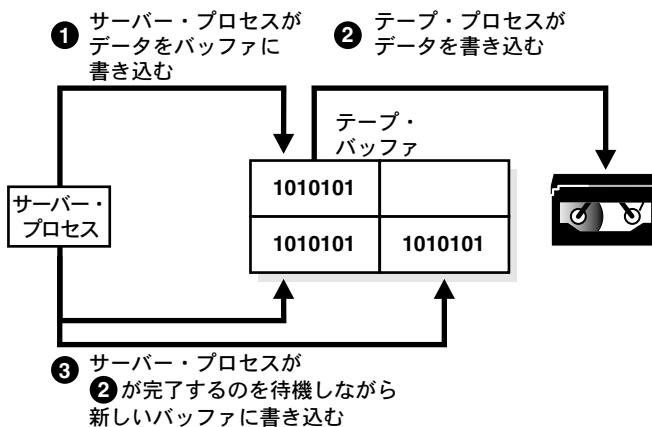


図 14-4 は、テープ・バックアップでの**非同期 I/O**を示します。次の手順が発生します。

1. サーバー・プロセスがテープ・バッファにブロックを書き込みます。
2. テープ・プロセスがテープにデータを書き込みます。テープ・プロセスが書き込みを実行している間、他のサーバー・プロセスは入力ブロックを処理し、出力バッファにデータを入れることができます。
3. 最初のテープ・プロセスがテープへの書き込みを実行中に、起動された 2 つのサーバー・プロセスがテープ・バッファに書き込みを行います。

図 14-4 非同期 I/O



## テープへのバックアップ速度に影響する要因

テープへのバックアップ・スピードには、次の要因が影響します。

- システム固有の転送レート
- テープの圧縮
- テープ・ストリーム
- 物理テープ・ブロック・サイズ

### システム固有の転送レート

テープ固有の転送レートは、圧縮なしでテープに書き込む場合のスピードです。このスピードは、バックアップ・レートの上限を表します。バックアップ・パフォーマンスの上限は、すべてのテープ・ドライブの転送レートの集計となります。バックアップがすでにそのレートで実行されており、必要以上に CPU を使用していない場合は、チューニングしても Recovery Manager のパフォーマンスは変わりません。

### テープの圧縮

テープの圧縮レベルは、バックアップ・パフォーマンスに重大な影響を与えます。テープの圧縮レベルが高い場合、管理バックアップ・レートが高くなります。たとえば、圧縮比が 2:1 で、テープ・ドライブ固有の転送レートが 6MB/s の場合、結果的なバックアップ・スピードは 12MB/s となります。

### テープ・ストリーム

バックアップ・パフォーマンスに関して最も重要な問題の 1 つは、テープ・ストリームです。現在市販されているテープ・ドライブは、ほぼすべてが固定スピードのストリーム・テープ・ドライブです。つまり、これらのドライブではデータ書き込み速度を変更できません。そのため、テープに書き込むデータがなくなると、低速になってから停止します。たとえば、デバイスのバッファが空になっても、テープ移動が高速すぎて実際には通り越してしまい、書き込みを停止した時点まで巻き戻す必要があります。

### 物理テープ・ブロック・サイズ

物理テープ・ブロック・サイズは、バックアップ・パフォーマンスに影響する可能性があります。ブロック・サイズは、メディア管理ソフトウェアが 1 回の書き込み操作でテープに書き込むデータ量です。原則として、テープ・ブロック・サイズが大きいほどバックアップが高速になります。物理テープ・ブロック・サイズを制御するのは、Recovery Manager でも Oracle サーバーでもなく、メディア管理ソフトウェアであることに注意してください。物理テープ・ブロック・サイズが大きいほど、バックアップは高速になります。物理テープ・ブロック・サイズは、メディア管理ソフトウェアによって制御されます。

## チャネルのチューニング・オプション

CONFIGURE CHANNEL コマンドと ALLOCATE CHANNEL コマンドに、割当て済みのサーバー・セッションにより実行される操作に適用される様々なチャネル制限パラメータを設定することができます。

これらのパラメータを使用して、次のことができます。

- バックアップ・ピースのサイズを制限します。
- Recovery Manager がコンシュームするディスク帯域幅を制限します。
- 各チャネルの多重化レベルを判断します。

表 14-2 に説明されているチャネル・パラメータを指定できます。

表 14-2 Recovery Manager のチャネル・パラメータのチューニング

パラメータ	説明
MAXPIECESIZE	バックアップ・ピースの最大サイズを指定します。このパラメータは、バックアップ・セット内に複数のバックアップ・ピースを作成するように Recovery Manager に強制するために使用します。Recovery Manager は、このパラメータに指定されている値より小さいサイズのバックアップ・ピースを作成します。
RATE	Recovery Manager がこのチャネルで 1 秒間に読み込むバイト数を指定します。このパラメータは、読み込みバイト数の上限を設定し、Recovery Manager がディスク帯域幅をコンシュームしすぎてオンライン・パフォーマンスを低下させないようにするために使用します。  たとえば、RATE=1500K と設定します。各ディスク・ドライブのデータ転送速度が 3MB/ 秒の場合、Recovery Manager はディスク帯域幅の一部をオンライン・システム用に残します。
MAXOPENFILES	バックアップまたはコピー操作で特定の時点にオープンできる入力ファイルの最大数を決定します（デフォルト値は 8）。14-3 ページの「ディスク・バッファの割当て」で説明したように、MAXOPENFILES は Recovery Manager の多重化レベルを決定する 1 つの要因です。表 14-1 に示されているように、多重化レベルにより Recovery Manager がディスク・バッファを割り当てる方法が決まります。多重化とは、同時に読み込まれて同一バックアップ・ピースに書き込まれる入力ファイル数です。

関連項目：

- ALLOCATE CHANNEL コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- CONFIGURE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- SET コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

バックアップのチューニング・オプション

BACKUP コマンドを使用すると、Recovery Manager がバックアップ・セットへ入力するファイルを選択する方法に影響するパラメータを設定できます。これらのパラメータを設定すると、次のことができます。

- Recovery Manager で 1 つのバックアップ・セットが複数のボリュームに書き込まれないようにします。
- Recovery Manager で 1 度に多数のディスクからの読み込みが行われないようにします。
- 同時に読み込まれるディスク・ドライブの数を制限します。

表 14-3 に説明されているパラメータを指定できます。

表 14-3 Recovery Manager のバックアップ・パラメータのチューニング

パラメータ	説明
MAXSETSIZE	バックアップ・セットの最大サイズ（バイト単位）を指定します。このパラメータは、1 つのバックアップ・セットが複数のボリュームに拡散することを防止するときに使用します。
FILESERSET	1 つのバックアップ・セットに含めるファイルの最大数を指定します。 FILESERSET のデフォルト値は、64 とファイル / チャネル数の小さい方の値です。チャネルを 1 つのみ割り当てる場合は、このパラメータを使用して、Recovery Manager で複数のバックアップ・セットを作成できます。  たとえば、入力データ・ファイルが 50 個あり、チャネルが 2 つの場合は、FILESERSET=5 と設定して、バックアップ・セットを 10 個作成します。この方法で、バックアップ・セットが複数のテープに分割されることを防止できます。
DISKRATIO	バックアップに含めるドライブ数を指定します。  データ・ファイルが 5 台のディスクに配置されており、各ディスクは 1 秒間に 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ムの維持のため 1 秒間に 20 バイトのデータを要求している場合を想定します。DISKRATIO=2 と設定した場合、Recovery Manager は 1 度に 2 つのドライブから読み込み、バックアップ負荷が分散されます。

**注意：** BACKUP コマンドまたは CONFIGURE コマンドに FILESPERSET を設定して、チャンネルがアクセスするデータ・ファイルの数を制御します。

**関連項目：** BACKUP および CONFIGURE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## バックアップ・パフォーマンスの向上

バックアップのパフォーマンスに影響する要因は多数あります。低速のバックアップに対する解決策を発見しようとした場合、試行錯誤のプロセスになることがよくあります。バックアップを最高のパフォーマンスで行うには、ここでお薦めしている手順を実行してください。

- 手順 1: RATE パラメータを調整する
- 手順 2: テープにバックアップする場合は、BACKUP\_TAPE\_IO\_SLAVES を設定する
- 手順 3: 同期ディスクを使用する場合は、DBWR\_IO\_SLAVES を設定する
- 手順 4: 共有メモリの割当てに失敗する場合は、LARGE\_POOL\_SIZE を設定する
- 手順 5: 多重化レベルを調整する
- 手順 6: ファイルが空であるか変更が少ないかを判断する
- 手順 7: BLKSIZE パラメータを使用してテープ・バッファ・サイズを変更する
- 手順 8: V\$ ビューを問い合わせでボトルネックを識別する

### 手順 1: RATE パラメータを調整する

表 14-2 に説明されているように、RATE パラメータが ALLOCATE CHANNEL コマンドまたは CONFIGURE CHANNEL コマンドに設定されていないことを確認します。RATE パラメータは、バックアップをバックグラウンドで実行して OLTP 操作に対する影響を最低限に抑えられるように、バックアップの速度を抑えるためのパラメータです。

RATE パラメータは、1 秒当たりのバイト数を単位として指定します。問合せのパフォーマンスが向上し、かつ Recovery Manager がバックアップを妥当な時間内に完了できるような値を、テストして見つけます。RATE はバックアップのスループットを上げるのではなく、他のデータベース操作でより多くのディスク帯域幅を使用できるように、バックアップのスループットを下げるためのパラメータなので注意してください。

## 手順 2: テープにバックアップする場合は、BACKUP\_TAPE\_IO\_SLAVES を設定する

sbt デバイスにバックアップする場合（この場合のみ）は、BACKUP\_TAPE\_IO\_SLAVES を true に設定し、テープ・バッファが SGA から割り当てられるようにします。バッファ・サイズを制御するには、ALLOCATE CHANNEL コマンドまたは CONFIGURE CHANNEL コマンドに PARMS パラメータを使用します。

BACKUP\_TAPE\_IO\_SLAVES 初期化パラメータは、テープ I/O の完了を待機する追加プロセスを起動し、テープ I/O の完了待機中にプライマリ・プロセスが追加のディスク・ブロックを処理できるようにすることで非同期テープ I/O をシミュレートします。このパラメータを設定しない場合、テープ・レイヤーへの I/O が同期になります。これは、テープへの書き込みが終了するまで他の作業は実行できないことを意味します。

BACKUP\_TAPE\_IO\_SLAVES パラメータは、それぞれのディスク I/O またはテープ I/O のバッファが 2 つのプロセス間で共有できるように、共有メモリー (SGA) からの割当てを要求します。したがって、このメモリー使用に合せて十分大きい SGA を割り当てます。BACKUP\_TAPE\_IO\_SLAVES パラメータを設定する場合は、LARGE\_POOL\_SIZE パラメータも設定します。

## 手順 3: 同期ディスクを使用する場合は、DBWR\_IO\_SLAVES を設定する

ディスクが非同期 I/O をサポートしない場合（この場合のみ）は、DBWR\_IO\_SLAVES 初期化パラメータをゼロ以外の値に設定します。DBWR\_IO\_SLAVES にゼロ以外の値を設定することにより、バックアップおよびリストアに決まった数（4 つ）のディスク I/O スレーブが使用され、これにより非同期 I/O がシミュレートされます。I/O スレーブが使用される場合、I/O バッファは SGA（または、ラージ・プールが構成されている場合はラージ・プール）から取得されます。

## 手順 4: 共有メモリーの割当てに失敗する場合は、LARGE\_POOL\_SIZE を設定する

この初期化パラメータは、十分なメモリーがないので I/O スレーブを起動しないというエラーが alert.log にレポートされた場合にのみ設定します。次に、メッセージの例を示します。

```
ksfqxcrc: failure to allocate shared memory means sync I/O will be used whenever async I/O to file not supported natively
```

I/O スレーブ用の共有バッファを取得するときに、Oracle は次のことを行います。

- LARGE\_POOL\_SIZE が設定されている場合、Oracle はラージ・プールからメモリーを取得しようとします。この値が十分でない場合でも、共有プールからはバッファを取得しようとしません。
- LARGE\_POOL\_SIZE が設定されていない場合、Oracle は共有プールからメモリーを取得しようとします。

- 十分なメモリーが取得できない場合は、ローカル・プロセス・メモリーから I/O バッファ・メモリーを取得し、このバックアップに対して同期 I/O が使用されることを示すメッセージを `alert.log` に書き込みます。

ラージ・プールからのメモリーは、共有サーバー（以前はマルチスレッド・サーバーと呼ばれていたもの）、パラレル問合せおよび **Recovery Manager** の I/O スレーブ・バッファなどの多くの機能に使用されます。ラージ・プールを構成すると、同一メモリーに関して **Recovery Manager** と他のサブシステムとの競合が防止されます。

共有プールから連続メモリーを割り当てる要求は、通常小さいサイズ（5KB 未満）です。ただし、大きな連続メモリーを割り当てる要求では、失敗するか、必要量の連続メモリーを解放するためのかなりのクリーン・アップ作業が必要になる可能性があります。共有プールではこのメモリー要求を満たすことはできませんが、ラージ・プールでは可能です。ラージ・プールには最低使用頻度（LRU）リストがありません。**Oracle** ではラージ・プールのメモリーはエージ・アウトしません。

ラージ・プールを構成するには、`LARGE_POOL_SIZE` 初期化パラメータを使用します。オブジェクトのメモリーがどのプール（共有プールまたはラージ・プール）にあるかを判断するには、`V$SGASTAT.POOL` に問い合わせます。

**Oracle9i** で `LARGE_POOL_SIZE` を設定する場合の計算式は、次のとおりです。

```
LARGE_POOL_SIZE = number_of_allocated_channels * (16 MB + ( 4 * size_of_tape_buffer ) )
```

ディスクへのバックアップの場合、テープ・バッファ・サイズは明らかに 0 のため、`LARGE_POOL_SIZE` を 16MB に設定します。テープへのバックアップの場合、1 つのテープ・バッファのサイズは **Recovery Manager** のチャネル・パラメータ `BLKSIZE` で定義します。デフォルトは 256KB です。2 つのテープ・ドライブにバックアップする場合を考えます。テープ・バッファのサイズが 256KB の場合は、`LARGE_POOL_SIZE` を 18MB に設定します。`BLKSIZE` を 512KB に増やす場合は、`LARGE_POOL_SIZE` を 20MB に増やします。

**関連項目：** ラージ・プールの詳細は、『**Oracle9i データベース概要**』を参照してください。初期化パラメータの詳細は、『**Oracle9i データベース・リファレンス**』を参照してください。

## 手順 5: 多重化レベルを調整する

5-20 ページの「[多重バックアップ・セット](#)」に説明されているように、多重化のレベルは次の要因により決まります。

- 各バックアップ・セットに含まれるファイル数。これは、各チャネルにより読み込まれるファイル数と `FILESERSET` の設定値の小さい方です。
- `MAXOPENFILES` の設定値と各バックアップ・セットに含まれるファイル数の小さい方の値。

サーバー上のディスク構成を反映させるように、多重化レベルを調整する必要があります。ストライプ・ディスク構成ではハードウェアの多重化が含まれるため、**Recovery Manager** での多重化レベルはそれほど高くする必要はありません。たとえば、次のディスク構成例で考えてみます。

- 多数のディスクにデータ・ファイルをストライプ化する場合、ハードウェアがすでに多重化を実行していることにより、**Recovery Manager** で複数のディスクに分散されているデータを効率的に読み込めるため、**Recovery Manager** の多重化レベルは 1 に設定します。
- 少数（2、3 台）のディスクにデータ・ファイルをストライプ化する場合は、**Recovery Manager** で多重化作業を実行する必要があるため、**Recovery Manager** の多重化レベルは 4 ～ 8 の間に設定します。
- ディスクをストライプ化しない場合は、**Recovery Manager** が多重化自体を実行する必要があるため、**Recovery Manager** の多重化レベルは少なくとも 8 に設定します。

たとえば、各チャネルが 15 個のファイルを読み込み、`FILESERSET=10` および `MAXOPENFILES=8` に設定してある場合、多重化レベルは次のように計算します。

```
min( min( 15, 10 ), 8 ) = 8
```

データ・ファイルが 2 台のディスクにストライプ化される場合は、この多重化レベルは高すぎます。この場合は、`MAXOPENFILES` をたとえば 6 などの低い値に設定する必要があります。



## 手順 6: ファイルが空であるか変更が少ないかを判断する

ほとんど空のファイルを全体バックアップするとき、または変更ブロックが少ない場合に増分バックアップを行うときは、テープのストリーム化に必要な速度でデータを提供できないことがあります。どちらの場合も、多重化レベルを高くするとパフォーマンスを改善できます。

---

**注意：** テープ・バックアップのパフォーマンスが空のファイルのために低速になっているかどうかをテストする方法としては、アーカイブ・ログのバックアップを試してみる方法があります。アーカイブ・ログにはデータのみが含まれています（つまり、空の領域は含まれていません）。

---

増分バックアップは Recovery Manager によるバックアップであり、修正されたブロックのみがバックアップされます。Oracle は増分バックアップを取るためにデータ・ファイル全体を読み込むため、増分バックアップが完全バックアップより高速であるとは限りません。テープ・ドライブがローカルに接続されていない場合は、増分バックアップは高速になる可能性があります。テープ書込み用帯域幅とディスク読込み用帯域幅がどの程度あるかを考慮する必要があります。ディスク帯域幅に比べてテープ帯域幅の方が制限されている場合は、増分バックアップが役立ちます。

増分バックアップで変更ブロックが少ない場合、バッファがいっぱいになってテープに書き込まれるまでに、データ・ファイルから多数のバッファを入力してブロックを多数蓄積する必要があります。このため、テープ・ドライブはストリーム化されない可能性があります。

多重化レベル（14-14 ページの「[手順 5: 多重化レベルを調整する](#)」を参照）を大きい値に設定すると、多数のデータ・ファイルを並列してスキャンでき、テープ・ドライブ用の出力バッファはすぐいっぱいになります。ドライブがストリーム化されるように頻繁に書き込みます。FILESERSET の値は MAXOPENFILES の値より小さいか等しくする必要があります。たとえば、この 2 つのパラメータを 8 に設定し、テープ・ドライブがストリーム化されない場合にこの値を上げます。増分バックアップの場合、多重化レベルの値としては 50 が適しています。完全バックアップまたはレベル 0 の増分バックアップの場合は、多重化レベルは 4 や 8 などの低い値にします。

## 手順 7: BLKSIZE パラメータを使用してテープ・バッファ・サイズを変更する

テープがストリーム化されていないときに、問題の原因が増分バックアップや空のファイルのバックアップではない場合は、テープ・バッファのブロック・サイズを調整してみます。各テープ・バッファのサイズを変更するには、ALLOCATE CHANNEL または CONFIGURE CHANNEL コマンドの PARMS パラメータを使用します。使用プラットフォームで PARMS に BLKSIZE がサポートされている場合は、各バッファの希望サイズを設定できます。たとえば、sbt チャンネルを次のように構成します。

```
CONFIGURE CHANNEL DEVICE TYPE sbt PARMS="BLKSIZE=524288";
```

通常は、メディア・マネージャのテープ・ブロック・サイズより少し低い値に BLKSIZE を設定します。「少し低い」程度は、メディア・マネージャに依存します。たとえば、テープ・ブロック・サイズが 512KB で、メディア・マネージャのヘッダー・サイズが 16KB の場合は、BLKSIZE=49600 に設定できます。

メディア管理の物理テープ・ブロック・サイズを大きくすることも役立つため注意してください。たとえば、BLKSIZE パラメータを 512KB に設定し、物理テープ・ブロック・サイズを 32KB のままにする必要はありません。

## 手順 8: V\$ ビューを問い合せてボトルネックを識別する

前述のどの手順でもバックアップ・パフォーマンスが改善されない場合は、ボトルネックの正確な原因を判断するようにします。バックアップおよびリストアのボトルネックの原因を判断し、バックアップ・ジョブの進行状況を詳しく調べるには、V\$BACKUP\_SYNC\_IO および V\$BACKUP\_ASYNC\_IO を使用します。

V\$BACKUP\_SYNC\_IO には、バックアップを実行するプロセス（プラットフォームによってはスレッド）に対して I/O が同期しているときに行が表示されます。V\$BACKUP\_ASYNC\_IO には、I/O が非同期の場合に行が表示されます。非同期 I/O が得られるのは、I/O プロセスに伴うか、基本オペレーティング・システムが非同期 I/O をサポートしている場合です。

この項では、次の項目について説明します。

- 同期 I/O でのボトルネックの識別
- 非同期 I/O でのボトルネックの識別

**関連項目：** これらのビューについて、詳細は、『Oracle9i データベース・リファレンス』を参照してください。

同期 I/O の場合にテープがストリーム化されているかどうかを判断するには、V\$BACKUP\_SYNC\_IO ビューまたは V\$BACKUP\_ASYNC\_IO ビューの EFFECTIVE\_BYTES\_PER\_SECOND 列を問い合せます。表 14-4 に、この列の使用方法を示します。

表 14-4 V\$BACKUP\_SYNC\_IO ビュー

EFFECTIVE_BYTES_PER_SECOND 列の値 .. 処理 ..	
ハードウェアのロー・キャパシティより小さい	テープはストリーム化されていません。テープがストリーム化されていないと、通常、パフォーマンスは大きく低下します。
ハードウェアのロー・キャパシティより大きい	テープは、データの圧縮率によってはストリーム化されている可能性があります。

同期 I/O でのボトルネックの識別

同期 I/O の場合は、同期 I/O すべてがプロセスのボトルネックになるため、特定のボトルネックを識別することは困難です。同期 I/O をチューニングするには、I/O レート（バイト / 秒）とデバイスの最大スループット・レートを比較する方法しかありません。この値が、デバイスで指定されている値より低い場合は、バックアップおよびリストア・プロセスのこの側面をチューニングすることを考えます。V\$BACKUP\_SYNC\_IO ビューの DISCRETE\_BYTES\_PER\_SECOND 列にこの I/O 値が表示されます。V\$BACKUP\_SYNC\_IO でデータを表示する場合、問題は、非同期 I/O を使用可能にしていないこと、またはディスク I/O スレーブを使用していないことであるため注意してください。

非同期 I/O でのボトルネックの識別

ロング・ウェイトとは、バックアップ・プロセスまたはリストア・プロセスがオペレーティング・システムに対して I/O が完了するまで待機するように指示した回数です。ショート・ウェイトとは、バックアップ・プロセスまたはリストア・プロセスがオペレーティング・システム・コールを実行して、非ブロック化モードで I/O の完了をポーリングした回数です。レディとは、I/O の使用準備が整っていて、I/O の完了をポーリングするためにオペレーティング・システム・コールを実行する必要がなかった回数です。

ボトルネックを識別する場合に最も簡単な方法は、V\$BACKUP\_ASYNC\_IO を問い合せて、LONG\_WAITS を IO\_COUNT で除算した比が最も大きいデータ・ファイルを調べることです。

**注意：** 同期 I/O で、BACKUP\_DISK\_IO\_SLAVES を設定してある場合、I/O は V\$BACKUP\_ASYNC\_IO ビューに表示されます。

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



---

## トラブルシューティング

この章では、Recovery Manager のトラブルシューティングについて説明します。この章の項目は、次のとおりです。

- [Recovery Manager のメッセージ出力の解釈](#)
- [Media Management API のテスト](#)
- [Recovery Manager コマンドの終了](#)
- [V\\$ ビューを介した Recovery Manager の監視](#)
- [トラブルシューティング例](#)

# Recovery Manager のメッセージ出力の解釈

Recovery Manager では、詳細なエラー・メッセージによって、トラブルシューティングを支援します。また、Oracle データベース・サーバーとサード・パーティのメディア・ベンダーは、それぞれ有益なデバッグ出力を生成しています。この項では、次の項目について説明します。

- [メッセージ出力のタイプの識別](#)
- [Recovery Manager のエラー・メッセージ・スタックの認識](#)
- [エラー・コードの識別](#)
- [Recovery Manager のエラー・スタックの解析](#)
- [Recovery Manager のリターン・コードの識別](#)

## メッセージ出力のタイプの識別

失敗または停止した Recovery Manager のジョブのトラブルシューティングに役立つ出力は、次の表で説明するように、いくつかの異なる位置に格納されます。

出力のタイプ	作成元	位置	説明
Recovery Manager メッセージ	Recovery Manager	出力先は次のとおりです。 <ul style="list-style-type: none"><li>■ 標準出力（通常は端末）</li><li>■ コマンドラインの LOG オプションまたは SPOOL LOG コマンドで指定したログ・ファイル</li><li>■ UNIX の &gt; 演算子など、コマンドライン・オプションによって Recovery Manager の出力をリダイレクトすることで作成されたファイル</li></ul>	Recovery Manager のジョブに関連するアクションに加え、Recovery Manager、サーバーおよびメディア・ベンダーにより生成されたエラー・メッセージが含まれます。Recovery Manager のエラー・メッセージには、RMAN- という接頭辞が付加されます。通常のアクションの説明では接頭辞は付加されません。
alert_SID.log	Oracle データ ベース・サー バー	USER_DUMP_DEST 初期化パラメータで名前付けされたディレクトリ。	エラー、初期化パラメータの設定および管理操作の時系列ログが含まれています。上書きされた制御ファイル・レコードの値が記録されています（16-37 ページの「 <a href="#">制御ファイル・レコードの上書きの監視</a> 」を参照してください）。

出力のタイプ	作成元	位置	説明
Oracle トレース・ファイル	Oracle データベース・サーバー	USER_DUMP_DEST 初期化パラメータで指定されたディレクトリ。	Oracle サーバー・プロセスで生成された詳細な出力が含まれています。このファイルは、ORA-00600 または ORA-03113 のエラー・メッセージが発生すると作成されます。このエラー・メッセージは、Recovery Manager がチャネルを割り当てられない場合および Oracle がメディア管理ライブラリのロードに失敗した場合に発生します。
sbtio.log	サード・パーティのメディア管理ソフトウェア	USER_DUMP_DEST 初期化パラメータで指定されたディレクトリ。	メディア管理ソフトウェアが書き込んだ、ベンダー固有の情報が含まれています。このファイルには、Oracle サーバーまたは Recovery Manager のエラーは含まれないので、注意してください。
メディア・マネージャのログ・ファイル	サード・パーティのメディア管理ソフトウェア	メディア・マネージャのログのファイル名は、sbtio.log を除いてメディア・マネージャ・ソフトウェアによって決められます。	メディア管理デバイスの機能に関する情報が含まれています。

## Recovery Manager のエラー・メッセージ・スタックの認識

Recovery Manager が、コマンドを正常に実行したかどうかを判断することは、様々な場面で重要になります。たとえば、Recovery Manager を使用した自動バックアップのスクリプトを作成するときに、バックアップが成功したか、あるいは失敗したかを確認する場合があります。

Recovery Manager でエラーが発生したかどうかを判断する方法の 1 つは、リターン・コードを調べることです（15-9 ページの「[Recovery Manager のリターン・コードの識別](#)」を参照）。また、Recovery Manager の出力から、文字列 RMAN-00569 を検索して判断することもできます。これは、エラー・スタック・バナーのメッセージ番号です。Recovery Manager 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 メッセージがない場合、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: 入力コマンド解析中にエラーが見つかりました。
RMAN-01005: syntax error: found ")": expecting one of:"archivelog, backup, backupset,
controlfilecopy, current, database, datafile, datafilecopy, (, plus, ;, tablespace"
RMAN-01007: 行: 1、列: 18、ファイル: 標準入力
```

## エラー・コードの識別

通常、Recovery Manager のメッセージ・スタックには、次のタイプのエラー・コードがあります。

- 接頭辞が RMAN- のエラー
- 接頭辞が ORA- のエラー
- 追加情報：の後に表示されるエラー

**関連項目：** RMAN および ORA のエラー・コードの説明は、『Oracle9i データベース・エラー・メッセージ』を参照してください。

## Recovery Manager のエラー・メッセージ番号

表 15-1 は、一般的な Recovery Manager のエラー・メッセージのエラー範囲を示します。これらはすべて、『Oracle9i データベース・エラー・メッセージ』で説明されています。

表 15-1 Recovery Manager のエラー・メッセージの範囲

エラーの範囲	原因
0550-0999	コマンドラインのインタプリタ
1000-1999	キーワードのアナライザ
2000-2999	構文のアナライザ
3000-3999	主レイヤー
4000-4999	サービス・レイヤー
5000-5499	RESTORE または RECOVER コマンドのコンパイル
5500-5999	DUPLICATE コマンドのコンパイル
6000-6999	通常のコンプイル
7000-7999	通常の実行
8000-8999	PL/SQL プログラム
9000-9999	低レベルのキーワードのアナライザ
10000-10999	サーバー側の実行
11000-11999	PL/SQL と Recovery Manager 間のインタフェース・エラー
12000-12999	リカバリ・カタログ・パッケージ



## メディア・マネージャのエラー番号

Media Management API でエラーが発生した場合、Recovery Manager は、次の接頭辞が付いたエラー・メッセージ番号を戻します。

Additional information:

表 15-2 のリストは、メディア・マネージャのメッセージ番号と、対応するエラー・メッセージです。エラー・コード内の *O/S* は、オペレーティング・システムの略です。アスタリスク (\*) が付いたエラーは内部エラーで、通常の操作中表示されることはありません。

**表 15-2 メディア・マネージャのエラー・メッセージの範囲**

原因	番号	説明
sbtopen	7000	バックアップ・ファイルが見つかりません（読み込み時のみ戻されます）。
	7001	ファイルは存在します（書き込み時のみ戻されます）。
	7002*	不正なモードが指定されました。
	7003	無効なブロック・サイズが指定されました。
	7004	テープ・デバイスがありません。
	7005	デバイスは見つかりましたがビジーです。後で再試行してください。
	7006	テープ・ボリュームがありません。
	7007	テープ・ボリュームは使用中です。
	7008	I/O エラー。
	7009	メディア・マネージャと接続できません。
	7010	許可されません。
	7011	O/S エラー。例: malloc、fork のエラー
	7012*	sbtopen への引数が無効です。
sbtclose	7020*	ファイル・ハンドルが無効またはファイルがオープンされていません。
	7021*	sbtclose へのフラグが無効です。
	7022	I/O エラー。
	7023	O/S エラー。
	7024*	sbtclose への引数が無効です。
	7025	メディア・マネージャと接続できません。

表 15-2 メディア・マネージャのエラー・メッセージの範囲（続き）

原因	番号	説明
sbtwrite	7040*	ファイル・ハンドルが無効またはファイルがオープンされていません。
	7041	ボリュームの終わりに達しました。
	7042	I/O エラー。
	7043	O/S エラー。
	7044*	sbtwrite への引数が無効です。
sbtread	7060*	ファイル・ハンドルが無効またはファイルがオープンされていません。
	7061	EOF に達しました。
	7062	ボリュームの終わりに達しました。
	7063	I/O エラー。
	7064	O/S エラー。
	7065*	sbtread への引数が無効です。
sbtremove	7080	バックアップ・ファイルが見つかりません。
	7081	バックアップ・ファイルは使用中です。
	7082	I/O エラー。
	7083	メディア・マネージャと接続できません。
	7084	許可されません。
	7085	O/S エラー。
	7086*	sbtremove への引数が無効です。
sbtinfo	7090	バックアップ・ファイルが見つかりません。
	7091	I/O エラー。
	7092	メディア・マネージャと接続できません。
	7093	許可されません。
	7094	O/S エラー。
	7095*	sbtinfo への引数が無効です。
sbtinit	7110*	sbtinit への引数が無効です。
	7111	O/S エラー。

## Recovery Manager のエラー・スタックの解析

Recovery Manager のエラー・スタック内で有効なメッセージを識別するのが困難な場合があります。次のヒントと提案を参考にしてください。

- エラー・スタック内のほとんどのメッセージは、トラブルシューティングとは関係がないため、最も重要な 1 つか 2 つのエラーを識別します。
- Additional information の後に整数が表示される行をチェックしてください。この行は、メディア管理のエラーを示します。後に続く整数は、エラー・メッセージのテキストで説明されているコードを参照しています。
- メッセージは、Recovery Manager が発行した順に並べられているため、下から読んでください。スタックに表示されるエラーの中で、通常、最後の 1 つまたは 2 つのエラーが最も重要です。
- バナーの真下にある RMAN-03002 または RMAN-03009 メッセージを調べてください (RMAN-03009 は RMAN-03002 と同じですが、チャンネル ID が含まれています)。この 2 つのメッセージは、どのコマンドが失敗したかを示します。構文エラーがあると、RMAN-00558 が生成されます。
- 表 15-1 にあるエラー範囲のグラフに従って基本的なエラーのタイプを識別し、最も重要なメッセージについては、詳細を『Oracle9i データベース・エラー・メッセージ』で確認してください。

### Recovery Manager のエラーの解釈 : 例

表領域 users のバックアップを試行したところ、次のメッセージを受け取りました。

```
Starting backup at 29-AUG-01
using channel ORA_DISK_1
RMAN-00571: =====
RMAN-00569:===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: backup コマンドが 08/29/2001 15:14:03 で失敗しました
RMAN-20202: 表領域がリカバリ・カタログに見つかりません
RMAN-06019: 表領域名 "USESR" を変換できませんでした
```

RMAN-03002 エラーは、BACKUP コマンドが失敗したことを示します。メッセージ・スタックの最後の 2 つのメッセージを読むと、表領域名を正しく入力しなかったため、リカバリ・カタログ内に `usesr` という名前の表領域が見つからないことがすぐにわかります。

## サーバーのエラーの解釈 : 例

表領域のリカバリを試行したところ、次のエラーを受け取ったとします。

```
RMAN> RECOVER TABLESPACE users;
```

```
Starting recover at 29-AUG-01
```

```
using channel ORA_DISK_1
```

```
starting media recovery
```

```
media recovery failed
```

```
RMAN-00571: =====
```

```
RMAN-00569:===== ERROR MESSAGE STACK FOLLOWS =====
```

```
RMAN-00571: =====
```

```
RMAN-03002: recover コマンドが 08/29/2001 15:18:43 で失敗しました
```

```
RMAN-11003: SQL 文: alter database recover if needed tablespace USERS の解析 / 実行中に障害が発生しました。
```

```
ORA-00283: エラーによってリカバリ・セッションは取り消されました。
```

```
ORA-01124: データ・ファイル 8 はリカバリできません - ファイルは使用中またはリカバリ中です
```

```
ORA-01110: データ・ファイル 8: '/oracle/oradata/trgt/users01.dbf'
```

前述のとおり、下から読み始めます。ORA-01110 メッセージは、データ・ファイル users01.dbf のリカバリで問題が発生したことを示します。2 番目のエラーは、データ・ファイルが使用中か、すでにリカバリ中であるために、Oracle はそのデータ・ファイルをリカバリできないことを示します。その他の Recovery Manager エラーは、サーバー・エラーのためにリカバリ・セッションが取り消されたことを示します。このデータ・ファイルは、リカバリを完了していないため、オンラインになっていることが問題だと判断できます。これをオフラインにした上でバックアップのリストアを行います。

## メディア管理のエラーの解釈 : 例

Recovery Manager のメッセージ出力の中で、メディア管理のエラーは少なくありません。テープ・ドライブを使用して、バックアップ・ジョブ中に次の出力を受け取ったとします。

```
RMAN-00571: =====
```

```
RMAN-00569:===== ERROR MESSAGE STACK FOLLOWS =====
```

```
RMAN-00571: =====
```

```
RMAN-03009: backup コマンド (c1 チャンネル上) が 09/04/2001 13:18:19 で失敗しました
```

```
ORA-19506: シーケンシャル・ファイルの作成に失敗しました。ファイル名 ="07d36ecp_1_1"、パラメータ ="
```

```
ORA-27007: ファイルのオープンに失敗しました。
```

```
SVR4 Error: 2: No such file or directory
```

```
Additional information: 7005
```

```
Additional information: 1
```

```
ORA-19511: メディア・マネージャ・レイヤーからのエラーを受け取りました。エラー・テキスト:
```

```
SBT error = 7005, errno = 2, sbtopen: system error
```

前述のエラー・メッセージ・スタックを読むときの提案事項に従って、追加情報の行を探して確認します。

```
Additional information: 7005
```

エラー 7005 は、メディア管理デバイスがビジーであることを意味します。したがって、メディア管理ソフトウェアは、デバイスが使用中であるか、デバイスに問題があるため、デバイスに書き込めません。

---

---

**注意：** `sbtio.log` に含まれる情報は、メディア管理ソフトウェアが書き出したものであり、Oracle データベース・サーバーが書き出したものではありません。エラー・コードおよびメッセージの解釈は、メディア・ベンダーのドキュメントを参照してください。

---

---

## Recovery Manager のリターン・コードの識別

Recovery Manager にエラーが発生したかどうかを判断する 1 つの方法として、リターン・コードを調べます。Recovery Manager は、エラーが発生していない場合にはオペレーティング・システムに 0 を戻し、エラーが発生した場合には 0 以外の値を戻します。たとえば、C シェルを使用して UNIX を実行している場合、Recovery Manager の完了時には `$status` というシェル変数にリターン・コードが入ります。

## Media Management API のテスト

特定のプラットフォームに対して、Oracle は、`sbttest` と呼ばれる診断ツールを提供しています。このユーティリティは、Oracle データベース・サーバーとして動作し、メディア・マネージャとの通信を試行して、メディア管理ソフトウェアの簡単なテストを実行します。

### sbttest ユーティリティの取得

UNIX では、`sbttest` ユーティリティは、`$ORACLE_HOME/bin` にあります。なんらかの理由で、このユーティリティがプラットフォームに含まれていない場合は、オラクル社カスタマ・サポート・センターに連絡して、このプログラムの C バージョンを取得してください。このバージョンのプログラムは、すべての UNIX プラットフォームでコンパイルできます。

Solaris などのプラットフォームでは、`sbttest` の使用時に再リンクを行う必要はありません。その他のプラットフォームでは、再リンクが必要になる場合があります。

## sbttest 用オンライン・マニュアルの取得

sbttest のオンライン・ドキュメントを取得するには、コマンドラインで次のコマンドを発行します。

