

## **Oracle® Database**

バックアップおよびリカバリ・ユーザーズ・ガイド

11g リリース 1 (11.1)

部品番号 : E05700-03

2008 年 10 月

Recovery Manager のバックアップとリカバリ、Recovery Manager のデータ転送、Oracle フラッシュバック技術、ユーザー管理のバックアップとリカバリなどの Oracle Database のバックアップおよびリカバリに関するガイド

Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド, 11g リリース 1 (11.1)

部品番号 : E05700-03

Oracle Database Backup and Recovery User's Guide, 11g Release 1 (11.1)

原本部品番号 : B28270-03

原本著者 : Lance Ashdown

原本協力者 : Antonio Romero, Katherine Weill, Tammy Bednar, Anand Beldalker, Timothy Chien, Mark Dilman, Senad Dizdar, Raymond Guzman, Stephan Haisley, Wei Hu, Alex Hwang, Ashok Joshi, Vasudha Krishnaswamy, J. William Lee, Valarie Moore, Muthu Olagappan, Vsevolod Panteleenko, Cris Pedregal-Martin, Samitha Samaranayake, Francisco Sanchez, Vivian Schupmann, Vinay Srihari, Margaret Susairaj, Mike Stewart, Steven Wertheimer, Wanli Yang, Rob Zijlstra

Copyright © 2003, 2008, Oracle. All rights reserved.

#### 制限付権利の説明

このプログラム（ソフトウェアおよびドキュメントを含む）には、オラクル社およびその関連会社に所有権のある情報が含まれています。このプログラムの使用または開示は、オラクル社およびその関連会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と工業所有権に関する法律により保護されています。

独立して作成された他のソフトウェアとの互換性を得るために必要な場合、もしくは法律によって規定される場合を除き、このプログラムのリバース・エンジニアリング、逆アセンブル、逆コンパイル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場合があります。誤りを見つけた場合は、オラクル社までご連絡ください。オラクル社およびその関連会社は、このドキュメントに誤りが無いことの保証は致し兼ねます。これらのプログラムのライセンス契約で許諾されている場合を除き、プログラムを形式、手段（電子的または機械的）、目的に関係なく、複製または転用することはできません。

このプログラムが米国政府機関、もしくは米国政府機関に代わってこのプログラムをライセンスまたは使用する者に提供される場合は、次の注意が適用されます。

#### U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

このプログラムは、核、航空、大量輸送、医療あるいはその他の本質的に危険を伴うアプリケーションで使用されることを意図しておりません。このプログラムをかかるとして使用する際、上述のアプリケーションを安全に使用するために、適切な安全装置、バックアップ、冗長性（**redundancy**）、その他の対策を講じることは使用者の責任となります。万一かかるプログラムの使用に起因して損害が発生いたしましても、オラクル社およびその関連会社は一切責任を負いかねます。

Oracle、JD Edwards、PeopleSoft、Siebel は米国 Oracle Corporation およびその子会社、関連会社の登録商標です。その他の名称は、他社の商標の可能性がります。

このプログラムは、第三者の Web サイトへリンクし、第三者のコンテンツ、製品、サービスへアクセスすることがあります。オラクル社およびその関連会社は第三者の Web サイトで提供されるコンテンツについては、一切の責任を負いかねます。当該コンテンツの利用は、お客様の責任になります。第三者の製品またはサービスを購入する場合は、第三者と直接の取引となります。オラクル社およびその関連会社は、第三者の製品およびサービスの品質、契約の履行（製品またはサービスの提供、保証義務を含む）に関しては責任を負いかねます。また、第三者との取引により損失や損害が発生いたしましても、オラクル社およびその関連会社は一切の責任を負いかねます。

---

---

# 目次

|  |      |
|--|------|
| はじめに .....                               | xvii |
| バックアップおよびリカバリの新機能 .....                  | xxi  |
| <b>第1部 バックアップおよびリカバリの概要</b>              |      |
| <b>1 バックアップおよびリカバリの概要</b>                |      |
| バックアップおよびリカバリの目的 .....                   | 1-2  |
| データ保護 .....                              | 1-2  |
| データの保持 .....                             | 1-3  |
| データ送信 .....                              | 1-3  |
| Oracle のバックアップおよびリカバリのソリューション .....      | 1-4  |
| Oracle フラッシュバック技術 .....                  | 1-6  |
| 論理フラッシュバック機能 .....                       | 1-6  |
| フラッシュバック・データベース .....                    | 1-7  |
| データ・リカバリ・アドバイザ .....                     | 1-8  |
| バックアップおよびリカバリのドキュメント・ロードマップ .....        | 1-8  |
| Recovery Manager のドキュメント・ロードマップ .....    | 1-10 |
| ユーザー管理のバックアップおよびリカバリのドキュメント・ロードマップ ..... | 1-10 |
| <b>2 Recovery Manager の概要</b>            |      |
| Recovery Manager 環境の概要 .....             | 2-2  |
| Recovery Manager の起動およびデータベースへの接続 .....  | 2-3  |
| Recovery Manager のデフォルト構成の表示 .....       | 2-4  |
| データベースのバックアップ .....                      | 2-4  |
| ARCHIVELOG モードでのデータベースのバックアップ .....      | 2-5  |
| NOARCHIVELOG モードでのデータベースのバックアップ .....    | 2-5  |
| 一般的なバックアップ・オプション .....                   | 2-6  |
| 増分バックアップの作成 .....                        | 2-6  |
| データベース・ファイルおよびバックアップの検証 .....            | 2-8  |
| Recovery Manager 操作のスキプト化 .....          | 2-8  |
| Recovery Manager 操作に関するレポート .....        | 2-9  |
| バックアップの表示 .....                          | 2-9  |
| データベース・ファイルおよびバックアップに関するレポート .....       | 2-10 |
| Recovery Manager バックアップの保持 .....         | 2-11 |

|   |             |
|---|-------------|
| バックアップのクロスチェック .....                        | 2-11        |
| 不要なバックアップの削除 .....                          | 2-11        |
| <b>データ・リカバリ・アドバイザを使用した障害の診断および修復 .....</b>  | <b>2-12</b> |
| 障害の表示および修復オプションの決定 .....                    | 2-12        |
| 障害の修復 .....                                 | 2-13        |
| <b>フラッシュバック・データベースを使用したデータベースの巻戻し .....</b> | <b>2-14</b> |
| <b>データベース・ファイルのリストアおよびリカバリ .....</b>        | <b>2-15</b> |
| データベース・ファイルのリストアおよびリカバリの準備 .....            | 2-15        |
| データベース全体のリカバリ .....                         | 2-16        |
| 表領域のリカバリ .....                              | 2-16        |
| 個々のデータ・ブロックのリカバリ .....                      | 2-17        |

## 第 II 部 Recovery Manager の起動および構成

### 3 Recovery Manager のアーキテクチャ

|  |            |
|--|------------|
| <b>Recovery Manager 環境 .....</b>                 | <b>3-2</b> |
| <b>Recovery Manager コマンドライン・クライアント .....</b>     | <b>3-3</b> |
| <b>Recovery Manager チャンネル .....</b>              | <b>3-4</b> |
| チャンネルおよびデバイス .....                               | 3-4        |
| 自動チャンネルおよび手動チャンネル .....                          | 3-5        |
| <b>Recovery Manager リポジトリ .....</b>              | <b>3-5</b> |
| <b>メディア管理 .....</b>                              | <b>3-6</b> |
| Recovery Manager とメディア・マネージャの相互作用 .....          | 3-6        |
| Oracle Secure Backup .....                       | 3-7        |
| Backup Solutions Program .....                   | 3-7        |
| <b>フラッシュ・リカバリ領域 .....</b>                        | <b>3-7</b> |
| <b>Data Guard 環境での Recovery Manager .....</b>    | <b>3-8</b> |
| Data Guard 環境での Recovery Manager の構成 .....       | 3-8        |
| Data Guard 環境での Recovery Manager によるファイル管理 ..... | 3-8        |