```
% sbttest
```

このプログラムで使用可能な引数のリストが表示されます。

```
Error: backup file name must be specified
Usage: sbttest backup_file_name      # this is the only required parameter
      <-dbname database_name>
      <-trace trace_file_name>
      <-remove_before>
      <-no_remove_after>
      <-read_only>
      <-no_regular_backup_restore>
      <-no_proxy_backup>
      <-no_proxy_restore>
      <-file_type n>
      <-copy_number n>
      <-media_pool n>
      <-os_res_size n>
      <-pl_res_size n>
      <-block_size block_size>
      <-block_count block_count>
      <-proxy_file os_file_name bk_file_name
                        [os_res_size pl_res_size block_size block_count]>
```

各引数の意味も表示されます。たとえば、2つのオプション・パラメータの説明例を次に示します。

Optional parameters:

- dbname specifies the database name which will be used by SBT to identify the backup file. The default is "sbtddb"
- trace specifies the name of a file where the Media Management software will write diagnostic messages.

## sbtttest ユーティリティの使用法

sbtttest を使用して、メディア・マネージャの簡単なテストを実行します。次の表は、出力の解釈について説明します。

sbtttest の戻り値	処理 ..
0	プログラムはエラーなしで実行されました。つまり、メディア・マネージャがインストールされ、データ・ストリームを受け入れ、要求されると同じデータを戻します。
0 以外	プログラムはエラーを検出しました。メディア・マネージャがインストールされていないか、または正しく構成されていません。

### sbtttest の使用法

1. コマンドラインから sbtttest を入力し、このプログラムがインストールされてシステム・パスに含まれていることを確認します。

```
% sbtttest
```

プログラムが操作可能な場合には、オンライン・マニュアルが表示されます。

2. オンライン・マニュアルに説明されている任意の引数を指定して、プログラムを実行します。たとえば、次のように入力して、テスト・ファイル some\_file.f を作成し、出力を sbtio.log に書き込みます。

```
% sbtttest some_file.f -trace sbtio.log
```

既存のデータ・ファイルのバックアップもテストできます。たとえば、このコマンドで、データベース prod のデータ・ファイル tbs\_33.f をテストします。

```
% sbtttest tbs_33.f -dbname prod
```

3. 出力をチェックします。プログラムがエラーを検出した場合には、障害を説明するメッセージが表示されます。たとえば、Oracle でライブラリが見つからない場合は、次のようなメッセージが表示されます。

```
libobk.so could not be loaded. Check that it is installed properly, and that LD_LIBRARY_PATH environment variable (or its equivalent on your platform) includes the directory where this file can be found. Here is some additional information on the cause of this error:
ld.so.1: sbtttest: fatal: libobk.so: open failed: No such file or directory
```

`sbttest` が機能しても、Recovery Manager のバックアップが機能しない場合があります。次のような理由が考えられます。

- `sbttest` を実行したユーザーが、Oracle プロセスの所有者ではない場合。
- `sbttest` は機能していても、Oracle サーバーがメディア管理ライブラリにリンクされていない場合。
- `sbttest` プログラムではシェルからのすべての環境変数の受渡しが行われているが、Recovery Manager では行われていない場合。

## Recovery Manager コマンドの終了

実行中の Recovery Manager コマンドを終了するには、次の方法があります。

- デフォルトの方法として、Recovery Manager インタフェースで、[Ctrl] キーを押しながら [C]（または、使用しているシステムの "attention" キーの組合せ）を押します。この操作では、テープのマウントを待っている場合など、割り当てられたチャネルがメディア管理操作で待機していないかぎり、そのチャネルも終了します。
- SQL の ALTER SYSTEM 文の実行による Recovery Manager のチャネルに対応するサーバー・セッションを強制終了します。
- オペレーティング・システム上の Recovery Manager のチャネルに対応するサーバー・セッションを終了します。

## ALTER SYSTEM によるセッションの終了

Recovery Manager のチャネル用の Oracle セッション ID は、Recovery Manager ログの中の、次のような書式のメッセージにより識別できます。

```
channel ch1: sid=15 devtype=SBT_TAPE
```

`sid` および `devtype` は、それぞれに割り当てられたチャネルを示します。Oracle の `sid` は、オペレーティング・システム・プロセスの ID とは異なるので、注意してください。セッションは SQL 文に `sid` を指定して強制終了できますが、このコマンドはオペレーティング・システム・プロセスの `kill` コマンドと同じものではありません。

`sid` は、SQL 文の ALTER SYSTEM KILL SESSION コマンドで指定します。このコマンドは、2 つの引数 (Recovery Manager に出力される `sid` とシリアル番号) を使用しますが、この 2 つの引数は `V$SESSION` を問い合わせることで取得できます。たとえば、次の文を実行します。`sid_in_rman_output` は、Recovery Manager メッセージから得られた番号です。

```
SELECT SERIAL# FROM V$SESSION WHERE SID=sid_in_rman_output;
```



代入した `sid_in_rman_output` と問合せで取得したシリアル番号で次の文を実行します。

```
ALTER SYSTEM KILL SESSION 'sid_in_rman_output,serial#';
```

この操作は、プロセスがメディア・マネージャで停止した場合、オペレーティング・システム・レベルの強制終了と同等であることに注意してください。

## オペレーティング・システム・レベルでのセッション終了

サーバー・セッションに対応付けられたプロセスの検出および強制終了は、オペレーティング・システム固有のものです。プラットフォームによっては、サーバー・セッションがどのプロセスともまったく対応付けられていません。詳細は、使用しているオペレーティング・システム固有のマニュアルを参照してください。

## メディア・マネージャで停止した Recovery Manager セッションの終了

メディア・マネージャで停止した Recovery Manager ジョブの強制終了が必要になる場合があります。チャンネルの接続がメディア・マネージャで停止したとき、Recovery Manager を終了する最善の方法は、メディア・マネージャのセッションを強制終了することです。この処理で問題が解決しない場合は、接続を行っている Oracle プロセスを強制終了できるかどうかを試します。Oracle プロセスの強制終了は、メディア・マネージャに問題を発生させる場合がありますため、注意してください。

この項では、次の項目について説明します。

- [Recovery Manager のセッションのコンポーネント](#)
- [ジョブ停止中のプロセス動作](#)
- [Recovery Manager のセッションの終了: 基本手順](#)

### Recovery Manager のセッションのコンポーネント

Recovery Manager のセッションの特性は、オペレーティング・システムによって異なります。UNIX には、Recovery Manager のセッションに関連する次のようなセッションがあります。

- **Recovery Manager のプロセス自体。**
- リカバリ・カタログ・データベースへの**カタログ接続**（リカバリ・カタログを使用する場合のみ）。
- 補助インスタンスへの**補助接続**（DUPLICATE または TSPITR を実行する場合のみ）。
- ターゲット・データベースへの初期接続。**デフォルト・チャンネル**とも呼ばれます。

- 様々な割当てチャンネルで実行する Recovery Manager コマンドの監視に使用される、ターゲット・データベースへの**ポーリング**接続。デフォルトでは、Recovery Manager が行うポーリング接続は 1 つです。Recovery Manager では、ALLOCATE CHANNEL または CONFIGURE CHANNEL コマンドで様々な接続文字列を使用して、ポーリング接続を追加します。ALLOCATE CHANNEL または CONFIGURE CHANNEL コマンドで使用した、各個別の接続文字列に対して、1 つのポーリング接続が存在します。
- 割り当てられた各チャンネルに対応するターゲット・データベースへの 1 つの**ターゲット接続**。

### ジョブ停止中のプロセス動作

通常、Recovery Manager は、チャンネル接続の 1 つが、メディア・マネージャ・コードでテープ・リソースを待っていることが原因で停止します。カタログ接続とデフォルトのチャンネルは、Recovery Manager からの指示を待っているため、停止しているように見えます。ポーリング接続は、Recovery Manager のプロセスの制御下で RPC をポーリングする間、無限にループしているように見えます。

Recovery Manager のプロセス自体を強制終了する場合は、カタログ接続、補助接続、デフォルトのチャンネルおよびポーリング接続も強制終了します。メディア・マネージャ・コードで停止していない、ターゲット接続および補助接続も終了します。メディア管理レイヤーで実行している、ターゲット接続および補助接続のみがアクティブのままです。セッションを終了してもこのプロセスは強制終了されないため、手動で強制終了する必要があります。終了した後も、メディア・マネージャは、Oracle プロセスが終了したことを認識していないため、リソースをビジーのままにしたり、処理を継続する場合があります。この動作は、使用しているメディア・マネージャによって異なります。

カタログ接続を終了しても、Recovery Manager は終了しません。バックアップまたはリストアの進行中には Recovery Manager がカタログ操作を実行していないからです。デフォルトのチャンネルとポーリング接続を削除すると、Recovery Manager のプロセスは、チャンネルの 1 つが終了したことを検出して、終了の処理を行います。この場合は、前述のように、停止したチャンネルへの接続はアクティブのままです。

## Recovery Manager のセッションの終了 : 基本手順

割り当てられたチャネルの接続がメディア・マネージャで停止したとき、Recovery Manager を終了する最善の方法は、接続を行っている Oracle プロセスを強制終了することです。Recovery Manager のプロセスはこの終了を検出し、メディア管理レイヤーで操作できるターゲット接続を除くすべての接続を削除して、終了処理を行います。この場合も、メディア・マネージャのリソースに関する通告が適用されます。

### メディア・マネージャで停止した Oracle プロセスの終了

1. V\$SESSION と V\$SESSION\_WAIT を問い合わせます。手順については、15-23 ページの「[Recovery Manager とメディア・マネージャの相互作用の監視](#)」を参照してください。たとえば、次の問合せを実行します。

```
COLUMN EVENT FORMAT a10
COLUMN SECONDS_IN_WAIT FORMAT 999
COLUMN STATE FORMAT a20
COLUMN CLIENT_INFO FORMAT a30

SELECT p.SPID, EVENT, SECONDS_IN_WAIT AS SEC_WAIT,
       STATE, CLIENT_INFO
FROM V$SESSION_WAIT sw, V$SESSION s, V$PROCESS p
WHERE sw.EVENT LIKE 'sbt%'
      AND s.SID=sw.SID
      AND s.PADDR=p.ADDR
/
```

SQL 出力を調べて、どの sbt ファンクションが待機中であるかを判断します。たとえば、出力は次のようになります。

SPID	EVENT	SEC_WAIT	STATE	CLIENT_INFO
8642	sbtwrite2	600	WAITING	rman channel=ORA_SBT_TAPE_1
8374	sbtwrite2	600	WAITING	rman channel=ORA_SBT_TAPE_2

2. 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、停止したプロセスを強制終了します。たとえば、Solaris では、kill -9 コマンドを実行します。

```
% kill -9 8642 8374
```

3. メディア・マネージャがプロセスも消去していることをチェックします。消去していないと、次のバックアップまたはリストア処理で、以前の停止が原因で再度停止する場合があります。一部のメディア・マネージャでは、メディア・マネージャをシャットダウンして再起動することが唯一の解決策となる場合があります。メディア・マネージャのドキュメントで解決できない場合は、メディア・マネージャの技術サポートに連絡して、適切に解決してください。

**関連項目：** 関連するコマンドの詳細は、使用しているオペレーティング・システム固有のドキュメントを参照してください。

## V\$ ビューを介した Recovery Manager の監視

LIST、REPORT および SHOW を使用すると、Recovery Manager のほとんどの情報（バックアップ、コピー、データベース・インカネーション、構成設定など）を取得できます。ただし、V\$ ビューを使用して、これらのコマンドでは表示されない情報を表示する場合もあります。

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と有益な場合があります。次の表で説明するように、Recovery Manager のジョブの進行状況を監視し、関連する情報の取得を支援する、いくつかのビューにアクセスすることができます。

ビュー	説明
V\$PROCESS	現在アクティブなプロセスを識別します。
V\$RECOVER_FILE	リカバリを必要とするデータ・ファイルを識別します。
V\$SESSION	現在アクティブなセッションを識別します。Oracle データベース・サーバー・セッションと、Recovery Manager の割当てチャネルの対応関係の判断に、このビューを使用します。
V\$SESSION_LONGOPS	Recovery Manager のバックアップおよびリストア・ジョブに関する進行状況レポートを提供します。
V\$SESSION_WAIT	セッションが待機しているイベントまたはリソースをリスト表示します。
V\$BACKUP_SYNC_IO	バックアップを実行するプロセス（プラットフォームによってはスレッド）に対して I/O が同期しているときに行が表示されます。
V\$BACKUP_ASYNC_IO	バックアップを実行するプロセス（プラットフォームによってはスレッド）に対して I/O が非同期のときに行が表示されます。

V\$BACKUP\_SYNC\_IO には、バックアップを実行するプロセス（プラットフォームによってはスレッド）に対して I/O が同期しているときに行が表示されます。V\$BACKUP\_ASYNC\_IO には、I/O が非同期の場合に行が表示されます。非同期 I/O が得られるのは、I/O プロセスに伴うか、基本オペレーティング・システムが非同期 I/O をサポートしている場合です。

Recovery Manager を使用すると、次の項に説明されている様々なチェックを実行できます。

- [サーバー・セッションと Recovery Manager チャネルの相関](#)
- [Recovery Manager ジョブの進行状況の監視](#)
- [Recovery Manager とメディア・マネージャの相互作用の監視](#)
- [Recovery Manager ジョブのパフォーマンスの監視](#)
- [リカバリを必要とするデータ・ファイルの判断](#)

## サーバー・セッションと Recovery Manager チャネルの相関

どのサーバー・セッションがどの Recovery Manager チャネルに相関付けられているかを識別するために、V\$SESSION および V\$PROCESS を問い合わせることができます。V\$PROCESS の SPID 列には、プロセスまたはスレッドのオペレーティング・システム ID 番号が表示されます。たとえば、UNIX では SPID 列にプロセス ID が表示され、Windows NT では SPID 列にスレッド ID が表示されます。この情報を取得するには、同時に複数の Recovery Manager セッションがアクティブになっているかどうかに応じて、次の 2 つの基本方法があります。

### 1 つの Recovery Manager セッションがアクティブになっている場合のサーバー・セッションとチャネルの一致

Recovery Manager セッションが 1 つのみアクティブな場合、Recovery Manager チャネルのサーバー・セッション ID を判断する最も簡単な方法は、Recovery Manager ジョブの実行中にターゲット・データベースで次の問合せを実行することです。

```
COLUMN CLIENT_INFO FORMAT a30
COLUMN SID FORMAT 999
COLUMN SPID FORMAT 9999

SELECT s.SID, p.SPID, s.CLIENT_INFO
FROM V$PROCESS p, V$SESSION s
WHERE p.ADDR = s.PADDR
AND CLIENT_INFO LIKE 'rman%'
/
```

Recovery Manager ジョブで SET COMMAND ID コマンドを実行しないと、次の書式で CLIENT\_INFO 列が表示されます。

```
rman channel=channel_id
```

サンプルの出力を次に示します。

```
SID SPID          CLIENT_INFO
-----
14 8374          rman channel=ORA_SBT_TAPE_1
```

複数の Recovery Manager セッションでのサーバー・セッションとチャネルの一致

複数の Recovery Manager セッションがアクティブな場合は、各セッションで1つのチャネルについて V\$SESSION.CLIENT\_INFO 列に同じ情報が表示される場合があります。次に例を示します。

SID	SPID	CLIENT_INFO
14	8374	rman channel=ORA_SBT_TAPE_1
9	8642	rman channel=ORA_SBT_TAPE_1

この場合は、次の方法でどのチャネルがどの SID 値に対応しているかを判断します。

**Recovery Manager 出力からのチャネル ID の取得** この方法では、最初に Recovery Manager 出力から sid 値を取得し、その値を SQL 問合せで使用する必要があります。

バックアップ時のプロセスとチャネルの関連付け

- 1. アクティブなセッションの1つで Recovery Manager ジョブを通常どおり実行し、出力を調べてチャネルの sid を取得します。サンプル出力を次に示します。

```
Starting backup at 21-AUG-01
allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: sid=14 devtype=SBT_TAPE
```

- 2. Recovery Manager のジョブの実行中に、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います。たとえば、次のように入力します。

```
COLUMN CLIENT_INFO FORMAT a30
COLUMN SID FORMAT 999
COLUMN SPID FORMAT 9999

SELECT s.SID, p.SPID, s.CLIENT_INFO
FROM V$PROCESS p, V$SESSION s
WHERE p.ADDR = s.PADDR
AND CLIENT_INFO LIKE 'rman%'
/
```

最初の手順で取得した sid 値を使用して、どのチャネルがどのサーバー・セッションに対応しているかを判断します。

SID	SPID	CLIENT_INFO
14	2036	rman channel=ORA_SBT_TAPE_1
12	2066	rman channel=ORA_SBT_TAPE_1

**SET COMMAND ID を使用したサーバー・セッションとチャネルの相関** この方法では、Recovery Manager バックアップ・スクリプトでコマンド ID 文字列を指定します。この文字列を V\$SESSION.CLIENT\_INFO に問合せできます。

#### バックアップ時のプロセスとチャネルの関連付け

1. 各セッションで、チャネルの割当て後に COMMAND ID を異なる値に設定してから、必要なオブジェクトをバックアップします。たとえば、セッション 1 で次のように入力します。

```
RUN
{
  ALLOCATE CHANNEL c1 TYPE sbt;
  SET COMMAND ID TO 'sess1';
  BACKUP DATABASE;
}
```

セッション 2 で実行中のジョブで、コマンド ID を sess2 などの文字列に設定します。

```
RUN
{
  ALLOCATE CHANNEL c1 TYPE sbt;
  SET COMMAND ID TO 'sess2';
  BACKUP DATABASE;
}
```

2. Recovery Manager のジョブの実行中に、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います。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=sess%';
```

Recovery Manager ジョブで SET COMMAND ID コマンドを実行すると、次の書式で CLIENT\_INFO 列が表示されます。

```
id=command_id,rman channel=channel_id
```

サンプルの出力を次に示します。

SID	SPID	CLIENT_INFO
11	8358	id=sess1
15	8638	id=sess2
14	8374	id=sess1,rman channel=c1
9	8642	id=sess2,rman channel=c1

文字列 `rman channel` を含む行は、バックアップを実行中のチャンネルを示します。残りの行は、ターゲット・データベースへの接続に関する情報を示します。

**関連項目：** `SET COMMAND ID` の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。`V$SESSION` および `V$PROCESS` の詳細は、『Oracle9i データベース・リファレンス』を参照してください。

Recovery Manager ジョブの進行状況の監視

バックアップ、コピーおよびリストアの進行状況を監視するには、ビュー `V$SESSION_LONGOPS` に問合せを行います。Recovery Manager は、`V$SESSION_LONGOPS` で詳細行および集計行という 2 種類の行を使用します。詳細行は 1 つのジョブ・ステップで処理されているファイルの説明を示し、集計行は Recovery Manager コマンドのすべてのジョブ・ステップで処理されたファイルの説明を示します。ジョブ・ステップは、バックアップ・セットまたはデータ・ファイルのコピーの作成またはリストアです。詳細行は、バックアップ・ステップ中に読み込み / 書き込みされた各バッファで更新されるため、更新の粒度は小さくなります。集計行は各ジョブ・ステップが完了すると更新されるため、更新の粒度は大きくなります。

表 15-3 に、`V$SESSION_LONGOPS` のうち Recovery Manager に最も関係の深い列を示します。通常は、集計行ではなく詳細行を表示して、各バックアップ・セットの進行状況を判断します。

表 15-3 V\$SESSION\_LONGOPS の Recovery Manager 関連の列

列	詳細行の説明
SID	Recovery Manager チャンネルに対応するサーバー・セッション ID。
SERIAL#	サーバー・セッションのシリアル番号。この値は、サーバー・セッションが再利用されるたびに変更されます。
OPNAME	行のテキスト記述。詳細行の例は、 <code>RMAN: datafile copy</code> 、 <code>RMAN: full datafile backup</code> および <code>RMAN: full datafile restore</code> などです。 <b>注意：</b> <code>RMAN: aggregate input</code> および <code>RMAN: aggregate output</code> は、集計のみの行です。
CONTEXT	バックアップの出力行の場合、この値は 2 です。プロキシ・コピー（この列が更新されません）を除き、他のすべての行の場合、この値は 1 です。
SOFAR	イメージ・コピーの場合は読み込まれたブロック数、バックアップの入力行の場合は、バックアップ中のファイルから読み込まれたブロック数、バックアップの出力行の場合は、バックアップ・ピースに書き込まれたブロック数です。リストアの場合は、この 1 つのジョブ・ステップでリストアされるファイルに対して処理されたブロック数、プロキシ・コピーの場合はコピーされたファイル数です。



表 15-3 V\$SESSION\_LONGOPS の Recovery Manager 関連の列 (続き)

列	詳細行の説明
TOTALWORK	イメージ・コピーの場合はファイル内のブロックの合計数、バックアップの入力行の場合は、このジョブ・ステップで処理されたすべてのファイルから読み込まれるブロックの合計数です。バックアップの出力行の場合、Recovery Manager ではバックアップ・ピースに書き込まれるブロック数が不明のため、この値は 0 です。リストアの場合はこのジョブ・ステップでリストアされた全ファイル内のブロックの合計数、プロキシ・コピーの場合はこのジョブ・ステップでコピーされるファイルの合計数です。

バックアップ、リストアまたはコピーを実行する各サーバー・セッションは、ジョブ・ステップに要する全作業量に関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データベース・リストアを行い、各チャンネルでバックアップ・セットを 2 つずつ (合計 4 セット) リストアする場合、それぞれのサーバー・セッションは、1 つのバックアップ・セットの経過に関する進行状況をレポートします。そのセットが完全にリストアされた後、Recovery Manager は、次にリストアするセットに関して進行状況レポートを開始します。

ジョブの進行状況の監視

1. ターゲット・データベースに接続後、必要に応じて、リカバリ・カタログ・データベースに接続後、Recovery Manager のジョブを起動します。たとえば、次のように入力します。

```
RESTORE DATABASE;
```

2. ジョブの実行中に、次の SQL 文を含んだスクリプトを実行します。

```
SELECT SID, SERIAL#, CONTEXT, SOFAR, TOTALWORK,
       ROUND(SOFAR/TOTALWORK*100,2) "%_COMPLETE"
FROM V$SESSION_LONGOPS
WHERE OPNAME LIKE 'RMAN%'
AND OPNAME NOT LIKE '%aggregate%'
AND TOTALWORK != 0
AND SOFAR <> TOTALWORK
/
```

リストアの進行中に問合せを繰り返すと、次のような出力が表示されます。

```
SQL> @longops
      SID      SERIAL#      CONTEXT      SOFAR  TOTALWORK  %_COMPLETE
-----
          8          19          1       10377       36617       28.34

SQL> @longops
      SID      SERIAL#      CONTEXT      SOFAR  TOTALWORK  % COMPLETE
-----
```

```

      8      19      1      21513      36617      58.75

SQL> @longops
      SID      SERIAL#      CONTEXT      SOFAR      TOTALWORK % COMPLETE
-----
      8      19      1      29641      36617      80.95

SQL> @longops
      SID      SERIAL#      CONTEXT      SOFAR      TOTALWORK % COMPLETE
-----
      8      19      1      35849      36617      97.9

SQL> @longops
no rows selected
```

- 3. 2 分以上の間隔でスクリプトを実行し、%\_COMPLETE 列が増加しない場合は、Recovery Manager に問題が発生しています。詳細は、15-23 ページの「[Recovery Manager とメディア・マネージャの相互作用の監視](#)」を参照してください。

また、UNIX コマンドラインから次の Perl スクリプトを実行することもできます。このスクリプトは一時的な SQL 問合せスクリプトを作成して無限 while ループ内で実行します。ジョブが完了すると、Perl スクリプトは取り消されます。

```
#!/usr/local/bin/perl5
open (TMP, ">/tmp/test.sql")
|| die "can't open /tmp/test.sql\n";
print TMP
'SET FEEDBACK OFF
SPOOL /tmp/sql.out
SELECT SID, SERIAL#, CONTEXT, SOFAR, TOTALWORK,
      ROUND(SOFAR/TOTALWORK*100,2) "% COMPLETE"
FROM V$SESSION_LONGOPS WHERE OPNAME LIKE \'RMAN\''
AND OPNAME NOT LIKE \'%aggregate%\'' AND TOTALWORK != 0
AND SOFAR <> TOTALWORK;
SPOOL OFF
EXIT;
EOF';
close TMP;
while (1)
{
  # replace with your sqlplus connection information
  'sqlplus "/ AS SYSDBA" @/tmp/test';
  open(SQLOUT, "</tmp/sql.out")
  || die "can't open /tmp/sql.out\n";
  @whole = <SQLOUT>;
  foreach (@whole) {print if defined;}
  close(SQLOUT);
}
```

## Recovery Manager とメディア・マネージャの相互作用の監視

動的パフォーマンス・イベント・ビューでイベント名を使用すると、Media Management API への Recovery Manager コールを監視できます。次の例のように、イベント名と sbt ファンクションは 1 対 1 の対応関係を持っています。

```
sbtinit
sbtopen
sbtread
sbtwrite
sbtbackup
```

Media Management API のファンクションをコールする前に、サーバーは文字列 WAIT を含む STATUS 列を使用して、V\$SESSION\_WAIT に 1 行を追加します。V\$SESSION\_WAIT.SECONDS\_IN\_WAIT 列は、このコールの戻りをサーバーが待機していた秒数を示します。メディア・マネージャから sbt ファンクションが戻されると、この行は表示されなくなります。

V\$SESSION\_WAIT 内で sbt イベント名に対応する行はサーバーによって実行時に更新されるため、これらの行は問題を示しません。コールが行われるたびに行が表示され、戻されるたびに表示されなくなります。ただし、SECONDS\_IN\_WAIT 列の値が大きい場合は、メディア・マネージャが停止している可能性があります。

sbt イベントを監視するには、次の SQL 問合せを実行できます。

```
COLUMN EVENT FORMAT a10
COLUMN SECONDS_IN_WAIT FORMAT 999
COLUMN STATE FORMAT a20
COLUMN CLIENT_INFO FORMAT a30

SELECT p.SPID, EVENT, SECONDS_IN_WAIT AS SEC_WAIT,
       STATE, CLIENT_INFO
FROM V$SESSION_WAIT sw, V$SESSION s, V$PROCESS p
WHERE sw.EVENT LIKE 'sbt%'
      AND s.SID=sw.SID
      AND s.PADDR=p.ADDR
/
```

SQL 出力を調べて、どの sbt ファンクションが待機中であるかを判断します。たとえば、次の出力は、Recovery Manager が sbtbackup ファンクションの戻りを 10 分間待機していたことを示します。

SPID	EVENT	SEC_WAIT	STATE	CLIENT_INFO
8642	sbtbackup	600	WAITING	rman channel=ORA_SBT_TAPE_1

**注意：** V\$SESSION\_WAIT ビューには、Oracle のイベントのみが表示され、メディア・マネージャのイベントは表示されません。

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

## Recovery Manager ジョブのパフォーマンスの監視

バックアップとリストアのパフォーマンスを監視するには、V\$BACKUP\_SYNC\_IO および V\$BACKUP\_ASYNC\_IO に問合せを行います。

**関連項目：** これらの V\$ ビューの詳細は、『Oracle9i データベース・リファレンス』を参照してください。これらのビューを使用してバックアップのパフォーマンスをチューニングする方法は、14-16 ページの「[手順 8: V\\$ ビューを問い合わせてボトルネックを識別する](#)」を参照してください。

## リカバリを必要とするデータ・ファイルの判断

リカバリを必要とするファイル、およびそのファイルにリカバリが必要な理由を判断するために、動的パフォーマンス・ビュー V\$RECOVER\_FILE を使用することが多くあります。次の問合せにより、リカバリが必要なデータ・ファイルのファイル番号、リカバリの理由（認識されている場合）、およびリカバリを開始する必要がある SCN と時刻が表示されます。

```
COL FILE# FORMAT 999
COL ERROR FORMAT a10
SELECT * FROM V$RECOVER_FILE
/
```

FILE#	ONLINE	ONLINE_	ERROR	CHANGE#	TIME
-----					
4	ONLINE	ONLINE	FILE NOT FOUND	0	
5	ONLINE	ONLINE	FILE NOT FOUND	0	
8	OFFLINE	OFFLINE	OFFLINE NORMAL	0	

**注意：** 現在使用中の制御ファイルが、リストアされたバックアップか、メディア障害が発生した後に作成された新規制御ファイルの場合は、このビューは役に立ちません。リストアされた制御ファイルまたは再作成された制御ファイルには、V\$RECOVER\_FILE を正確に更新するために必要な情報が含まれていません。

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

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

NAME	NAME
-----	-----
/oracle/oradata/trgt/drsys01.dbf	DRSYS
/oracle/oradata/trgt/example01.dbf	EXAMPLE
/oracle/oradata/trgt/users01.dbf	USERS

これらの問合せは、次の SQL\*Plus スクリプトにまとめることができます（サンプル出力も含まれています）。

```
COL df# FORMAT 999
COL df_name FORMAT a35
COL tbsp_name FORMAT a10
COL status FORMAT a7
COL error FORMAT a10
```

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

DF#	DF_NAME	TBSP_NAME	STATUS	ERROR	CHANGE#	TIME
-----	-----	-----	-----	-----	-----	-----
4	/oracle/oradata/trgt/drsys01.dbf	DRSYS	ONLINE	FILE NOT FOUND	0	
5	/oracle/oradata/trgt/example01.dbf	EXAMPLE	ONLINE	FILE NOT FOUND	0	
8	//oracle/oradata/trgt/users01.dbf	USERS	OFFLINE	OFFLINE NORMAL	0	

## トラブルシューティング例

この項では、次の項目について説明します。

- メディア・マネージャのインストール後に、**Recovery Manager** のチャンネルの割当てに失敗した場合 : 例
- バックアップ・ジョブが停止する場合 : 例
- **Recovery Manager** が RPC コールの起動に失敗する場合 : 例
- 無効な **RECID** エラーでバックアップが失敗する場合 : 例
- 制御ファイルのエンキューが原因でバックアップが失敗する場合 : 例
- **Recovery Manager** がすべてのアーカイブ・ログの削除に失敗する場合 : 例
- **Recovery Manager** でアーカイブ・ログが見つからないためにバックアップが失敗する場合 : 例
- **Recovery Manager** がターゲット・データベース・キャラクタ・セットを設定できない場合 : 例
- **Recovery Manager** がキャラクタ・セット名を識別しない場合 : 例
- **Recovery Manager** がターゲット・データベースへのログオンを拒否する場合 : 例
- ログの消失が原因でデータベースの複製が失敗する場合 : 例
- 複数の **RMAN-06023** エラーが表示されて複製が失敗する場合 : 例
- リカバリ・カタログに **UNKNOWN** というデータベース名が表示される場合 : 例

## メディア・マネージャのインストール後に、Recovery Manager のチャネルの割当てに失敗した場合：例

この例では、8-2 ページの「[メディア・マネージャにバックアップを作成する場合の Recovery Manager の構成](#)」で説明したようにメディア・マネージャのインストールおよびテストを行っていますが、Recovery Manager でテープへのバックアップができません。たとえば、sbt チャネルの割当て後、次のようなエラー・スタックを受け取ります。

```

RMAN-00571: =====
RMAN-00569:===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03009: allocate コマンド (c1 チャネル上) が 08/29/2001 17:16:54 で失敗しました
ORA-19554: デバイスの割当て中にエラーが発生しました。デバイス型: SBT_TAPE、デバイス名:
ORA-27211: メディア管理ライブラリのロードに失敗しました
Additional information: 25

```

## メディア・マネージャのインストール後に、Recovery Manager のチャネルの割当てに失敗した場合：診断

ORA-27211 のエラーは、Oracle がメディア管理ライブラリをロードしていないために、チャネルの割当てに失敗したことを示しています。Oracle では、チャネルの割当てに失敗した場合、チャネル割当ての失敗を引き起こしたエラーを格納したトレース・ファイルを、USER\_DUMP\_DEST の位置に生成します。トレース・ファイルには、他のすべてのメディア・マネージャのエラーまたはオペレーティング・システムのエラーに加え、Oracle がロードしたメディア管理ライブラリの完全なパス名も含まれます。たとえば、UNIX 上のトレース・ファイルは /oracle/rdbms/log/prod1\_ora\_16226.trc などの名前です。次に示す内容を含みます。

```

*** 2001-08-29 17:16:54.385
SKGPFQ OSD: Error in function sbtinit on line 2396
SKGPFQ OSD: Look for SBT Trace messages in file /oracle/rdbms/log/sbtio.log
SBT Initialize failed for oracle.static

```

この出力の最後の行では、Oracle はインストールしたメディア管理ライブラリのかわりに、デフォルトの静的ライブラリをロードしていることが示されています。

メディア管理ライブラリのロードをテストするには、PARMS パラメータ SBT\_LIBRARY を使用して、メディア管理ライブラリの強制ロードを行い、チャネルの割当てを実行します。たとえば、/vendor/lib/some\_mm\_lib.so というライブラリが、\$ORACLE\_HOME/lib/libobk.so と示されている場合には、使用しているメディア・マネージャに必要な PARMS をすべて設定し、次のコマンドを実行します。

```

RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt
    PARMS='SBT_LIBRARY=?/lib/libobk.so',
    'ENV=(NSR_SERVER=tape_svr,NSR_CLIENT=oracleclnt,NSR_GROUP=oracle_tapes)';
}

```

チャンネルの割当てに失敗した場合には、再度トレース・ファイルをチェックして、他に新しい原因があるかどうかを調べてください。SBT\_LIBRARY によるチャンネル割当てに成功し、通常の sbt チャンネル割当てには失敗した場合、Oracle はインストールしたライブラリ (UNIX の場合は \$ORACLE\_HOME/lib/libobk.so、NT の場合は %ORACLE\_HOME%\bin\orasbt.dll) 以外のライブラリをロードしようとしている可能性があります。オペレーティング・システムのパスに 2 つ以上のライブラリがある場合、Oracle は間違ったライブラリをロードしています。

### **メディア・マネージャのインストール後に、Recovery Manager のチャンネルの割当てに失敗した場合：解決策**

Oracle が正しいライブラリをロードしていないという問題の場合には、ライブラリに正しい名前が付けられているかどうかを確認してください。たとえば、UNIX の場合には、ライブラリの名前を \$ORACLE\_HOME/rdbms/libobk.so とするか、この名前ですymbolic linkを作成して使用するライブラリを指してください。また、Windows NT の場合には、ライブラリの名前を %ORACLE\_HOME%\bin\orasbt.dll としてください。これらのファイルはデフォルト以外のディレクトリに置くことも可能ですが、Oracle がその位置を特定できるように、必ずシステム・パスに含めてください。

---

---

**関連項目：** 有効な PARMS パラメータの説明は、『Oracle9i Recovery Manager リファレンス』を参照してください。

---

---



## バックアップ・ジョブが停止する場合：例

この例では、Recovery Manager のバックアップ・ジョブを通常通り起動しましたが、停止してしまいました。

```
Recovery Manager: Release 9.2.0.0.0 - Production
connected to target database: TRGT
connected to recovery catalog database
```

```
RMAN> BACKUP TABLESPACE SYSTEM, tools;
```

```
allocated channel: t1
channel t1: sid=16 devtype=SBT_TAPE
```

```
channel t1: starting datafile backupset
set_count=15 set_stamp=338309600
channel t1: including datafile 2 in backupset
channel t1: including datafile 1 in backupset
channel t1: including current controlfile in backupset
# Hanging here for 30 minutes now
```

## バックアップ・ジョブが停止する場合：診断

バックアップ・ジョブが停止した場合、つまり処理が進行しない場合は、いくつかの原因が考えられます。

- ジョブが異常終了した場合
- サーバー側またはメディア管理のエラーが発生した場合
- 新しいカセットをテープ・デバイスに挿入するなどのイベントを Recovery Manager が待機している場合

15-23 ページの「[Recovery Manager とメディア・マネージャの相互作用の監視](#)」で説明したように、sbt 待機イベントを問い合わせで詳細情報を取得できます。たとえば、ターゲット・インスタンスに対して次の問合せを実行します。

```
COLUMN EVENT FORMAT a10
COLUMN SECONDS_IN_WAIT FORMAT 999
COLUMN STATE FORMAT a20
COLUMN CLIENT_INFO FORMAT a30

SELECT p.SPID, EVENT, SECONDS_IN_WAIT AS SEC_WAIT,
       STATE, CLIENT_INFO
FROM V$SESSION_WAIT sw, V$SESSION s, V$PROCESS p
WHERE sw.EVENT LIKE 'sbt%'
       AND s.SID=sw.SID
       AND s.PADDR=p.ADDR

/
```

SQL 出力を調べて、どの sbt ファンクションが待機中であることを判断します。たとえば、出力は次のようになります。

SPID	EVENT	SEC_WAIT	STATE	CLIENT_INFO
8642	sbtbackup	300	WAITING	rman channel=ORA_SBT_TAPE_1

## バックアップ・ジョブが停止する場合：解決策

バックアップ・ジョブが停止する原因はいくつか考えられるため、解決策もいくつかあります。たとえば、テープ・デバイスで使用中のカセットがいっぱいになり、単に新しいテープの挿入を待っているため、バックアップ・ジョブが停止している例が多くあります。sbt 待機イベントの間合せて問題が示されれば理想的です。

sbt 待機イベントの間合せて解決できない場合には、メディア・マネージャのプロセス、ログおよびトレース・ファイルを確認して、異常終了などのエラーが発生していないか調べます（メッセージ・ファイルの詳細は、15-2 ページの「[メッセージ出力のタイプの識別](#)」を参照してください）。

---

**関連項目：** 停止中の Recovery Manager セッションを強制終了する方法は、15-15 ページの「[Recovery Manager のセッションの終了：基本手順](#)」を参照してください。

---

## Recovery Manager が RPC コールの起動に失敗する場合：例

この例では、バックアップ・ジョブを実行し、次の同じようなメッセージ出力を受け取ります。

```
channel c8: including datafile number 47 in backupset
RPC call appears to have failed to start on channel c9
RPC call ok on channel c9
channel c3: including datafile number 18 in backupset
```

## Recovery Manager が RPC コールの起動に失敗する場合：診断

「RPC コールの起動がチャネル xx で失敗しました。」というメッセージは、通常は問題を示しません。このメッセージが示す意味は、次のいずれかです。

- ターゲット・データベースのインスタンスの処理が遅い場合
- タイミングの問題が発生した場合

タイミングの問題は、次のような場合に発生します。Recovery Manager は RPC を開始するときに、V\$SESSION パフォーマンス・ビューをチェックします。RPC はビュー内の情報を更新し、起動時刻と終了時刻を示します。Recovery Manager は、RPC が起動したことを示す前に V\$SESSION をチェックする場合があります、この結果として次のメッセージが生成されます。

RPC call appears to have failed

メッセージ文字列「RPC コールはチャンネル xx で OK です。」が表示されずに、即時に、メッセージ文字列「RPC コールの起動がチャンネル xx で失敗しました。」が表示された場合には、バックアップ・ジョブに問題が発生しています。

## 無効な RECID エラーでバックアップが失敗する場合：例

この例では、バックアップを試みて、次のエラー・メッセージを受け取ります。

RMAN-3014: リカバリ・カタログの暗黙的再同期に失敗しました  
 RMAN-6038: リカバリ・カタログでパッケージ・エラーが検出されました。  
 RMAN-20035: 高レコード ID が無効です。エラー。

## 無効な RECID エラーでバックアップが失敗する場合：診断

一般的な例として、Oracle 以外のメカニズムで作成されたバックアップ制御ファイルをリストアし、RESETLOGS オプションを指定せずにデータベースをオープンした可能性があります。Recovery Manager の BACKUP コマンドまたは SQL の ALTER DATABASE BACKUP CONTROLFILE 文を使用してバックアップ制御ファイルを作成した場合、Oracle ではオンライン・ログのリセットが必要です。

制御ファイルとリカバリ・カタログは、現在同期化されていません。データベース制御ファイルは、リカバリ・カタログより以前のものです。これは、リカバリ・カタログが古い現行の制御ファイルで再同期化されており、データベースは現在、バックアップ制御ファイルを使用しているためです。Recovery Manager は、現在使用中の制御ファイルが、以前に再同期化で使用した制御ファイルより古いことを検出します。

また、次のようにターゲット・データベースを新しいマシンにコピーする場合にも、問題が発生します。

1. マシン 1 で、データベースを停止し、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制御ファイルのコピーを作成します。この制御ファイルのコピーを、CATALOG を使用せずにリポジトリに追加します。
2. 制御ファイルのコピーをマシン 2 に送信します。
3. マシン 2 で、新規の初期化パラメータ・ファイルと新規のデータベース・インスタンスを作成します。
4. 制御ファイルのコピーをマシン 2 にマウントします。Oracle では、この制御ファイルはバックアップ制御ファイルとして認識されず、現行の制御ファイルとみなされます。
5. Recovery Manager を起動し、マシン 2 の新規ターゲット・データベースおよびリカバリ・カタログに接続します。制御ファイルは Recovery Manager で作成されておらず、制御ファイルのコピーとしてカタログ化されていないため、Recovery Manager ではマシン 2 のデータベースがマシン 1 のデータベースとみなされます。

6. マシン 2 の新規データベースに対してデータベースのリストアおよびリカバリを実行してオープンします。その結果、リストアおよびリカバリ中に各種のレコードがリカバリ・カタログに追加されます。たとえば、リカバリ・カタログの最大の RECID が 90 から 100 になります。
7. マシン 1 で、**Recovery Manager** を起動し、オリジナルのターゲット・データベースおよびリカバリ・カタログに接続します。このリカバリ・カタログは最大の RECID が 100 であることを示しますが、制御ファイルは最大の RECID が 90 であることを示します。制御ファイルの RECID は常にリカバリ・カタログの RECID 以上であることが必要のため、**Recovery Manager** は RMAN-20035 を発行します。

## 無効な RECID エラーでバックアップが失敗する場合：解決策 1

この解決策が最も安全であり、使用することをお勧めします。

### Recovery Manager によるデータベースのリセット

1. SQL\*Plus を使用して、ターゲット・データベースに接続します。たとえば、次のように入力します。
 

```
% sqlplus '/ AS SYSDBA'
```
2. データベースがマウントされていない場合にはマウントします。たとえば、次のように入力します。
 

```
ALTER DATABASE MOUNT;
```
3. バックアップ制御ファイルを使用して取消ベースのリカバリを起動し、これを取り消します。取消しを行う理由は、**USING BACKUP CONTROLFILE** 句が制御ファイルをバックアップとしてスタンプした後で、**OPEN RESETLOGS** を許可するためです。たとえば、次のように入力します。
 

```
ALTER DATABASE RECOVER DATABASE UNTIL CANCEL USING BACKUP CONTROLFILE;
ALTER DATABASE RECOVER CANCEL;
```
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。たとえば、次のように入力します。
 

```
ALTER DATABASE OPEN RESETLOGS;
```
5. **Recovery Manager** を使用して、ターゲット・データベースとリカバリ・カタログに接続します。たとえば、次のように入力します。
 

```
% rman TARGET SYS/oracle@trgt CATALOG rman/cat@catdb
```
6. データベースをリセットします。たとえば、次のように入力します。
 

```
RESET DATABASE;
```

7. 新しいバックアップをとると、必要に応じてデータベースをリカバリできます。たとえば、次のように入力します。

```
BACKUP DATABASE PLUS ARCHIVELOG;
```

## 無効な RECID エラーでバックアップが失敗する場合：解決策 2

この解決策の方が解決策 1 より困難です。

### SQL\*Plus による制御ファイルの作成

1. SQL\*Plus を使用して、ターゲット・データベースに接続します。たとえば、次のように入力します。

```
% sqlplus 'SYS/oracle@trgt AS SYSDBA'
```

2. データベースをマウントします（まだマウントしていない場合）。

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

3. トレース・ファイルに制御ファイルをバックアップします。

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

4. 必要に応じて、トレース・ファイルを編集します。次に、トレース・ファイルの関連セクションの例を示します。

```
# 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 "TRGT" NORESETLOGS ARCHIVELOG
-- STANDBY DATABASE CLUSTER CONSISTENT AND UNPROTECTED
    MAXLOGFILES 32
    MAXLOGMEMBERS 2
    MAXDATAFILES 32
    MAXINSTANCES 1
    MAXLOGHISTORY 226
LOGFILE
GROUP 1 '/oracle/oradata/trgt/redo01.log' SIZE 25M,
GROUP 2 '/oracle/oradata/trgt/redo02.log' SIZE 25M,
GROUP 3 '/oracle/oradata/trgt/redo03.log' SIZE 500K
-- STANDBY LOGFILE
DATAFILE
    '/oracle/oradata/trgt/system01.dbf',
    '/oracle/oradata/trgt/undotbs01.dbf',
    '/oracle/oradata/trgt/cwmlite01.dbf',
    '/oracle/oradata/trgt/drsys01.dbf',
```

```

'/oracle/oradata/trgt/example01.dbf',
'/oracle/oradata/trgt/indx01.dbf',
'/oracle/oradata/trgt/tools01.dbf',
'/oracle/oradata/trgt/users01.dbf'
CHARACTER SET WE8DEC
;
# Take files offline to match current control file.
ALTER DATABASE DATAFILE '/oracle/oradata/trgt/tools01.dbf' OFFLINE;
ALTER DATABASE DATAFILE '/oracle/oradata/trgt/users01.dbf' OFFLINE;
# Configure RMAN configuration record 1
VARIABLE RECNO NUMBER;
EXECUTE :RECNO := SYS.DBMS_BACKUP_RESTORE.SETCONFIG('CHANNEL','DEVICE TYPE DISK
DEBUG 255');
# 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;
# Commands to add tempfiles to temporary tablespaces.
# Online tempfiles have complete space information.
# Other tempfiles may require adjustment.
ALTER TABLESPACE TEMP ADD TEMPFILE '/oracle/oradata/trgt/temp01.dbf' REUSE;
# End of tempfile additions.

```

5. データベースを停止します。

```
SHUTDOWN IMMEDIATE
```

6. スクリプトを実行して、制御ファイルの作成、リカバリ（必要な場合）、ログのアーカイブおよびデータベースのオープンを行います。

```

STARTUP NOMOUNT
CREATE CONTROLFILE ...;
EXECUTE ...;
RECOVER DATABASE
ALTER SYSTEM ARCHIVE LOG CURRENT;
ALTER DATABASE OPEN ...;

```

---

---

**注意：** RESETLOGS オプションを使用してオープンしていない場合、1つのログ順序番号に対して2つのアーカイブ REDO ログのコピーが存在する場合があります（内容がまったく異なっている）。たとえば、1つのログは元のホストで作成され、他の1つのログは新しいホストで作成される場合があります。メディア・リカバリ中に誤ってログが入れ違っていると、データベースの破損につながりますが、Oracle および Recovery Manager ではこの問題を検出できません。

---

---

## 制御ファイルのエンキューが原因でバックアップが失敗する場合：例

この例では、Recovery Manager でスナップショット制御ファイルが作成できないために、バックアップ・ジョブが失敗します。メッセージ・スタックは次のようになります。

```

RMAN-00571: =====
RMAN-00569:===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: backup コマンドが 08/30/2001 22:48:44 で失敗しました
ORA-00230: 処理不可 : スナップショット制御ファイル・エンキューが使用できません。

```

## 制御ファイルのエンキューが原因でバックアップが失敗する場合：診断

Recovery Manager は、制御ファイルのバックアップまたは再同期化が必要な場合、最初に制御ファイルの**スナップショット**または一貫性のあるイメージを作成します。ある Recovery Manager 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は、次のようなメッセージが表示されることがあります。

```
waiting for snapshot controlfile enqueue
```

通常、制御ファイルのエンキューを待つ必要のあるジョブは、わずかの間待機した後に、そのエンキューの取得に成功します。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。

### 競合するエンキューを保持するジョブを確認する手順

1. 最初のメッセージ文字列「RMAN-08512: スナップショット制御ファイル・エンキューを待っています」が表示された後、ターゲット・データベース上で新規の SQL\*Plus セッションを開始します。

```
% sqlplus 'SYS/oracle@trgt AS SYSDBA'
```

2. 次の問合せを実行して、待機しているジョブを確認します。

```
SELECT s.SID, USERNAME AS "User", PROGRAM, MODULE,
       ACTION, LOGON_TIME "Logon", l.*
FROM V$SESSION s, V$ENQUEUE_LOCK l
WHERE l.SID = s.SID
AND l.TYPE = 'CF'
AND l.ID1 = 0
AND l.ID2 = 2;
```

次と同じような出力が表示されます（この出力例は一部分です）。

SID	User	Program	Module	Action	Logon
9	SYS	rman@h13	(TNS V1-V3)	backup full datafile: c1	0000210 STARTED 21-JUN-01

**制御ファイルのエンキューが原因でバックアップが失敗する場合：解決策**

エンキューを作成しているジョブを確認した後で、次のいずれかの処理を実行できます。

- エンキューを作成しているジョブが完了するまで待機します。
- 現行のジョブを取り消し、エンキューを作成しているジョブが完了してから再起動します。
- エンキューを作成しているジョブを取り消します。

通常、エンキュー状態は、ジョブがテープ・ドライブに書込みを行っているとき、テープ・ドライブが新しいカセットの挿入を待っている場合に発生します。この状況で新しいジョブを起動すると、新しいテープが挿入されるまで最初のジョブは完了できないため、エンキュー・メッセージを受け取る場合があります。

**Recovery Manager がすべてのアーカイブ・ログの削除に失敗する場合：例**

この例では、データベースが自動的に ?/oradata/trgt/arch と ?/oradata/trgt/arch2 の 2 つのディレクトリにアーカイブします。次のスクリプトで、バックアップを実行した後に、入力されたアーカイブ REDO ログを削除するように Recovery Manager に指示します。

```
BACKUP ARCHIVELOG ALL DELETE INPUT;
```

クロスチェックを実行して、ログが削除されたことを確認すると、次のメッセージが表示されます。

```
CROSSCHECK ARCHIVELOG ALL;

validation succeeded for archived log
archivelog filename=/oracle/oradata/trgt/arch2/archive1_964.arc recid=19 stamp=368726072
```

Recovery Manager は、1 セットのログを削除しますが、他のセットは削除しません。



## Recovery Manager がすべてのアーカイブ・ログの削除に失敗する場合：診断

この問題はエラーではありません。ALL キーワードを付けずに DELETE INPUT を指定した場合、Recovery Manager は入力ログのうち、1 つのコピーのみを削除します。5 つの宛先にアーカイブを実行した場合でも、Recovery Manager は 1 つのディレクトリからのログのみを削除します。

## Recovery Manager がすべてのアーカイブ・ログの削除に失敗する場合：解決策

存在するすべてのアーカイブ REDO ログを Recovery Manager で強制的に削除するには、BACKUP コマンドの DELETE ALL INPUT 句を使用します。たとえば、次のように入力します。

```
BACKUP ARCHIVELOG ALL DELETE ALL INPUT;
```

## Recovery Manager でアーカイブ・ログが見つからないためにバックアップが失敗する場合：例

この例では、アーカイブ REDO ログの定期的なバックアップをスケジュールします。次回バックアップを行うとき、次のエラーを受け取ります。

RMAN-6089: アーカイブ・ログ NAME が見つからない、またはカタログと同期していません

## Recovery Manager でアーカイブ・ログが見つからないためにバックアップが失敗する場合：診断

この問題は、Recovery Manager が検索しているアーカイブ・ログにアクセスできない場合や、リカバリ・カタログを再同期化する必要がある場合に発生します。多くの場合、このエラーは、オペレーティング・システムのコマンドを使用してアーカイブ・ログを削除したときに発生します。これは、Recovery Manager ではこの削除処理が認識されないためです。リポジトリではログがまだ存在すると示しており、Recovery Manager がバックアップを試みるため、RMAN-6089 エラーが発生します。

## Recovery Manager でアーカイブ・ログが見つからないためにバックアップが失敗する場合：解決策

アーカイブ・ログが指定したディレクトリに存在していること、Recovery Manager のカタログが同期化されていることを確認してください。次の項目をチェックします。

1. RMAN-6089 エラーで特定されたアーカイブ・ログ・ファイルが、正しいディレクトリに存在していることを確認してください。
2. オペレーティング・システムの権限がアーカイブ・ログに対して正しく設定されているかチェックして (owner = oracle, group = DBA)、Recovery Manager がそのファイルにアクセスできることを確認してください。

3. ファイルに問題がない場合は、次のコマンドを Recovery Manager プロンプトから実行して、カタログの同期を行います。

```
RESYNC CATALOG;
```

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ログを削除したために、そのログが使用不可能と認識している場合には、次のコマンドを Recovery Manager プロンプトから実行して、Recovery Manager メタデータを更新します。

```
CROSSCHECK ARCHIVELOG ALL;
```

常に適切なログの削除を行うために、オペレーティング・システムのユーティリティは使用しないで、Recovery Manager を使用してください。最も簡単な方法は、アーカイブ・ログをバックアップするときに DELETE INPUT オプションを指定して、不要なログを削除することです。たとえば、次のように入力します。

```
BACKUP DEVICE TYPE sbt
ARCHIVELOG ALL
DELETE ALL INPUT;
```

## Recovery Manager がターゲット・データベース・キャラクタ・セットを設定できない場合 : 例

この例では、リリース 8.0.4 のターゲット・データベースに接続します。ターゲット・データベースに接続しようとする、次のエラー・メッセージを受け取ります。

```
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
PLS-00201: 識別子 'DBMS_BACKUP_RESTORE.SET_CHARSET' を宣言してください。
ORA-06550: 行 1、列 7: PL/SQL: Statement ignored
RMAN-04015: WE8ISO8859P1 へのターゲット・データベース・キャラクタ・セットの設定でエラーが発生しました。
```

## Recovery Manager がターゲット・データベース・キャラクタ・セットを設定できない場合 : 診断

通常、このエラー・メッセージは、データベースのインストール中に DBMS\_BACKUP\_RESTORE パッケージが作成されなかったことを意味します。次の原因が考えられます。

- インストレーション・スクリプトにエラーが含まれていたため。
- Recovery Manager で必要な PL/SQL オプションがインストールされていないため。

## Recovery Manager がターゲット・データベース・キャラクタ・セットを設定できない場合：解決策

PL/SQL オプションがインストールされていなかった場合は、これをインストールします。PL/SQL オプションがインストールされている場合は、SYSDBA 権限で SQL\*Plus に接続し、次のスクリプトを実行して、必要なパッケージを作成します。

```
SQL> @?/rdbs/admin/dbmsbkrs.sql
SQL> @?/rdbs/admin/prvtbkrs.plb
```

## Recovery Manager がキャラクタ・セット名を識別しない場合：例

この例では、オープンされていないターゲット・データベースに接続して、Recovery Manager 操作を実行しようとしています。この場合、次のエラーを受け取ります。

PLS-00553: キャラクタ・セット名が認識されません。

## Recovery Manager がキャラクタ・セット名を識別しない場合：診断

このメッセージは、通常、Recovery Manager 実行可能ファイルを実行しているクライアント環境のキャラクタ・セットと、ターゲット・データベース環境のキャラクタ・セットが異なることを意味します。

## Recovery Manager がキャラクタ・セット名を識別しない場合：解決策

1. ターゲット・データベースに問合せを行い、NLS\_CHARACTERSET パラメータの値を判断します。たとえば、次の問合せを実行します。

```
SQL> SELECT VALUE FROM V$NLS_PARAMETERS WHERE PARAMETER='NLS_CHARACTERSET';
```

2. クライアントでキャラクタ・セットの環境変数を、サーバーのキャラクタ・セットの環境変数と同じ値に設定します。たとえば、UNIX システムでは、次のように環境変数 NLS\_LANG を設定します。

```
% setenv NLS_LANG american_america.we8dec
% setenv NLS_DATE_FORMAT "MON DD YYYY HH24:MI:SS"
```

## Recovery Manager がターゲット・データベースへのログオンを拒否する場合：例

ターゲット・データベースへの接続を試みると、次のエラーが表示されて Recovery Manager が失敗します。

```
% rman
Recovery Manager: Release 9.2.0.0.0 - Production

RMAN> CONNECT TARGET sys/change_on_install@inst1

RMAN-00571: =====
RMAN-00569:===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
ORA-01017: ユーザー名 / パスワードが無効です。ログオンは拒否されました
```

### 原因の診断

Recovery Manager は自動的に、SYSDBA でターゲット・データベースに接続するよう要求します。SYSDBA でターゲット・データベースに接続するには、次のいずれかの条件が必要です。

- ターゲット・データベースに関して、オペレーティング・システムの DBA グループに属していること（これにより、SYSDBA 権限でのターゲット・データベース接続が可能になり、パスワードが不要になります）。
- パスワード・ファイルを、orapwd コマンドおよび初期化パラメータ REMOTE\_LOGIN\_PASSWORDFILE を使用して作成していること。

ターゲット・データベースにパスワード・ファイルがない場合は、ログインするユーザー名が有効かどうかを、オペレーティング・システム認証を使用して検証する必要があります。

### 解決策

ターゲット・データベースのパスワード・ファイルを作成するか、オペレーティング・システムの管理者リストに自分自身を追加します。

**関連項目：** パスワード・ファイルの作成方法は、『Oracle9i データベース管理者ガイド』を参照してください。

## ログの消失が原因でデータベースの複製が失敗する場合：例

この例では、DUPLICATE コマンドを使用してデータベースを複製しようと試みますが、次のエラー・スタックを受け取ります。

```

RMAN-00571: =====
RMAN-00569:===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: Duplicate Db コマンドが 09/04/2001 12:11:29 で失敗しました
RMAN-03015: ストアド・スクリプト Memory Script にエラーが発生しました
RMAN-06053: ログが見つからないためメディア・リカバリが実行できません
RMAN-06025: ログ・スレッド： 1、seq： 16、scn： 145858 をリストアするためのバックアップが見つかりません。

```

## ログの消失が原因でデータベースの複製が失敗する場合：診断

問題は、Recovery Manager が完全リカバリに必要なすべてのアーカイブ・ログを適用できないことです。たとえば、順序 15 までのログのみをバックアップしていても、最新のアーカイブ・ログが順序 16 の場合、DUPLICATE は失敗します。

## ログの消失が原因でデータベースの複製が失敗する場合：解決策

複製スクリプトの作成時に、SET UNTIL コマンドを使用して、不完全リカバリに対するログ順序番号を指定します。たとえば、ログ順序番号 15 を適用してからリカバリを終了するには、次のように入力します。

```

RUN
{
  SET UNTIL SEQUENCE 16 THREAD 1; # recovers up to but not including log 16
  DUPLICATE TARGET DATABASE TO 'dupdb';
}

```

**関連項目：** 複製操作中の不完全リカバリ実行の詳細は、12-24 ページの「[過去の複製データベースの作成：例](#)」を参照してください。

## 複数の RMAN-06023 エラーが表示されて複製が失敗する場合 : 例

この例では、データベースをバックアップしてから DUPLICATE コマンドを実行します。この場合、次のエラー・スタックを受け取ります。

```
RMAN-00571: =====
RMAN-00569:===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: Duplicate Db コマンドが 09/04/2001 13:55:11 で失敗しました
RMAN-03015: ストアド・スクリプト Memory Script にエラーが発生しました
RMAN-06026: 見つからないターゲットがあります - リストアを中止します
RMAN-06023: データ・ファイル 8 をリストアするためのバックアップまたはコピーが見つかりません
RMAN-06023: データ・ファイル 7 をリストアするためのバックアップまたはコピーが見つかりません
RMAN-06023: データ・ファイル 6 をリストアするためのバックアップまたはコピーが見つかりません
RMAN-06023: データ・ファイル 5 をリストアするためのバックアップまたはコピーが見つかりません
RMAN-06023: データ・ファイル 4 をリストアするためのバックアップまたはコピーが見つかりません
RMAN-06023: データ・ファイル 3 をリストアするためのバックアップまたはコピーが見つかりません
RMAN-06023: データ・ファイル 2 をリストアするためのバックアップまたはコピーが見つかりません
RMAN-06023: データ・ファイル 1 をリストアするためのバックアップまたはコピーが見つかりません
```

### 複数の RMAN-06023 エラーが表示されて複製が失敗する場合 : 診断

DUPLICATE コマンドを実行すると、アーカイブ REDO ログまでリカバリされますが、オンライン REDO ログはリカバリできません。したがって、オンライン REDO ログを適用しなければリストアされたバックアップが一貫した状態にならない場合、Recovery Manager は最新のアーカイブ・ログより前に作成されたバックアップを検索するため、RMAN-06023 エラーが表示されて複製が失敗します。

### 複数の RMAN-06023 エラーが表示されて複製が失敗する場合 : 解決策

バックアップの作成後に次の文を発行します。

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

この文により、オンライン REDO ログ内のすべてのレコードがアーカイブされるため、Recovery Manager は最新のアーカイブ REDO ログを適用してバックアップをリカバリできるようになります。

# リカバリ・カタログに UNKNOWN というデータベース名が表示される場合 : 例

この例では、リカバリ・カタログに登録したデータベースのインカネーションをリスト表示し、UNKNOWN という名前のデータベースを調べます。

```
LIST INCARNATION OF DATABASE;
```

```
RMAN-03022: compiling command:list
List of Database Incarnations
DB Key  Inc Key  DB Name  DB ID      CUR    Reset SCN  Reset Time
-----  -
56      57       TRGT     4052472287 YES     1          Sep 03 2001 06:45:51
1       19       UNKNOWN  4141147584 NO      1          Jan 08 2001 14:47:28
1       2        TRGT     4141147584 YES    14602     Jan 15 2001 15:32:57
```

## リカバリ・カタログに UNKNOWN というデータベース名が表示される場合 : 診断

DB\_NAME に UNKNOWN が表示されるのは、RESETLOGS オプションを使用してすでにオープンしたデータベースに登録するときです。DB\_NAME は RESETLOGS 操作中に変更できるため、Recovery Manager は、データベースの古いインカネーションに対する DB\_NAME が何かを認識しません。これは、その時点でリカバリ・カタログに登録されていないためです。その結果として、Recovery Manager は、DBINC レコードを作成するとき、DB\_NAME 列に UNKNOWN と設定します。

## リカバリ・カタログに UNKNOWN というデータベース名が表示される場合 : 解決策

UNKNOWN という名前のエントリがあることは、RESETLOGS 操作の後では予測される動作です。UNKNOWN のエントリをリカバリ・カタログから削除しないでください。





# 第 IV 部

---

## Recovery Manager のリポジトリの メンテナンス

第 IV 部では、ターゲット・データベースの制御ファイルや、必要に応じてリカバリ・カタログに格納されている Recovery Manager メタデータの、メンテナンスおよび問合せ方法を説明します。この部では、次の章について説明します。

- 第 16 章「Recovery Manager のリポジトリの管理」
- 第 17 章「Recovery Manager のリポジトリの問合せ」
- 第 18 章「Recovery Manager のメンテナンスの実行」



---

## Recovery Manager のリポジトリの管理

この章では、Recovery Manager のリポジトリの管理方法を説明します。Recovery Manager の実装方法によっては、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。この章の項目は、次のとおりです。

- リカバリ・カタログの作成
- リカバリ・カタログ内のターゲット・データベース・レコードの管理
- リカバリ・カタログの再同期化
- リカバリ・カタログに格納された Recovery Manager のスクリプトの管理
- リカバリ・カタログ使用時の制御ファイルの管理
- リカバリ・カタログのバックアップとリカバリ
- リカバリ・カタログのエクスポート
- リカバリ・カタログの可用性の増大
- リカバリ・カタログのスキーマのバージョンの判断
- 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 Recovery Manager リポジトリの管理方法

## リカバリ・カタログの作成

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・データベースの構成](#)
- [リカバリ・カタログ所有者の作成](#)
- [リカバリ・カタログの作成](#)

**関連項目：** Recovery Manager への接続方法は、3-3 ページの「[基本的な Recovery Manager コマンドの使用法](#)」を参照してください。リカバリ・カタログの保持について、その長所と短所は、3-2 ページの「[Recovery Manager 認証方式の選択](#)」を参照してください。

## リカバリ・カタログ・データベースの構成

Recovery Manager でリカバリ・カタログを使用する場合、リカバリ・カタログのスキーマをメンテナンスする必要があります。リカバリ・カタログは、スキーマのデフォルト表領域に格納されます。ただし、SYS はリカバリ・カタログ所有者になることはできないので、注意してください。

リカバリ・カタログのスキーマのインストールに使用するデータベースを決定します。そのデータベースのバックアップ方式も決定してください。カタログは、ターゲット・データベースにはインストールしないでください。インストールすると、カタログの目的がなくなってしまう。カタログ・データベースを、ARCHIVELOG モードで運用するかどうかも決定します。このモードでの運用をお勧めします。

## リカバリ・カタログのスキーマのサイズの計画

カタログのスキーマで使用する領域を割り当てる必要があります。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
- カatalogに格納された Recovery Manager のスクリプトの数とサイズに依存
- 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

たとえば、trgt データベースに 100 ファイルがあり、このデータベースを 1 日に 1 回バックアップして、それぞれ 1 つのバックアップ・ピースを含む 50 のバックアップ・セットを生成するとします。バックアップ・ピースの表の各行で最大量の領域が使用されると想定すると、1 回の日次バックアップによってリカバリ・カタログ内で使用される領域は 170KB です。そこで、この日次バックアップを 1 年間行くと、この期間の合計記憶領域は約 62MB となります。また、アーカイブ・ログの場合もほぼ同量を想定します。したがって、最悪の場合、メタデータの格納には年間ほぼ 120MB が使用されます。通常の場合は、バックアップ・ピースの行領域の一部のみが使用され、年ごとの現実的な見積りは 15MB となります。

リカバリ・カタログ・データベース用のディスク領域の割当て

カタログ・データベースを新規に作成した後や、カタログを格納する既存のデータベースを検出した後に、次の目的のためにディスク領域を割り当てます。

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

リカバリ・カタログ・データベースで使用する領域のほとんどは、SYSTEM、一時、ロールバックまたは UNDO などの表領域のサポートに使用されます。表 16-1 で、一般的な領域要件を説明します。

表 16-1 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	領域要件
SYSTEM 表領域	90MB
一時表領域	5MB
ロールバックまたは UNDO 表領域	5MB
リカバリ・カタログ表領域	15MB
オンライン REDO ログ	各 1MB (3 グループ、各グループに 2 メンバー)

**注意：** 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場合、データベースを消失した場合、リカバリ・カタログも失うことになります。

## リカバリ・カタログ所有者の作成

リカバリ・カタログ・データベースを選択し、必要な領域を作成すると、リカバリ・カタログ所有者の作成と、このユーザーに必要な権限を付与する準備が整います。

次に示すバックグラウンド情報は、次の項の指示に関するものです。

- パスワード `oracle` のユーザー `SYS` には、リカバリ・カタログ・データベース `catdb` の `SYSDBA` 権限があります。
- リカバリ・カタログ・データベース `catdb` には表領域 `tools` があり、この表領域にリカバリ・カタログを格納します。予約語を表領域名として使用する場合、その語を引用符で囲み、大文字フォントを使用する必要があります。
- リカバリ・カタログ・データベースには、`temp` という表領域が存在します。
- このデータベースは、あらゆる標準的なデータベースと同じ方法で構成される必要があります。たとえば、`catalog.sql` と `catproc.sql` が正常に実行されている必要があります。

リカバリ・カタログ・データベースに、リカバリ・カタログのスキーマを作成します。

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、管理者権限で接続します。たとえば、次のように入力します。

```
CONNECT SYS/oracle@catdb AS SYSDBA
```

2. リカバリ・カタログのユーザーとスキーマを作成します。たとえば、次のように入力します。

```
CREATE USER rman IDENTIFIED BY cat
TEMPORARY TABLESPACE temp
DEFAULT TABLESPACE tools
QUOTA UNLIMITED ON tools;
```

3. `RECOVERY_CATALOG_OWNER` ロールを、スキーマの所有者に付与します。このロールは、リカバリ・カタログのメンテナンスと同カタログへの問合せの権限をユーザーに与えます。

```
SQL> GRANT RECOVERY_CATALOG_OWNER TO rman;
```

4. その他の必要な権限を `Recovery Manager` ユーザーに付与します。

```
SQL> GRANT CONNECT, RESOURCE TO rman;
```

## リカバリ・カタログの作成

カタログ所有者を作成後、Recovery Manager の CREATE CATALOG コマンドを実行して、カタログを作成します。このコマンドでは、カタログ所有者のデフォルト表領域にカタログが作成されます。

### リカバリ・カタログの作成

1. カatalogを含むデータベースに、カタログ所有者として接続します。たとえば、オペレーティング・システムのコマンドラインから、次のように入力します。

```
% rman CATALOG rman/cat@catdb
```

Recovery Manager プロンプトから接続することもできます。

```
% rman
RMAN> CONNECT CATALOG rman/cat@catdb
```

2. CREATE CATALOG コマンドを実行して、カタログを作成します。カタログ表領域が、このユーザーのデフォルト表領域の場合、次のコマンドが実行できます。

```
CREATE CATALOG;
```

カタログの作成には時間を要する場合がありますことに注意してください。

3. 必要に応じて SQL\*Plus を起動し、リカバリ・カタログに問い合せて、作成された表を調べます。

```
SQL> SELECT TABLE_NAME FROM USER_TABLES;
```

**関連項目：** GRANT 文と CREATE USER 文の SQL 構文は、『Oracle9i SQL リファレンス』を参照してください。CREATE CATALOG コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## リカバリ・カタログ内のターゲット・データベース・レコードの管理

この項では、ターゲット・データベース・レコードのリカバリ・カタログへの登録方法、登録解除方法およびリセット方法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。

- データベースのリカバリ・カタログへの登録
- リカバリ・カタログからのターゲット・データベースの登録解除
- リカバリ・カタログ内のデータベース・インカネーションのリセット

### データベースのリカバリ・カタログへの登録

Recovery Manager をリカバリ・カタログで使用する前に、ターゲット・データベースをリカバリ・カタログに登録する必要があります。Recovery Manager は、ターゲット・データベースの登録に必要なすべての情報を、データベース自体から取得します。

各ターゲット・データベースに異なる DBID があるかぎり、同じリカバリ・カタログに複数のターゲット・データベースを登録できます。たとえば、ターゲット・データベース prod1、prod2 および prod3 を、カタログのスキーマ rman1 へ登録できます。特定のデータベースを何度も同一のリカバリ・カタログへ登録することはできません。登録は 1 回のみです。ただし、ターゲット・データベースを複数のリカバリ・カタログのスキーマへ登録することは可能です。たとえば、ターゲット・データベース prod1 は、カタログのスキーマ rman1 へも、カタログのスキーマ rman2 へも登録できます。

指定されたカタログへ登録される各データベースには、一意のデータベース識別子 (DBID) が必要です。ただし、データベース名は必ずしも一意である必要はありません。ユーザー管理の方法でデータベースをコピーすると、コピー・データベースの DBID はオリジナル・データベースと同じになります。DBNEWID ユーティリティを使用すると、手動でコピーしたデータベースの DBID とデータベース名を変更できます。ただし、Recovery Manager によって新規 DBID が与えられるため、データベースのコピー操作には DUPLICATE コマンドを使用することをお勧めします。

---

**注意：** Oracle Enterprise Manager を使用する場合は、メンテナンス・ウィザードを使用してデータベースを登録することもできます。

---



## ターゲット・データベースの登録

1. ターゲット・データベースとリカバリ・カタログ・データベースに接続します。たとえば、次のコマンドを発行し、ユーザー `rman` (カタログ・スキーマの所有者) としてカタログ・データベース `catdb` に接続します。

```
% rman TARGET / CATALOG rman/cat@catdb
```

2. ターゲット・データベースがマウントされていない場合は、マウントまたはオープンします。たとえば、次のように発行します。

```
STARTUP MOUNT;
```

リカバリ・カタログ・データベースの状態は、必ずオープンです。

3. ターゲット・データベースで **Recovery Manager** を使用するには、まずデータベースを登録する必要があります。次のコマンドを実行してください。

```
REGISTER DATABASE;
```

**REGISTER DATABASE** の実行後、**Recovery Manager** は、ターゲット・データベースの情報を含むリポジトリに行を作成します。次に、**Recovery Manager** はカタログで完全再同期化を実行します。カタログでは、**Recovery Manager** がターゲット・データベースに関するすべての適切なデータを制御ファイルから転送し、そのデータをカタログに保存します。

4. **REPORT SCHEMA** を実行して、登録が正常終了するかどうかをテストします。このコマンドは、データベースがリポジトリに格納されている場合のデータベース構造を表示します。次に例を示します。

```
RMAN> REPORT SCHEMA;
```

```
Report of database schema
File K-bytes    Tablespace      RB segs Datafile Name
-----
1          307200 SYSTEM          ***    /oracle/oradata/trgt/system01.dbf
2          20480 UNDOTBS          ***    /oracle/oradata/trgt/undotbs01.dbf
3          10240 CWMLITE          ***    /oracle/oradata/trgt/cwmlite01.dbf
4          10240 DRSYS          ***    /oracle/oradata/trgt/drsys01.dbf
5          10240 EXAMPLE          ***    /oracle/oradata/trgt/example01.dbf
6          10240 INDX          ***    /oracle/oradata/trgt/indx01.dbf
7          10240 TOOLS          ***    /oracle/oradata/trgt/tools01.dbf
8          10240 USERS          ***    /oracle/oradata/trgt/users01.dbf
```

5. ユーザーが Oracle リリース 8.0 以上で作成した、データベースのコピーまたはアーカイブ・ログがディスクにある場合は、CATALOG コマンドを使用して、それらをリカバリ・カタログに追加できます。たとえば、次のように入力します。

```
CATALOG DATAFILECOPY '/tmp/users01.dbf';  
CATALOG ARCHIVELOG '/tmp/archive1_731.dbf', '/tmp/archive1_732.dbf';
```