### 4 Recovery Manager クライアントの起動および操作

|  |            |
|--|------------|
| <b>Recovery Manager の起動および終了 .....</b>                             | <b>4-2</b> |
| <b>Recovery Manager 出力の場所の指定 .....</b>                             | <b>4-2</b> |
| <b>Recovery Manager のグローバルゼーション・サポート環境変数の設定 .....</b>              | <b>4-3</b> |
| <b>Recovery Manager コマンドの入力 .....</b>                              | <b>4-3</b> |
| Recovery Manager プロンプトでの Recovery Manager コマンドの入力 .....            | 4-3        |
| Recovery Manager でのコマンド・ファイルの使用 .....                              | 4-4        |
| Recovery Manager コマンド・ファイルへのコメントの入力 .....                          | 4-4        |
| コマンド・ファイルでの置換変数の使用 .....   | 4-5        |
| Recovery Manager 構文の確認 .....                                       | 4-6        |
| <b>Recovery Manager によるデータベース接続の確立 .....</b>                       | <b>4-8</b> |
| Recovery Manager によるデータベース接続 .....                                 | 4-8        |
| オペレーティング・システムのコマンドラインからの Recovery Manager による<br>データベース接続の確立 ..... | 4-9        |
| Recovery Manager プロンプトからのデータベース接続の確立 .....                         | 4-11       |
| Recovery Manager による補助データベースへの接続 .....                             | 4-11       |
| コマンド・ファイル内での Recovery Manager によるデータベース接続の確立 .....                 | 4-12       |



|   |      |
|---|------|
| Recovery Manager による接続で発生した問題の診断 .....          | 4-13 |
| <b>Recovery Manager のパイプ・インタフェースの使用</b> .....   | 4-13 |
| パイプを介した複数の Recovery Manager コマンドの連続実行の例 .....   | 4-14 |
| パイプを介した Recovery Manager コマンドの単一ジョブでの実行の例 ..... | 4-14 |

## 5 Recovery Manager 環境の構成

|  |      |
|--|------|
| <b>Recovery Manager バックアップの環境の構成</b> .....               | 5-2  |
| Recovery Manager の永続的な構成の表示およびクリア .....                  | 5-2  |
| バックアップ用のデフォルト・デバイスの構成: ディスクまたは SBT .....                 | 5-3  |
| バックアップ用のデフォルト・タイプの構成: バックアップ・セットまたはコピー .....             | 5-4  |
| チャンネルの構成 .....   | 5-5  |
| 制御ファイルおよびサーバー・パラメータ・ファイルの自動バックアップの構成 .....               | 5-7  |
| <b>メディア・マネージャにバックアップするための Recovery Manager の構成</b> ..... | 5-9  |
| Recovery Manager をメディア・マネージャとともに使用するための前提条件 .....        | 5-9  |
| メディア管理ライブラリの場所の確認 .....                                  | 5-10 |
| Recovery Manager のバックアップに対するメディア管理ソフトウェアの構成 .....        | 5-10 |
| メディア・マネージャ・ライブラリが正常に統合されたかどうかのテスト .....                  | 5-11 |
| メディア・マネージャで使用する SBT チャンネルの構成 .....                       | 5-13 |
| <b>フラッシュ・リカバリ領域の構成</b> .....                             | 5-14 |
| フラッシュ・リカバリ領域の概要 .....                                    | 5-15 |
| フラッシュ・リカバリ領域の有効化 .....                                   | 5-17 |
| 制御ファイルおよび REDO ログの場所の構成 .....                            | 5-20 |
| フラッシュ・リカバリ領域内での Recovery Manager によるファイル作成の構成 .....      | 5-21 |
| <b>バックアップの保存方針の構成</b> .....                              | 5-22 |
| 冗長性に基づく保存方針の構成 .....                                     | 5-22 |
| リカバリ期間に基づく保存方針の構成 .....                                  | 5-23 |
| 保存方針の無効化 .....   | 5-23 |
| <b>バックアップの最適化の構成</b> .....                               | 5-23 |
| バックアップの最適化の概要 .....                                      | 5-23 |
| SBT バックアップのバックアップの最適化への保存方針の影響 .....                     | 5-25 |
| バックアップの最適化の構成 .....                                      | 5-26 |
| <b>アーカイブ REDO ログの削除方針の構成</b> .....                       | 5-27 |
| アーカイブ REDO ログの削除方針 .....                                 | 5-27 |
| アーカイブ REDO ログの削除方針の有効化 .....                             | 5-28 |
| <b>Oracle Flashback Database およびリストア・ポイントの構成</b> .....   | 5-28 |
| リストア・ポイントおよびフラッシュバック・データベース .....                        | 5-29 |
| フラッシュバック・データベースおよび保証付きリストア・ポイントの前提条件 .....               | 5-33 |
| フラッシュバック・データベースの有効化 .....                                | 5-34 |
| 通常のリストア・ポイントと保証付きリストア・ポイントの作成 .....                      | 5-35 |
| 最適なフラッシュバック・データベースのパフォーマンスのための環境の構成 .....                | 5-35 |
| <b>Data Guard 環境での Recovery Manager の構成</b> .....        | 5-36 |

## 6 Recovery Manager 環境の構成: 高度なトピック

|                                |     |
|--------------------------------|-----|
| <b>高度なチャンネル・オプションの構成</b> ..... | 6-2 |
| チャンネル制御オプション .....             | 6-2 |
| 固有のチャンネル・パラメータの構成 .....        | 6-2 |

|   |      |
|---|------|
| 高度なバックアップ・オプションの構成 .....                  | 6-4  |
| バックアップ・セットの最大サイズの構成 .....                 | 6-4  |
| バックアップ・ピースの最大サイズの構成 .....                 | 6-5  |
| バックアップの多重化の構成 .....                       | 6-5  |
| 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 .....         | 6-6  |
| バックアップ圧縮アルゴリズムの構成 .....                   | 6-7  |
| バックアップの暗号化の構成 .....                       | 6-7  |
| 補助インスタンスのデータファイル名の構成 .....                | 6-10 |
| スナップショット制御ファイルの場所の構成 .....                | 6-11 |
| スナップショット制御ファイルの構成場所の表示 .....              | 6-11 |
| スナップショット制御ファイルの場所の設定 .....                | 6-11 |
| 共有サーバーで使用するための Recovery Manager の構成 ..... | 6-12 |
| 消失書込みの検出の有効化 .....                        | 6-13 |

### 第 III 部 データのバックアップおよびアーカイブ

## 7 Recovery Manager バックアップの概要

|  |      |
|--|------|
| Recovery Manager の一貫性バックアップおよび非一貫性バックアップ ..... | 7-2  |
| 一貫性バックアップ .....                                | 7-2  |
| 非一貫性バックアップ .....                               | 7-2  |
| オンライン・バックアップおよびバックアップ・モード .....                | 7-3  |
| バックアップ・セット .....                               | 7-3  |
| バックアップ・セットおよびバックアップ・ピース .....                  | 7-4  |
| バックアップ・セットの圧縮 .....                            | 7-4  |
| バックアップ・セットの暗号化 .....                           | 7-5  |
| バックアップ・ピースのファイル名 .....                         | 7-5  |
| バックアップ・ピースの数およびサイズ .....                       | 7-6  |
| バックアップ・セットの数およびサイズ .....                       | 7-6  |
| 多重バックアップ・セット .....                             | 7-7  |
| プロキシ・コピー .....                                 | 7-8  |
| イメージ・コピー .....                                 | 7-8  |
| Recovery Manager で作成したイメージ・コピー .....           | 7-9  |
| ユーザー管理イメージ・コピー .....                           | 7-10 |
| Recovery Manager を使用したバックアップの複数のコピー .....      | 7-10 |
| 多重バックアップ・セット .....                             | 7-10 |
| バックアップのバックアップ .....                            | 7-11 |
| 制御ファイルおよびサーバー・パラメータ・ファイルの自動バックアップ .....        | 7-12 |
| Recovery Manager が制御ファイルの自動バックアップを実行する場合 ..... | 7-12 |
| Recovery Manager による制御ファイルの自動バックアップの実行方法 ..... | 7-13 |
| 増分バックアップ .....                                 | 7-13 |
| マルチレベル増分バックアップ .....                           | 7-14 |
| ブロック・チェンジ・トラッキング .....                         | 7-16 |
| 増分バックアップのアルゴリズム .....                          | 7-17 |
| 増分バックアップを使用したリカバリ .....                        | 7-17 |
| バックアップの保存方針 .....                              | 7-18 |
| リカバリ期間 .....                                   | 7-19 |
| バックアップ冗長性 .....                                | 7-20 |

|                                      |      |
|--------------------------------------|------|
| 不要なバックアップのバッチ削除 .....                | 7-20 |
| バックアップ保存方針およびフラッシュ・リカバリ領域の削除規則 ..... | 7-21 |

## 8 データベースのバックアップ

|  |      |
|--|------|
| <b>Recovery Manager バックアップの概要</b> .....                              | 8-2  |
| Recovery Manager バックアップの目的 .....                                     | 8-2  |
| Recovery Manager バックアップの基本的な概念 .....                                 | 8-2  |
| <b>バックアップ出力オプションの指定</b> .....  | 8-3  |
| Recovery Manager バックアップ用のデバイス・タイプの指定 .....                           | 8-3  |
| ディスクへの Recovery Manager バックアップ用のバックアップ・セットまたは<br>バックアップ・コピーの指定 ..... | 8-3  |
| Recovery Manager バックアップのフォーマットの指定 .....                              | 8-4  |
| Recovery Manager バックアップのタグの指定 .....                                  | 8-5  |
| 圧縮バックアップの作成 .....  | 8-6  |
| <b>Recovery Manager を使用したデータベース・ファイルのバックアップ</b> .....                | 8-7  |
| Recovery Manager を使用したデータベース全体のバックアップの作成 .....                       | 8-8  |
| Recovery Manager を使用した表領域およびデータファイルのバックアップ .....                     | 8-8  |
| Recovery Manager を使用した制御ファイルのバックアップ .....                            | 8-9  |
| 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 .....                   | 8-11 |
| NOARCHIVELOG モードでのデータベースのバックアップ .....                                | 8-11 |
| <b>Recovery Manager を使用したアーカイブ REDO ログのバックアップ</b> .....              | 8-12 |
| アーカイブ REDO ログのバックアップ .....   | 8-12 |
| アーカイブ REDO ログ・ファイルのバックアップ .....                                      | 8-13 |
| バックアップが必要なアーカイブ REDO ログのみのバックアップ .....                               | 8-14 |
| バックアップ後のアーカイブ REDO ログの削除 .....                                       | 8-14 |
| <b>増分バックアップの作成および更新</b> .....  | 8-15 |
| 増分バックアップの目的 .....  | 8-16 |
| 増分バックアップ計画の設計 .....  | 8-16 |
| 増分バックアップの作成 .....  | 8-17 |
| 増分更新バックアップ .....   | 8-18 |
| ブロック・チェンジ・トラッキングを使用した、増分バックアップのパフォーマンスの向上 .....                      | 8-21 |
| <b>長期格納用のデータベース・バックアップの作成</b> .....                                  | 8-24 |
| アーカイブ・バックアップの目的 .....  | 8-24 |
| アーカイブ・バックアップの基本的な概念 .....  | 8-25 |
| 長期格納用のアーカイブ・バックアップの作成 .....  | 8-26 |
| 一時的なアーカイブ・バックアップの作成 .....  | 8-27 |
| <b>Recovery Manager バックアップのバックアップ</b> .....                          | 8-28 |
| バックアップのバックアップ .....  | 8-28 |
| Recovery Manager を使用したバックアップ・セットのバックアップ .....                        | 8-30 |
| Recovery Manager を使用したイメージ・コピーのバックアップ .....                          | 8-31 |

## 9 データベースのバックアップ: 高度なトピック

|  |     |
|--|-----|
| <b>Recovery Manager バックアップ・セットのサイズの制限</b> .....    | 9-2 |
| バックアップ・セットのサイズ .....                               | 9-2 |
| BACKUP ... MAXSETSIZE を使用したバックアップ・セットのサイズの制限 ..... | 9-2 |
| セクションへの大規模なデータファイルのバックアップの分割 .....                 | 9-3 |

|   |      |
|---|------|
| バックアップの最適化を使用したファイルのスキップ .....                    | 9-4  |
| 1つのテープへの毎日のアーカイブ・ログ・バックアップの最適化の例 .....            | 9-4  |
| 複数のメディア・ファミリーへの毎日のアーカイブ・ログ・バックアップの最適化の例 .....     | 9-4  |
| アーカイブ・ログの週ごとのセカンダリ・バックアップの作成の例 .....              | 9-5  |
| オフライン・ファイル、読取り専用ファイルおよびアクセスできないファイルのスキップ .....    | 9-7  |
| バックアップ・セットの多重化 .....                              | 9-7  |
| CONFIGURE BACKUP COPIES を使用したバックアップ・セットの多重化 ..... | 9-8  |
| BACKUP ... COPIES を使用したバックアップ・セットの多重化 .....       | 9-9  |
| Recovery Manager を使用したミラーの分割によるバックアップの実行 .....    | 9-9  |
| Recovery Manager バックアップの暗号化 .....                 | 9-11 |
| Recovery Manager バックアップの暗号化設定 .....               | 9-11 |
| 透過モード暗号化バックアップの作成 .....                           | 9-12 |
| パスワード・モード暗号化バックアップの作成 .....                       | 9-12 |
| デュアル・モード暗号化バックアップの作成 .....                        | 9-12 |
| Recovery Manager バックアップの再開 .....                  | 9-13 |
| 再開可能バックアップ .....                                  | 9-13 |
| 部分的に完了したバックアップの再開 .....                           | 9-13 |
| バックアップ期間の管理 .....                                 | 9-14 |
| バックアップ期間 .....                                    | 9-14 |
| バックアップ期間の指定 .....                                 | 9-14 |
| バックアップ期間での部分バックアップの許可 .....                       | 9-15 |
| バックアップの負荷および時間の最小化 .....                          | 9-15 |