Oracle7 データベースとオペレーティング・システムで作成したデータ・ファイルのコピーは、カタログに追加できる場合があります。使用可能にするために、コピーが Oracle7 の REDO ログを要求してリカバリを行うことがないようにします。つまり、コピーは次のようになります。

- データ・ファイルのコピーは、データベースが一貫してシャットダウンした時に作成されます。移行の前に、データベースを再度オープンしないようにします。
- データ・ファイルのコピーは、表領域が NORMAL モードでオフラインされるか、読取り専用になった後で作成されます。移行の前に、表領域をオンラインにする、または再度読み込み / 書き込みに行わないようにします。

---

**注意：** 制御ファイルのログ・レコードが古いかどうかを判断するには、ディスク上のログ数と、V\$ARCHIVED\_LOG 内のレコード数を比較します。

---

### 関連項目：

- REGISTER コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- Recovery Manager のリストアおよびリカバリの詳細は、『Oracle Enterprise Manager 管理者ガイド』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、『Oracle9i データベース移行ガイド』を参照してください。

## DBID の問題のトラブルシューティング

Oracle では、DBID という一意に生成した内部番号を使用して、データベースを区別しています。DBID は、データベースの作成時に生成されます。Recovery Manager では、DBID がデータベース間の区別に使用されます。CREATE DATABASE 文や DUPLICATE コマンドを使用せず、Recovery Manager 以外の技法（たとえば、ユーザー管理のバックアップおよびリストアなど）でデータベースを作成すると、問題が発生する可能性があります。このような場合、Recovery Manager は重複したデータベース識別子を検出し、REGISTER DATABASE コマンドはエラーになります。

登録済みのデータベースと同じ DBID を持つコピーしたデータベースを登録できるようにするには、DBNEWID を使用してコピー・データベースの DBID を変更します。また、このユーティリティを使用すると、データベース名を変更できます。ただし、最善の解決策は、DUPLICATE コマンドを使用して問題をまとめて回避することです。このコマンドを実行すると、新規データベース識別子が自動的に生成されます。

### 関連項目：

- CATALOG コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- DUPLICATE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- DBNEWID ユーティリティを使用して DBID を変更する方法は、『Oracle9i データベース・ユーティリティ』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、『Oracle9i データベース移行ガイド』を参照してください。

## リカバリ・カタログからのターゲット・データベースの登録解除

Recovery Manager では、データベースの登録と、登録解除が実行できます。この処理は、必要かどうかを確認して行ってください。この処理を誤ると、データベースの再登録が必要になります。この場合、ターゲット・データベースの制御ファイルで設定した CONTROLFILE\_RECORD\_KEEP\_TIME より古いメタデータは、すべて失われます。

### データベースの登録解除

1. Recovery Manager を起動し、ターゲット・データベースに接続します。Recovery Manager を使用してターゲット・データベースに接続するとき、DBID の値が表示されます。この値は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 / CATALOG rman/cat@catdb

connected to target database: RDBMS (DBID=1237603294)
connected to recovery catalog database
```

2. リポジトリに記録されているコピーおよびバックアップ・セットをリスト表示します (17-3 ページの「[Recovery Manager のバックアップ、コピーおよびデータベース・インカネーションのリスト表示](#)」を参照)。たとえば、次のように入力します。

```
LIST BACKUP SUMMARY;
```

List of Backups  
=====

Key	TY	LV	S	Device	Type	Completion Time	#Pieces	#Copies	Tag
19	B	A	A	DISK		08-FEB-02	1	1	TAG20020208T155239
20	B	F	A	DISK		08-FEB-02	1	1	TAG20020208T155242
21	B	A	A	DISK		08-FEB-02	1	1	TAG20020208T155331
22	B	A	A	DISK		08-FEB-02	1	1	TAG20020208T155604

3. DELETE 文を実行して、既存の物理バックアップをすべて削除します (18-5 ページの「[バックアップおよびコピーの削除](#)」を参照)。たとえば、次のように入力します。

```
DELETE BACKUP DEVICE TYPE sbt;  
DELETE BACKUP DEVICE TYPE DISK;
```

Recovery Manager は削除するバックアップをリストします。また、バックアップを削除する前に、確認のプロンプトを表示します。

4. 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、そのリカバリ・カタログで次の問合せを実行して、DB 表の正しい行を検索します。DB\_ID の値は手順 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。

```
% sqlplus rman/cat@catdb  
SQL> SELECT DB_KEY, DB_ID FROM DB WHERE DB_ID = 1237603294;
```

この問合せで戻される行は 1 行です。

DB_KEY	DB_ID
1	1237603294

1 row selected.

5. リカバリ・カタログに接続した状態で、次のように入力します。DB\_KEY と DB\_ID は、手順 4 の問合せで得られた行に対応する列です。

```
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)
```

## リカバリ・カタログ内のデータベース・インカネーションのリセット

Recovery Manager コマンドまたは SQL 文 `ALTER DATABASE OPEN RESETLOGS` を実行すると、データベースの新規インカネーションが作成されます。新規インカネーションのレコードは、ターゲット・データベースの `V$DATABASE_INCARNATION` ビューで確認できます。

Recovery Manager コマンド (SQL 文ではない) の `ALTER DATABASE OPEN RESETLOGS` を実行すると、Recovery Manager は自動的に、新規のデータベース・インカネーション・レコードをリカバリ・カタログに作成します。Recovery Manager では、`RESET DATABASE` コマンドが暗黙的かつ自動的に発行され、データベースのこの新規インカネーションが現行のインカネーションであることが指定されます。Recovery Manager は、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とログ・アーカイブを新しいデータベース・インカネーションに関連付けます。

SQL 文 (Recovery Manager コマンドではない) の `ALTER DATABASE OPEN RESETLOGS` 文を発行しても、Recovery Manager で `RESET DATABASE` コマンドが自動的に実行されることはありません。Recovery Manager はリカバリ・カタログにアクセスできません。これは、Recovery Manager には `RESETLOGS` コマンドと、古い制御ファイルが誤ってリストアされた場合との区別ができないためです。この問題を解決するためには、SQL 文の `ALTER DATABASE OPEN RESETLOGS` を実行した後、`RESET DATABASE` コマンドを Recovery Manager で手動で実行する必要があります。RESET DATABASE コマンドはリポジトリを更新して、ターゲット・データベースが `RESETLOGS` オプションでオープンされたことを示します。

データベースの前のインカネーションのバックアップをリストアする場合は、`RESET DATABASE TO INCARNATION key` コマンドを使用して、現行のインカネーションを古いインカネーションに変更します。たとえば、最後の `RESETLOGS` の直後に誤って表を削除した場合は、データベースを最後の `RESETLOGS` の直前の状態までリカバリしてから、`RESETLOGS` オプションを使用してオープンします。これにより、新規インカネーションが作成されます。

---

**注意：** Oracle Enterprise Manager を使用する場合は、メンテナンス・ウィザードを使用してデータベース・インカネーションをリセットすることもできます。

---

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の主キーを指定します。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       | TRGT    | 1224038686 | NO  | 1         | 02-JUL-01  |
| 1      | 582     | TRGT    | 1224038686 | YES | 59727     | 10-JUL-01  |
2. データベースを古いインカーネーションにリセットします。たとえば、次のように入力します。
- RESET DATABASE TO INCARNATION 2;
3. データベースをシャットダウンし、マウントしないで起動します。たとえば、次のように入力します。
- SHUTDOWN IMMEDIATE
- STARTUP NOMOUNT
4. 制御ファイルを古いインカーネーションからリストアします。タグ付きの制御ファイルがある場合は、タグを指定します。その他の場合は、次の例のように、SET UNTIL コマンドが実行できます。
- RUN
- {
- SET UNTIL 'SYSDATE-45';
- RESTORE CONTROLFILE;
- }
5. 制御ファイルをマウントします。
- ALTER DATABASE MOUNT;
6. RESTORE コマンドと RECOVER コマンドを実行して、前のインカーネーションからデータベース・ファイルをリストアし、リカバリします。次に、そのデータベースを RESETLOGS オプションでオープンします。たとえば、次のように入力します。
- RESTORE DATABASE;
- RECOVER DATABASE;
- ALTER DATABASE OPEN RESETLOGS;

**関連項目：**

- RESET DATABASE コマンドの構文および LIST コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。  
『Oracle9i Recovery Manager リファレンス』
- メンテナンス・ウィザードの詳細は、『Oracle Enterprise Manager 管理者ガイド』を参照してください。

## リカバリ・カタログの再同期化

再同期化を実行する場合、Recovery Manager はリカバリ・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バックアップ制御ファイルのいずれかと比較し、欠落または変更された情報に基づいて、リカバリ・カタログを更新します。

この項では次の項目について説明します。

- [再同期化](#)
- [再同期化されるレコードのタイプ](#)
- [再同期化を実行する時期](#)
- [リカバリ・カタログの完全再同期化の強制実行](#)
- [CONTROLFILE\\_RECORD\\_KEEP\\_TIME 値の設定](#)

## 再同期化

再同期化は、完全でも部分でも実行できます。部分再同期化の場合、Recovery Manager は現行の制御ファイルを読み取って、新しいバックアップや新しいアーカイブ・ログなどに関する変更された情報を更新します。ただし、Recovery Manager はデータベースの物理スキーマに関するメタデータは再同期化しません。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。一方、完全再同期化の場合、Recovery Manager 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：** 制御ファイルがバックアップの場合、Recovery Manager は、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。

---

---

再同期化の実行時に、Recovery Manager は次の処理を行います。

1. スナップショット制御ファイルを作成します。
2. リカバリ・カタログをスナップショットと比較します。
3. 欠落または変更された情報に基づいてリカバリ・カタログを更新します。

特定のコマンド（BACKUP および COPY を含む）を実行すると、Recovery Manager 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。このため、RESYNC CATALOG を頻繁に実行する必要はありません。これは、BACKUP および COPY コマンドが、カタログを自動的に再同期化するからです。ただし、バックアップはほとんど NOCATALOG モードで実行し、カタログへの接続は定期的のみに行う、といった場合もあります。この場合、完全再同期化を確実に実行するには、RESYNC CATALOG コマンドを実行します。

---

---

**注意：** Oracle Enterprise Manager を使用する場合は、メンテナンス・ウィザードを使用してカタログの再同期化を実行することもできます。

---

---

### 関連項目：

- RESYNC コマンドの詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- メンテナンス・ウィザードの詳細は、『Oracle Enterprise Manager 管理者ガイド』を参照してください。



## 再同期化されるレコードのタイプ

Recovery Manager の再同期化の対象となるレコードのタイプを表 16-2 で説明します。

**表 16-2 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・スイッチの発生時に作成されます。
アーカイブ REDO ログ	オンライン・ログのアーカイブ、既存のアーカイブ・ログのコピーまたはアーカイブ REDO ログのバックアップ・セットのリストアによって作成されたアーカイブ・ログに関連したレコードです。Recovery Manager は、この情報を追跡して、どのアーカイブ・ログの検出が必要かを判断します。
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です。RESYNC CATALOG コマンドは、BACKUP コマンドまたは COPY コマンドの実行時に、これらのレコードを更新します。
物理スキーマ	<p>データ・ファイルと表領域に関連したレコードです。ターゲット・データベースがオープン状態の場合は、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、新しく作成された制御ファイル、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、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場合も、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

Recovery Manager 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。ほとんどの Recovery Manager コマンド、たとえば BACKUP、COPY、DELETE などは、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ていて、リカバリ・カタログ・データベースが使用可能な場合、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を自動的にを行います。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・カタログが使用不可能な場合の再同期化

リカバリ・カタログが使用不可能な状況で、部分再同期化を実行する Recovery Manager コマンドを発行した場合は、後からカタログ・データベースをオープンし、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。

たとえば、ターゲット・データベースがニューヨークにあり、リカバリ・カタログ・データベースが日本にあると想定します。この場合、毎日 CATALOG モードでターゲット・データベースのバックアップは行いません。これは、地理的に離れたデータベースの可用性に依存するからです。1つのソリューションでは、週1回カタログに接続し、RESYNC CATALOG コマンドを実行します。

### バックアップ頻度が低い場合の ARCHIVELOG モードでの再同期化

次の実行例を想定します。

- ARCHIVELOG モードでデータベースを実行
- データベースを低頻度で（たとえば、データベースのバックアップ間で 1,000 ログをアーカイブする）バックアップ
- 数多くのログ・スイッチ（たとえば、カタログの同期化間で 1,000 スイッチ）を毎日生成

この場合、リカバリ・カタログは定期的に、手動で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、自動的に更新されません。ただし、ログ・スイッチとアーカイブ REDO ログに関する情報は制御ファイルに格納されます。この情報は、定期的にリカバリ・カタログに伝播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、REDO ログがアーカイブされる頻度によって決まります。この運用コストは、前回の再同期化以降に挿入または変更された制御ファイル内の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ない場合、再同期化のコストは非常に低くなり、多数のレコードが挿入または変更された場合は再同期化の所要時間が長くなります。

## 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更新されます。

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

## リカバリ・カタログの完全再同期化の強制実行

RESYNC CATALOG コマンドを発行して、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. SQL\*Plus でリカバリ・カタログ・データベースに接続し、まだデータベースをオープンしていない場合はオープンします。たとえば、次のように入力します。

```
STARTUP
```

2. Recovery Manager からターゲットおよびリカバリ・カタログ・データベースへの接続後に、ターゲット・データベースがマウントされていない場合は、マウントします。たとえば、次のように入力します。

```
STARTUP MOUNT;
```

3. RESYNC CATALOG コマンドを、Recovery Manager プロンプトで実行します。

```
RESYNC CATALOG;
```

**関連項目：** RESYNC CATALOG コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## CONTROLFILE\_RECORD\_KEEP\_TIME 値の設定

リカバリ・カタログをメンテナンスする場合は、Recovery Manager の RESYNC CATALOG コマンドを使用して、制御ファイル・レコードが再使用される前にカタログを再同期化します。この方法により、レコードは制御ファイルからリカバリ・カタログに伝播されるため、Recovery Manager でのレコード消失はありません。Recovery Manager では、バックアップとコピーの作成時にカタログが自動的に再同期化されることに注意してください。

制御ファイル・レコードがリカバリ・カタログへの追加前に上書きされないように、CONTROLFILE\_RECORD\_KEEP\_TIME を、バックアップまたは再同期化の間隔より少し低い値（1 週間など）に設定します。たとえば、各月の第 1 週と第 3 週にデータベースをバックアップし、第 2 週と第 4 週に RESYNC CATALOG を実行する場合は、CONTROLFILE\_RECORD\_KEEP\_TIME を 10 または 14 に設定することを検討してください。

**関連項目：** 16-37 ページ「[制御ファイル・レコードの上書きの監視](#)」

## リカバリ・カタログに格納された Recovery Manager のスクリプトの管理

この項では、次の項目について説明します。

- [リカバリ・カタログのストアド・スクリプト](#)
- [リカバリ・カタログへのスクリプトの格納](#)
- [ストアド・スクリプトの実行](#)
- [ストアド・スクリプトの置換](#)
- [ストアド・スクリプトの削除](#)

**関連項目：** Recovery Manager のストアド・スクリプトの問合せ方法は、17-24 ページの「[リカバリ・カタログに格納されたスクリプトの出力](#)」を参照してください。

## リカバリ・カタログのストアド・スクリプト

ストアド・スクリプトとは、リカバリ・カタログに格納されている一連の Recovery Manager コマンドのことです。ストアド・スクリプトは、頻繁に実行する Recovery Manager コマンドの、共通リポジトリを提供します。

たとえば、毎晩バックアップを実行するために必要な Recovery Manager コマンドを、`nightly_bkup` という名前の、1 つのスクリプトにまとめることができます。スクリプトを、オペレーティング・システムのテキスト・ファイルではなく、リカバリ・カタログに格納しておくと、Recovery Manager がどのマシン上で実行されていても、DBA は Recovery Manager を使用してスクリプトにアクセスできるので便利です。

リカバリ・カタログがある場合は、Recovery Manager を使用して次のことができます。

- スクリプトを作成し、リカバリ・カタログに格納
- ストアド・スクリプトを実行
- 既存のストアド・スクリプトを置換
- ストアド・スクリプトをリカバリ・カタログから削除
- ストアド・スクリプトの内容をメッセージ・ログ・ファイルまたは画面へ出力
- 全ストアド・スクリプトのリストを取得（17-24 ページの「[リカバリ・カタログに格納されたスクリプトの出力](#)」を参照）

ストアド・スクリプトの作成については、次の重要な側面に注意してください。

- スクリプト・コマンドの実行時には、リカバリ・カタログに接続する必要があります。
- スクリプト・コマンドの実行時には、ターゲット・データベースに接続する必要があります。スクリプトは単一のターゲット・データベース専用で作成されるため、スクリプトを置換、出力または削除するときには、同じターゲット・データベースに接続する必要があります。

## リカバリ・カタログへのスクリプトの格納

CREATE SCRIPT コマンドを使用して Recovery Manager のスクリプトを作成し、リカバリ・カタログに格納します。

### ストアド・スクリプトの作成

1. Recovery Manager をターゲット・データベースとリカバリ・カタログに接続後、CREATE SCRIPT コマンドを使用して、必要なスクリプトを構成します。たとえば、このスクリプトは、データベースとアーカイブ REDO ログをバックアップします。

```
CREATE SCRIPT b_whole
{
    BACKUP DATABASE PLUS ARCHIVELOG;
    DELETE OBSOLETE;
}
```

2. 出力をチェックします。RMAN-08085 というメッセージが表示された場合、スクリプトの作成およびリカバリ・カタログへの格納は成功です。

```
created script b_whole
```

このスクリプトは、その作成時に接続していたターゲット・データベースに関連付けられていることに注意してください。

## ストアド・スクリプトの実行

EXECUTE SCRIPT コマンドを使用して、リカバリ・カタログに格納した Recovery Manager のスクリプトを実行します。

### ストアド・スクリプトの実行

リカバリ・カタログおよびターゲット・データベースへの接続後、RUN コマンドを発行して、必要なスクリプトを実行します。たとえば、次のように入力します。

```
RUN { EXECUTE SCRIPT b_whole; }
```

Recovery Manager はスクリプトの内容を、RUN の大カッコの間に挿入します。スクリプト内でのチャンネル割当てがすでに終了している場合や、自動チャンネルの構成が行われている場合には、ALLOCATE CHANNEL を実行する必要はありませんので、注意してください。

**関連項目：** EXECUTE SCRIPT コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ストアド・スクリプトの置換

REPLACE SCRIPT コマンドを使用して、カタログのスクリプトをリライトします。スクリプトが存在しない場合は、Recovery Manager によって作成されます。REPLACE SCRIPT を使用してスクリプトの 1 行を置換することはできないことに注意してください。すべての行をリライトしてください。

### ストアド・スクリプトの置換

Recovery Manager をリカバリ・カタログに接続後、REPLACE SCRIPT コマンドを発行して、ストアド・スクリプトを置換します。たとえば、このコマンドでは、スクリプト b\_whole が次のように置換されます。

```
REPLACE SCRIPT b_whole
{
    BACKUP DATABASE PLUS ARCHIVELOG;
}
```

**関連項目：** REPLACE SCRIPT コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## ストアド・スクリプトの削除

DELETE SCRIPT コマンドを使用して、リカバリ・カタログのスクリプトをリライトします。

### ストアド・スクリプトの削除

Recovery Manager をリカバリ・カタログに接続後、DELETE SCRIPT コマンドを発行して、ストアド・スクリプトを削除します。

```
DELETE SCRIPT 'b_whole';
```

**関連項目：** DELETE SCRIPT コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## リカバリ・カタログ使用時の制御ファイルの管理

目標は、リカバリ・カタログ内のメタデータを常に現行の状態に保つことです。リカバリ・カタログはメタデータをターゲット制御ファイルから取得するため、カタログ内のデータが最新の状態かどうかは、制御ファイル内のデータが最新の状態かどうかによって左右されます。制御ファイル内のバックアップのメタデータが新規レコードで上書きされる前に、カタログに記録されていることを確認する必要があります。

制御ファイル内のレコードが上書き候補となる前に保持される最小日数は、`CONTROL_FILE_RECORD_KEEP_TIME` 初期化パラメータによって決定されます。そのため、制御ファイルのレコードが消去される前に、これらのレコードとリカバリ・カタログを再同期化する必要があります。16-18 ページの「[CONTROLFILE\\_RECORD\\_KEEP\\_TIME 値の設定](#)」のように、次のいずれかのアクションを `CONTROL_FILE_RECORD_KEEP_TIME` 設定より短い間隔で実行してください。

- バックアップまたはコピーを作成し、リカバリ・カタログの暗黙的な再同期化を実行
- `RESYNC CATALOG` コマンドを使用してリカバリ・カタログを手動で再同期化

したがって、リカバリ・カタログの情報が最新のものであることを確実にするには、再同期化の頻度を `CONTROL_FILE_RECORD_KEEP_TIME` 初期化パラメータの値に関連付ける必要があります。

制御ファイルが大きくなりすぎると、1 つ問題が発生する可能性があります。ターゲット・データベースの制御ファイルのサイズは、次に項目の数に応じて増加します。

- 実行するバックアップの回数
- 生成されるアーカイブ REDO ログの数
- この情報を制御ファイル内に格納しておく日数

16-37 ページの「[制御ファイル・レコードの上書きの監視](#)」のように、制御ファイルが大きくなりすぎて、最大ブロック数または最大レコード数に達したために拡張できなくなった場合、Oracle は期間が `CONTROL_FILE_RECORD_KEEP_TIME` 設定より短い場合も、最も古いレコードを上書きできます。この場合は、アラート・ログにメッセージが書き込まれます。この状況が頻繁に発生していることが検出された場合は、`CONTROL_FILE_RECORD_KEEP_TIME` の値を小さくし、再同期化の頻度を大きくしてください。

---

---

**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。通常、最大サイズは 20,000 Oracle ブロックです。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。

---

---



**関連項目：**

- `CONTROL_FILE_RECORD_KEEP_TIME` パラメータの詳細は、『Oracle9i データベース・リファレンス』を参照してください。
- バックアップおよびリカバリ時の制御ファイルの管理方法は、『Oracle9i ユーザー管理バックアップおよびリカバリ・ガイド』を参照してください。
- 制御ファイル管理の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

## リカバリ・カタログのバックアップとリカバリ

バックアップおよびリカバリ計画には、リカバリ・カタログを含めてください。リカバリ・カタログをバックアップしないと、リカバリ・カタログ・データベースが破損するようなディスク障害が発生した場合、カタログ内のメタデータを失う可能性があります。このような結果を避けるため、リカバリ・カタログのバックアップ方法とリカバリ方法を決定します。

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・カタログのリカバリ](#)
- [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てる場合は、一般的なガイドラインに従ってください。この項では、次の項目について説明します。

- [定期的なバックアップの励行](#)
- [物理バックアップに適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](#)

### 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てください。たとえば、ターゲット・データベースに対してデータベース全体のバックアップを週 1 回行っている場合は、すべてのターゲット・データベースをバックアップした直後に、リカバリ・カタログもバックアップしてください。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録されています。したがって、カタログのリストアが必要な場合は、これを使用してターゲット・バックアップもリストアできます。

### 物理バックアップに適したメソッドの選択

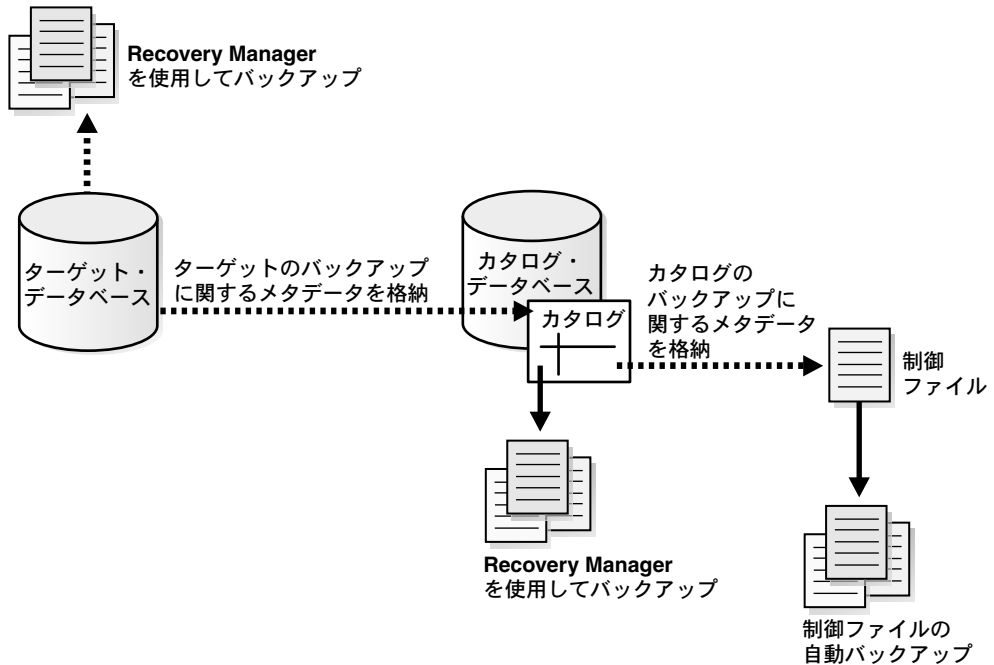
リカバリ・カタログ・データベースのバックアップ時には、Recovery Manager を使用したバックアップが可能です。[図 16-1](#) に示すように、Recovery Manager は NOCATALOG オプションを使用して起動します。これにより、リカバリ・カタログのリポジトリは、カタログ・データベース内の制御ファイルになります。

リカバリ・カタログ・データベースの、Recovery Manager を使用したバックアップ計画を立てる場合は、次のガイドラインに従ってください。

- リカバリ・カタログ・データベースを ARCHIVELOG モードで実行します。これにより、必要に応じて Point-in-Time リカバリを行うことができます。
- 保存方針を REDUNDANCY の値に設定します。2 以上に設定します。
- データベースのバックアップを 2 つの別メディア（たとえば、ディスクとテープ）に作成します。バックアップの作成時には、BACKUP COPIES 2 が指定できます。
- 一定の間隔で BACKUP DATABASE PLUS ARCHIVELOG を、使用可能な場合はメディア・マネージャへ、または単にディスクへ実行します。
- 他のリカバリ・カタログを、バックアップ用リポジトリとして使用しないでください。
- 制御ファイルの自動バックアップ機能を ON に構成します。

この方法により、制御ファイルの自動バックアップ機能で、リカバリ・カタログ・データベースは、いつでも確実にリカバリできるようになります。

図 16-1 制御ファイルをカタログのバックアップ用リポジトリとして使用する方法



**関連項目：** SQL 文を使用した制御ファイルのバックアップ方法は、『Oracle9i ユーザー管理バックアップおよびリカバリ・ガイド』を参照してください。制御ファイルの自動バックアップによるリカバリの詳細は、10-37 ページの「障害時リカバリの実行」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 Recovery Manager リポジトリを含んだリカバリ・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や、ターゲット・データベースと同じディスク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` には格納できません。`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` がメディア全体障害に陥り、`prod1` のリカバリ・カタログも `prod1` に格納されていた場合は、リカバリに使用するカタログが失われます。`prod1` に対して、制御ファイルの自動バックアップをリストアし、これをデータベースのリストアおよびリカバリに使用する必要があります。

ターゲットとカタログ・データベースの分割は、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、特に重要です。次に、禁止項目の事例を示します。たとえば、次の例を考えてみます。

- ターゲット・データベース `prod1` とカタログ・データベース `catdb` は、別のホストにあります。
- `catdb` には、ターゲット・データベース `prod1` のリカバリ・カタログ・リポジトリが含まれています。

カタログを使用して `catdb` をバックアップすると決定し、カタログの作成場所は決まっていない状況だとします。`catdb` のリポジトリを含むカタログを、データベース `catdb` に作成した場合、メディア障害が原因で `catdb` を失うと、`catdb` のリストアが困難になり、結局はカタログのないまま、`prod1` をリストアに使用することになります。したがって、ターゲット・データベースのリカバリ・カタログは、絶対にターゲット・データベース自体には格納しないでください。格納した場合、リカバリ・カタログの目的が完全に失われます。

### リカバリ・カタログのリカバリ

カタログのリカバリに Recovery Manager を使用する場合、カタログのバックアップのメタデータを格納する場所によって、手順が異なります。10-5 ページの「[基本的な Recovery Manager メディア・リカバリの実行](#)」にある、リカバリのオプションに関する解説の表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・データベースが消失または破損し、そのリカバリ・カタログ・データベースが Oracle の標準的なリカバリ手順でリカバリできない場合は、カタログの再作成を実行する必要があります。この最悪の状態の例には、次のようなものがあります。

- バックアップされたことのないカタログ・データベース
- バックアップが行われていても、データ・ファイルまたはアーカイブ・ログが使用不可能なため、リカバリ不可能なカタログ・データベース

欠落したカタログの内容を部分的に再作成するには、2つのオプションがあります。

- CATALOG コマンドを発行して、アーカイブ REDO ログ、バックアップ制御ファイルおよびデータ・ファイルのコピーを再カタログ化します。CATALOG コマンドは、バックアップ・ピースまたはバックアップ・セットの再カタログ化をサポートしていないため、RESYNC CATALOG コマンドの使用によってのみバックアップ・セットに関する情報を再作成できます。
- RESYNC CATALOG コマンドを使用して、制御ファイルからメタデータを抽出し、リカバリ・カタログを再作成します。古い制御ファイルのレコードに含まれていたメタデータは自動的に失われますので、注意してください。

ターゲットの制御ファイルの状態により、次の方法が可能です。

- 現行の制御ファイルから再同期化
- 制御ファイルのコピーから再同期化

### 関連項目：

- CATALOG コマンドの詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- CROSSCHECK コマンドの詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- 制御ファイルへの情報の記録方法については、16-22 ページの「[リカバリ・カタログ使用時の制御ファイルの管理](#)」を参照してください。

## リカバリ・カタログのエクスポート

リカバリ・カタログを、あるデータベースから他のデータベースへ移動するには、インポートまたはエクスポート・ユーティリティを実行します。カタログは、サポートされているバージョンの Oracle データベース・サーバーにのみ、インポートできます。通常、カタログは同じリリース以上のデータベースにインポートできます。

この項では次の項目について説明します。

- [カタログ・データ移動時の考慮事項](#)
- [リカバリ・カタログのエクスポート](#)
- [リカバリ・カタログのインポート](#)

## カタログ・データ移動時の考慮事項

リカバリ・カタログのスキーマを含まないスキーマには、リカバリ・カタログのみをインポートします。つまり、すでにカタログのスキーマを所有しているユーザーは、インポートされたリカバリ・カタログのスキーマを所有することはできません。たとえば、ユーザー `rman` がデータベース `catdb` にリカバリ・カタログを所有しており、`catdb` のリカバリ・カタログをエクスポートしてデータベース `catdb2` にインポートする場合、`rman` は `catdb2` に、事前にリカバリ・カタログを所有していることはできません。新しいリカバリ・カタログ所有者を `catdb2` に作成するか、現行の `rman` を `catdb2` から削除して、ユーザーを再作成してください。リカバリ・カタログと既存のリカバリ・カタログはマージできません。

カタログをプライマリ・データベースからエクスポートし、2 次データベースにインポートする基本手順は、次のとおりです。

1. エクスポート・ユーティリティを使用して、カタログ・データをプライマリ・データベースからエクスポートします。16-29 ページの「[リカバリ・カタログのエクスポート](#)」を参照してください。
2. 2 次データベースにユーザーを作成し（16-4 ページの「[リカバリ・カタログ所有者の作成](#)」を参照）、必要な権限をユーザーに付与します。
3. インポート・ユーティリティを使用して、カタログ・データを前の手順で作成したスキーマにインポートします。16-29 ページの「[リカバリ・カタログのインポート](#)」を参照してください。

カタログを 2 次データベースにインポートする前後には、`CREATE CATALOG` コマンドを実行しないでください。カタログ・データを新スキーマにインポートすることにより、カタログを効果的に、2 次データベースに作成できます。

## リカバリ・カタログのエクスポート

エクスポート・ユーティリティに関する概念および手順については、『Oracle9i データベース・ユーティリティ』を参照してください。

### コマンドラインからリカバリ・カタログの論理エクスポートを実行する方法

1. オペレーティング・システムのコマンドラインでエクスポート・ユーティリティを実行します。その際、次の処理を必ず実行してください。
  - a. リカバリ・カタログ所有者として接続
  - b. OWNER オプションを指定
  - c. 出力ファイルを指定

たとえば、データベース `catdb` のカタログ所有者が `rman` の場合、UNIX のコマンドラインで次のコマンドを発行して、カタログをファイル `cat.dmp` にエクスポートできます。

```
% exp rman/cat@catdb FILE=cat.dmp OWNER=rman
```

2. 出力をチェックし、成功したことを確認します。

```
Export terminated successfully without warnings.
```

## リカバリ・カタログのインポート

エクスポート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、インポート・ユーティリティを使用して、他のデータベースへインポートします。

### コマンドラインからリカバリ・カタログの論理インポートを実行する方法

1. 別のデータベースに新規ユーザーを作成します。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 SQL の構文は、16-2 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・データをエクスポート・ファイルからインポートします。コマンドラインでインポートを実行します。その際、次の処理を必ず実行してください。
  - a. リカバリ・カタログの新規所有者で接続します。
  - b. FROMUSER パラメータで、旧所有者を指定します。
  - c. TOUSER パラメータで、新規所有者を指定します。
  - d. インポート・ファイルを指定します。

たとえば、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を rman とします。
- 新規リカバリ・カタログ・データベース catdb2 内のユーザーは、rman2 とします。
- カatalogのエクスポートが含まれたファイルは、cat.dmp とします。

コマンドは次のようになります。

```
% imp USERID=rman2/cat2@catdb2 FILE=cat.dmp FROMUSER=rman TOUSER=rman2
```

3. インポートしたカタログ・データを使用して、ターゲット・データベースのリストアとリカバリを行います。

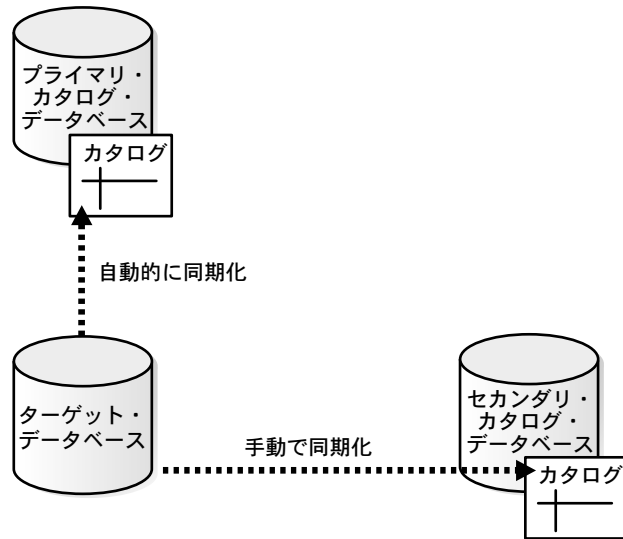
## リカバリ・カタログの可用性の増大

カタログ・データベースの高可用性を維持する、本番システムを持っている場合があります。たとえば、100 のターゲット・データベースがリカバリ・カタログに登録されている場合です。プライマリ・データベースがダウンした場合、2 番目のリカバリ・カタログを別のデータベースに格納することによって、冗長性を作成できます。これを図 16-2 に示します。ターゲット・データベースは、2 番目のカタログに登録する必要があります。

この可用性の例では、主カタログは定期バックアップの間、通常どおり同期化されていますが、2 番目のカタログは RESYNC CATALOG コマンドで、定期的に同期化されます。主カタログ・データベースがダウンする、またはルーチンのメンテナンスを必要とした場合、2 番目のカタログを再同期化し、これを新規の主カタログとして暫定的に使用できます。



図 16-2 1つのデータベースを2つのリカバリ・カタログに登録する方法



関連項目： 16-2 ページ「[リカバリ・カタログの作成](#)」

## リカバリ・カタログのスキーマのバージョンの判断

リカバリ・カタログのスキーマ・バージョンは、リカバリ・カタログ自体に格納されます。この情報が重要になるのは、本番システムにバージョンの異なる複数のデータベースがあり、カタログのスキーマのバージョンを特定のターゲット・データベース・バージョンで使用するかどうかを判断する必要がある場合です。

**関連項目：** Recovery Manager 環境を管理する互換性ルールの詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。

### リカバリ・カタログのスキーマのバージョンの判別

1. SQL\*Plus を起動し、カタログ所有者としてリカバリ・カタログ・データベースに接続します。たとえば、次のように入力します。

```
% sqlplus rman/cat@catdb
```

2. RCVER 表を問い合せて、次の例（出力例も含む）のようにスキーマのバージョンを取得します。

```
SELECT *  
FROM RCVER;  
  
VERSION  
-----  
09.02.00
```

RCVER 表に複数の行が表示される場合は、この表で最上位のバージョンが現行のカタログ・スキーマのバージョンです。この表に格納されるのはバージョン番号のみで、パッチ番号は格納されません。たとえば、RCVER 表に次の行が表示されるとします。