## 第 IV 部 Recovery Manager のバックアップの管理

### 10 Recovery Manager 操作に関するレポート

|  |       |
|--|-------|
| Recovery Manager レポートの概要 .....               | 10-2  |
| Recovery Manager レポートの目的 .....               | 10-2  |
| Recovery Manager レポートの基本的な概念 .....           | 10-2  |
| バックアップおよびリカバリ関連オブジェクトの表示 .....               | 10-4  |
| LIST コマンド .....                              | 10-4  |
| バックアップおよびコピーの表示 .....                        | 10-5  |
| 選択したバックアップおよびコピーの表示 .....                    | 10-8  |
| データベース・インカネーションの表示 .....                     | 10-10 |
| リストア・ポイントの表示 .....                           | 10-10 |
| バックアップおよびデータベース・スキーマに関するレポート .....           | 10-11 |
| Recovery Manager バックアップに関するレポート .....        | 10-11 |
| 保存方針に基づくバックアップが必要なファイルに関するレポート .....         | 10-12 |
| リカバリ不能な操作によって影響を受けるデータファイルに関するレポート .....     | 10-13 |
| 不要なバックアップに関するレポート .....                      | 10-14 |
| データベース・スキーマに関するレポート .....                    | 10-15 |
| V\$ ビューを使用したバックアップ・メタデータの問合せ .....           | 10-16 |
| 過去および現行の Recovery Manager ジョブの詳細の問合せ .....   | 10-16 |
| バックアップ・ピースの暗号化ステータスの確認 .....                 | 10-18 |
| リカバリ・カタログ・ビューの問合せ .....                      | 10-18 |
| リカバリ・カタログ・ビュー .....                          | 10-18 |
| カタログ・ビューへのターゲット DB_KEY 値または DBID 値の問合せ ..... | 10-20 |
| RC_BACKUP_FILES の問合せ .....                   | 10-21 |

## 11 Recovery Manager バックアップおよびリポジトリ・レコードのメンテナンス

|  |       |
|--|-------|
| Recovery Manager バックアップおよびリポジトリのメンテナンスの概要 .....  | 11-2  |
| バックアップおよびリポジトリのメンテナンスの目的 .....                   | 11-2  |
| バックアップおよびリポジトリのメンテナンスの基本的な概念 .....               | 11-2  |
| 制御ファイルのリポジトリのメンテナンス .....                        | 11-4  |
| 制御ファイル・レコード .....                                | 11-4  |
| 制御ファイル・レコードの消失の防止 .....                          | 11-5  |
| 制御ファイルの保護 .....                                  | 11-6  |
| フラッシュ・リカバリ領域のメンテナンス .....                        | 11-7  |
| フラッシュ・リカバリ領域の規則の削除 .....                         | 11-7  |
| フラッシュ・リカバリ領域の領域使用状況の監視 .....                     | 11-7  |
| フラッシュ・リカバリ領域でのフラッシュバック・ログの領域の管理 .....            | 11-8  |
| フラッシュ・リカバリ領域が一杯になった場合の対応 .....                   | 11-8  |
| リストア・ポイントの削除 .....                               | 11-9  |
| 新しい場所へのフラッシュ・リカバリ領域の変更 .....                     | 11-10 |
| フラッシュ・リカバリ領域の無効化 .....                           | 11-10 |
| ファイル作成時のインスタンスのクラッシュへの対応 .....                   | 11-10 |
| フラッシュバック・データベースのパフォーマンスの影響の監視 .....              | 11-11 |
| I/O エラーが発生した場合のフラッシュバック・ライター (RVWR) の動作 .....    | 11-11 |
| Recovery Manager リポジトリの更新 .....                  | 11-12 |
| Recovery Manager リポジトリのクロスチェック .....             | 11-12 |
| バックアップおよびコピーのリポジトリ・ステータスの変更 .....                | 11-15 |
| Recovery Manager リポジトリへのバックアップ・レコードの追加 .....     | 11-16 |
| Recovery Manager リポジトリからのレコードの削除 .....           | 11-19 |
| Recovery Manager バックアップおよびアーカイブ REDO ログの削除 ..... | 11-20 |
| Recovery Manager の削除の概要 .....                    | 11-20 |
| すべてのバックアップおよびコピーの削除 .....                        | 11-22 |
| 指定したバックアップおよびコピーの削除 .....                        | 11-23 |
| 期限切れの Recovery Manager バックアップおよびコピーの削除 .....     | 11-24 |
| 保存方針に基づく不要な Recovery Manager バックアップの削除 .....     | 11-24 |
| データベースの削除 .....                                  | 11-25 |

## 12 リカバリ・カタログの管理

|  |       |
|--|-------|
| リカバリ・カタログの概要 .....                         | 12-2  |
| リカバリ・カタログの目的 .....                         | 12-2  |
| リカバリ・カタログの基本的な概念 .....                     | 12-2  |
| リカバリ・カタログの管理の基本手順 .....                    | 12-4  |
| リカバリ・カタログの作成 .....                         | 12-5  |
| リカバリ・カタログ・データベースの構成 .....                  | 12-5  |
| リカバリ・カタログのスキーマ所有者の作成 .....                 | 12-6  |
| CREATE CATALOG コマンドの実行 .....               | 12-7  |
| リカバリ・カタログへのデータベースの登録 .....                 | 12-7  |
| リカバリ・カタログへのデータベースの登録 .....                 | 12-7  |
| REGISTER DATABASE コマンドを使用したデータベースの登録 ..... | 12-8  |
| リカバリ・カタログへのバックアップの追加 .....                 | 12-9  |
| 仮想プライベート・カタログの作成および管理 .....                | 12-10 |

|   |              |
|---|--------------|
| 仮想プライベート・カタログ .....                       | 12-10        |
| 仮想プライベート・カタログ所有者の作成および権限の付与 .....         | 12-11        |
| 仮想プライベート・カタログの作成 .....                    | 12-12        |
| 仮想プライベート・カタログ所有者からの権限の取消し .....           | 12-12        |
| 仮想プライベート・カタログの削除 .....                    | 12-13        |
| <b>リカバリ・カタログの保護 .....</b>                 | <b>12-13</b> |
| リカバリ・カタログのバックアップ .....                    | 12-13        |
| リカバリ・カタログのリカバリ .....                      | 12-15        |
| <b>ストアド・スクリプトの管理 .....</b>                | <b>12-16</b> |
| ストアド・スクリプト .....                          | 12-16        |
| ストアド・スクリプトの作成 .....                       | 12-16        |
| ストアド・スクリプトの置換え .....                      | 12-17        |
| ストアド・スクリプトの実行 .....                       | 12-18        |
| 動的ストアド・スクリプトの作成および実行 .....                | 12-19        |
| ストアド・スクリプトの出力 .....                       | 12-20        |
| ストアド・スクリプト名の表示 .....                      | 12-20        |
| ストアド・スクリプトの削除 .....                       | 12-21        |
| Recovery Manager の起動時のストアド・スクリプトの実行 ..... | 12-21        |
| <b>リカバリ・カタログのメンテナンス .....</b>             | <b>12-22</b> |
| リカバリ・カタログのメンテナンス .....                    | 12-22        |
| リカバリ・カタログの再同期化 .....                      | 12-22        |
| DB_UNIQUE_NAME を変更した後のリカバリ・カタログの更新 .....  | 12-25        |
| 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の解除 .....      | 12-26        |
| リカバリ・カタログのデータベース・インカネーションの再設定 .....       | 12-28        |
| リカバリ・カタログのアップグレード .....                   | 12-30        |
| リカバリ・カタログのインポートおよび移動 .....                | 12-32        |
| リカバリ・カタログの削除 .....                        | 12-34        |