```
VERSION  
-----  
08.01.07  
09.00.01  
09.02.00
```

これらの行は、カタログがリリース 8.1.7 の実行可能ファイルで作成された後、リリース 1 (9.0.1) にアップグレードされ、最後にリリース 2 (9.2) にアップグレードされたことを示しています。カタログのスキーマの現行バージョンは 2 (9.2) です。

## リカバリ・カタログのアップグレード

Recovery Manager の実行可能ファイルが必要とするバージョンより古いリカバリ・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、リリース 8.1 の Recovery Manager 実行可能ファイルで、リリース 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します。

Recovery Manager の実行可能ファイルが必要とするバージョンより新しいバージョンのリカバリ・カタログを使用している場合は、UPGRADE CATALOG を発行するとエラーが表示されます。リカバリ・カタログが現行のバージョンで、アップグレードが不要な場合、Recovery Manager で UPGRADE CATALOG コマンドを実行できます。このコマンドの実行により、パッケージを必要に応じて随時作成できます。アップグレード時に生成されたエラー・メッセージに関してメッセージ・ログをチェックしてください。

## リカバリ・カタログの現行バージョンのチェック

カタログのスキーマの、現行のリリースを判別するには、SQL 問合せを実行する必要があります。カタログのバージョンは、Recovery Manager の新しいリリースごとに変更する必要はありません。カタログ間に非互換性が発見された場合に更新します。つまり、アップグレードが必要な時期を Recovery Manager に通知させることが、最良のバックアップ計画になるわけです。

### リカバリ・カタログの現行リリースの判別

1. SQL\*Plus セッションを使用し、カタログ所有者としてリカバリ・カタログ・データベースに接続します。たとえば、次のように入力します。

```
% sqlplus rman/cat@catdb
```

2. RCVER カタログ表に問い合わせます。たとえば、次の問合せを実行します。

```
SELECT * FROM RCVER;
```

```
VERSION
```

```
-----
```

```
09.02.00
```

## リカバリ・カタログ・スキーマのアップグレード

Recovery Manager の UPGRADE CATALOG コマンドを使用して、カタログを新リリースに変更します。

### リカバリ・カタログをアップグレードする方法

1. Recovery Manager を起動し、ターゲット・データベースとリカバリ・カタログ・データベースに接続します。たとえば、次のように入力します。

```
% rman TARGET / CATALOG rman/cat@catdb
```

```
connected to recovery catalog database
PL/SQL package rcat.DBMS_RVCAT version 08.00.04 in RVCAT database is too old
```

2. UPGRADE CATALOG コマンドを発行します。

```
UPGRADE CATALOG;
```

```
recovery catalog owner is rman
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します。

```
UPGRADE CATALOG;
```

```
recovery catalog upgraded to version 09.02.00
DBMS_RCVMAN package upgraded to version 09.02.00
DBMS_RVCAT package upgraded to version 09.02.00
```

### 関連項目：

- UPGRADE CATALOG コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- リカバリ・カタログの互換性の詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- 互換性と移行の詳細は、『Oracle9i データベース移行ガイド』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、このカタログが管理している全ターゲット・データベースの全バックアップが使用不可能となります。

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、データベースを1つ登録解除する場合、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・データベースのメタデータを削除しようとする、全ターゲット・データベースのメタデータ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. Recovery Manager を起動し、ターゲット・データベースとリカバリ・カタログ・データベースに接続します。

```
% rman TARGET / CATALOG rman/cat@catdb
```

2. 確認のため、DROP CATALOG コマンドを2回発行します。

```
DROP CATALOG;
```

```
recovery catalog owner is rman  
enter DROP CATALOG command again to confirm catalog removal
```

```
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削除するには、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、『Oracle9i Recovery Manager リファレンス』を参照してください。カタログからデータベースを登録解除する方法は、16-9 ページの「[リカバリ・カタログからのターゲット・データベースの登録解除](#)」を参照してください。

## リカバリ・カタログを使用しない Recovery Manager リポジトリの管理方法

Recovery Manager 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、実際には全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、追加の管理タスクをいくつか実行する必要があります。この項では、次の項目について説明します。

- [カタログ専用コマンドの制限](#)
- [制御ファイル・レコードの上書きの監視](#)
- [制御ファイル・リポジトリのメンテナンス](#)
- [制御ファイルのバックアップおよびリストア](#)

### 関連項目：

- 制御ファイルの概説は、『Oracle9i データベース管理者ガイド』を参照してください。
- バックアップ時とリカバリ時の制御ファイルの重要性は、[第 16 章「Recovery Manager のリポジトリの管理」](#)を参照してください。
- 制御ファイルの管理の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

## カタログ専用コマンドの制限

リカバリ・カタログを使用しない場合でも、Recovery Manager を非常に効果的に利用することができます。Recovery Manager 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しています。ただし、制御ファイルを Recovery Manager メタデータの単一リポジトリとして使用している場合、一部のコマンドは使用できません。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。

- CREATE CATALOG、UPGRADE CATALOG、DROP CATALOG
- CREATE SCRIPT、DELETE SCRIPT、REPLACE SCRIPT、PRINT SCRIPT
- LIST INCARNATION
- REGISTER DATABASE
- REPORT SCHEMA AT TIME
- RESET DATABASE
- RESYNC CATALOG

## 制御ファイル・レコードの上書きの監視

リカバリ・カタログを使用しない場合、制御ファイルは、Recovery Manager のバックアップとコピーに関する唯一の情報ソースです。バックアップとコピーを作成すると、新規レコードが制御ファイルに追加されます。これらのレコードは循環しながら再使用されます。つまり、古いレコードは上書きされます。

次の初期化パラメータによって、レコードを上書きできるまでの最低日数を決定します。

```
CONTROL_FILE_RECORD_KEEP_TIME = integer
```

たとえば、パラメータ値が 14 の場合、14 日以上経過したレコードは再利用の候補になります。上書きされたレコードの情報は消失します。

Oracle で新規レコードを制御ファイルに追加する必要があり、最も古いレコードが CONTROL\_FILE\_RECORD\_KEEP\_TIME で指定した日数をまだ経過していない場合は、次のような処理が行われます。

1. 制御ファイルのサイズを拡張しようとします。ただし、この処理は、基礎になるオペレーティング・システム・ファイルが拡張できる場合にのみ可能です。
2. 制御ファイルを拡張できない場合は、Oracle は CONTROL\_FILE\_RECORD\_KEEP\_TIME で指定した日数を経過していなくても、最も古いレコードを上書きして、この処理を alert.log に記録します。

リカバリ・カタログを使用していない場合は、CONTROL\_FILE\_RECORD\_KEEP\_TIME の値を、保存する必要のある最も古いファイルより少し大きい値に設定します。たとえば、データベースを週に 1 回バックアップする場合は、各バックアップを少なくとも 1 週間は保存する必要があります。CONTROL\_FILE\_RECORD\_KEEP\_TIME を、10 または 14 などの値に設定します。

### 制御ファイル・レコードの上書きの管理：例

次の例を想定してみます。

- リカバリ・カタログを使用しません。
- CONTROL\_FILE\_RECORD\_KEEP\_TIME を 14 に設定します。
- 制御ファイル内の現在のレコードは、すべて 1 日～13 日間経過しています。
- 制御ファイルは、オペレーティング・システムで許容可能な最大サイズです。

データベースのバックアップを作成します。オペレーティング・システムのファイル・サイズ制限を超えて制御ファイルを拡張することは不可能なため、制御ファイル内のレコードで、13 日を経過したレコードから上書きが開始されます。上書きされる各レコードについては、次のように alert.log にエントリが記録されます。

```
krcpwnc: following controlfile record written over:
RECID #72 Recno 72 Record timestamp
07/28/00 22:15:21
Thread=1 Seq#=3460
Backup set key: stamp=372031415, count=17
Low scn: 0x0000.3af33f36
07/27/00 21:00:08
Next scn: 0x0000.3af3871b
07/27/00 23:23:54
Resetlogs scn and time
scn: 0x0000.00000001
08/05/99 10:46:44
Block count=102400 Blocksize=512
```

このような状況が発生しないように、リカバリ・カタログを使用してください。カタログを使用できない場合は、可能ならば次の処理を行ってください。

- 制御ファイルは、ロー・ディスクではなくファイル・システムに格納すると、必要に応じて拡張できます。
- alert.log を監視して、制御ファイル・レコードが上書きされていないことを確認します。

**関連項目：** 制御ファイル・レコードの概説は、4-16 ページの「[制御ファイル内のレコードのタイプ](#)」を参照してください。

## 制御ファイル・リポジトリのメンテナンス

Recovery Manager には、バックアップとコピーを物理的に削除するコマンドの他に、バックアップのレコードのチェックまたは削除を行うコマンドがいくつか用意されています。

**関連項目：**

- これらのメンテナンス手順の詳細は、16-6 ページの「[リカバリ・カタログ内のターゲット・データベース・レコードの管理](#)」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・不使用に関係なく有効です。
- カタログが必要なコマンドのリストは、16-36 ページの「[カタログ専用コマンドの制限](#)」を参照してください。



## 制御ファイルのバックアップおよびリストア

制御ファイルを Recovery Manager メタデータの専用のリポジトリとして使用する場合は、多重化またはオペレーティング・システム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 Recovery Manager のバックアップとコピーの情報もすべて失われます。このため、CONFIGURE CONTROLFILE AUTOBACKUP を ON に設定する必要があります。

制御ファイルの自動バックアップが使用可能な場合にかぎり、Recovery Manager はデータベースをマウントおよびリストアできます。制御ファイルのマウント後には、データベースの残りの部分をリストアできます。制御ファイルに格納された、永続的な構成の設定は失われますが、自動バックアップをリストアした後は、構成の設定を復元できます。

### 関連項目：

- 制御ファイルの自動バックアップ機能については、5-52 ページの「[制御ファイルとサーバー・パラメータ・ファイルの自動バックアップ](#)」を参照してください。
- 制御ファイルの手動および自動バックアップについては、9-9 ページの「[Recovery Manager を使用した制御ファイルのバックアップ](#)」を参照してください。
- 現行の制御ファイルおよびリカバリ・カタログが使用不可能な場合の、データベースのリストア方法については、10-18 ページの「[バックアップ制御ファイルを使用してリカバリ・カタログを使用しないリカバリの実行](#)」を参照してください。
- 制御ファイルのユーザー管理バックアップの作成方法については、『Oracle9i ユーザー管理バックアップおよびリカバリ・ガイド』を参照してください。



---

## Recovery Manager のリポジトリの問合せ

この章では、リポジトリから Recovery Manager に関する情報を取得する方法について説明します。この章の項目は、次のとおりです。

- [Recovery Manager のメタデータの問合せ](#)
- [Recovery Manager のバックアップ、コピーおよびデータベース・インカネーションのリスト表示](#)
- [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](#)
- [Recovery Manager の構成設定の表示](#)
- [リカバリ・カタログに格納されたスクリプトの出力](#)
- [リカバリ・カタログ・ビューへの問合せ](#)
- [Recovery Manager リポジトリの問合せの例](#)

# Recovery Manager のメタデータの問合せ

Recovery Manager リポジトリから情報を取得するには、いくつかの方法があります。基本的なオプションを次の表に示します。

方法	カタログの 必要性	説明
LIST コマンド	不要	このコマンドは、バックアップ、コピーおよびデータベース・インカネーションをリストするために使用します。このコマンドの出力には、CHANGE、CROSSCHECK および DELETE コマンドで操作されたファイルが表示されます。
REPORT コマンド	不要	このコマンドは、バックアップの必要なファイル、不要になったバックアップ、スキーマ内にあるファイルなどを確認するために使用します。
SHOW コマンド	不要	このコマンドは、Recovery Manager の永続的な構成を表示するために使用します。
PRINT SCRIPT コマンド	必要	このコマンドは、リカバリ・カタログに格納されているスクリプトの名前を表示するために使用します。
リカバリ・ カタログの固定 ビュー	必要	これらのビューは、カタログ自体にアクセスするときに問い合わせます。ストアド・スクリプトの名前や内容などの情報は、カタログ・ビューからのみ取得できます。
V\$ 固定ビュー	不要	これらのビューは、ターゲット・データベースの制御ファイルにアクセスするときに問い合わせます。Recovery Manager では、リカバリ・カタログのメタデータを制御ファイルから取得します。V\$DATAFILE_HEADER、V\$PROCESS および V\$SESSION などの V\$ ビューには、カタログ・ビューにはない情報が含まれています。

Recovery Manager に関する情報の主なソースは、REPORT および LIST コマンドの出力です。これらのコマンドを使用して Recovery Manager リポジトリに問い合わせ、何をバックアップしたか、バックアップする必要があるのは何かを確認します。この情報は、効果的なバックアップ計画の策定に非常に役立ちます。

LIST コマンドを実行すると、Recovery Manager のバックアップ（バックアップ・セットとプロキシ・コピー）およびコピーがすべて表示されます。REPORT コマンドを使用すると、さらに複雑な分析を行うことができます。たとえば、REPORT コマンドを使用して、バックアップが必要なデータ・ファイルと不要になったバックアップ・ピースに関するレポートを生成できます。Recovery Manager は、REPORT コマンドと LIST コマンドの出力を、標準出力またはログ・ファイルへ書き込みます。

SHOW コマンドは、永続的な構成の設定を表示します。たとえば、CONFIGURE コマンドを使用して自動チャネルを割り当てると、これらのチャネルの設定が SHOW コマンドの出力に表示されます。

**関連項目：**

- Recovery Manager リポジトリを、現行の状態に保持する方法は、[第 16 章「Recovery Manager のリポジトリの管理」](#)を参照してください。
- LIST コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- REPORT コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## Recovery Manager のバックアップ、コピーおよびデータベース・インカネーションのリスト表示

LIST コマンドは、リカバリ・カタログまたは制御ファイルに問合せを行い、その中に記録されているバックアップ、コピー、アーカイブ REDO ログおよびデータベース・インカネーションのリストを生成します。これらのファイルは、CHANGE、CROSSCHECK および DELETE コマンドを実行するときに指定できます。

この項では、次の項目について説明します。

- [Recovery Manager のリスト](#)
- [バックアップごとのリスト表示](#)
- [ファイルによるバックアップのリスト表示](#)
- [コピーのリスト表示](#)
- [サマリー・モードでのバックアップのリスト表示](#)
- [制限付きのバックアップおよびコピーのリスト表示](#)
- [データベース・インカネーションのリスト表示](#)

## Recovery Manager のリスト

BY BACKUP オプションおよび BY FILE オプションを使用し、SUMMARY オプションと VERBOSE オプションのいずれかを選択すると、出力の表示方法を制御できます。

LIST コマンドの主な目的は、どのバックアップまたはコピーが使用可能かを判断することです。リポジトリに格納されるのは、正常に完了したバックアップとコピーのみです。たとえば、次のようにリストできます。

- **Recovery Manager** リポジトリに記録されたバックアップ（バックアップ・セットとプロキシ・コピー）またはイメージ・コピー
- 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
- 期限切れのバックアップおよびイメージ・コピー
- 時間、パス名、デバイス・タイプ、タグまたはリカバリ能力などのオプションに基づいて限定したバックアップとイメージ・コピー
- 指定したデータベースのインカネーション

バックアップが必要なものを確認するには **Recovery Manager** リポジトリを使用します。具体的には、次の点を確認します。

- 出力表の STATUS 列に、バックアップとイメージ・コピーが AVAILABLE としてリスト表示されている
- バックアップが必要なデータ・ファイル、アーカイブ REDO ログおよび制御ファイルがすべて出力に含まれている
- リポジトリに記録されているバックアップおよびコピーが最新のものである

## バックアップごとのリスト表示

Recovery Manager はデフォルトではバックアップごとにリスト表示します。つまり、各バックアップ・セットまたはプロキシ・コピーを順番にリストしてから、バックアップに含まれているファイルを識別します。デフォルトでは、Recovery Manager はバックアップおよびコピーを冗長モードでリストします。つまり、複数行にわたる詳しい情報を表示します。

### バックアップごとのリスト表示

1. ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続後、LIST BACKUP コマンドを発行します。listObjList 句を使用して必要なオブジェクトを指定します。たとえば、次のように入力できます。

```
LIST BACKUP; # lists backup sets, backup pieces, and proxy copies
```

必要に応じて EXPIRED キーワードを指定し、クロスチェック中に見つけられなかったバックアップを識別します。

```
LIST EXPIRED BACKUP;
```

2. 出力をチェックします（LIST 出力の様々な列ヘッダーの説明は、『Oracle9i Recovery Manager リファレンス』を参照してください）。サンプル出力を次に示します。

```
LIST BACKUP;
```

```
List of Backup Sets
=====
```

BS Key	Size	Device Type	Elapsed Time	Completion Time
19	201K	DISK	00:00:01	08-FEB-02
BP Key: 30    Status: AVAILABLE    Tag: TAG20020208T155239				
Piece Name: /oracle/dbs/0jdg9v28_1_1				

```
List of Archived Logs in backup set 19
```

Thrd	Seq	Low SCN	Low Time	Next SCN	Next Time
1	21	98086	08-FEB-02	98461	08-FEB-02
1	22	98461	08-FEB-02	98464	08-FEB-02
1	23	98464	08-FEB-02	98469	08-FEB-02
1	24	98469	08-FEB-02	98472	08-FEB-02
1	25	98472	08-FEB-02	98475	08-FEB-02

BS Key	Type	LV	Size	Device	Type	Elapsed Time	Completion Time
-----							
20	Full		197M	DISK		00:00:42	08-FEB-02
BP Key: 31    Status: AVAILABLE    Tag: TAG20020208T155242							
Piece Name: /oracle/dbs/0kdg9v2b_1_1							
SPFILE Included: Modification time: 08-FEB-02							
List of Datafiles in backup set 20							
File	LV	Type	Ckp	SCN	Ckp Time	Name	
-----							
1		Full	98512		08-FEB-02	/oracle/oradata/trgt/system01.dbf	
2		Full	98512		08-FEB-02	/oracle/oradata/trgt/undotbs01.dbf	
3		Full	98512		08-FEB-02	/oracle/oradata/trgt/cwmlite01.dbf	
4		Full	98512		08-FEB-02	/oracle/oradata/trgt/drsys01.dbf	
5		Full	98512		08-FEB-02	/oracle/oradata/trgt/example01.dbf	
6		Full	98512		08-FEB-02	/oracle/oradata/trgt/indx01.dbf	
7		Full	98512		08-FEB-02	/oracle/oradata/trgt/tools01.dbf	
8		Full	98512		08-FEB-02	/oracle/oradata/trgt/users01.dbf	
BS Key	Size		Device	Type	Elapsed Time	Completion Time	
-----							
21	1K		DISK		00:00:02	08-FEB-02	
BP Key: 32    Status: AVAILABLE    Tag: TAG20020208T155331							
Piece Name: /oracle/dbs/0ldg9v3r_1_1							
List of Archived Logs in backup set 21							
Thrd	Seq	Low SCN		Low Time	Next SCN	Next Time	
-----							
1	34	98509		08-FEB-02	98529	08-FEB-02	



## ファイルによるバックアップのリスト表示

データ・ファイル、制御ファイルおよびアーカイブ・ログのコピーをリストできます。  
`listObjList` 句または `recordSpec` 句を使用して必要なオブジェクトを指定します  
(『Oracle9i Recovery Manager リファレンス』を参照してください)。オブジェクトを指定しない場合、Recovery Manager はすべてのデータベース・ファイルおよびアーカイブ REDO ログのコピーを表示します。デフォルトでは、Recovery Manager はバックアップを冗長モードでリストします。つまり、複数行にわたる詳しい情報を表示します。

### ファイルによるバックアップのリスト表示

1. ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続後、  
BY FILE オプションを指定した LIST BACKUP コマンドを実行します。必要なオブジェクトおよびオプションを指定します。たとえば、次のように入力できます。

```
LIST BACKUP BY FILE;
```

必要に応じて EXPIRED キーワードを指定し、クロスチェック中に見つけられなかったバックアップを識別します。

```
LIST EXPIRED BACKUP BY FILE;
```

2. 出力をチェックします (LIST 出力の様々な列ヘッダーの説明は、『Oracle9i Recovery Manager リファレンス』を参照してください)。サンプル出力を次に示します。

```
List of Datafile Backups
=====
```

File	Key	TY	LV	S	Ckp SCN	Ckp Time	#Pieces	#Copies	Tag
----	----	-	-	-	-----	-----	-----	-----	----
1	20	B	F	A	98512	08-FEB-02 1	1	1	TAG20020208T155242
2	20	B	F	A	98512	08-FEB-02 1	1	1	TAG20020208T155242
3	20	B	F	X	98512	08-FEB-02 1	1	1	TAG20020208T155242
4	20	B	F	U	98512	08-FEB-02 1	1	1	TAG20020208T155242

```
List of Archived Log Backups
=====
```

Thrd	Seq	Low SCN	Low Time	BS Key	S	#Pieces	#Copies	Tag
----	----	-----	-----	-----	-	-----	-----	----
1	21	98086	08-FEB-02 22	A 1	1	1	1	TAG20020208T155604
				19	A 1	1	1	TAG20020208T155239
1	22	98461	08-FEB-02 22	A 1	1	1	1	TAG20020208T155604
				19	A 1	1	1	TAG20020208T155239
1	23	98464	08-FEB-02 22	A 1	1	1	1	TAG20020208T155604
				19	A 1	1	1	TAG20020208T155239

```
List of Controlfile Backups
=====

CF Ckp SCN Ckp Time  BS Key  S #Pieces #Copies Tag
-----
98510      08-FEB-02 20      A 1      1      TAG20020208T155242

List of SPFILE Backups
=====

Modification Time BS Key  S #Pieces #Copies Tag
-----
08-FEB-02      20      A 1      1      TAG20020208T155242
```

コピーのリスト表示

バックアップ・セットおよびプロキシ・コピーの他に、イメージ・コピーもリストできます。`listObjList` 句、`recordSpec` 句または `archiveLogRecordSpecifier` 句を使用して必要なオブジェクトを指定します。オブジェクトを指定しない場合、`LIST COPY` によりすべてのデータ・ファイルのコピー、制御ファイルのコピーおよびアーカイブ REDO ログが表示されます。**Recovery Manager** では、アーカイブ REDO ログとそのイメージ・コピーが、どちらもコピーとみなされるため注意してください。デフォルトでは、**Recovery Manager** はバックアップを冗長モードでリストします。つまり、複数行にわたる詳しい情報を表示します。

イメージ・コピーのリスト表示

- 1. ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続後、`LIST COPY` コマンドを実行します。必要なオブジェクトおよびオプションを指定します。たとえば、次のように入力できます。  
  
`LIST COPY; # lists all datafile copies, control file copies, and archived logs`  
`LIST ARCHIVELOG ALL; # lists all archived logs`  
  
必要に応じて `EXPIRED` キーワードを指定し、クロスチェック中に見つけられなかったコピーを識別します。  
  
`LIST EXPIRED COPY;`

2. 出力をチェックします (LIST 出力の様々な列ヘッダーの説明は、『Oracle9i Recovery Manager リファレンス』を参照してください)。サンプル出力を次に示します。

```
LIST ARCHIVELOG ALL;
```

List of Archived Log Copies						
Key	Thrd	Seq	S	Low Time	Name	
-----	-----	-----	-----	-----	-----	
8	1	28	A	NOV 07 2001 10:50:07	/oracle/oradata/trgt/arch/archive1_28.dbf	
9	1	29	A	NOV 07 2001 11:54:10	/oracle/oradata/trgt/arch/archive1_29.dbf	
10	1	30	A	NOV 07 2001 12:00:22	/oracle/oradata/trgt/arch/archive1_30.dbf	
11	1	31	A	NOV 07 2001 12:01:28	/oracle/oradata/trgt/arch/archive1_31.dbf	
12	1	32	A	NOV 07 2001 12:44:00	/oracle/oradata/trgt/arch/archive1_32.dbf	
13	1	33	A	NOV 07 2001 12:59:37	/oracle/oradata/trgt/arch/archive1_33.dbf	

サマリー・モードでのバックアップのリスト表示

LIST コマンドの出力はデフォルトでは非常に詳しい情報ですが、サマリー形式で出力を表示するように指定することもできます。listObjList 句または recordSpec 句を使用して必要なオブジェクトを指定します。オブジェクトを指定しない場合、LIST BACKUP によりすべてのバックアップが表示されます。デフォルトでは、Recovery Manager は冗長モードでリスト表示します。

サマリー・モードでバックアップをリスト表示する方法

1. ターゲット・データベースおよびリカバリ・カタログ (使用している場合) に接続後、LIST BACKUP コマンドを発行します。必要なオブジェクトおよびオプションを指定します。たとえば、次のように入力できます。

```
LIST BACKUP SUMMARY;
```

必要に応じて EXPIRED キーワードを指定し、クロスチェック中に見つけられなかったコピーを識別します。

```
LIST EXPIRED BACKUP SUMMARY;
```

2. 出力をチェックします (LIST 出力の様々な列ヘッダーの説明は、『Oracle9i Recovery Manager リファレンス』を参照してください)。サンプル出力を次に示します。

List of Backups  
=====

Key	TY	LV	S	Device	Type	Completion Time	#Pieces	#Copies	Tag
387	B	0	A	SBT_TAPE		08-FEB-01	1	2	TAG20020208T155604
396	B	0	A	SBT_TAPE		08-FEB-01	3	1	TAG20020208T155604
423	B	0	X	SBT_TAPE		08-FEB-01	1	1	TAG20020208T155604
427	B	0	U	SBT_TAPE		08-FEB-01	1	1	TAG20020208T155604

## 制限付きのバックアップおよびコピーのリスト表示

複数の異なる条件を指定すると、LIST の出力を絞り込めます。

### オブジェクトまたは他の条件で限定して、コピーとバックアップのリストを生成する方法

1. ターゲット・データベースおよびリカバリ・カタログ (使用している場合) に接続後、*listObjList* または *recordSpec* 条件を指定して LIST COPY または LIST BACKUP コマンドを実行します。たとえば、次のように入力します。

```
# lists backups of all files in database
LIST BACKUP OF DATABASE;

# lists copy of specified datafile
LIST COPY OF DATAFILE '?/oradata/trgt/system01.dbf';

# lists specified backup set
LIST BACKUPSET 213;

# lists datafile copy
LIST DATAFILECOPY '/tmp/tools01.dbf';

2. maintQualifier または RECOVERABLE 句を指定して、検索を制限することもできます。たとえば、次のように入力します。

# specify a backup by tag
LIST BACKUP TAG 'weekly_full_db_backup';

# specify a backup or copy by device type
LIST COPY OF DATAFILE '?/oradata/trgt/system01.dbf' DEVICE TYPE sbt;

# specify a backup or copy by directory or path
LIST BACKUP LIKE '/tmp/%';

# specify a backup or copy by a range of completion dates
LIST COPY OF DATAFILE 2 COMPLETED BETWEEN '10-DEC-2001' AND '17-DEC-2001';
```

```
# specify logs backed up at least 2X to tape
LIST ARCHIVELOG ALL BACKED UP 2 TIMES TO DEVICE TYPE sbt;
```

3. 出力をチェックします。たとえば、次のような datafile 1 のコピーのリストのサンプル出力が表示されます。

```
LIST COPY OF DATAFILE 1;

List of Datafile Copies
Key      File S Completion time Ckp SCN      Ckp time      Name
-----
3        1      A 18-JUL-00      114148      17-JUL-01      /tmp/system01.dbf
```

**関連項目：** *listObjList* の構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。LIST 出力の様々な列の説明は、『Oracle9i Recovery Manager リファレンス』を参照してください。

データベース・インカネーションのリスト表示

ターゲット・データベースのオンライン REDO ログをリセットするたびに、データベースの新規インカネーションが作成されます。インカネーションは、LIST コマンドの INCARNATION オプションを使用して追跡できます。

データベース・インカネーションのリスト表示

1. ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続後、LIST INCARNATION を実行します。

```
LIST INCARNATION;

リカバリ・カタログを使用して、同じカタログに複数のターゲット・データベースを登録している場合は、OF DATABASE オプションを使用して各ターゲット・データベースを区別できます。

LIST INCARNATION OF DATABASE prod3;
```

2. 出力をチェックします（LIST 出力の様々な列ヘッダーの説明は、『Oracle9i Recovery Manager リファレンス』を参照してください）。サンプル出力を次に示します。

```
LIST INCARNATION OF DATABASE;

List of Database Incarnations
DB Key  Inc Key DB Name  DB ID      CUR Reset SCN  Reset Time
-----
1       12     TRGT    1335481537 NO 1         NOV 12 2001 03:06:41
1       2      TRGT    1335481537 YES 164378      NOV 12 2001 17:54:26
```

前述の出力は、SCN 164378 でデータベース trgt に対して RESETLOGS が実行され、新規インカネーションが作成されたことを示しています。インカネーションは、インカネーション・キーで区別されます。

## 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

この項では次の項目について説明します。

- [Recovery Manager のレポート](#)
- [バックアップに必要なオブジェクトのレポート](#)
- [不要なバックアップとコピーのレポート](#)
- [リカバリ不能なバックアップおよびコピーのレポート](#)
- [データベース・スキーマのレポート](#)

### Recovery Manager のレポート

Recovery Manager リポジトリから詳しい情報を取得する場合は、レポートを生成します。次の項目を確認するには、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
- リカバリ不能操作が実行されたファイルはどれか
- 不要なバックアップやコピーはどれか。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
- 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

---

---

**注意：** レポートの正確性を期すために、Recovery Manager リポジトリは制御ファイルと同期させる必要があります。また、CHANGE、UNCATALOG および CROSSCHECK の各コマンドを実行して、すべてのバックアップとコピーの状態を更新しておく必要があります。Recovery Manager リポジトリのメンテナンス方法は、[第 16 章「Recovery Manager のリポジトリの管理」](#)を参照してください。

---

---

レポートから取得した情報は、バックアップおよびリカバリ計画を立てる上で非常に重要です。特に、REPORT NEED BACKUP コマンドおよび REPORT UNRECOVERABLE コマンドを実行して、次のことが確実に行われるようにします。

- リカバリに必要なバックアップが利用できるようにします。
- リカバリが妥当な時間内で完了するようにします。つまり、平均リカバリ時間 (MTTR) を最小限に抑えるようにします。

## バックアップに必要なオブジェクトのレポート

NEED BACKUP キーワードを指定すると、バックアップの必要なオブジェクトに関するレポートを作成できます。REDUNDANCY パラメータは、データ・ファイルが、バックアップの必要がない範囲に存在すると判断されるのに必要な、バックアップまたはコピーの最小数を指定します。このパラメータを使用しない場合、REDUNDANCY のデフォルト値は 1 です。DAYS パラメータは、integer で指定した日数以上経過しているログを使用して、リカバリを開始する必要があることを示します。INCREMENTAL パラメータは、完全リカバリのために integer で指定した数より多い増分バックアップが必要であることを示します。

---

---

**注意：** 保存方針を使用禁止にした場合、他のオプションを指定しないで REPORT NEED BACKUP を実行すると、エラー・メッセージが生成されます。

---

---

### バックアップが必要なオブジェクトのレポート作成

1. ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続後、必要に応じて CROSSCHECK コマンドを実行し、バックアップとコピーの状態を更新します。クロスチェック・セッションの例を次に示します。

```
# allocate maintenance channel for crosscheck if automatic channels not
configured
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt;
CROSSCHECK BACKUP; # crosschecks all backups
CROSSCHECK COPY; # crosschecks all copies
```

2. 保存方針が構成してある場合は、他のオプションを指定しないで **REPORT NEED BACKUP** を実行すると、バックアップが必要なファイルを判別できます（サンプル出力を次に示します）。

```
REPORT NEED BACKUP;

RMAN retention policy will be applied to the command
RMAN retention policy is set to redundancy 1
Report of files with less than 1 redundant backups
File #bkps Name
-----
2      0      /oracle/oradata/trgt/undotbs01.dbf
3      0      /oracle/oradata/trgt/cwmlite01.dbf
4      0      /oracle/oradata/trgt/drsys01.dbf
7      0      /oracle/oradata/trgt/tools01.dbf
```

3. 保存方針をオーバーライドするには（または、保存方針が使用可能になっていない場合は）、**REPORT NEED BACKUP DAYS** を実行します。**DAYS** パラメータ値より古いファイルには新規バックアップが必要です。これは、これらのファイルのバックアップには、指定された **DAYS** 日分のリカバリ用アーカイブ・ログが必要になるためです。たとえば、次のように実行します。

```
REPORT NEED BACKUP DAYS = 7 DATABASE; # needs min 7 days of logs to recover
REPORT NEED BACKUP DAYS = 30 TABLESPACE SYSTEM;
REPORT NEED BACKUP DAYS = 14 DATAFILE '?/oradata/trgt/tools01.dbf';
```

4. 増分バックアップを必要とするファイルを判断するには、**INCREMENTAL** パラメータを指定します。データ・ファイルの完全リカバリのために、指定した数より多くの増分バックアップが必要な場合、**Recovery Manager** はそのデータ・ファイルには新規バックアップが必要であると判断します。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 '?/oradata/trgt/users01.dbf';
```

**関連項目：** **REPORT** 出力の様々な列ヘッダーの説明は、『**Oracle9i Recovery Manager** リファレンス』を参照してください。



## 不要なバックアップとコピーのレポート

OBSOLETE キーワードを指定すると、不要なオブジェクトに関するレポートを作成できます。他のオプションを指定しない場合、REPORT OBSOLETE は現在の保存方針により不要とマークされたバックアップとコピーを表示します。デフォルトでは、保存方針の REDUNDANCY は 1 に構成されます。

REPORT OBSOLETE コマンドは、コマンド・レベルで RECOVERY WINDOW オプションと REDUNDANCY オプションをサポートします。この 2 つのオプションの意味は、CONFIGURE コマンドに使用する同名のオプションと同じです。

### 不要なバックアップとコピーのレポート方法

1. ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続後、必要に応じて CROSSCHECK コマンドを発行し、バックアップとコピーの状態を更新します。クロスチェック・セッションの例を次に示します。

```
# allocate maintenance channel for crosscheck if automatic channels not
configured
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt;
CROSSCHECK BACKUP; # crosschecks all backups
CROSSCHECK COPY; # crosschecks all copies
RELEASE CHANNEL;
```

2. OBSOLETE オプションを使用して、今後のリカバリには不要となったバックアップを識別します。たとえば、次のように入力します。

```
# lists backups or copies that are superfluous because they are not needed to
recover
# the database to a random point within the past week
REPORT OBSOLETE RECOVERY WINDOW OF 7 DAYS;
# lists backups or copies that are superfluous because more than 2 copies of the
# files exist on tape
REPORT OBSOLETE REDUNDANCY = 2 DEVICE TYPE sbt;
```

3. ORPHAN オプションを使用して、現行のインカネーションの直接の親でないインカネーションに属した、使用不可能のバックアップおよびコピーをリスト表示します（7-5 ページの「[孤立したバックアップのレポート](#)」を参照してください）。たとえば、次のように入力します。

```
REPORT OBSOLETE ORPHAN;
```

4. 必要に応じて、これらの不要なバックアップを削除します。DELETE OBSOLETE コマンドを発行すると、不要なバックアップとコピーを自動的に削除できます。たとえば、次のように入力できます。

```
# delete obsolete backups and copies displayed when you issue REPORT OBSOLETE
DELETE OBSOLETE;
# delete obsolete backups and copies according to a specified recovery window
DELETE OBSOLETE RECOVERY WINDOW OF 7 DAYS;
# delete obsolete backups and copies according to a specified redundancy
DELETE OBSOLETE REDUNDANCY = 2;
```

Recovery Manager では、ファイルを削除する前に削除を確認するプロンプトが表示されます。このプロンプトを表示しないようにするには、DELETE コマンドの NOPROMPT オプションを指定します。ファイルを削除し、ファイルの有無に関係なく、リポジトリ・レコードを削除するには、FORCE を指定します。Recovery Manager では、削除されたオブジェクトに関する I/O エラーは無視されます。

**関連項目：** 不要なバックアップのレポートの概要は、7-4 ページの「[不要なバックアップのレポート](#)」を参照してください。保存方針の概要は、5-55 ページの「[バックアップの保存方針](#)」を参照してください。

## リカバリ不能なバックアップおよびコピーのレポート

REPORT UNRECOVERABLE コマンドを発行して、最新のバックアップ後にデータ・ファイル内に存在するオブジェクトに対してリカバリ不能操作が行われたオブジェクトを含んだデータ・ファイルを確認します。

ALTER TABLE employee ... NOLOGGING 文を発行し、表 employee に対してリカバリ不能な操作を実行したとします。employee 表がデータ・ファイル 3 にある場合、REPORT コマンドは、このデータ・ファイルのバックアップにリカバリ不能というフラグを立てます。

### リカバリ不能なバックアップおよびコピーのレポート作成

ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続後、REPORT UNRECOVERABLE を発行します。たとえば、次のように入力します。

```
REPORT UNRECOVERABLE DATABASE;           # examines all datafiles
REPORT UNRECOVERABLE TABLESPACE 'users'; # examines a specific tablespace
```

## データベース・スキーマのレポート

データベース・ファイルの識別に、V\$ ビューやリカバリ・カタログ・ビューを使用する必要はありません。REPORT SCHEMA を発行して、ファイルをリスト表示します。リカバリ・カタログを使用すると、過去のデータベース・スキーマに関する履歴レポートも生成されます。現在のスキーマに関するレポートの作成には、リカバリ・カタログは必要ありません。