## 第 V 部 障害の診断および対応

### 13 Recovery Manager のデータ修復の概要

|  |             |
|--|-------------|
| <b>Recovery Manager のデータ修復の概要 .....</b>  | <b>13-2</b> |
| データ修復が必要な問題 .....                        | 13-2        |
| Recovery Manager のデータ修復方法 .....          | 13-2        |
| <b>Recovery Manager のリストア操作 .....</b>    | <b>13-3</b> |
| バックアップの選択 .....                          | 13-4        |
| リストア・フェイルオーバー .....                      | 13-4        |
| リストアの最適化 .....                           | 13-5        |
| <b>Recovery Manager のメディア・リカバリ .....</b> | <b>13-5</b> |
| 増分バックアップおよびアーカイブ REDO ログの選択 .....        | 13-5        |
| データベース・インカネーション .....                    | 13-6        |

### 14 データ・リカバリ・アドバイザーを使用した障害の診断および修復

|                                 |             |
|---------------------------------|-------------|
| <b>データ・リカバリ・アドバイザーの概要 .....</b> | <b>14-2</b> |
| データ・リカバリ・アドバイザーの目的 .....        | 14-2        |
| データ・リカバリ・アドバイザーの基本的な概念 .....    | 14-2        |

|                                    |       |
|------------------------------------|-------|
| 障害の診断および修復の基本手順 .....              | 14-6  |
| <b>障害の表示</b> .....                 | 14-7  |
| すべての障害の表示 .....                    | 14-7  |
| 障害のサブセットの表示 .....                  | 14-8  |
| <b>データベースの検証によるブロック破損の確認</b> ..... | 14-9  |
| <b>修復オプションの決定</b> .....            | 14-10 |
| すべての障害に対する修復オプションの決定 .....         | 14-11 |
| 障害のサブセットに対する修復オプションの決定 .....       | 14-12 |
| <b>障害の修復</b> .....                 | 14-13 |
| 障害の修復の概要 .....                     | 14-13 |
| 障害の修復 .....                        | 14-14 |
| <b>障害のステータスおよび優先順位の変更</b> .....    | 14-15 |

## 15 データベース・ファイルおよびバックアップの検証

|  |      |
|--|------|
| <b>Recovery Manager の検証の概要</b> .....             | 15-2 |
| Recovery Manager の検証の目的 .....                    | 15-2 |
| Recovery Manager の検証の基本的な概念 .....                | 15-2 |
| <b>VALIDATE コマンドによるブロック破損の確認</b> .....           | 15-4 |
| データファイルの検証の平行化 .....                             | 15-6 |
| <b>BACKUP VALIDATE を使用したデータベース・ファイルの検証</b> ..... | 15-6 |
| <b>リストアする前のバックアップの検証</b> .....                   | 15-7 |

## 16 フラッシュバックおよびデータベースの Point-in-Time リカバリの実行

|   |       |
|---|-------|
| <b>フラッシュバック技術およびデータベースの Point-in-Time リカバリの概要</b> ..... | 16-2  |
| フラッシュバックおよびデータベースの Point-in-Time リカバリの目的 .....          | 16-2  |
| Point-in-Time リカバリおよびフラッシュバック機能の基本的な概念 .....            | 16-2  |
| <b>フラッシュバック表を使用した表の巻戻し</b> .....                        | 16-4  |
| フラッシュバック表の前提条件 .....                                    | 16-4  |
| フラッシュバック表操作の実行 .....                                    | 16-5  |
| <b>フラッシュバック・ドロップを使用した DROP TABLE 操作の巻戻し</b> .....       | 16-7  |
| フラッシュバック・ドロップ .....                                     | 16-7  |
| フラッシュバック・ドロップの前提条件 .....                                | 16-8  |
| フラッシュバック・ドロップ操作の実行 .....                                | 16-8  |
| <b>フラッシュバック・データベースを使用したデータベースの巻戻し</b> .....             | 16-11 |
| フラッシュバック・データベースの前提条件 .....                              | 16-11 |
| フラッシュバック・データベース操作の実行 .....                              | 16-12 |
| <b>データベースの Point-in-Time リカバリの実行</b> .....              | 16-15 |
| データベースの Point-in-Time リカバリの前提条件 .....                   | 16-15 |
| データベースの Point-in-Time リカバリの実行 .....                     | 16-15 |
| <b>フラッシュバックおよびデータベースの Point-in-Time リカバリの例</b> .....    | 16-17 |
| フラッシュバック・データベースを使用した OPEN RESETLOGS 操作の巻戻し .....        | 16-17 |
| 取り消されたインカネーション・ブランチ内の SCN へのデータベースの巻戻し .....            | 16-18 |
| 祖先インカネーションへのデータベースのリカバリ .....                           | 16-20 |

## 17 データベースの完全リカバリの実行

|                                  |       |
|----------------------------------|-------|
| データベースの完全リカバリの概要 .....           | 17-2  |
| データベースの完全リカバリの目的 .....           | 17-2  |
| この章の概要 .....                     | 17-2  |
| データベースの完全リカバリの準備 .....           | 17-3  |
| リストアまたはリカバリするデータファイルの識別 .....    | 17-3  |
| データベースの DBID の確認 .....           | 17-6  |
| リストア操作で使用されるバックアップのプレビュー .....   | 17-6  |
| リストアする前のバックアップの検証 .....          | 17-9  |
| リカバリに必要なアーカイブ REDO ログのリストア ..... | 17-9  |
| データベースの完全リカバリの実行 .....           | 17-10 |
| データベースの完全リカバリ .....              | 17-10 |
| データベース全体の完全リカバリの実行 .....         | 17-11 |
| 表領域の完全リカバリの実行 .....              | 17-14 |
| コピーへの切替え後の完全リカバリの実行 .....        | 17-16 |

## 18 ブロック・メディア・リカバリの実行

|  |      |
|--|------|
| ブロック・メディア・リカバリの概要 .....                            | 18-2 |
| ブロック・メディア・リカバリの目的 .....                            | 18-2 |
| ブロック・メディア・リカバリの基本的な概念 .....                        | 18-2 |
| ブロック・メディア・リカバリの前提条件 .....                          | 18-4 |
| 個々のブロックのリカバリ .....                                 | 18-4 |
| V\$DATABASE_BLOCK_CORRUPTION 内のすべてのブロックのリカバリ ..... | 18-5 |

## 19 Recovery Manager のリカバリの実行：高度な例

|  |       |
|--|-------|
| 増分バックアップを使用した NOARCHIVELOG モードのデータベースのリカバリ ..... | 19-2  |
| サーバー・パラメータ・ファイルのリストア .....                       | 19-2  |
| 制御ファイルの自動バックアップからのサーバー・パラメータ・ファイルのリストア .....     | 19-4  |
| Recovery Manager を使用した初期化パラメータ・ファイルの作成 .....     | 19-4  |
| バックアップ制御ファイルを使用したリカバリの実行 .....                   | 19-5  |
| バックアップ制御ファイルを使用したリカバリの実行 .....                   | 19-5  |
| 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 .....    | 19-7  |
| 障害リカバリの実行 .....                                  | 19-9  |
| 障害リカバリの前提条件 .....                                | 19-9  |
| 障害発生後のデータベースのリカバリ .....                          | 19-9  |
| 新しいホストへのデータベースのリストア .....                        | 19-11 |
| 新しいホストへのデータベースのリストアの準備 .....                     | 19-11 |
| 新しいホストへのデータベースのリストアのテスト .....                    | 19-13 |

## 20 Recovery Manager の表領域の Point-in-Time リカバリ (TSPITR) の実行

|   |      |
|---|------|
| Recovery Manager の TSPITR の概要 .....     | 20-2 |
| Recovery Manager の TSPITR の目的 .....     | 20-2 |
| Recovery Manager の TSPITR の基本的な概念 ..... | 20-2 |
| Recovery Manager の TSPITR の基本手順 .....   | 20-3 |
| TSPITR の前提条件および結果 .....                 | 20-4 |
| TSPITR の結果 .....                        | 20-4 |



|   |              |
|---|--------------|
| リカバリ・カタログを使用しない場合の特別な考慮事項 .....   | 20-5         |
| <b>TSPITR の計画および準備 .....</b>  | <b>20-5</b>  |
| TSPITR での正しい目標時点の選択 .....   | 20-5         |
| リカバリ・セットの決定 .....   | 20-6         |
| TSPITR の実行後に消失するオブジェクトの確認および保存 .....  | 20-7         |
| <b>完全に自動化された Recovery Manager の TSPITR の実行 .....</b>  | <b>20-8</b>  |
| <b>Recovery Manager 管理の補助インスタンスを使用した Recovery Manager の<br/>    カスタマイズ TSPITR の実行 .....</b> | <b>20-10</b> |
| TSPITR での Oracle Managed Files の名前の変更 .....   | 20-10        |
| SET NEWNAME を使用した TSPITR のリカバリ・セットのデータファイル名の変更 .....  | 20-10        |
| TSPITR の補助セットのデータファイル名の指定 .....   | 20-11        |
| イメージ・コピーを使用した Recovery Manager の TSPITR の高速化 .....  | 20-14        |
| TSPITR における自動補助インスタンスの初期化パラメータのカスタマイズ .....   | 20-16        |
| <b>独自の補助インスタンスを使用した Recovery Manager の TSPITR の実行 .....</b>                                 | <b>20-18</b> |
| Recovery Manager の TSPITR に使用する独自の補助インスタンスの準備 .....   | 20-18        |
| 独自の補助インスタンスを使用した TSPITR に使用する Recovery Manager コマンドの準備 .....                                | 20-20        |
| 独自の補助インスタンスを使用した TSPITR の実行 .....   | 20-20        |
| 独自の補助インスタンスを使用した TSPITR の実行例 .....  | 20-21        |
| <b>Recovery Manager の TSPITR のトラブルシューティング .....</b>   | <b>20-23</b> |
| ファイル名の競合のトラブルシューティング .....  | 20-23        |
| UNDO セグメントを含む表領域の識別のトラブルシューティング .....   | 20-23        |
| TSPITR の失敗後の手動による補助インスタンスの再起動のトラブルシューティング .....   | 20-23        |

## 第 VI 部 チューニングおよびトラブルシューティング

### 21 Recovery Manager のパフォーマンスのチューニング

|   |       |
|---|-------|
| Recovery Manager のパフォーマンスのチューニングの目的 .....                   | 21-2  |
| Recovery Manager のパフォーマンスのチューニングの基本的な概念 .....               | 21-2  |
| 読取りフェーズ .....   | 21-4  |
| コピー・フェーズ .....  | 21-7  |
| SBT への書込みフェーズ .....   | 21-7  |
| ディスクへの書込みフェーズ .....   | 21-11 |
| Recovery Manager のパフォーマンスの問題を診断するための V\$ ビューの使用法 .....      | 21-11 |
| V\$SESSION_LONGOPS による Recovery Manager ジョブの進捗状況の監視 .....   | 21-11 |
| V\$BACKUP_SYNC_IO および V\$BACKUP_ASYNC_IO によるボトルネックの特定 ..... | 21-13 |
| Recovery Manager のバックアップのパフォーマンスのチューニング .....               | 21-15 |
| 手順 1: チャネル設定からの RATE パラメータの削除 .....                         | 21-15 |
| 手順 2: 同期ディスク I/O を使用する場合の DBWR_IO_SLAVES の設定 .....          | 21-15 |
| 手順 3: 共有メモリの割当てに失敗する場合の LARGE_POOL_SIZE の設定 .....           | 21-16 |
| 手順 4: 読取り、書込みおよびコピーのフェーズのチューニング .....                       | 21-16 |

### 22 Recovery Manager 操作のトラブルシューティング

|   |      |
|---|------|
| Recovery Manager のメッセージ出力の解釈 .....        | 22-2 |
| メッセージ出力タイプの識別 .....                       | 22-2 |
| Recovery Manager のエラー・メッセージ・スタックの識別 ..... | 22-2 |
| エラー・コードの識別 .....                          | 22-3 |

|   |       |
|---|-------|
| Recovery Manager エラー・スタックの解釈 .....                      | 22-5  |
| Recovery Manager のリターン・コードの識別 .....                     | 22-7  |
| <b>Recovery Manager のトラブルシューティングでの V\$ ビューの使用</b> ..... | 22-8  |
| Recovery Manager とメディア・マネージャの相互作用の監視 .....              | 22-8  |
| サーバー・セッションと Recovery Manager チャンネルの関連付け .....           | 22-9  |
| <b>Media Management API のテスト</b> .....                  | 22-11 |
| sbttest ユーティリティの入手 .....                                | 22-11 |
| sbttest ユーティリティのオンライン・ドキュメントの取得 .....                   | 22-12 |
| sbttest ユーティリティの使用 .....                                | 22-12 |
| <b>Recovery Manager コマンドの終了</b> .....                   | 22-13 |
| ALTER SYSTEM KILL SESSION によるセッションの終了 .....             | 22-13 |
| オペレーティング・システム・レベルでのセッションの終了 .....                       | 22-14 |
| メディア・マネージャでハングアップした Recovery Manager セッションの終了 .....     | 22-14 |