### 指定した時点のデータベース・スキーマのレポート作成

1. ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続後、REPORT SCHEMA を発行し、現時点でターゲット・データベースに含まれているすべてのデータ・ファイルと表領域のリストを取得します。

```
REPORT SCHEMA;
```

リカバリ・カタログを使用する場合は、`atClause` 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できます。

```
REPORT SCHEMA AT TIME 'SYSDATE-14';      # schema as it existed two weeks ago
REPORT SCHEMA AT SCN 1000;                # schema as it existed at scn 1000
REPORT SCHEMA AT SEQUENCE 100 THREAD 1;   # schema as it existed at log sequence
100
```

2. 出力をチェックします。サンプル出力を次に示します。

```
REPORT SCHEMA AT SCN 1000;
```

```
Report of database schema
File K-bytes      Tablespace      RB segs Datafile Name
-----
1          307200 SYSTEM          YES    /oracle/oradata/trgt/system01.dbf
2           20480 UNDOTBS          YES    /oracle/oradata/trgt/undotbs01.dbf
3           10240 CWMLITE          NO     /oracle/oradata/trgt/cwmlite01.dbf
4           10240 DRSYS           NO     /oracle/oradata/trgt/drsys01.dbf
5           10240 EXAMPLE          NO     /oracle/oradata/trgt/example01.dbf
6           10240 INDX            NO     /oracle/oradata/trgt/indx01.dbf
7           10240 TOOLS           NO     /oracle/oradata/trgt/tools01.dbf
8           10240 USERS            NO     /oracle/oradata/trgt/users01.dbf
```

このタイプの情報は、リカバリする時点のデータベースのスキーマを確認できるため、不完全リカバリに役立ちます。

**関連項目：** REPORT コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## Recovery Manager の構成設定の表示

CONFIGURE コマンドで指定された永続的な構成を表示するには、SHOW コマンドを実行します。この構成の設定は、すべての Recovery Manager セッションでの使用を目的に構成されているという点で、永続的です。

SHOW コマンドを使用すると、次の項に説明されている問合せを実行できます。

- [すべての Recovery Manager の構成設定の表示](#)
- [Recovery Manager の保存方針の構成設定の表示](#)
- [自動チャネルの構成設定の表示](#)
- [BACKUP コマンドの構成設定の表示](#)
- [スナップショット制御ファイルのファイル名の表示](#)
- [補助チャネル用に構成されたデフォルトのファイル名の表示](#)

**関連項目：** CONFIGURE コマンドおよび SHOW コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## すべての Recovery Manager の構成設定の表示

Recovery Manager 環境に様々な永続的な構成の設定を指定するには、CONFIGURE コマンドを使用できます。SHOW ALL コマンドは、発行した CONFIGURE コマンドと Recovery Manager のデフォルト構成の両方を表示します。CONFIGURE ...CLEAR コマンドを実行すると、CONFIGURE コマンドで行った設定をデフォルト設定に戻すことができます。

### すべての Recovery Manager の構成の設定の表示

ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続後、SHOW ALL コマンドを実行します。たとえば、次のように入力します。

```
SHOW ALL;      # shows all CONFIGURE settings, both user-entered and default
```

SHOW ALL のサンプル出力は、次のとおりです。

```
RMAN configuration parameters are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION ON;
CONFIGURE DEFAULT DEVICE TYPE TO 'SBT';
CONFIGURE DEVICE TYPE 'SBT' PARALLELISM 1;
CONFIGURE DEVICE TYPE DISK PARALLELISM 2;
CONFIGURE DATAFILE BACKUP COPIES FOR DISK TO 2;
CONFIGURE DATAFILE BACKUP COPIES FOR SBT TO 1; #default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR SBT TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DISK TO 1; # default
CONFIGURE MAXSETSIZE TO 3072K;
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/oracle/dbs/cf_snap.f';
```

```
CONFIGURE EXCLUDE FOR TABLESPACE 'example';
```

表示される出力は、スクリプト内にペーストして、**Recovery Manager** コマンド・ファイルとして実行できます。このため、構成全体を容易に変更できます。このスクリプトを別のターゲット・データベースに対して実行することもできます。

## Recovery Manager の保存方針の構成設定の表示

**CONFIGURE RETENTION POLICY** コマンドを使用すると、リカバリ・ウィンドウの日数または冗長性レベルを指定できます。デフォルトでは、保存方針は **REDUNDANCY = 1** に設定されます。

### 構成の方針の表示

ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続後、**SHOW RETENTION POLICY** コマンドを実行します。たとえば、次のように入力します。

```
SHOW RETENTION POLICY;    # shows the CONFIGURE setting for the retention policy
```

**SHOW RETENTION POLICY** のサンプル出力は、次のとおりです。

```
RMAN configuration parameters are:  
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;
```

## 自動チャネルの構成設定の表示

**CONFIGURE** コマンドを使用すると、次のことを設定できます。

- 自動チャネル
- 自動チャネルにより使用されるデフォルト・デバイス・タイプ
- 割り当てられる自動チャネルのデフォルトの数

### 自動チャネルの設定の表示

自動的に割り当てられたすべてのチャネルの設定を表示するには、**SHOW CHANNEL** コマンドを発行します。

#### 自動チャネルの設定の表示

ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続後、**SHOW CHANNEL** コマンドを発行します。たとえば、次のように入力します。

```
SHOW CHANNEL;    # shows the CONFIGURE setting for the automatic channels
```

**SHOW CHANNEL** のサンプル出力は、次のとおりです。

```
RMAN configuration parameters are:  
CONFIGURE CHANNEL DEVICE TYPE 'SBT' RATE 1500K;
```

## 構成済みデバイス・タイプの表示

構成済みデバイスとそのパラレル化設定を表示するには、SHOW DEVICE TYPE コマンドを発行します。DISK デバイス・タイプは、事前に構成されています。

### 自動チャンネルのデフォルト・デバイス・タイプの表示

ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続後、SHOW DEVICE TYPE コマンドを実行します。たとえば、次のように入力します。

```
SHOW DEVICE TYPE;      # shows the CONFIGURE DEVICE TYPE ... PARALLELISM settings
```

SHOW DEVICE TYPE のサンプル出力は、次のとおりです。

```
RMAN configuration parameters are:
CONFIGURE DEVICE TYPE 'SBT' PARALLELISM 1;
CONFIGURE DEVICE TYPE DISK PARALLELISM 2;
```

## デフォルト・デバイス・タイプの表示

自動チャンネルにより使用されるデフォルト・デバイス・タイプの設定を表示するには、SHOW DEFAULT DEVICE TYPE コマンドを発行します。BACKUP コマンドを発行すると、CONFIGURE DEFAULT DEVICE TYPE コマンドにより設定されているタイプのデフォルト・チャンネルのみが割り当てられます。このデフォルト・デバイス・タイプの設定は、BACKUP 以外のコマンドを使用した場合は有効になりません。デフォルト・デバイス・タイプを使用禁止にすることはできないので注意してください。これは常に DISK（デフォルト設定）または sbt のいずれかです。

### 自動チャンネルのデフォルト・デバイス・タイプの表示

ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続後、SHOW DEFAULT DEVICE TYPE コマンドを実行します。たとえば、次のように入力します。

```
SHOW DEFAULT DEVICE TYPE;      # shows the CONFIGURE DEFAULT DEVICE TYPE setting
```

SHOW DEFAULT DEVICE TYPE のサンプル出力は、次のとおりです。

```
RMAN configuration parameters are:
CONFIGURE DEFAULT DEVICE TYPE TO 'SBT';
```

## BACKUP コマンドの構成設定の表示

BACKUP コマンドに対して次の項目を設定するには、CONFIGURE コマンドを使用します。

- データベース全体のバックアップから除外される表領域（CONFIGURE EXCLUDE）
- Recovery Manager が作成する各バックアップの同一のコピー数（CONFIGURE ... BACKUP COPIES）
- 自動チャネルを使用するジョブのためのデフォルトの最大バックアップ・セット・サイズ（CONFIGURE MAXSETSIZE）
- 同一のファイルがすでにバックアップされている場合に、Recovery Manager がファイルをバックアップするかどうか（CONFIGURE BACKUP OPTIMIZATION）

### バックアップから除外された表領域の表示

データベース全体のバックアップから表領域を除外するには、CONFIGURE EXCLUDE コマンドを使用できます。

#### データベース全体のバックアップから除外された表領域の表示

ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続後、SHOW EXCLUDE コマンドを実行します。たとえば、次のように入力します。

```
SHOW EXCLUDE;      # shows the CONFIGURE EXCLUDE setting
```

SHOW EXCLUDE のサンプル出力は、次のとおりです。

```
RMAN configuration parameters are:
CONFIGURE EXCLUDE FOR TABLESPACE 'OLD_ACCOUNTS';
```

### 各バックアップの同じ内容のコピーの数の表示

Recovery Manager が各バックアップに作成する同一のコピー数を設定するには、CONFIGURE ... BACKUP COPIES コマンドを使用します。たとえば、この値が 3 の場合、Recovery Manager はバックアップ・セット内の各バックアップ・ピースごとに合計 3 つのまったく同じコピーを生成します。

#### 各バックアップの同じ内容のコピーの数の表示

ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続後、SHOW ARCHIVELOG BACKUP COPIES あるいは SHOW DATAFILE BACKUP COPIES コマンドを実行します。たとえば、次のように入力します。

```
SHOW DATAFILE BACKUP COPIES;      # shows the CONFIGURE DATAFILE BACKUP COPIES setting
```

SHOW DATAFILE BACKUP COPIES のサンプル出力は、次のとおりです。

```
RMAN configuration parameters are:
CONFIGURE DATAFILE BACKUP COPIES FOR DISK TO 2;
CONFIGURE DATAFILE BACKUP COPIES FOR SBT TO 1; #default
```

## バックアップ・セットのデフォルト最大サイズの表示

CONFIGURE MAXSETSIZE コマンドを実行して、Recovery Manager のバックアップ・セットに最大サイズを設定できます。バックアップ・セットのサイズは、含まれているバックアップ・ピースの合計バイト数です。

### Recovery Manager のバックアップ・セットの最大サイズの表示

ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続後、SHOW MAXSETSIZE コマンドを発行します。たとえば、次のように入力します。

```
SHOW MAXSETSIZE;          # shows the CONFIGURE MAXSETSIZE settings
```

SHOW MAXSETSIZE のサンプル出力は、次のとおりです。

```
RMAN configuration parameters are:
CONFIGURE MAXSETSIZE TO 3072K;
```

## バックアップの最適化の可能、不可能の表示

バックアップの最適化を使用可能および使用禁止にするには、CONFIGURE BACKUP OPTIMIZATION コマンドを使用できます。

### バックアップの最適化の状態の表示

ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続後、SHOW BACKUP OPTIMIZATION コマンドを発行します。たとえば、次のように入力します。

```
SHOW BACKUP OPTIMIZATION;
```

SHOW BACKUP OPTIMIZATION のサンプル出力は、次のとおりです。

```
RMAN configuration parameters are:
CONFIGURE BACKUP OPTIMIZATION ON;
```



## スナップショット制御ファイルのファイル名の表示

CONFIGURE SNAPSHOT CONTROLFILE コマンドを使用して、スナップショット制御ファイルにデフォルト値を設定できます。SHOW SNAPSHOT CONTROLFILE コマンドを発行して、この値を表示します。

---

---

**注意：** Oracle9i より前のリリースでは、CONFIGURE SNAPSHOT CONTROLFILE コマンドは、SET SNAPSHOT CONTROLFILE と呼ばれていました。

---

---

### スナップショット制御ファイルのファイル名の表示

ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続後、SHOW SNAPSHOT CONTROLFILE コマンドを実行します。たとえば、次のように入力します。

```
SHOW SNAPSHOT CONTROLFILE NAME;      # shows the CONFIGURE SNAPSHOT CONTROLFILE setting
```

SHOW SNAPSHOT CONTROLFILE のサンプル出力は、次のとおりです。

```
RMAN configuration parameters are:
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/oracle/dbs/cf_snap.f';
```

**関連項目：** スナップショット制御ファイルとその機能の詳細は、8-29 ページの「[スナップショット制御ファイルの場所の設定](#)」を参照してください。

## 補助チャネル用に構成されたデフォルトのファイル名の表示

補助チャネルに永続的なファイル名を設定するには、CONFIGURE AUXNAME コマンドを使用できます。たとえば、複製データ・ファイルまたはスタンバイ・データ・ファイル、あるいは TSPITR 操作でのデータ・ファイルに新しいファイル名を付けられます。SHOW AUXNAME コマンドを発行して、これらのファイル名を表示します。

---

**注意：** Oracle9i より前のリリースでは、CONFIGURE AUXNAME コマンドは、SET AUXNAME と呼ばれていました。

---

### 補助ファイル名の永続的な設定の表示

ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続後、SHOW AUXNAME コマンドを発行します。たとえば、次のように入力します。

```
SHOW AUXNAME;      # shows the CONFIGURE AUXNAME setting
```

SHOW AUXNAME のサンプル出力は、次のとおりです。

```
RMAN configuration parameters are:
CONFIGURE AUXNAME FOR DATAFILE '/oracle/oradata/trgt/tools01.dbf' TO '/tmp/tools01.dbf';
```

## リカバリ・カタログに格納されたスクリプトの出力

指定されたストアド・スクリプトのテキストを出力するには、PRINT SCRIPT コマンドを実行するか、RC\_STORED\_SCRIPT\_LINE カタログ・ビューに問い合わせます。Recovery Manager のストアド・スクリプトのリストは、RC\_STORED\_SCRIPT カタログ・ビューに問い合わせて表示します。

この項では、次の項目について説明します。

- [PRINT SCRIPT を使用したストアド・スクリプトのテキスト表示](#)
- [RC\\_STORED\\_SCRIPT\\_LINE の問合せによるストアド・スクリプトのテキスト表示](#)
- [RC\\_STORED\\_SCRIPT の問合せによるストアド・スクリプトのリスト表示](#)

## PRINT SCRIPT を使用したストアド・スクリプトのテキスト表示

PRINT SCRIPT コマンドを使用して、ストアド・スクリプトのテキストを表示します。必要に応じて、Recovery Manager のログ・ファイルに出力を保存できます。

### ストアド・スクリプトのメッセージ・ログへの出力

1. Recovery Manager を起動して、リカバリ・カタログ・データベースとターゲット・データベースに接続し、メッセージ・ログに出力する場合は、LOG 引数を指定します。たとえば、次のように入力して、rman\_log を指定します。

```
% rman TARGET / CATALOG rman/cat@catdb LOG = rman_log
```

スクリプトの作成時に接続していたターゲット・データベースに接続する必要があるため注意してください。

2. PRINT SCRIPT コマンドを発行して、スクリプトをログに書き込みます。

```
PRINT SCRIPT b_whole;
```

3. Recovery Manager を終了し、テキスト・エディタを使用してスクリプトを参照します。たとえば、次のように入力します。

```
RMAN> EXIT  
% vi rman_log
```

**関連項目：** PRINT SCRIPT コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## RC\_STORED\_SCRIPT\_LINE の問合せによるストアド・スクリプトのテキスト表示

RC\_STORED\_SCRIPT\_LINE ビューには、リカバリ・カタログに登録されているターゲット・データベースのすべてのインカネーションに関するすべてのストアド・スクリプトのテキストが含まれます。

### 特定のスクリプトのテキストのリスト表示

1. SQL\*Plus を起動し、カタログ所有者としてリカバリ・カタログ・データベースに接続します。たとえば、次のように入力します。

```
% sqlplus rman/cat@catdb
```

2. 次の問合せを実行します。 *database\_key* はターゲット・データベースの数値主キーに置き換え、 *script\_name* はスクリプトの名前に置き換えます。

```
SELECT TEXT
FROM RC_DATABASE_INCARNATION i, RC_STORED_SCRIPT_LINE l
WHERE i.DB_KEY = database_key
AND SCRIPT_NAME = script_name
AND i.DB_KEY = s.DB_KEY
AND i.CURRENT_INCARNATION = 'YES'
/
```

サンプル出力を次に示します。

```
TEXT
-----
{ backup database plus archivelog;}
```

## RC\_STORED\_SCRIPT の問合せによるストアド・スクリプトのリスト表示

RC\_STORED\_SCRIPT ビューには、カタログに登録されているターゲット・データベースのすべてのインカーネーションに関するすべてのストアド・スクリプトの情報が含まれます。

### ターゲット・データベースの現在のインカーネーションに関するスクリプトのリスト表示

1. SQL\*Plus を起動し、カタログ所有者としてリカバリ・カタログ・データベースに接続します。たとえば、次のように入力します。

```
% sqlplus rman/cat@catdb
```

2. SQL\*Plus で次の問合せを実行します。 *database\_key* はターゲット・データベースの数値主キーに置き換えます。

```
SELECT DISTINCT SCRIPT_NAME
FROM RC_DATABASE_INCARNATION i, RC_STORED_SCRIPT s
WHERE i.DB_KEY = database_key
AND i.DB_KEY = s.DB_KEY
/
```

サンプル出力を次に示します。

```
SCRIPT_NAME
-----
backup_db
backup_system
```

**関連項目：** RC\_STORED\_SCRIPT のビュー詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## リカバリ・カタログ・ビューへの問合せ

LIST、REPORT および SHOW コマンドにより、必要なリポジトリ情報がすべて提供されます。ただし、リカバリ・カタログ・ビューからも役に立つ情報を取得できることがあります。リカバリ・カタログ・ビューは、カタログ・スキーマ内のビューで、接頭辞の RC\_ が付きます。

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・ビューへの問合せ](#)
- [ターゲット DB\\_KEY または DBID 値を使用したカタログ・ビューへの問合せ](#)

**関連項目：** リカバリ・カタログ・ビューの詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## リカバリ・カタログ・ビューへの問合せ

リカバリ・カタログ・ビューは正規化されていませんが、ユーザー問合せではなく、Recovery Manager で使用するための最適化が行われています。Recovery Manager は、ターゲット・データベースの制御ファイルからバックアップおよびリカバリ情報を取得し、この情報をカタログ表に格納します。

一般に、リカバリ・カタログ・ビューは、Recovery Manager のレポート作成コマンドよりは使い勝手が劣ります。たとえば、Recovery Manager を起動し、ターゲット・データベースに接続したとき、このターゲット・データベースに関する情報は、LIST、REPORT および SHOW コマンドを発行したときにのみ取得されます。1 つのリカバリ・カタログに 10 の異なるターゲット・データベースが登録されている場合、カタログ・ビューには、このカタログに登録されている全データベースのすべてのインカネーションに関する情報が表示されます。特定のターゲット・データベースの特定のインカネーションを区別するには、ビューの結合が必要になることがあります。

カタログ・ビューのほとんどは、データベース・サーバー内に対応する動的パフォーマンス・ビューが存在します。たとえば、RC\_BACKUP\_PIECE は、V\$BACKUP\_PIECE に対応します。カタログ・ビューとサーバー・ビューの主な違いは、カタログ・ビューにはカタログに登録されているすべてのデータベースの情報が含まれ、サーバー・ビューにはそのサーバー自体の情報のみが含まれるということです。この 2 種類のビューでは、行を一意に識別するために異なる主キーが使用されることがあります。

## カタログ・ビューでのデータベースの識別

ほとんどのカタログ・ビューには、DB\_KEY および DBINC\_KEY 列が含まれています。各ターゲット・データベースは主キーにより一意に識別されますが、これが DB\_KEY 列および DBID 列（32 ビットの一意データベース識別子）の値です。各ターゲット・データベースの各インカネーションは、DBINC\_KEY 主キーにより一意に識別されます。ターゲット・データベースの特定のインカネーションに関する情報を問い合わせるときは、これらの列を使用してデータベースを指定します。次に、他のカタログ・ビューとの結合を実行して、該当する情報を取得します。

## カタログ・ビューでのデータベース・オブジェクトの識別

カタログ・ビューと V\$ ビューの主な違いは、バックアップおよびリカバリ用オブジェクトに使用される一意識別子のシステムが異なるということです。たとえば、V\$ARCHIVED\_LOG などのほとんどの V\$ ビューでは、RECID 列と STAMP 列を使用して連結主キーが作成されます。対応するカタログ・ビューでは、導出値が主キーとして使用され、この値が 1 つの列に格納されます。たとえば、RC\_ARCHIVED\_LOG の主キーは、AL\_KEY 列です。AL\_KEY 列の値が、Recovery Manager で LIST コマンドの出力に表示される主キーです。

## ターゲット DB\_KEY または DBID 値を使用したカタログ・ビューへの問合せ

ターゲット・データベースの主キーである DB\_KEY 値は、リカバリ・カタログでのみ使用されます。DB\_KEY を取得する最も簡単な方法は、Recovery Manager をターゲット・データベースに接続するたびに表示される、ターゲット・データベースの DBID を使用することです。DBID は、すべての Oracle データベースに付与される一意のシステム定義番号ですが、これにより、Recovery Manager メタデータ内で個々のターゲット・データベースが識別されます。

リカバリ・カタログに登録されているターゲット・データベースの 1 つに関する情報を取得するとします。Recovery Manager がこのデータベースに接続したときに表示される出力を参照するか、次のように V\$ ビューに問い合わせて、このデータベースの DBID を簡単に判別できます。

```
SELECT DBID
FROM V$DATABASE;
```

```
DBID
-----
598368217
```

次の問合せを実行すると、ターゲット・データベースの DB\_KEY を取得できます。  
`dbid_of_target` は、前述の手順で取得した DBID です。

```
SELECT DB_KEY
FROM RC_DATABASE
WHERE DBID = dbid_of_target;
```

ターゲット・データベースの現在のインカネーションに関する情報を取得するには、ターゲット・データベースの DB\_KEY 値を指定し、RC\_DATABASE\_INCARNATION との結合を実行します。これには、WHERE 条件を使用して、CURRENT\_INCARNATION 列の値が YES に設定されていることを指定します。たとえば、DB\_KEY 値が 1 のターゲット・データベースの現在のインカネーション内にあるバックアップ・セットに関する情報を取得するには、次のスクリプトを実行します。

```
SELECT BS_KEY, BACKUP_TYPE, COMPLETION_TIME
FROM RC_DATABASE_INCARNATION i, RC_BACKUP_SET b
WHERE i.DB_KEY = 1
AND i.DB_KEY = b.DB_KEY
AND i.CURRENT_INCARNATION = 'YES';
```

データベースの指定に DB\_NAME 列を使用するのは、複数のデータベースがリカバリ・カタログに同一の DB\_NAME で登録されていない場合のみです。Recovery Manager では、同一のデータベース名で複数のデータベースを登録できます。ただし、DBID 値は一意である必要があります。たとえば、DB\_NAME 値が prod1 のデータベースを 10 個登録できますが、各データベースの DBID は一意です。DBID はメタデータ内の各データベースの一意識別子であるため、この値を使用して DB\_KEY を取得し、次に DB\_KEY を使用してデータベースを一意に識別します。

## Recovery Manager リポジトリの問合せの例

この項では、次の項目について説明します。

- [制限のあるオブジェクトのリスト表示 : 例](#)
- [リカバリ・ウィンドウを必要としないバックアップおよびコピーのレポート : 例](#)
- [冗長なバックアップおよびコピーのレポート : 例](#)
- [データベース・スキーマの履歴レポートの生成 : 例](#)
- [データベース・インカネーションのリスト表示 : 例](#)

## 制限のあるオブジェクトのリスト表示 : 例

リカバリ・カタログを使用していない場合は、LIST コマンドを使用して、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の内容を問い合わせることができます。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、2000 年 6 月 11 日より後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 users BY FILE COMPLETED BEFORE 'JUN 11 2001 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をリスト表示します。

```
LIST BACKUP OF DATABASE SUMMARY DEVICE TYPE sbt;
```

次の例では、/copy ディレクトリ内のタグ df2\_copy が付いているデータ・ファイル 2 の全コピーをリスト表示します。

```
LIST COPY OF DATAFILE 2 TAG df2_copy LIKE '/copy/%';
```

## リカバリ・ウィンドウを必要としないバックアップおよびコピーのレポート : 例

不要で削除可能なバックアップとコピーを判断するには、REPORT コマンドを使用します。たとえば、過去 2 週間以内のいずれかの時点までデータベースをリカバリできればよい場合は、次のコマンドを発行します。

```
REPORT OBSOLETE RECOVERY WINDOW OF 14 DAYS;
```

さらに、次のコマンドを発行して不要なバックアップとコピーを削除できます。

```
DELETE OBSOLETE RECOVERY WINDOW OF 14 DAYS;
```

## 冗長なバックアップおよびコピーのレポート : 例

次のコマンドは、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、不要と判断されたディスク上のバックアップとコピー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、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、不要と判断されたテープ上のバックアップをすべてレポートします。

```
REPORT OBSOLETE REDUNDANCY 2 UNTIL TIME 'SYSDATE-7' DEVICE TYPE sbt;
```



## データベース・スキーマの履歴レポートの生成 : 例

次のコマンドは、現在、1 週間前、2000 年 9 月 20 日時点の各データベース・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('09/20/01','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 SEQUENCE 12 THREAD 2;
```

## データベース・インカネーションのリスト表示 : 例

データベースで RESETLOGS 操作を実行するたびに、新規のインカネーションが作成されます。次の例では、リカバリ・カタログに登録されている trgt のデータベース・インカネーションをすべてリスト表示します。

```
LIST INCARNATION OF DATABASE trgt;
```

List of Database Incarnations

DB Key	Inc Key	DB Name	DB ID	CUR	Reset SCN	Reset Time
-----	-----	-----	-----	---	-----	-----
1	2	TRGT	1224038686	NO	1	02-JUL-01
1	582	TRGT	1224038686	YES	59727	10-JUL-01

ターゲット・データベースの V\$DATABASE\_INCARNATION と、リカバリ・カタログ・データベースの RC\_DATABASE\_INCARNATION を問い合わせると、同じ情報にアクセスできます。

**関連項目：** LIST OUTPUT 内の UNKNOWN データベース名の詳細は、15-43 ページの「リカバリ・カタログに UNKNOWN というデータベース名が表示される場合 : 例」を参照してください。



---

## Recovery Manager のメンテナンスの実行

この章では、Recovery Manager のリポジトリの管理方法を説明します。Recovery Manager の実装方法によっては、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。この章の項目は、次のとおりです。

- バックアップおよびコピーのクロスチェック
- バックアップおよびコピーの削除
- 複数の Recovery Manager のチャンネルでのクロスチェックおよび削除
- バックアップまたはコピー・レコードの可用性のステータスの変更
- バックアップまたはコピーの保存方針からの除外
- アーカイブ・ログおよびユーザー管理コピーのカタログ化
- Recovery Manager のレコードのカタログからの削除

## バックアップおよびコピーのクロスチェック

リカバリ・カタログや制御ファイル内のバックアップ・セットおよびイメージ・コピーに関するデータと、ディスクやメディア管理カタログ内の対応するデータとの間で同期が保たれていることを確認するために、**クロスチェック**を実行します。CROSSCHECK 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ているファイルに対してのみ有効です。

この項では、次の項目について説明します。

- [Recovery Manager のクロスチェック](#)
- [特定のバックアップおよびコピーのクロスチェック](#)
- [データベース・ファイルのバックアップおよびコピーのクロスチェック](#)

**関連項目：** クロスチェックの概説は、7-8 ページの「[Recovery Manager のバックアップおよびコピーのクロスチェック](#)」を参照してください。

## Recovery Manager のクロスチェック

クロスチェックを使用して、リポジトリ・レコードが物理的なステータスと一致しないバックアップとコピーについて、期限切れのリポジトリ情報を更新できます。たとえば、オペレーティング・システム・コマンドを使用してディスクからアーカイブ・ログを削除すると、実際には削除されていても、リポジトリはこのログがディスク上に存在することを示します。

バックアップまたはコピーがディスク上に存在する場合、CROSSCHECK コマンドでは、ファイルのヘッダーが有効かどうかを判断します。バックアップがテープに存在する場合、このコマンドは単にバックアップの存在をチェックします。

バックアップとコピーのステータスに可能な値は、AVAILABLE、UNAVAILABLE および EXPIRED です。LIST コマンドの出力またはリカバリ・カタログ・ビューを調べて、バックアップとコピーのステータスを判断します。

---

**注意：** CROSSCHECK コマンドでは、オペレーティング・システムのファイルやリポジトリ・レコードを削除できません。これらの操作には、DELETE コマンドを使用する必要があります。

---

**関連項目：** ファイルの削除方法およびリポジトリ・レコードの更新方法は、18-5 ページの「[バックアップおよびコピーの削除](#)」を参照してください。CROSSCHECK コマンドの構文とリポジトリのステータス値の詳細は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## 特定のバックアップおよびコピーのクロスチェック

LIST コマンドを発行して、作成したバックアップおよびコピーに関するレポートを取得し、次に CROSSCHECK コマンドを使用して、これらのファイルがまだ存在することをチェックできます。DELETE EXPIRED コマンドを実行すると、クロスチェックに失敗したコピーおよびバックアップのリポジトリ・レコードを削除できます。

### 指定したバックアップおよびコピーのクロスチェック

1. 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・コピーを指定します。たとえば、次のように発行します。

```
LIST BACKUP;  
LIST COPY;
```

2. 自動チャネルが構成されていない場合は、MAINTENANCE タイプのチャネルを1つ以上割り当てます。構成されている場合は、この手順をスキップします。たとえば、次のように発行します。

```
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK;  
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt;
```

3. 指定したバックアップまたはコピーの存在をチェックします。たとえば、次のように入力します。

```
CROSSCHECK BACKUPSET 1338, 1339, 1340;  
CROSSCHECK BACKUPPIECE TAG = 'nightly_backup';  
CROSSCHECK CONTROLFILECOPY '/tmp/control01.ctl';  
CROSSCHECK DATAFILECOPY 113, 114, 115;  
CROSSCHECK PROXY 789;
```

バックアップまたはコピーがすでに使用不能の場合、Recovery Manager は EXPIRED のマークを付けます。EXPIRED とマークされていても、現在は使用可能なバックアップ・セットに対しては AVAILABLE のマークを付けます。

4. メンテナンス・チャネルを手動で1つ以上割り当ててある場合は、これらのチャネルを解放します。

```
RELEASE CHANNEL;
```

## データベース・ファイルのバックアップおよびコピーのクロスチェック

LIST コマンドを発行して、どのデータベース・オブジェクトをバックアップまたはコピーしたかを判断し、次に CROSSCHECK コマンドを使用して、これらのバックアップとコピーが存在するかどうかをチェックできます。

### 指定しファイルのたバックアップおよびコピーのクロスチェック

1. 自動チャンネルが割り当てられていない場合は、MAINTENANCE タイプのチャンネルを1つ以上割り当てます。割り当てられている場合は、この手順をスキップします。たとえば、次のように発行します。

```
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK;  
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt;
```

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をチェックします。時間順序に従って、クロスチェックの範囲を限定します。たとえば、データ・ファイル `?/oradata/trgt/system01.dbf` の過去6か月間のバックアップすべてと、テープ上のすべてのログおよびサーバー・パラメータ・ファイルをチェックするには、次のように入力します。

```
CROSSCHECK BACKUP OF DATAFILE "?/oradata/trgt/system01.dbf"  
COMPLETED AFTER 'SYSDATE-180';  
CROSSCHECK BACKUP OF ARCHIVELOG ALL SPFILE;
```

3. 割り当てたメンテナンス・チャンネルをリリースします。

```
RELEASE CHANNEL;
```

**関連項目：** CROSSCHECK コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## バックアップおよびコピーの削除

Recovery Manager を使用して、バックアップ、コピーおよびアーカイブ・ログを削除できます。Recovery Manager は指定されたファイルを削除し、そのファイルのリポジトリ・レコードを削除します。

この項では、次の項目について説明します。

- [特定のバックアップおよびコピーの削除](#)
- [期限切れのバックアップおよびコピーの削除](#)
- [不要なバックアップとコピーの削除](#)

---

---

**注意：** DELETED ステータスのバックアップとコピーは、LIST コマンドの出力には表示されません。かわりに、v\$ 制御ファイル・ビューに問い合わせます。

---

---

### 関連項目：

- Recovery Manager の削除コマンドの概説は、[第 7 章「概念 III: メンテナンス」](#)を参照してください。
- DELETE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。
- リカバリ・カタログ・ビューの詳細は、『Oracle9i Recovery Manager リファレンス』参照してください。

## 特定のバックアップおよびコピーの削除

一般に、保持する必要のないバックアップおよびコピーの削除には DELETE コマンドを使用します。このコマンドは、物理ファイルを削除し、(カタログが使用されている場合は) カタログ・レコードを削除し、ターゲット制御ファイル内のレコードのステータスを DELETED に更新します。

### バックアップおよびコピーの削除と各リポジトリ・レコードの削除

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. LIST コマンドを実行して、バックアップおよびコピーの主キーを取得します。

```
LIST BACKUP OF DATABASE ARCHIVELOG ALL; # lists backups of db files and logs
LIST COPY;
```

2. 自動チャンネルが構成されていない場合は、次のようにメンテナンス・チャンネルを割り当てます。

```
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK;
```

3. DELETE コマンドを実行して、指定した物理ファイルおよびそのリポジトリ・レコードを削除します。recordSpec 句に指定するすべてのタイプのオブジェクトを削除できます。たとえば次のように指定します。

```
DELETE BACKUPPIECE 101;
DELETE CONTROLFILECOPY '/tmp/control01.ctl';
DELETE NOPROMPT ARCHIVELOG UNTIL SEQUENCE = 300;
```

4. さらに、次の例に示すように、DELETE BACKUP、DELETE COPY または DELETE ARCHIVELOG を使用してファイルを削除することもできます。

```
DELETE BACKUP OF TABLESPACE users DEVICE TYPE sbt; # deletes only tape backups
DELETE COPY OF CONTROLFILE LIKE '/tmp/%'; # LIKE specifies name of the copy
DELETE NOPROMPT BACKUP OF SPFILE COMPLETED BEFORE 'SYSDATE-7';
DELETE NOPROMPT ARCHIVELOG ALL
    BACKED UP 3 TIMES TO sbt; # backs up logs only if already backed up 3X to
    tape
```

Recovery Manager を対話的に実行する場合は、ファイルを削除する前に削除を確認するプロンプトが表示されます。NOPROMPT と指定すると、確認のプロンプトは表示されません。

5. 割り当てたメンテナンス・チャンネルをリリースします。

```
RELEASE CHANNEL;
```



## 期限切れのバックアップおよびコピーの削除

CROSSCHECK コマンドを使用して、リポジトリに記録されているバックアップおよびコピーがディスクまたはテープにまだ存在するかどうかを判断できます。バックアップおよびコピーが見つからない場合は、リポジトリ・レコードが EXPIRED ステータスに更新されます。その後、DELETE EXPIRED コマンドを使用して、期限切れのレコードを削除します。なんらかの理由により期限切れファイルが残っていると、DELETE EXPIRED コマンドはエラー・メッセージを返して中断します。

### 期限切れリポジトリ・レコードの削除

1. 自動チャンネルが構成されていない場合は、次のように、メンテナンス・チャンネルを1つ以上割り当てます。

```
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK;  
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt;
```

2. 最近クロスチェックを実行していない場合は、CROSSCHECK コマンドを発行します。たとえば、次のように発行します。

```
CROSSCHECK BACKUP;  
CROSSCHECK COPY;
```

3. 期限切れのバックアップを削除します。たとえば、次のように発行します。

```
DELETE EXPIRED BACKUP;  
DELETE EXPIRED COPY;
```

4. 割り当てたメンテナンス・チャンネルをリリースします。

```
RELEASE CHANNEL;
```

## 不要なバックアップとコピーの削除

不要になった（削除対象の）バックアップおよびコピーを削除するには、`DELETE OBSOLETE` コマンドを使用します。バックアップまたはコピーを不要とする基準は、次の方法で決定できます。

- `CONFIGURE` コマンドを使用して保存方針を構成します。
- `DELETE OBSOLETE` コマンドにオプションを使用します。

`DELETE OBSOLETE` コマンドは、物理ファイルを削除し、（カタログが使用されている場合は）カタログ・レコードを削除し、ターゲット制御ファイル内のレコードのステータスを `DELETED` に更新します。

### 保存方針により不要と判断されたバックアップまたはコピーの削除

`DELETE OBSOLETE` コマンドに他のオペランドを指定しない場合、保存方針により不要と定義されたバックアップおよびコピーがすべて削除されます。

#### 保存方針により不要と判断されたバックアップまたはコピーの削除

1. 自動チャネルが構成されていない場合は、バックアップまたはコピーを作成した各デバイス・タイプに対してメンテナンス・チャネルを割り当てます。`REPORT OBSOLETE` コマンドは、ディスク上のバックアップとコピーおよびテープ上のバックアップの不要状態に関してレポートします。たとえば、次のように入力します。

```
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt;  
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK;
```

2. 保存方針により不要と判断されたオブジェクトを削除するには、オプションを指定しないで `DELETE OBSOLETE` コマンドを発行します。

```
DELETE OBSOLETE;
```

`Recovery Manager` を対話的に実行する場合は、ファイルを削除する前に削除を確認するプロンプトが表示されます。`NOPROMPT` と指定すると、確認のプロンプトは表示されません。

3. 割り当てたメンテナンス・チャネルをリリースします。