## 第 VII 部 Recovery Manager を使用したデータの送信

### 23 データベースの複製

|   |       |
|---|-------|
| <b>Recovery Manager データベースの複製の概要</b> .....            | 23-2  |
| データベースの複製の目的 .....                                    | 23-2  |
| データベースの複製の基本的な概念 .....                                | 23-2  |
| データベースの複製の基本手順 .....                                  | 23-3  |
| <b>バックアップおよびアーカイブ・ログから複製インスタンスへのアクセス可能化</b> .....     | 23-4  |
| SBT バックアップから複製インスタンスへのアクセス可能化 .....                   | 23-4  |
| ディスク・バックアップから複製インスタンスへのアクセス可能化 .....                  | 23-4  |
| <b>複製ファイルの名前の指定方法の選択</b> .....                        | 23-6  |
| <b>補助インスタンスの準備</b> .....                              | 23-8  |
| 手順 1: 補助インスタンス用の Oracle パスワード・ファイルの作成 .....           | 23-8  |
| 手順 2: Oracle Net から補助インスタンスへの接続の確立 .....              | 23-9  |
| 手順 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 .....                | 23-9  |
| 手順 4: SQL*Plus を使用した補助インスタンスの起動 .....                 | 23-10 |
| <b>Recovery Manager の起動および複製前の構成</b> .....            | 23-11 |
| 手順 1: Recovery Manager の起動およびデータベース・インスタンスへの接続 .....  | 23-11 |
| 手順 2: ソース・データベースのマウントまたはオープン .....                    | 23-11 |
| 手順 3: 複製で使用するための Recovery Manager チャンネルの構成 .....      | 23-12 |
| <b>データベースの複製</b> .....                                | 23-12 |
| 同じディレクトリ構造を持つリモート・ホストでのデータベースの複製 .....                | 23-12 |
| 異なるディレクトリ構造を持つリモート・ホストでのデータベースの複製 .....               | 23-13 |
| ローカル・ホストでの複製データベースの作成 .....                           | 23-14 |
| Oracle Managed Files または自動ストレージ管理を使用したデータベースの複製 ..... | 23-15 |
| <b>代替方法を使用した複製ファイルの名前の指定</b> .....                    | 23-17 |
| 複製制御ファイルの名前の指定 .....                                  | 23-17 |
| 複製オンライン REDO ログ・ファイルのネーミング .....                      | 23-17 |
| 複製データファイルの名前の指定 .....                                 | 23-18 |
| 複製一時ファイルの名前の指定 .....                                  | 23-19 |
| <b>Recovery Manager による複製の例</b> .....                 | 23-20 |
| ソース・データベース表領域のサブセットの複製 .....                          | 23-20 |

|   |       |
|---|-------|
| DUPLICATE を使用したアーカイブ・バックアップのリストア .....    | 23-21 |
| SET NEWNAME を使用した複製ファイルの名前の指定 .....       | 23-22 |
| CONFIGURE AUXNAME を使用した複製ファイルの名前の指定 ..... | 23-24 |
| <b>24   トランスポートابل表領域セットの作成</b>           |       |
| トランスポートابل表領域セットの作成の概要 .....              | 24-2  |
| トランスポートابل表領域セットの作成の目的 .....              | 24-2  |
| トランスポートابل表領域セットの基本的な概念 .....             | 24-3  |
| トランスポートابل表領域セットの作成の基本手順 .....            | 24-5  |
| <b>補助インスタンスの初期化パラメータのカスタマイズ</b> .....     | 24-5  |
| 補助インスタンスの初期化パラメータの設定 .....                | 24-6  |
| 補助インスタンスのパラメータ・ファイルの場所の設定 .....           | 24-7  |
| トランスポートابل表領域セットの作成 .....                 | 24-7  |
| トランスポートابل表領域セットの作成のトラブルシューティング .....     | 24-8  |
| トランスポートابل表領域セットの使用例 .....                | 24-9  |
| 指定した時点または SCN でのトランスポートابل表領域セットの作成 ..... | 24-9  |
| データ・ポンプ・ファイルの場所の指定 .....                  | 24-10 |
| 補助ファイルの場所の指定 .....                        | 24-11 |
| <b>25   プラットフォーム間でのデータ転送</b>              |       |
| クロス・プラットフォーム・データ転送の概要 .....               | 25-2  |
| クロス・プラットフォーム・データ転送の目的 .....               | 25-2  |
| クロス・プラットフォーム・データ転送の基本的な概念 .....           | 25-2  |
| ソース・ホストでのクロス・プラットフォーム表領域変換の実行 .....       | 25-3  |
| ソース・ホストでのクロス・プラットフォーム・データファイル変換の実行 .....  | 25-5  |
| 転送先ホストでのクロス・プラットフォーム・データファイル変換 .....      | 25-5  |
| CONVERT DATAFILE を使用したデータファイル形式の変換 .....  | 25-5  |
| クロス・プラットフォーム・データベース変換の前のデータベースの確認 .....   | 25-7  |
| データベースを転送する場合のソース・ホストでのデータファイルの変換 .....   | 25-9  |
| データベースを転送する場合の転送先ホストでのデータファイルの変換 .....    | 25-11 |
| ソース・ホストでのデータファイル初期変換手順の実行 .....           | 25-11 |
| 転送先ホストでのデータファイルの変換 .....                  | 25-13 |
| <b>26   ASM でのデータの移行の実行</b>               |       |
| ASM でのデータの移行の概要 .....                     | 26-2  |
| ASM でのデータの移行の目的 .....                     | 26-2  |
| ASM でのデータの移行の基本的な概念 .....                 | 26-2  |
| ASM へのデータ移行の基本手順 .....                    | 26-3  |
| ASM へのデータベースの移行の準備 .....                  | 26-3  |
| ASM へのデータベースの移行 .....                     | 26-5  |
| ASM から代替ストレージへのデータベースの移行 .....            | 26-8  |
| ASM ディスク・グループ間でのデータファイルの移動 .....          | 26-9  |

## 第 VIII 部 ユーザー管理のバックアップおよびリカバリの実行

### 27 ユーザー管理データベース・バックアップの作成

|   |       |
|---|-------|
| V\$ ビューの問合せによるバックアップ情報の取得 .....                       | 27-2  |
| バックアップ前のデータベース・ファイルの表示 .....                          | 27-2  |
| オンライン表領域バックアップのデータファイルのステータスの確認 .....                 | 27-2  |
| データベース全体のユーザー管理バックアップの作成 .....                        | 27-3  |
| 一貫性のあるデータベース全体のバックアップの作成 .....                        | 27-4  |
| 表領域およびデータファイルのユーザー管理バックアップの作成 .....                   | 27-4  |
| オフラインの表領域およびデータファイルのユーザー管理バックアップの作成 .....             | 27-4  |
| オンラインの表領域およびデータファイルのユーザー管理バックアップの作成 .....             | 27-5  |
| 制御ファイルのユーザー管理バックアップの作成 .....                          | 27-11 |
| 制御ファイルのバイナリ・ファイルへのバックアップ .....                        | 27-11 |
| 制御ファイルのトレース・ファイルへのバックアップ .....                        | 27-11 |
| アーカイブ REDO ログのユーザー管理バックアップの作成 .....                   | 27-12 |
| SUSPEND モードでのユーザー管理バックアップの作成 .....                    | 27-12 |
| 一時停止 / 再開機能 .....                                     | 27-12 |
| 一時停止中のデータベースのバックアップの作成 .....                          | 27-13 |
| RAW デバイスへのユーザー管理バックアップの作成 .....                       | 27-14 |
| Linux および UNIX での RAW デバイスへのバックアップ .....              | 27-14 |
| Windows での RAW デバイスへのバックアップ .....                     | 27-16 |
| Volume Shadow Copy Service (VSS) を使用したバックアップの作成 ..... | 27-18 |
| ユーザー管理データファイル・バックアップの検証 .....                         | 27-18 |
| データファイル・バックアップのリストアのテスト .....                         | 27-18 |
| DBVERIFY ユーティリティの実行 .....                             | 27-18 |