```
RELEASE CHANNEL;
```

## DELETE コマンドを使用した、不要と判断されるバックアップおよびコピーの削除

DELETE OBSOLETE REDUNDANCY または DELETE OBSOLETE RECOVERY WINDOW コマンドを発行することで、不要なバックアップとコピーを削除できます。DELETE コマンドの冗長性設定またはリカバリ・ウィンドウ設定は、CONFIGURE RETENTION POLICY コマンドの設定をオーバーライドします。

### DELETE OBSOLETE コマンドにより不要とされたバックアップの削除

1. 自動チャネルが構成されていない場合は、次のようにメンテナンス・チャネルを割り当てます。

```
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt;
```

2. REDUNDANCY または RECOVERY WINDOW オプションを指定して DELETE OBSOLETE コマンドを実行します。これらのオプションは、削除ジョブで何が不要かを定義します。たとえば、次のように入力します。

```
DELETE OBSOLETE REDUNDANCY = 3;
DELETE OBSOLETE RECOVERY WINDOW OF 7 DAYS;
```

3. 割り当てたメンテナンス・チャネルをリリースします。

```
RELEASE CHANNEL;
```

## バックアップおよびコピーの削除の強制実行

Recovery Manager リポジトリでは、オブジェクトがあるステータスであることを示していても、メディア上での実際のステータスが異なる場合があります。たとえば、Recovery Manager リポジトリではバックアップ・セットが AVAILABLE に設定されていても、実際にはメディア管理カタログから消失している場合があります。このオブジェクトの削除を試みると、次のような警告を受け取ります。

RMAN-06207: 警告: DISK チャネルに対して 1 オブジェクトを削除できませんでした。

RMAN-06208: 一致しないステータスのため。CROSSCHECK コマンドを使用してステータスを修正してください。

List of Mismatched objects

=====

Object Type	Filename/Handle
Backup Piece	01d270ud_1_1

FORCE キーワードを指定すると、Recovery Manager にすべてのオブジェクトとそのリポジトリ・レコードを強制的に削除させることができます。Recovery Manager では、I/O エラーは無視されます。たとえば、次のように入力します。

```
DELETE FORCE NOPROMPT BACKUPSET TAG 'weekly_bkup';
```

**関連項目：**

- 7-15 ページ「リポジトリとメディアが対応していない場合の DELETE コマンドの動作」
- DELETE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## 複数の Recovery Manager のチャンネルでのクロスチェックおよび削除

この項では、次の項目について説明します。

- [メンテナンス・コマンド用の複数の Recovery Manager のチャンネルの割当て](#)
- [Recovery Manager による複数のチャンネルでのクロスチェックおよび削除の方法](#)
- [1 つのコマンドでのディスクおよびテープ・チャンネルのクロスチェック：例](#)
- [複数の Oracle Real Application Clusters ノードでのクロスチェック：例](#)
- [1 つの DELETE コマンドでのディスクおよびテープ・チャンネルの削除：例](#)
- [複数のチャンネルの割当ての解除：例](#)

## メンテナンス・コマンド用の複数の Recovery Manager のチャンネルの割当て

CROSSCHECK または DELETE コマンドを発行する前に、複数のメンテナンス・チャンネルを構成するか手動で割り当てることができます。Recovery Manager は、バックアップの作成に使用したチャンネルと同じデバイス・タイプを持つすべてのチャンネルで、各バックアップまたはコピーを検索します。複数チャンネル機能は、次のような使用を目的に設計されています。

- テープとディスクの両方に存在する、すべてのバックアップ・ピースまたはプロキシ・コピーのクロスチェックと削除を単一のコマンドで可能にします。
- 各バックアップ・ピースまたはプロキシ・コピーが 1 つのノードのみに存在する Oracle Real Application Clusters 構成で、クロスチェックおよび削除作業を正しく実行させます。

## Recovery Manager による複数のチャンネルでのクロスチェックおよび削除の方法

複数のメンテナンス・チャンネルを構成するか手動で割り当てて CROSSCHECK または DELETE コマンドを実行すると、該当するデバイス・タイプのすべてのチャンネル上でクロスチェックまたは削除が実行されます。

たとえば、構成済みチャンネルを使用するのではなく、手動でチャンネルを割り当てるとします。メディア・マネージャは構成済みですが、テープへのバックアップはまだ作成されていません。次のように、データベースのバックアップを 1 つのみディスクに作成してあります。

```
RUN
{
  ALLOCATE CHANNEL ch1 DEVICE TYPE DISK CONNECT 'SYS/sys_pwd@node2';
  BACKUP DATABASE;
}
```

次の一連のコマンドを、Recovery Manager プロンプトから発行するとします。

```
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK CONNECT 'SYS/oracle@node1';
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK CONNECT 'SYS/oracle@node2';
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt;
CROSSCHECK BACKUP OF DATABASE;
```

Recovery Manager は、最初の 2 つのチャンネルのデバイス・タイプがディスクであるため、この 2 つのチャンネルをチェックし、2 番目のチャンネルにバックアップを発見します。ただし、Recovery Manager では 3 番目の sbt チャンネルではクロスチェックを実行しません。これは、メディア・マネージャを使用してバックアップを作成したことがまだないためです。

---

**注意：** EXPIRED としてマークされたオブジェクトが実際に存在する場合は、DELETE EXPIRED コマンドを実行すると警告が発行されます。まれに、オブジェクトが存在している場合でも、リポジトリでそのオブジェクトが EXPIRED とマークされることがあります。たとえば、オブジェクトを含むディレクトリがクロスチェック時に破損していてその後修復された場合や、メディア・マネージャが正しく構成されていないために、実際は存在するバックアップが存在しないとレポートされている場合などです。

---

## 1つのコマンドでのディスクおよびテープ・チャンネルのクロスチェック： 例

Recovery Manager では、1つのコマンドで複数のメディアにクロスチェックを実行できます。sbt チャンネルが次のように構成されているとします。

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 1;  
CONFIGURE DEFAULT DEVICE TYPE sbt;
```

この場合は、次のコマンドを実行して、DISK と sbt の両方にクロスチェックを実行できます。

```
CROSSCHECK BACKUP OF DATABASE;
```

Recovery Manager では、sbt チャンネルと事前に構成されている DISK チャンネルの両方を使用して、クロスチェックを実行します。サンプル出力を次に示します。

```
allocated channel: ORA_SBT_TAPE_1  
channel ORA_SBT_TAPE_1: sid=12 devtype=SBT_TAPE  
channel ORA_SBT_TAPE_1: WARNING: Oracle Test Disk API  
using channel ORA_DISK_1  
crosschecked backup piece: found to be 'AVAILABLE'  
backup piece handle=/oracle/dbs/16c5esv4_1_1 recid=36 stamp=408384484  
crosschecked backup piece: found to be 'AVAILABLE'  
backup piece handle=/oracle/dbs/c-674966176-20000915-01 recid=37 stamp=408384496  
crosschecked backup piece: found to be 'AVAILABLE'  
backup piece handle=l2c5erb2_1_1 recid=32 stamp=408382820  
crosschecked backup piece: found to be 'AVAILABLE'  
backup piece handle=l3c5erba_1_1 recid=33 stamp=408382829  
crosschecked backup piece: found to be 'AVAILABLE'  
backup piece handle=l4c5erce_1_1 recid=34 stamp=408382863  
crosschecked backup piece: found to be 'AVAILABLE'  
backup piece handle=c-674966176-20000915-00 recid=35 stamp=408382869
```

自動 sbt チャンネルが構成されていない場合は、次の例のように、メンテナンス・チャンネルをディスクとテープに手動で割り当てることもできます。

```
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt;  
CROSSCHECK BACKUP OF DATABASE;
```

Recovery Manager では事前に構成されているディスク・チャンネルが使用されるため、ディスク・チャンネルを手動で割り当てる必要はありません。

## 複数の Oracle Real Application Clusters ノードでのクロスチェック : 例

この機能は、Oracle Real Application Clusters 構成で、テープ・バックアップがクラスタ内の様々なノード上にあり、バックアップが作成されたノード上でのみバックアップを表示できる場合に役立ちます。

複数ノード上でクロスチェックを実行するときは、バックアップが作成されたすべてのノードでチャンネルを割り当てることが重要です。ノードのチャンネルを省略するか、並列度をノード数より小さい値に設定すると、そのノード上で作成されたバックアップはリポジトリ内で EXPIRED としてマークされます。たとえば、次のように指定して、Recovery Manager を Oracle Real Application Clusters ノードとともに使用するように構成します。

```
CONFIGURE DEVICE TYPE DISK PARALLELISM 2;
CONFIGURE CHANNEL 1 DEVICE TYPE DISK CONNECT 'SYS/oracle@node_1';
CONFIGURE CHANNEL 2 DEVICE TYPE DISK CONNECT 'SYS/oracle@node_2';
```

次のコマンドを使用してこれらのクラスタ・ノードをクロスチェックします。

```
CROSSCHECK BACKUP;
```

## 1 つの DELETE コマンドでのディスクおよびテープ・チャンネルの削除 : 例

割り当て済みのすべてのチャンネルで削除を実行することもできます。次の例では、sbt チャンネルを構成します。

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 1;
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
```

次に、ディスクおよびテープからバックアップ・セットをすべて削除するコマンドを実行します。

```
DELETE BACKUPSET;
```

Recovery Manager では、削除の際に sbt チャンネルと事前に構成されている DISK チャンネルの両方を使用します（サンプル出力を次に示します）。Recovery Manager では、ファイルを削除する前に削除を確認するプロンプトが表示されます。

```
using channel ORA_SBT_TAPE_1
using channel ORA_DISK_1
```

List of Backup Pieces

BP Key	BS Key	Pc#	Cp#	Status	Device Type	Piece Name
388	387	1	1	AVAILABLE	SBT_TAPE	12c5erb2_1_1
397	396	1	1	UNAVAILABLE	SBT_TAPE	13c5erba_1_1
424	423	1	1	AVAILABLE	SBT_TAPE	14c5erce_1_1
428	427	1	1	AVAILABLE	SBT_TAPE	c-674966176-20000915-00
433	432	1	1	AVAILABLE	DISK	/oracle/dbs/16c5esv4_1_1
437	436	1	1	AVAILABLE	DISK	/oracle/dbs/c-674966176-20000915-01

```
Do you really want to delete the above objects (enter YES or NO)? y
deleted backup piece
backup piece handle=/oracle/dbs/16c5esv4_1_1 recid=36 stamp=408384484
deleted backup piece
backup piece handle=/oracle/dbs/c-674966176-20000915-01 recid=37 stamp=408384496
deleted backup piece
backup piece handle=12c5erb2_1_1 recid=32 stamp=408382820
deleted backup piece
backup piece handle=13c5erba_1_1 recid=33 stamp=408382829
deleted backup piece
backup piece handle=14c5erce_1_1 recid=34 stamp=408382863
deleted backup piece
backup piece handle=c-674966176-20000915-00 recid=35 stamp=408382869
```

次の例では、DISK および sbt のメンテナンス・チャンネルを手動で割り当て、ディスクおよびテープから特定のバックアップ・セットを削除します。

```
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK;
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt;
DELETE BACKUPSET 1,2,3,4,5;
```

Recovery Manager はチャンネル上で特定のバックアップ・セットを検索し、見つかったバックアップ・セットを削除します。どのチャンネルでもバックアップが見つからない場合、Recovery Manager はそのオブジェクトを削除済みとして制御ファイルにマークし、リカバリ・カタログのレコードを削除します（リカバリ・カタログを使用している場合）。

## 複数のチャンネルの割り当ての解除：例

次のコマンドを実行すると、割り当てられているメンテナンス・チャンネルをすべて解放できます。

```
RELEASE CHANNEL;
```



## バックアップまたはコピー・レコードの可用性のステータスの変更

バックアップまたはコピーが見つからない場合や、サイト外に移された場合には、CHANGE ... UNAVAILABLE コマンドを実行します。RESTORE または RECOVER コマンドでは、UNAVAILABLE とマークされたファイルは使用されません。後でファイルが見つかった場合や、メイン・サイトに戻された場合は、CHANGE ... AVAILABLE を発行して、再び使用可能なマークを付けることができます。

この項では、次の項目について説明します。

- バックアップまたはコピーを UNAVAILABLE としてマーク
- バックアップまたはコピーを AVAILABLE としてマーク

### バックアップまたはコピーを UNAVAILABLE としてマーク

CHANGE ... UNAVAILABLE コマンドを発行して、バックアップまたはコピーに UNAVAILABLE とマークします。

#### リポジトリ内のファイルのステータスを UNAVAILABLE としてマークする方法

1. LIST コマンドを発行し、Recovery Manager のバックアップおよびコピーの可用性ステータスを判断します。たとえば、次のように発行します。

```
LIST BACKUP;  
LIST COPY;
```

2. CHANGE ... UNAVAILABLE コマンドを実行し、Recovery Manager リポジトリ内のバックアップまたはコピーを UNAVAILABLE としてマークします。たとえば、次のように入力します。

```
CHANGE DATAFILECOPY '/tmp/control01.ctl' UNAVAILABLE;  
CHANGE COPY OF ARCHIVELOG SEQUENCE BETWEEN 1000 AND 1012 UNAVAILABLE;  
CHANGE BACKUPSET 12 UNAVAILABLE;  
CHANGE BACKUP OF CONTROLFILE UNAVAILABLE;  
CHANGE BACKUP OF SPFILE TAG "TAG20020208T154556" UNAVAILABLE;
```

## バックアップまたはコピーを AVAILABLE としてマーク

CHANGE ... AVAILABLE コマンドを使用して、バックアップまたはコピーに AVAILABLE とマークします。このコマンドでは、ファイルの有無や妥当性はチェックされず、リポトリ・レコードが AVAILABLE に更新されるのみであることに注意してください。CROSSCHECK を実行すると、ファイルの妥当性をチェックできます。

### リポトリ内のファイルのステータスを AVAILABLE としてマークする方法

1. LIST コマンドを発行し、Recovery Manager のバックアップおよびコピーの可用性ステータスを判断します。たとえば、次のように発行します。

```
LIST BACKUP;  
LIST COPY;
```

2. 自動チャネルが割り当てられていない場合は、メンテナンス・チャネルを 1 つ以上割り当てます。割り当てられている場合は、この手順をスキップします。たとえば、次のように発行します。

```
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt;  
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK;
```

3. 今まで使用できなかったバックアップまたはコピーを使用できるようにする場合は、CHANGE ... AVAILABLE コマンドを発行し、Recovery Manager リポトリ内で AVAILABLE としてマークします。たとえば、次のように入力します。

```
CHANGE DATAFILECOPY '/tmp/system01.dbf' AVAILABLE;  
CHANGE BACKUPSET 12 AVAILABLE;  
CHANGE BACKUP OF SPFILE TAG "TAG20020208T154556" AVAILABLE;
```

**関連項目：** CHANGE コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## バックアップまたはコピーの保存方針からの除外

BACKUP ...KEEP コマンドは、ユーザーの保存方針で指示された時間よりも長い間、バックアップを（潜在的にはサイト外で）アーカイブできます。ただし、このバックアップは完全には有効なバックアップであり、他の Recovery Manager のバックアップと同様にリストアップ可能です。このタイプのバックアップは、**長期バックアップ**と呼ばれます。

---

**注意：** KEEP FOREVER 句を使用する場合は、リカバリ・カタログを使用する必要があります。

---

不完全リカバリのためのアーカイブ・ログを保存する場合は KEEP ... LOGS を、保存しない場合は KEEP ... NOLOGS を指定します。NOLOGS は非一貫性バックアップでは有効でないので注意してください。

バックアップまたはコピーの KEEP ステータスを変更するには、CHANGE コマンドを使用します。たとえば、長期バックアップを保持する必要がないと判断することがあります。BACKUP ... KEEP で使用できるオプションを、CHANGE ... KEEP でも使用できます。

### バックアップまたはコピー・レコードの KEEP ステータスの変更

バックアップまたはコピーの保存ステータスを変更します。保存方針からファイルを除外する場合は CHANGE ... KEEP を発行し、ファイルを保存方針に準拠させる場合は CHANGE ... NOKEEP を発行します。次の例では、保存方針によりバックアップ・セットが不要とマークすることが可能になります。

```
CHANGE BACKUPSET 231 NOKEEP;
```

この例では、データ・ファイルのコピーを保存方針から 180 日（6 か月）間除外します。

```
CHANGE DATAFILECOPY '/tmp/system01.dbf' KEEP UNTIL 'SYSDATE+180';
```

## アーカイブ・ログおよびユーザー管理コピーのカタログ化

リポジトリに記録されていないアーカイブ・ログの存在と、Recovery Manager 以外の手段で作成されたファイル・コピーの存在を Recovery Manager に認識させることができます。この項では次の項目について説明します。

- [アーカイブ・ログおよびユーザー管理コピーのカタログ化](#)
- [ユーザー管理コピーをカタログ化する時期](#)
- [ユーザー管理コピーのカタログ化](#)

## アーカイブ・ログおよびユーザー管理コピーのカタログ化

制御ファイルには、ターゲット・データベースにより生成されたすべてのアーカイブ・ログのレコードが保持されます。リカバリ・カタログを使用する場合は、Recovery Manager がアーカイブ・ログ情報を制御ファイルからカタログに伝播します。制御ファイルのバックアップをリストアすることが必要な場合、リカバリ中にアーカイブの宛先または形式を変更すると、リカバリに必要なアーカイブ・ログに関する情報はリポジトリに保持されません。このため、これらのログをリカバリに使用する場合は、ログをカタログする必要がある場合があります。

ユーザー管理コピー（オペレーティング・システム・コマンドにより作成されたコピー）を作成する場合は、別のケースになります。この場合は、リポジトリにそのレコードがありません。オペレーティング・システムのユーティリティ、たとえば UNIX の `cp` コマンドなどを使用してコピーを作成した場合は、手動で Recovery Manager に通知する必要があります。

次のような場合、Recovery Manager の CATALOG コマンドを実行します。

- ユーザー管理のデータ・ファイルのコピー、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・カタログと制御ファイルに追加する場合。
- 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場合。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て使用し、後で増分バックアップの実行が可能となります。
- Recovery Manager の使用前に作成された Oracle8 リリース 8.0 以上のデータベースのユーザー管理コピーの存在を記録する場合。
- Oracle8 リリース 8.0 以上に移行する前に、Oracle7 バックアップの存在を記録する場合。

## ユーザー管理コピーをカタログ化する時期

データベース・ファイルのユーザー管理コピーは、次のような場合にカタログ化します。

- 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・ファイルまたはアーカイブ・ログのコピーを作成する場合。
- 18-20 ページの「[Oracle7 データベース・ファイルのユーザー管理コピーのカタログ化](#)」に示されている基準に一致する Oracle7 データベース・ファイルのユーザー管理コピーをカタログ化する場合。これらのデータ・ファイルのコピーを使用すると、移行した後、移行済みデータベースのバックアップを作成する前にデータベースがクラッシュした場合でもデータベースをリカバリできます。

---

**注意：** バックアップ・セットまたはピースは、カタログ化できません。

---

## 一貫性および非一貫性のユーザー管理コピーのカタログ化

ユーザー管理コピーを作成する場合、たとえば 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などは、必ずコピーをカタログ化するようにします。ユーザー管理コピーを作成するときは、ALTER TABLESPACE ... BEGIN/END BACKUP 文を使用して、オンライン表領域からデータ・ファイルのコピーを作成できます。Recovery Manager ではこのようなデータ・ファイルのコピーは作成しませんが、CATALOG コマンドを使用してこれらのコピーをリカバリ・カタログに追加し、Recovery Manager で認識できるようにすることができます。

カタログ化するユーザー管理のコピーは、次の条件を満たしている必要があります。

- 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・ファイルのコピー、制御ファイル・コピーまたはアーカイブ REDO ログ・コピーであること

たとえば、データ・ファイルがミラー化されたディスク・ドライブに格納されている場合は、ミラー化を解除することでユーザー管理コピーを作成できます。この例では、ミラー化の解除後に、CATALOG コマンドを使用してユーザー管理コピーの存在を Recovery Manager に通知します。ミラー化を再作成する前に、CHANGE ... UNCATALOG コマンドを実行して、ファイルのコピーが存在しなくなったことを Recovery Manager に通知します。

## Oracle7 データベース・ファイルのユーザー管理コピーのカタログ化

Recovery Manager は、次のような特殊な状況を除き、Oracle7 のファイルをカタログ化することはできません。

- Oracle7 から現行のリリースへの移行時に、移行ユーティリティの実行前に Oracle7 データベースを完全にシャットダウンし、データ・ファイルをバックアップした場合。
- バックアップ時に OFFLINE NORMAL または読取り専用である表領域のバックアップを作成し、移行の前にこれらの表領域をオンラインまたは読込み / 書込みにしない場合。

これらのバックアップは、リカバリする際に Oracle7 データベースの REDO を必要としないため、Oracle7 のコピーでカタログ化することができます。Recovery Manager は、他のバックアップが存在しない場合は、これらのコピーを Oracle9i データベースでカタログ化およびリストアできます。

次の例は、Recovery Manager でのカタログ化が可能な、Oracle7 のデータ・ファイルのバックアップの生成を示しています。

1. Oracle7 データベースを完全に停止します。
2. オペレーティング・システムのユーティリティ（UNIX の `cp` コマンドや Windows の `COPY` コマンドなど）を使用して、データ・ファイルのユーザー管理コピーを作成します。
3. データベースを現行のリリースへ移行します。

移行前に作成されたバックアップは、移行後に作成されたユーザー管理コピーと同一であるため、これらは Recovery Manager でカタログ化できます。

**関連項目：** Oracle データベースの移行方法は、『Oracle9i データベース移行ガイド』を参照してください。

## ユーザー管理コピーのカタログ化

CATALOG コマンドを使用して、ユーザー管理のコピーの情報をリカバリ・カタログに伝播させます。

### ユーザー管理コピーのカタログ化

1. 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、コピーを作成します。バックアップの進行中、データベースがオープン状態でデータ・ファイルがオンラインの場合は、ALTER TABLESPACE BEGIN/END BACKUP が必要なので注意してください。この例では、オンライン・データ・ファイルをバックアップします。

```
SQL> ALTER TABLESPACE users BEGIN BACKUP;  
% cp $ORACLE_HOME/oradata/trgt/users01.dbf /tmp/users01.dbf;  
SQL> ALTER TABLESPACE users END BACKUP;
```

2. ターゲット・データベースおよび、必要に応じてリカバリ・カタログ・データベースに接続後、CATALOG コマンドを起動します。たとえば、次のように入力します。

```
CATALOG DATAFILECOPY '/tmp/users01.dbf';
```

**関連項目：** CATALOG コマンドの構文は、『Oracle9i Recovery Manager リファレンス』を参照してください。

## Recovery Manager のレコードのカタログからの削除

この項では次の項目について説明します。

- [Recovery Manager のレコードのカタログからの削除](#)
- [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削除したファイルのレコードの削除](#)
- [ステータス DELETED によるカタログ・レコードの削除](#)

## Recovery Manager のレコードのカタログからの削除

Recovery Manager のリポジトリ・レコードに次のようなアクションを実行する場合は、`CHANGE ... UNCATALOG` コマンドを実行します。

- リカバリ・カタログ（使用している場合）から特定のバックアップ・レコードまたはコピー・レコードを削除
- 制御ファイル・リポジトリ内のバックアップ・レコードまたはコピー・レコードのステータスを `DELETED` に更新

Recovery Manager では、指定された物理ファイルは操作しません。ファイルのリポジトリ・レコードを変更するのみです。

このコマンドは、Recovery Manager 以外の手段でバックアップまたはコピーを削除した場合に使用できます。たとえば、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アーカイブ REDO ログを削除した場合は、`CHANGE ARCHIVELOG ... UNCATALOG` を発行して、このログをリポジトリから削除します。

Oracle9i より前のリリースでは、Recovery Manager がリカバリ・カタログ内のレコードを削除せずに、レコードのステータスを `DELETED` に更新する場合があります。Oracle9i では、Recovery Manager はカタログのレコードを常に削除します。`DELETED` にはしません。このため、カタログ・レコードのステータスを `DELETED` としてマークするのは、カタログが更新された場合、またはカタログがバックアップ制御ファイルから再同期化された場合のみです。バックアップおよびコピーのリポジトリ・レコードで、ステータスが `DELETED` になっているレコードは、`prgrmanc.sql` スクリプトを使用してすべて削除できます。このスクリプトは、オペレーティング・システム指定の場所（UNIX では `$ORACLE_HOME/rdbms/admin`）に格納されています。

## 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削除したファイルのレコードの削除

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削除されたファイルのカタログ・レコードを削除するには、`CHANGE ... UNCATALOG` コマンドを使用します。

### オペレーティング・システムのユーティリティで削除したバックアップまたはコピーのカタログ・レコードの削除

1. オペレーティング・システムのコマンドを使用してオペレーティング・システムから削除したバックアップまたはコピーに対して、`CHANGE ... UNCATALOG` コマンドを実行します。この例では、制御ファイルとデータ・ファイル 1 のコピーへの参照をリポジトリから削除します。

```
CHANGE CONTROLFILECOPY '/tmp/control01.ctl' UNCATALOG;  
CHANGE DATAFILECOPY '/tmp/system01.dbf' UNCATALOG;
```



2. 必要に応じて、RC\_DATAFILE\_COPY や RC\_CONTROLFILE\_COPY など該当するリカバリ・カATALOG・ビューを表示し、指定したレコードが削除されたことを確認します。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。

```
SELECT CDF_KEY, STATUS
FROM RC_DATAFILE_COPY
WHERE CDF_KEY = 4833;

CDF_KEY    STATUS
-----
0 rows selected.
```

## ステータス DELETED によるカATALOG・レコードの削除

ステータスが DELETED のリカバリ・カATALOG・レコードを削除するには、prgrmanc.sql スクリプトを使用します。Oracle9i より前のリリースでは、Recovery Manager は物理ファイルを削除した後、リカバリ・カATALOGのレコードを削除せずに、レコードのステータスを DELETED に更新していました。

Oracle9i では、Recovery Manager は常にカATALOGのレコードを削除し、レコードのステータスを DELETED に更新することはありません。ただし、Oracle9i より前のリリースで作成されたカATALOGを現行のリリースにアップグレードすると、ステータスが DELETED のレコードがリカバリ・カATALOGに含まれることがあります。このような特殊ケースの場合は、prgrmanc.sql スクリプトを実行できます。

### ステータスが DELETED のコピー・レコードとバックアップ・レコードをすべて削除する方法

1. SQL\*Plus セッションを開始し、リカバリ・カATALOGに接続します。この例では、ユーザー rman でデータベース rcat に接続します。

```
% sqlplus rman/cat@catdb
```

2. prgrmanc.sql スクリプトを実行します。このスクリプトは、オペレーティング・システム指定の場所（UNIX では \$ORACLE\_HOME/rdbms/admin）に格納されています。

```
SQL> @prgrmanc
```

このスクリプトは、状態が DELETED のレコードを、すべてリカバリ・カATALOGから削除します。



---

# 索引

## A

---

ALTER DATABASE 文  
    OPEN RESETLOGS 句, 16-11  
    RENAME FILE 句, 10-21  
ALTER SYSTEM 文  
    KILL SESSION 句, 15-13  
ALTER TABLESPACE 文  
    BEGIN BACKUP 句, 18-19  
    END BACKUP 句, 18-19  
AVAILABLE オプション  
    CHANGE, 18-15

## B

---

BACKUP COPIES パラメータ  
    CONFIGURE コマンド, 8-25  
BACKUP OPTIMIZATION オプション  
    CONFIGURE, 9-31  
Backup Solutions Program (BSP), 4-19  
BACKUP\_TAPE\_IO\_SLAVES 初期化パラメータ,  
    8-25, 9-18  
BACKUP コマンド, 5-14, 9-57  
    BACKUPSET オプション, 5-28, 9-30  
    COPIES パラメータ, 5-23  
    CURRENT CONTROLFILE オプション, 9-11  
    DELETE ALL INPUT オプション, 5-17  
    DELETE INPUT オプション, 5-17, 7-15  
    FORMAT パラメータ, 5-31  
    KEEP オプション, 5-61  
    NOT BACKED UP SINCE 句, 5-68, 9-31  
    PLUS ARCHIVELOG オプション, 5-19  
    PROXY ONLY オプション, 5-15  
    PROXY オプション, 5-15

    SKIP OFFLINE オプション, 9-39  
    VALIDATE オプション, 5-75  
BEGIN BACKUP 句  
    ALTER TABLESPACE 文, 18-19  
BLOCKRECOVER コマンド, 6-11, 10-40  
BSP, 「Backup Solutions Program (BSP)」を参照

## C

---

catalog.sql スクリプト, 16-4  
CATALOG コマンド, 16-8, 18-18  
catproc.sql スクリプト, 16-4  
CHANGE コマンド, 7-8, 18-2  
    AVAILABLE オプション, 7-18, 18-15  
    KEEP オプション, 7-19  
    UNAVAILABLE オプション, 18-15  
    UNCATALOG オプション, 18-22  
CONFIGURE コマンド, 17-18  
    BACKUP COPIES パラメータ, 5-23  
    BACKUP OPTIMIZATION オプション, 8-24  
    CHANNEL オプション, 5-8  
    CLEAR オプション, 5-10, 8-17  
    DEFAULT DEVICE TYPE 句, 5-6  
    DEVICE TYPE 句, 5-5  
    EXCLUDE オプション, 8-26  
    RETENTION POLICY 句, 5-55, 5-56  
CONTROL\_FILE\_RECORD\_KEEP\_TIME 初期化パラ  
    メータ, 16-37  
    Recovery Manager レコードの上書きの防止, 16-37  
CONTROL\_FILES 初期化パラメータ, 10-22, 11-15  
COPIES オプション  
    BACKUP, 9-19  
COPY コマンド, 5-69  
    KEEP オプション, 5-72  
CROSSCHECK コマンド, 18-2

## D

---

DB\_FILE\_NAME\_CONVERT 初期化パラメータ, 11-14  
    Recovery Manager DUPLICATE コマンドで使用,  
    12-9

DB\_NAME 初期化パラメータ, 11-14

DBMS\_PIPE パッケージ, 4-11  
    Recovery Manager を使用, 2-12

DBNEWID ユーティリティ, 4-13, 16-6, 16-9

DB 識別子  
    DBNEWID で設定, 16-6, 16-9  
    DBNEWID で変更, 4-13  
    コピーしたデータベースの登録に関する問題, 16-9  
    障害時リカバリ中に設定, 10-18

DELETE コマンド, 7-13, 18-6, 18-19  
    EXPIRED オプション, 18-7  
    OBSOLETE オプション, 5-60

DUPLICATE コマンド, 6-14

## E

---

END BACKUP 句  
    ALTER TABLESPACE 文, 18-19

EXCLUDE オプション  
    CONFIGURE, 8-26

EXIT コマンド, 2-13

EXPIRED オプション  
    DELETE, 18-7

## F

---

FILESERSET パラメータ  
    BACKUP コマンド  
        バックアップ・セットのファイル数の指定, 5-33

FORCE オプション  
    DELETE コマンド, 7-15, 18-9

## I

---

INCARNATION オプション  
    LIST, 16-11  
    RESET DATABASE, 16-11

INCLUDE CURRENT CONTROLFILE オプション  
    BACKUP コマンド, 9-12

I/O エラー  
    削除中に無視, 7-15, 18-9  
    バックアップへの影響, 5-51

## K

---

KEEP オプション  
    BACKUP, 5-61, 18-17  
    CHANGE, 7-19

## L

---

LARGE\_POOL\_SIZE 初期化パラメータ, 14-13

LIST コマンド, 7-2, 17-3  
    INCARNATION オプション, 16-11

LOCK\_NAME\_SPACE 初期化パラメータ, 11-14

LOG\_FILE\_NAME\_CONVERT 初期化パラメータ,  
    11-14

## M

---

MAXPIECESIZE パラメータ  
    SET コマンド, 5-32, 8-7

MAXSETSIZE パラメータ  
    BACKUP コマンド, 5-37, 8-23  
    CONFIGURE コマンド, 8-23

MAXSIZE パラメータ  
    RECOVER コマンド, 6-9

## N

---

NOARCHIVELOG モード  
    バックアップ, 9-51

NOT BACKED UP SINCE 句  
    BACKUP コマンド, 9-31

## O

---

OPEN RESETLOGS 句  
    ALTER DATABASE 文, 16-11

Oracle Enterprise Manager  
    ジョブ・スクリプト, 4-11  
    バックアップ・ウィザード, 9-5  
    リカバリ・ウィザード, 10-2

Oracle Real Application Clusters  
    Recovery Manager を使用した接続, 2-8  
    Recovery Manager を使用したリカバリ, 10-27  
    バックアップ, 9-52  
    バックアップの自動位置検索, 10-27

## P

---

Point-in-Time リカバリ  
表領域, 11-2 ~ 11-12  
PRINT SCRIPT コマンド, 7-8  
PROXY ONLY オプション  
BACKUP, 5-15  
PROXY オプション  
BACKUP, 5-15

## Q

---

QUIT コマンド, 2-13

## R

---

RATE オプション  
ALLOCATE CHANNEL, 5-40  
CONFIGURE CHANNEL, 5-40  
Recovery Manager, 14-2  
DBMS\_PIPE パッケージ, 2-12  
Recovery Manager コマンドの使用法, 4-5  
RPC コール, 15-30  
sbttest プログラム, 4-19  
アーカイブ REDO ログ  
バックアップ, 9-13  
イメージ・コピー・バックアップ, 5-69  
エラー, 15-2, 15-3, 15-4  
解釈, 15-7  
エラー・コード  
メッセージ番号, 15-4  
概要, 1-6, 4-2, 4-5  
環境  
定義, 4-2  
監視, 15-16, 15-24  
起動, 2-4  
基本的なコマンドを使用した, 3-3  
構成  
デバイス・タイプ, 5-5  
デフォルト・デバイス・タイプ, 5-6  
表示, 7-7, 17-18  
コマンド  
BACKUP, 5-14, 5-15, 9-57  
CATALOG, 16-8, 18-18  
CHANGE, 7-8, 18-2  
DELETE, 18-19  
DUPLICATE, 13-2

EXECUTE SCRIPT, 16-19  
LIST, 16-11  
REGISTER, 16-9  
REPORT, 17-13  
RESTORE, 10-7  
RESYNC CATALOG, 16-27  
コマンド・ファイルの使用, 4-9  
ジョブ・コマンド, 4-8  
スタンドアロン・コマンド, 4-7  
タイプ, 4-7  
対話形式の使用法, 4-9  
コマンドのコンパイルおよび実行, 4-6  
コマンドの終了, 15-12  
コマンドの対話的な使用法, 4-9  
サンプル・スクリプト, 3-2  
時間を指定するパラメータの設定, 8-28  
実行可能ファイル  
定義, 1-6  
使用する理由, 1-2  
ジョブ  
進行状況の監視, 15-20  
スタンバイ・データベース  
DB\_FILE\_NAME\_CONVERT 初期化パラメータ,  
13-8  
LOG\_FILE\_NAME\_CONVERT 初期化パラ  
メータ, 13-8  
RC\_ARCHIVED\_LOG ビューの解析, 9-25  
Recovery Manager およびスタンバイ・インスタ  
ンスの起動, 13-12  
Recovery Manager の BACKUP コマンドを使用  
したスタンバイ制御ファイルの作成, 13-3  
Recovery Manager の COPY コマンドを使用した  
スタンバイ制御ファイルの作成, 13-5  
Recovery Manager を使用した準備, 13-2  
SQL を使用したスタンバイ制御ファイルの作成,  
13-5  
アーカイブ REDO ログのバックアップ, 9-26  
イメージ・コピーを使用した作成, 13-23  
作成, 6-16, 13-9  
作成について, 13-2  
作成の概要, 13-9  
スタンバイ制御ファイルのコピー, 13-6  
スタンバイ制御ファイルの作成, 13-3  
スタンバイ・データ・ファイルの命名, 13-7  
バックアップ, 9-23, 9-28  
バックアップの概要, 9-23  
バックアップの制限事項, 9-24

- ストアド・スクリプト, 4-10
- スナップショット制御ファイルの位置, 8-29
- 整合性チェック, 5-72
- 制約
  - バックアップ, 5-50
- 接続
  - コマンドラインのオプション, 2-4
- 切断, 2-13
- 増分バックアップ
  - 差分, 5-45
  - 累積, 5-47
  - レベル 0, 5-44
- 多重化
  - データ・ファイル, 5-20
- チャンネル, 5-2
  - 一般的な構成, 5-8
  - 特定の構成, 5-9
  - ネーミング規則, 5-7
- チャンネルの割当て, 14-9
- 定義, 1-2
- ディスク API のテスト, 8-5
- ディスク・バッファの割当て, 14-2
- データ・ファイルのコピー
  - バックアップ, 9-9
- データベース・キャラクタ・セット, 8-28
- データベース接続, 2-2
  - Oracle Real Application Clusters 構成, 2-8
    - カタログを使用, 2-6
    - カタログを使用しない, 2-5
    - ターゲットに必要な SYSDBA, 2-3
    - パスワードの非表示, 2-11
    - パスワード・ファイルを使用, 2-6
    - 複製データベース, 2-9
    - 補助データベース, 2-9
    - リカバリ・カタログを使用, 2-6, 2-9
- データベースの登録, 16-9
- テープ・バッファの割当て, 14-5
- 同期 I/O と非同期 I/O, 14-6
- 入力ファイルの割当て, 14-10
- パイプ・インタフェース, 4-11
- パイプを使用, 2-12
- 破損データ・ファイル・ブロック, 5-73, 5-74
  - I/O エラーの処理, 5-51
- バックアップ
  - アーカイブ REDO ログ, 9-13, 9-14, 9-15, 9-17
  - イメージ・コピー, 5-69
  - 概要, 9-4
  - 再起動可能, 5-68
  - 最適化, 5-62
  - スタンバイ・データベース, 9-29
  - 制御ファイル, 9-9, 9-11, 9-12
  - 制御ファイル・コピー, 9-12
  - 制御ファイルの自動バックアップ, 5-52, 8-18, 9-9
  - 増分, 9-18, 9-19, 9-20
  - タイプ, 5-41
  - タグの使用方法, 5-71
  - 妥当性チェック, 9-32
  - 長期, 5-61
  - データ・ファイル, 9-8, 9-9, 9-30, 9-31, 9-32
  - データベース全体, 9-7
  - テスト, 5-75, 9-32
  - バックアップ, 5-28
  - 表領域, 9-8, 9-9, 9-30, 9-31, 9-32
  - 不要のバッチ削除, 5-60
  - バックアップ・セット, 5-37
    - 概要, 5-14
    - バックアップ, 9-30
  - バックアップのタイプ, 5-13, 5-69
    - 二重バックアップ・セット, 5-23
  - バックアップのためのタグ, 5-71
  - バックアップの停止, 15-29
  - パフォーマンス, 14-11
    - 監視, 15-16
  - 表領域の Point-in-Time リカバリ, 6-10
  - ファイルの削除
    - 概要, 7-12
  - 不完全リカバリ
    - 現行の制御ファイル, 10-11
  - 複製データベース
    - 作成方法, 12-3
  - 分裂ブロックの検出, 5-74
  - 保存方針
    - 構成, 8-21
  - メタデータ, 4-12, 12-1, 16-1, 18-1
    - 制御ファイルへの格納, 4-16
  - メディア管理
    - Backup Solutions Program (BSP), 4-19
    - クロスチェック, 7-8
    - テスト, 4-19
    - ファイルのバックアップ, 4-18
    - ファイルのリストア, 4-18
    - プロキシ・コピー, 4-19
    - メディア・マネージャ、リンク, 8-3

- ユーザー管理リカバリとの機能の比較, 1-4
  - リカバリ
    - オープン状態のデータベース, 10-10
    - 全体的なメディア障害の後, 10-37
    - バックアップ制御ファイルを使用, 10-16
    - 不完全, 10-11
  - リカバリ・カタログ, 4-12
    - アップグレード, 16-33
    - 可用性, 16-30
    - クロスチェック, 7-9
    - 互換性, 4-16
    - サイズの管理, 16-22
    - 再同期化, 16-13
    - 使用した運用, 1-8, 4-12
    - 使用・不使用の決定, 3-14
    - 新規データベースへの移動, 16-28
    - スキーマ, 8-32
    - スキーマの変更後の更新, 16-17
    - スナップショット制御ファイル, 4-15
    - ターゲット・データベースの登録, 4-13, 16-7
    - 問合せ, 17-2
    - 同期, 4-14
    - 内容, 4-14
    - バックアップ, 16-24
    - リカバリ, 16-27
    - リカバリ・カタログなしでの運用, 4-16
    - レコードの削除, 18-22
  - リカバリ・カタログのクロスチェック, 7-9
  - リスト, 17-3
  - リストア, 10-5
    - アーカイブ REDO ログ, 10-23
    - 新しいホストへリストア, 10-25
    - データ・ファイル, 6-2
    - デフォルトではないファイル名の使用, 10-21
    - リカバリ・カタログを使用せずに制御ファイルを新しい場所にリストア, 10-22
  - リターン・コード, 15-9
  - リポジトリ
    - 定義, 1-7
  - 累積増分バックアップを使用, 14-11
  - レポート, 7-2, 17-12
    - 概要, 7-3
    - データベース・スキーマ, 17-17
    - バックアップが必要なオブジェクト, 17-13
    - 不要なバックアップ, 17-15, 17-16
  - Recovery Manager コマンドの終了, 15-12
  - Recovery Manager コマンドの取消, 15-12
  - Recovery Manager のエラー・スタックの解釈, 15-7
  - Recovery Manager の監視, 15-16
  - Recovery Manager の起動
    - データベースに接続しない, 2-4
  - Recovery Manager の構成の消去, 5-10, 8-17
  - Recovery Manager の終了, 2-13
  - Recovery Manager のスタンドアロン・コマンド, 4-7
  - Recovery Manager のテスト
    - Media Management API, 15-9
    - バックアップ, 5-75, 9-32
    - メディア・マネージャのテスト, 4-19
  - Recovery Manager のバックアップの再起動, 9-31
  - Recovery Manager のメタデータの管理, 12-1, 16-1, 18-1
  - Recovery Manager, 「Recovery Manager」を参照
  - RECOVERY WINDOW パラメータ
    - CONFIGURE コマンド, 8-21
  - RECOVER コマンド, 6-6, 6-9
  - REGISTER コマンド, 16-7, 16-9
  - RENAME DATABASE 句
    - ALTER DATABASE 文, 10-21
  - REPORT OBSOLETE コマンド, 5-60
  - REPORT コマンド, 17-12
    - NEED BACKUP オプション, 17-13
  - RESET DATABASE コマンド
    - INCARNATION オプション, 16-11
  - RESTORE コマンド, 6-2, 10-7
    - FORCE オプション, 6-5
  - RESYNC CATALOG コマンド, 4-14, 16-13
    - FROM CONTROLFILECOPY オプション, 16-27
- ## S
- 
- sbtio.log
    - Recovery Manager, 15-3
  - sbttest プログラム, 4-19, 15-9
  - SET コマンド
    - BACKUP COPIES パラメータ, 5-23
    - MAXCORRUPT オプション, 9-57
  - SHOW コマンド, 5-5, 7-7, 17-18
  - SKIP OFFLINE オプション
    - BACKUP, 9-39
  - SKIP READONLY オプション
    - BACKUP, 9-39
  - SYSDBA オプション
    - 暗黙的に Recovery Manager がターゲットへ接続することが前提, 2-3

## T

---

TSPITR, 「表領域の Point-in-Time リカバリ」を参照

## U

---

UNAVAILABLE オプション

CHANGE, 7-18, 18-15

UNCATALOG オプション

CHANGE, 18-22

リポジトリ・レコードの削除, 18-22

## V

---

V\$ARCHIVED\_LOG ビュー, 6-8

V\$BACKUP\_ASYNC\_IO, 15-16

V\$BACKUP\_CORRUPTION ビュー, 5-51

V\$BACKUP\_SYNC\_IO, 15-16

V\$COPY\_CORRUPTION ビュー, 5-51

V\$DATABASE\_BLOCK\_CORRUPTION ビュー, 9-32,  
10-42

V\$PROCESS ビュー, 15-16

V\$PROXY\_DATAFILE ビュー, 5-15

V\$SESSION\_LONGOPS ビュー, 15-16

V\$SESSION\_WAIT ビュー, 15-16

V\$SESSION ビュー, 15-16

## あ

---

アーカイブ REDO ログ

Recovery Manager で削除に失敗, 15-36

Recovery Manager を使用したリストア, 10-23

カタログ化, 16-8, 18-18

バックアップ, 5-19, 9-47

Recovery Manager を使用, 9-13, 9-14, 9-15,  
9-17

バックアップ後の削除, 5-17, 7-14

バックアップ時に生成, 5-19

リストア後の削除, 6-9

アラート・ログ

Recovery Manager で有効, 15-2

制御ファイル・レコードの上書きの監視, 16-38

制御ファイル・レコードのメッセージ, 16-37

## い

---

一貫性バックアップ

Recovery Manager を使用, 9-6

一般的なチャンネル

定義, 8-13

イメージ・コピー, 5-69

リストアのテスト, 10-43

## え

---

エラー

Recovery Manager, 15-3

Recovery Manager のバックアップ時, 9-57

エラー・コード

Recovery Manager, 15-2, 15-4

メッセージ番号, 15-4

メディア・マネージャ, 15-5

エラー・スタック

解釈, 15-7

エラー・メッセージ

Recovery Manager

解釈, 15-7

## お

---

オープン状態のデータベースのリカバリ

Recovery Manager を使用, 10-10

オペレーティング・システム・コピー

定義, 5-70

オペレーティング・システムのユーティリティ

ファイルのコピー, 9-53

オンライン REDO ログ

複製データベース, 12-6

## か

---

カタログ化

アーカイブ REDO ログ, 16-8

オペレーティング・システム・コピー, 9-53

データ・ファイル, 16-8

可用性

Recovery Manager のバックアップ, 18-15

環境、Recovery Manager

定義, 4-2



環境変数

NLS\_DATE\_FORMAT, 8-28

NLS\_LANG, 8-28

環境変数 NLS\_DATE\_FORMAT, 8-28

環境変数 NLS\_LANG, 8-28

完全リカバリ

Recovery Manager を使用, 10-9

## き

---

期限切れバックアップ

削除, 7-14, 18-7

機能、新規, xxix, xlv

キャラクタ・セット

Recovery Manager で使用するための設定, 8-28

Recovery Manager のエラー, 15-38

共有サーバー

Recovery Manager で使用するための構成, 8-31

## く

---

クロスチェック, 18-2

定義, 7-8

バックアップとコピー, 18-2

複数のチャネル, 18-10

リカバリ・カタログとメディア・マネージャ, 7-9

## こ

---

構成

Recovery Manager

SHOW コマンド, 7-7

共有サーバー, 8-31

自動チャネル, 8-11, 8-13

自動バックアップ, 8-18

消去, 5-10, 8-17

スナップショット制御ファイルの位置, 8-29

デバイス・タイプ, 5-5

デフォルト・デバイス・タイプ, 5-6, 8-12

特定のチャネル, 8-15

バックアップから表領域を除外, 8-26

バックアップ・セットのサイズ, 8-23

バックアップの最適化, 8-24

バックアップ保存方針, 8-21

パラレル化, 5-5

表示, 17-18

Recovery Manager で使用するためのメディア・マネージャ, 8-6

メディア・マネージャ

インストール, 8-2

前提条件, 8-2

互換性

リカバリ・カタログ, 4-16

コピー

クロスチェック, 18-2

長期, 5-72

コマンド・インタフェース

Recovery Manager, 4-5

コマンド・ファイル

Recovery Manager, 4-9

コマンド、Recovery Manager

BACKUP, 5-14, 9-57

PROXY ONLY オプション, 5-15

PROXY オプション, 5-15

SKIP OFFLINE オプション, 9-39

BACKUP CURRENT CONTROLFILE, 9-11

CATALOG, 16-8, 18-18

CHANGE, 7-8, 18-2

CONFIGURE, 5-8, 5-10, 17-18

COPY, 5-69

CROSSCHECK, 18-2

DELETE, 7-13, 18-19

DROP CATALOG, 16-35

DUPLICATE, 6-14, 13-2

EXECUTE SCRIPT, 16-19

LIST, 7-2, 17-3

INCARNATION オプション, 16-11

PRINT SCRIPT, 7-8

RECOVER, 6-6

Recovery Manager での解析方法, 4-5

REGISTER, 16-9

REPORT, 17-12

NEED BACKUP オプション, 17-13

RESET DATABASE

INCARNATION オプション, 16-11

RESTORE, 10-7

RESYNC CATALOG, 4-14, 16-13

FROM CONTROLFILECOPY オプション, 16-27

SET

MAXCORRUPT オプション, 9-57

SHOW, 5-5, 7-7, 17-18

UPGRADE CATALOG, 16-33

終了, 15-12

- 使用, 3-3
- スタンドアロン, 4-7
- タイプ, 4-7
- 対話形式, 4-9
- パイプの使用, 4-11
- バッチの実行, 4-9
- 孤立したバックアップ
  - レポート, 7-5
- コンパイル
  - Recovery Manager コマンドの実行, 4-6

## さ

---

- サーバー・セッション
  - Recovery Manager, 4-5
- サーバー・パラメータ・ファイル
  - 自動バックアップ, 5-52
  - 自動バックアップの構成, 8-18
  - バックアップ, 9-13
  - リストア, 10-13
- 再起動可能なバックアップ
  - 定義, 5-68, 9-31
- 削除
  - Recovery Manager を使用, 7-12
  - 期限切れバックアップ, 7-14
  - コピー, 18-5
  - バックアップ, 18-5, 18-6
    - 複数のチャネル, 18-10
  - バックアップ後のファイル, 7-15
  - 複数のデバイス・タイプ, 7-17
  - 不要なバックアップ, 7-13
- 作成
  - テストデータベース, 6-14
  - 複製データベース
    - リモート・ホスト上, 12-15
  - リカバリ・カタログ, 8-32
- 差分増分バックアップ, 5-45
- サンプル・スクリプト
  - Recovery Manager, 3-2

- 自動チャネル, 5-2
  - 一般的な
    - 構成, 5-8, 8-13
    - 定義, 8-13
  - オーバーライド, 8-11
  - 構成, 8-11, 8-12, 8-15, 9-2
  - 特定の構成, 5-9
  - ネーミング規則, 5-7
  - パラレル化, 8-12
  - 割当て, 5-3
- 自動バックアップ
  - サーバー・パラメータ・ファイル, 5-52
  - 制御ファイル, 5-52, 8-18, 9-9
  - 生成, 9-9
- 循環再利用レコード, 4-16
- 障害時リカバリ
  - Recovery Manager を使用, 6-11
- ショート・ウェイト
  - 定義, 14-17
- 初期化パラメータ
  - BACKUP\_TAPE\_IO\_SLAVES, 9-18
  - CONTROL\_FILE\_RECORD\_KEEP\_TIME, 16-37
  - CONTROL\_FILES, 10-22
  - DB\_FILE\_NAME\_CONVERT, 11-14, 12-7, 13-8
  - DB\_NAME, 11-14
  - LARGE\_POOL\_SIZE, 14-13
  - LOCK\_NAME\_SPACE, 11-14
  - LOG\_FILE\_NAME\_CONVERT, 11-14, 13-8
- 初期化パラメータ・ファイル, 6-6
- ジョブ
  - Recovery Manager, 4-8
    - 進行状況の監視, 15-20
    - パフォーマンスの監視, 15-24
- 新機能, xxix ~ xlv

## す

---

- スキーマ
  - 変更
    - リカバリ・カタログの更新, 16-17
- スタンバイ・データベース
  - DB\_FILE\_NAME\_CONVERT 初期化パラメータ, 13-8
  - LOG\_FILE\_NAME\_CONVERT 初期化パラメータ, 13-8
  - Recovery Manager およびスタンバイ・インスタンスの起動, 13-12

## し

---

- 時間を指定するパラメータ
  - Recovery Manager で使用するための設定, 8-28
- 自動位置検索
  - Oracle Real Application Clusters でのリストア, 6-5

Recovery Manager を使用した作成, 6-16, 13-9  
イメージ・コピーを使用, 13-23  
概要, 13-9  
Recovery Manager を使用した作成について, 13-2  
Recovery Manager を使用した準備, 13-2  
Recovery Manager を使用した制御ファイルのコピー, 13-6  
Recovery Manager を使用したデータ・ファイルの命名, 13-7  
Recovery Manager を使用したバックアップ, 9-23, 9-28, 9-29  
RC\_ARCHIVED\_LOG ビューの解析, 9-25  
概要, 9-23  
制限事項, 9-24  
ログのバックアップのガイドライン, 9-26  
SET AUXNAME コマンド, 13-9  
SET NEWNAME コマンド, 13-8, 13-9  
作成  
同じディレクトリ構造を持つリモート・ホストへの作成, 13-13  
異なるディレクトリ構造を持つリモート・ホストへの作成, 13-15  
ローカル・ホスト上に作成, 13-22  
スタンバイ制御ファイルのコピー, 13-6  
制御ファイルの作成  
Recovery Manager を使用, 13-3, 13-5  
SQL を使用, 13-5

## ストアド・スクリプト

Recovery Manager, 4-10  
Recovery Manager の作成, 16-20  
Recovery Manager の実行, 16-20  
管理, 16-19  
削除, 16-21  
実行, 16-20  
出力, 17-25  
置換, 16-21  
表示, 7-8, 17-24

スナップショット制御ファイル, 4-15  
位置の指定, 8-29

## せ

---

制御ファイル  
コピー

バックアップ, 9-12  
自動バックアップ, 5-52, 8-18, 9-9  
構成, 8-18

スナップショット  
位置の指定, 8-29  
バックアップ  
Recovery Manager を使用, 9-9, 9-11, 9-12  
使用したリカバリ, 10-15  
データベースのバックアップ内, 9-12  
バックアップおよびリカバリ, 10-16  
複製データベース, 12-5  
リカバリ・カタログの代替として使用, 4-16  
リストア, 10-22  
Recovery Manager を使用, 10-22  
SET DBID を使用, 10-46  
レコードの上書き, 16-37  
レコードのタイプ, 4-16

## 制御ファイル・コピー

バックアップ, 9-12  
制御ファイルの自動バックアップ  
構成, 8-18  
データベース構造の変更後, 8-18  
デフォルトの書式, 5-54  
リストア, 5-53

## 制御ファイル・レコード

上書き, 16-37  
制御ファイル・レコードの上書き, 16-37  
整合性チェック, 5-72  
セッション・アーキテクチャ

Recovery Manager, 4-5

## 切断

Recovery Manager, 2-13  
全体バックアップ, 5-42

## そ

---

増分バックアップ, 9-20, 9-49  
Recovery Manager での適用方法, 6-8  
Recovery Manager を使用, 9-18, 9-19, 9-20  
差分, 5-45

## た

---

ターゲット・データベース

定義, 1-6

タグ, 5-71

## 多重化

Recovery Manager を使用したデータ・ファイル,  
5-20

妥当性チェック  
バックアップ, 9-32  
ダミー API, 8-5

## ち

---

チャンネル

Recovery Manager, 5-2  
Recovery Manager のネーミング規則, 5-7  
一般的な構成, 5-8  
事前構成のディスク, 8-11  
自動のオーバーライド, 8-11  
自動の構成, 8-11  
手動チャンネルの平行化, 5-10  
手動と自動の違い, 5-4  
制御オプション, 5-11  
定義, 5-2  
特定の構成, 5-9  
バックアップのための構成, 9-2  
バックアップのための手動割当て, 9-2  
複数の  
クロスチェックおよび削除, 18-10  
複数割当て, 7-17

チューニング

Recovery Manager  
V\$ ビュー, 15-16  
長期バックアップ  
状態の変更, 7-19  
定義, 5-61

## て

---

ディスク API, 8-5  
ディスク API のテスト, 8-5  
ディスク・チャンネル  
事前構成された, 8-11  
データ・ファイル  
カタログ化, 16-8  
バックアップ  
Recovery Manager を使用, 5-14, 9-8, 9-9, 9-30,  
9-31, 9-32  
バックアップが必要、リスト表示, 17-30, 17-31  
複製データベース, 12-7  
リカバリ  
基本手順, 6-6  
必要に応じて判断, 15-24

リスト  
リカバリ不能, 17-13  
リストア, 6-2, 10-5  
デフォルトではないファイル名の使用, 10-21  
データ・ファイルのコピー  
Recovery Manager を使用したバックアップ, 9-9  
データ・ファイルのリカバリ  
定義, 6-6  
データベース  
バックアップ  
Recovery Manager を使用, 5-14  
複製の作成  
リモート・ホスト上, 12-15  
リカバリ・カタログからの登録解除, 16-9  
リカバリ・カタログへの登録, 16-6, 16-7  
データベース間での表領域の移動, 10-51  
データベース・スキーマ  
レポートの生成, 17-17  
データベース接続  
Recovery Manager  
Oracle Real Application Clusters 構成, 2-8  
カタログを使用, 2-6  
カタログを使用しない, 2-5  
パスワードの非表示, 2-11  
補助データベース, 2-9  
Recovery Manager 内のタイプ, 2-2  
Recovery Manager に必要な SYSDBA, 2-3  
データベース全体のバックアップ  
Recovery Manager を使用, 9-7  
データベースの Point-in-Time リカバリ (DBPITR)  
定義, 10-11  
データベースの複製, 6-14  
トラブルシューティング, 15-41  
テスト・データベース、作成, 6-14  
デバイス・タイプ  
Recovery Manager で構成, 8-12  
デフォルト・デバイス・タイプ  
構成, 8-12

## と

---

問合せ  
リカバリ・カタログ, 17-2  
トランスポートابل表領域, 10-51  
トレース・ファイル  
Recovery Manager, 15-3

## に

---

### 二重化

バックアップ・セット, 5-23, 8-25, 9-18

## は

---

### ハードウェアの多重化

定義, 5-13

### パイプ

Recovery Manager コマンドの実行に使用, 2-12

パイプ・インタフェース, 4-11

### パスワード

Recovery Manager への接続, 2-11

パスワード・ファイル

Recovery Manager への接続に使用, 2-6

破損データ・ファイル・ブロック, 5-73

Recovery Manager, 5-51

検出, 5-73, 5-74

制御ファイルのレコード, 5-51

バックアップの最大値の設定, 9-57

リカバリ, 10-42

破損の検出, 5-74

### バックアップ

BACKUP BACKUPSET 時のフェイルオーバー,  
5-29

I/O 速度の制限, 5-40

NOARCHIVELOG モード, 9-51

Oracle Real Application Clusters, 9-52

Recovery Manager エラー処理, 9-57

Recovery Manager の概要, 9-4

Recovery Manager のチャンネルの関連付け, 15-18,  
15-19

Recovery Manager のテスト, 5-75, 9-32

メディア・マネージャの使用, 8-8

RESETLOGS 以前のリカバリ, 10-48

アーカイブ REDO ログ, 9-47

Recovery Manager を使用, 9-13, 9-14, 9-15

Recovery Manager を使用したバックアップ,  
9-17

バックアップ後の削除, 7-14

必要な場合, 5-19

### 一貫性

Recovery Manager を使用した作成, 9-6

イメージ・コピー, 5-13, 5-69

オープン・ファイルの制限数, 5-22

可用性, 18-15

CHANGE コマンドで変更, 7-18

期限切れ

削除, 18-7

クロスチェック, 18-2

サーバー・パラメータ・ファイル, 9-13

再起動可能, 5-68, 9-31

最適化, 5-62

削除, 7-13

作成方法, 1-2

失敗した Recovery Manager, 15-37

失敗した Recovery Manager のトラブルシューティ  
ング, 15-27, 15-31, 15-35

自動ログ・スイッチ, 5-19

スタンバイ・データベース

RC\_ARCHIVED\_LOG ビューの解析, 9-25

Recovery Manager の概要, 9-23

Recovery Manager を使用, 9-23, 9-28, 9-29

アーカイブ REDO ログ, 9-26

制限事項, 9-24

ストアド・スクリプト, 16-19

### 制御ファイル

Recovery Manager を使用, 9-11, 9-12

リカバリのために使用, 10-16

制御ファイルの自動バックアップ, 9-9

全体, 5-42

増分, 5-43, 9-20, 9-49

Recovery Manager を使用, 9-18, 9-19, 9-20

差分, 5-45

タイプ, 5-13, 5-41

タグ, 5-71

妥当性チェック, 9-32

中断, 9-31

長期, 5-61

状態の変更, 7-19

停止, 15-29

### データ・ファイル

Recovery Manager を使用, 9-8, 9-9, 9-30, 9-31,  
9-32

データベース全体

Recovery Manager を使用, 9-7

二重化, 8-25, 9-18

バックアップ, 5-28

バックアップが必要なオブジェクトのレポート,  
17-13

バックアップから表領域を除外, 8-26

バックアップ・セット, 5-13, 9-30

- バックアップ・セット内のデータ・ファイルの制限, 5-22
- バックアップ・セットのファイル数の指定, 5-33
- 非一貫性
  - Recovery Manager を使用した作成, 9-6
- 表領域
  - Recovery Manager を使用, 9-8, 9-9, 9-30, 9-31, 9-32
- 非累積増分, 5-46, 5-57, 5-58
- 複数コピー, 8-25
- 不要
  - バッチ削除, 5-60
- 保存, 9-53
- 保存方針からの除外, 18-17
- ミラーの分割, 5-70
  - Recovery Manager を使用, 9-21
- ユーザー管理
  - カタログ化, 18-19
  - リカバリ・カタログ, 4-15, 16-24
- 累積増分, 5-47, 5-48, 6-7, 6-14, 6-15, 9-49, 14-4, 14-5, 14-7
- レポートの生成, 17-2
- バックアップおよびリカバリ
  - Recovery Manager とユーザー管理の方法の比較, 1-4
- バックアップ・セット
  - BACKUP コマンドを使用して作成, 5-37
  - Recovery Manager での生成方法, 5-33, 5-39
  - 概要, 5-14
  - 数, 5-39
  - 数の指定, 5-33
  - クロスチェック, 7-9
  - サイズ, 5-39
  - サイズの制限, 5-33
  - 最大サイズの設定, 8-23
  - 最大バイト数の指定, 5-32, 5-37
  - 作成中のエラー, 5-51
  - 多重化, 5-20
  - 定義, 5-14
  - 二重化, 9-18
  - ネーミング, 5-31
  - バックアップ, 5-28, 9-30
  - バックアップ時のフェイルオーバー, 5-29
  - ピース・サイズの制限, 5-32
  - 編成, 5-14
  - リストアのテスト, 10-43

- バックアップ・セットのサイズ
  - 設定, 5-32
- バックアップ・セットのネーミング, 5-31
- バックアップの最適化
  - 構成, 8-24
  - 冗長性, 5-67
  - 定義, 5-62, 9-31
  - 保存方針, 5-66
  - 無効化, 5-65, 8-24
  - 有効化, 5-65, 8-24
  - リカバリ・ウィンドウ, 5-66
- バックアップ・ピース
  - サイズの制限, 5-32
  - 定義, 5-14
- バックアップ保存方針
  - 定義, 5-55
- バックアップ・モード
  - 定義, 5-70
- パッケージ
  - DBMS\_PIPE, 2-12, 4-11
- パラレル化
  - Recovery Manager の構成, 5-5, 8-12
  - Recovery Manager のチャネルの手動割当て, 5-10
  - バックアップ, 5-25

## ひ

---

- 非一貫性バックアップ
  - Recovery Manager を使用, 9-6
- 非循環再利用レコード, 4-16
- ビュー
  - RC\_ARCHIVED\_LOG, 9-25
  - リカバリ・カタログ, 17-27
- 表領域
  - Recovery Manager のバックアップから除外, 8-26
  - Recovery Manager のバックアップの移動, 10-51
  - Recovery Manager を使用したバックアップ, 9-8
  - アクセス可能な表領域のリカバリ
    - オープン状態のデータベース, 10-10
  - 読取り専用
    - バックアップ, 9-39
- 表領域の Point-in-Time リカバリ
  - Recovery Manager を使用, 6-10
  - 概要, 11-2
  - 基本手順, 11-2
  - 計画, 11-4
  - 実行, 11-18

- 実行の理由, 11-2
- 制限事項, 11-5
- 制約事項, 11-5
- 補助インスタンスの準備, 11-13
- 用語, 11-4
- リカバリ・カタログなし, 11-5
- 表領域バックアップ
  - Recovery Manager を使用, 9-8, 9-9, 9-30, 9-31, 9-32
- 非累積増分バックアップ, 5-46, 5-57, 5-58

## ふ

---

- 不完全リカバリ
  - Recovery Manager を使用, 10-11
  - 概要, 6-10
  - 現行の制御ファイル, 10-11
  - 時間ベース, 10-12
- 複製データベース
  - DB\_FILE\_NAME\_CONVERT 初期化パラメータ, 12-7
  - NOFILENAMECHECK オプション, 12-9
  - NORMAL モードでオフラインされた表領域のスキップ, 12-8
  - Recovery Manager での作成方法, 12-3
  - SET NEWNAME コマンド, 12-7
  - オンライン REDO ログ, 12-6
  - 作成, 6-14
    - CONFIGURE AUXNAME を使用, 12-19
    - init.ora パラメータと LOGFILE を使用, 12-17
    - SET NEWNAME を使用, 12-18
    - 同じファイル・システムを持つリモート・ホスト上に作成, 12-15
    - 過去, 12-24
    - 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, 12-16
    - ローカル・ホスト上に作成, 12-21
  - 作成が失敗, 15-41
  - スナップショット制御ファイルの生成, 12-5
  - データ・ファイル, 12-7
  - 表領域の除外, 6-14, 12-4
  - ファイル名の生成, 12-5
  - 複製の準備, 12-10
  - 読取り専用表領域のスキップ, 12-7

- 不要なバックアップ
  - 期限が切れたバックアップとの違い, 5-55
  - 削除, 5-60, 7-13
  - レポート, 7-4
- プロキシ・コピー
  - 概要, 4-19
- ブロックの破損
  - V\$DATABASE\_BLOCK\_CORRUPTION に格納, 9-32
- ブロック・メディア・リカバリ, 10-40
  - ガイドライン, 6-12
  - 概要, 6-11
- 分裂ブロック
  - 検出, 5-74
  - 定義, 5-74

## へ

---

- 平均リカバリ時間 (MTTR)
  - 定義, 6-11

## ほ

---

- 補助セット
  - Recovery Manager の TSPITR, 11-4
  - 表領域内のデータ・ファイルの名前付け, 11-10
- 補助データベース
  - Recovery Manager の TSPITR
    - 変換済ファイル名, 11-12
- 保存方針
  - 構成, 8-21
  - 冗長性, 5-56, 5-59
  - 冗長性のための構成, 8-22
  - 除外されたバックアップ, 5-61, 18-17
  - 定義, 5-55
  - バックアップの最適化に与える影響, 5-66
  - 無効化, 5-56, 8-22
  - リカバリ・ウィンドウ, 5-56, 8-21
- ホット・バックアップ・モード, 「バックアップ・モード」を参照

## み

---

- ミラー化
  - バックアップ, 9-21
- ミラー化ファイル
  - 分割
    - Recovery Manager を使用, 9-21
- ミラーの分割
  - バックアップとして使用, 9-21

## め

---

- メタデータ
  - Recovery Manager
    - 問合せ, 17-2
  - Recovery Manager の管理, 4-12, 12-1, 16-1, 18-1
  - Recovery Manager の記憶領域, 4-5
  - Recovery Manager の問合せ, 7-2
  - 制御ファイルへの格納, 4-16
- メッセージ・スタック
  - Recovery Manager, 15-3
- メディア管理
  - API のテスト, 15-9
  - Backup Solutions Program, 4-19
  - sbttest プログラム, 4-19, 15-9
  - エラー・コード, 15-5
  - クロスチェック, 7-8
  - ソフトウェアへのリンク, 8-3
  - ファイルのバックアップ, 4-18
  - ファイルのリストア, 4-18
  - プロキシ・コピー, 4-19
- メディア管理ライブラリ (MML), 1-9
- メディア・マネージャ
  - Recovery Manager で使用するための構成, 8-6
  - インストール, 8-2
  - 構成のための前提条件, 8-2
  - 定義, 1-9
  - テスト, 8-6
  - トラブルシューティング, 8-8
  - バックアップのテスト, 8-8
  - リンク
    - テスト, 8-6
- メディア・リカバリ
  - Recovery Manager を使用, 6-6
  - データ・ファイル
    - 基本手順, 6-6

## ゆ

---

- ユーザー管理バックアップ
  - カタログ化, 18-19
- ユーティリティ
  - オペレーティング・システム、コピーの作成に使用, 9-53

## よ

---

- 読取り専用表領域
  - バックアップ, 9-39

## り

---

- リカバリ
  - Oracle Real Application Clusters, 10-27
  - Recovery Manager を使用して障害時に, 10-37
  - Recovery Manager を使用してデータベースをオープン, 10-10
  - 一般的な Recovery Manager, 10-5
  - 一般的な手順, 10-2
  - 完全, 10-9
  - 時間ベース, 10-12
  - 準備, 10-7
  - 消失または破損したリカバリ・カタログ, 16-27
  - 使用例の相違, 10-3
  - データ・ブロック, 6-11, 10-40
    - ガイドライン, 6-12
  - データベース
    - NOARCHIVELOG モード, 10-49
  - データベース全体
    - バックアップ制御ファイルを使用, 10-16
  - データベース・ファイル
    - Recovery Manager での変更の適用方法, 6-8
    - 概要, 6-6
    - 障害時, 6-11
  - バックアップがないデータ・ファイル, 10-50
  - バックアップ制御ファイルを使用, 10-15, 10-16
    - リカバリ・カタログなし, 10-18
    - リカバリ・カタログを使用, 10-16
  - 表領域, 10-10
  - 不完全, 10-11
  - リカバリ・カタログなし, 4-17
  - リカバリが必要なファイルの判断, 15-24



- リカバリ・ウィンドウ
    - 定義, 5-56
    - バックアップの最適化, 5-66
    - 保存方針のための構成, 8-21
    - リカバリ可能ポイント, 5-56
  - リカバリ・カタログ, 4-12
    - DB 識別子の問題, 16-9
    - UNKNOWN というデータベース名, 15-43
    - アップグレード, 16-33
    - 概要, 4-12
    - カタログ化
      - O/S バックアップ, 18-18
    - 可用性, 16-30
    - クロスチェック, 7-9
    - 更新
      - オペレーティング・システムでの削除後, 18-22
      - スキーマの変更後, 16-17
    - 互換性, 4-16
    - サイズの管理, 16-22
    - 再同期化, 16-13
    - 削除, 16-35
    - 作成, 8-32
    - 使用した運用, 1-8, 4-12
    - 使用した場合の効果, 3-15
    - 使用した不完全リカバリ, 10-11
    - 使用して Recovery Manager に接続する場合, 2-6, 2-9
    - 使用しない TSPITR の実行, 11-5
    - 新規データベースへの移動, 16-28
    - スキーマ
      - 設定, 8-32
    - ストアド・スクリプト
      - 作成, 16-20
      - 実行, 16-20
    - ストアド・スクリプトの表示, 17-24
    - スナップショット制御ファイル, 4-15
    - ターゲット・データベースの登録, 4-13, 16-6, 16-7
    - データベースの登録解除, 16-9
    - 問合せ, 17-2
    - 同期, 4-14
    - 内容, 4-14
    - バックアップ, 4-15, 16-24
    - バックアップの削除, 18-5
    - ビュー
      - 問合せ, 17-27
    - リカバリ, 16-27
    - リカバリ・カタログなしでの運用, 4-16
    - リフレッシュ, 16-13
    - 領域要件, 16-3
    - レコードの削除, 18-6, 18-22
    - ログ・スイッチのレコード, 16-23, 18-18
  - リカバリ・カタログからのデータベースの登録解除, 16-9
    - リカバリ・カタログからのレコードの削除, 18-22
    - リカバリ・カタログのアップグレード, 16-33
    - リカバリ・カタログの再同期化, 16-13
    - リカバリ・カタログの削除, 16-35
  - リカバリ可能ポイント
    - リカバリ・ウィンドウ, 5-56
  - リカバリ・セット
    - Recovery Manager の TSPITR, 11-4
  - リスト
    - 指定日より前に作成されたバックアップとコピー, 17-30
    - バックアップとコピー, 17-11
    - 例, 17-30
  - リストア
    - サーバー・パラメータ・ファイル, 10-13
    - 制御ファイル, 10-22
    - 制御ファイルのバックアップ
      - SET DBID を使用, 10-46
    - データ・ファイル, 10-5
      - デフォルトではないファイル名の使用, 10-21
    - データベース
      - 新しいホストヘリストア, 10-25
    - データベース・ファイル, 6-2
      - Recovery Manager での選択方法, 6-4
      - 仕組み, 6-3
      - リストアの最適化, 6-5
    - テスト, 10-43
  - リストアの最適化, 6-5
  - リストアの妥当性チェック, 10-43
  - リターン・コード
    - Recovery Manager, 15-9
  - リポジトリ
    - Recovery Manager, 4-5, 4-12
    - 問合せ, 17-2
- ## る
- 
- 累積増分バックアップ, 5-47, 9-49

## れ

---

例、Recovery Manager

NOARCHIVELOG バックアップ, 9-51

Oracle Real Application Clusters バックアップ,  
9-52

RESETLOGS 以前のバックアップのリカバリ,  
10-48, 10-49

アーカイブ REDO ログのバックアップ, 9-47

オペレーティング・システム・コピーのカタログ  
化, 9-53

全体的なメディア障害の後のリカバリ, 10-37

増分バックアップ, 9-49

データベース・スキーマのレポート, 17-31

バックアップ・エラーの処理, 9-57

バックアップおよびコピーのメンテナンス, 9-53

バックアップ・セットのサイズ設定, 9-43

バックアップ・セットの二重化, 9-18

バックアップとコピーのリスト表示, 17-30

不要なバックアップとコピーのリスト表示, 17-30

不要なバックアップのレポート, 17-30

累積増分バックアップ, 9-49

レベル 0 の増分バックアップ, 5-44

レポート, 7-2, 17-2, 17-12

概要, 7-3

孤立したバックアップ, 7-5

データベース・インカネーション, 17-31

データベース・スキーマ, 17-17, 17-31

バックアップが必要なオブジェクト, 17-13

不要なバックアップ, 7-4, 17-15, 17-16, 17-30

リカバリ不能なバックアップ, 17-15, 17-16

例, 17-30, 17-31

## ろ

---

ログ・スイッチ

リカバリ・カタログ・レコード, 16-23, 18-18

ロング・ウェイト

定義, 14-17