### 28 ユーザー管理のデータベースのフラッシュバックおよびリカバリの実行

|  |       |
|--|-------|
| SQL*Plus でのデータベースのフラッシュバックの実行 .....    | 28-2  |
| ユーザー管理のメディア・リカバリの概要 .....              | 28-3  |
| ユーザー管理のリストアおよびリカバリ .....               | 28-3  |
| RECOVER コマンドを使用した自動リカバリ .....          | 28-5  |
| アーカイブ・ログがデフォルトの場所にある場合のリカバリ .....      | 28-6  |
| アーカイブ・ログがデフォルト以外の場所にある場合のリカバリ .....    | 28-6  |
| リカバリの取消し .....                         | 28-8  |
| パラレル・メディア・リカバリ .....                   | 28-8  |
| データベースの完全リカバリの実行 .....                 | 28-8  |
| クローズしているデータベースのリカバリの実行 .....           | 28-9  |
| オープンしているデータベースのリカバリの実行 .....           | 28-12 |
| データベースの不完全リカバリの実行 .....                | 28-14 |
| 取消しベースの不完全リカバリの実行 .....                | 28-15 |
| 時間ベースまたは変更ベースの不完全リカバリの実行 .....         | 28-17 |
| NOARCHIVELOG モードでのデータベースのリカバリ .....    | 28-18 |
| メディア・リカバリのトラブルシューティング .....            | 28-19 |
| ユーザー管理のメディア・リカバリの問題 .....              | 28-19 |
| メディア・リカバリの問題の調査: フェーズ 1 .....          | 28-21 |
| ブロックを破損させない修正の試行: フェーズ 2 .....         | 28-22 |
| リカバリでブロックの破損を許可するかどうかの決定: フェーズ 3 ..... | 28-23 |

|                                 |       |
|---------------------------------|-------|
| リカバリでのブロックの破損の許可 : フェーズ 4 ..... | 28-24 |
| 試行リカバリの実行 .....                 | 28-25 |

## 29 ユーザー管理のリカバリの実行 : 高度な例

|   |       |
|---|-------|
| <b>現行の制御ファイルのサブセットが消失した場合の対応</b> .....        | 29-2  |
| 多重制御ファイルのデフォルトの場所へのコピー .....                  | 29-2  |
| 多重制御ファイルのデフォルト以外の場所へのコピー .....                | 29-2  |
| <b>現行の制御ファイルがすべて消失した場合のリカバリ</b> .....         | 29-3  |
| デフォルトの場所でのバックアップ制御ファイルを使用したりカバリ .....         | 29-4  |
| デフォルト以外の場所でのバックアップ制御ファイルを使用したりカバリ .....       | 29-5  |
| バックアップ制御ファイルを使用したデータファイルの追加を伴うリカバリ .....      | 29-5  |
| バックアップ制御ファイルを使用した読取り専用表領域のリカバリ .....          | 29-6  |
| <b>制御ファイルの再作成</b> .....                       | 29-6  |
| 作成された制御ファイルを使用した RESETLOGS を伴うリカバリ .....      | 29-8  |
| 再作成された制御ファイルを使用した読取り専用ファイルのリカバリ .....         | 29-8  |
| <b>バックアップが利用できない場合のデータファイルの再作成</b> .....      | 29-9  |
| <b>NOLOGGING 表および索引のリカバリ</b> .....            | 29-10 |
| トランスポータブル表領域のリカバリ .....                       | 29-10 |
| <b>オンライン REDO ログ・ファイルが消失した後のリカバリ</b> .....    | 29-11 |
| 多重オンライン REDO ログ・グループの一部のメンバーが消失した後のリカバリ ..... | 29-11 |
| オンライン REDO ログ・グループのすべてのメンバーが消失した後のリカバリ .....  | 29-12 |
| <b>フラッシュバック機能を使用しない、削除された表のリカバリ</b> .....     | 29-16 |
| <b>SQL*Plus でのデータベースの削除</b> .....             | 29-17 |

## 用語集

## 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
- ドキュメントのアクセシビリティについて
- 関連ドキュメント
- 表記規則
- サポートおよびサービス

## 対象読者

このマニュアルは、次の作業を実行するデータベース管理者を対象としています。

- Oracle Database のバックアップ、リストアおよびリカバリ
- データベース・ファイルのバックアップのメンテナンス
- Oracle Database をインストールする際のファイル・システムと ASM 間またはプラットフォーム間でのデータ送信

このマニュアルを読むには、次の知識が必要です。

- 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  
（『Oracle Database 概要』および『Oracle Database 管理者ガイド』を参照）
- データベースを実行しているオペレーティング・システム環境

## ドキュメントのアクセシビリティについて

オラクル社は、障害のあるお客様にもオラクル社の製品、サービスおよびサポート・ドキュメントを簡単にご利用いただけることを目標としています。オラクル社のドキュメントには、ユーザーが障害支援技術を使用して情報を利用できる機能が組み込まれています。HTML 形式のドキュメントで用意されており、障害のあるお客様が簡単にアクセスできるようにマークアップされています。標準規格は改善されつつあります。オラクル社はドキュメントをすべてのお客様がご利用できるように、市場をリードする他の技術ベンダーと積極的に連携して技術的な問題に対応しています。オラクル社のアクセシビリティについての詳細情報は、Oracle Accessibility Program の Web サイト <http://www.oracle.com/accessibility/> を参照してください。

### ドキュメント内のサンプル・コードのアクセシビリティについて

スクリーン・リーダーは、ドキュメント内のサンプル・コードを正確に読めない場合があります。コード表記規則では閉じ括弧だけを行に記述する必要があります。しかし JAWS は括弧だけの行を読まない場合があります。

### 外部 Web サイトのドキュメントのアクセシビリティについて

このドキュメントにはオラクル社およびその関連会社が所有または管理しない Web サイトへのリンクが含まれている場合があります。オラクル社およびその関連会社は、それらの Web サイトのアクセシビリティに関しての評価や言及は行っておりません。

### Oracle サポート・サービスへの TTY アクセス

アメリカ国内では、Oracle サポート・サービスへ 24 時間年中無休でテキスト電話 (TTY) アクセスが提供されています。TTY サポートについては、(800)446-2398 にお電話ください。アメリカ国外からの場合は、+1-407-458-2479 にお電話ください。

## 関連ドキュメント

詳細は、次の Oracle ドキュメントを参照してください。

- 『Oracle Database バックアップおよびリカバリ・リファレンス』
- 『Oracle Database ユーティリティ』
- 『Oracle Database ストレージ管理者ガイド』

Backup Solutions Program については、次の Web サイトを参照してください。

<http://www.oracle.com/technology/deploy/availability>



ドキュメント・セットの多くのマニュアルでは、Oracle Database ととも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。作成されるサンプル・スキーマの詳細と、ユーザーがサンプル・スキーマを使用する方法については、『Oracle Database サンプル・スキーマ』を参照してください。

Oracle エラー・メッセージのドキュメントは、HTML 版のみが存在します。Oracle Database JP Documentation Library を使用する場合は、エラー・メッセージを範囲ごとに参照できます。特定の範囲のページを表示したら、ブラウザの「このページの検索」機能を使用して特定のメッセージを検索できます。インターネットに接続できる場合、Oracle オンライン・ドキュメントのエラー・メッセージ検索機能を使用して特定のエラー・メッセージを検索できます。

## 表記規則

このマニュアルでは次の表記規則を使用します。

| 規則      | 意味   |
|---------|--|
| 太字      | 太字は、操作に関連する Graphical User Interface 要素、または本文中で定義されている用語および用語集に記載されている用語を示します。 |
| イタリック体  | イタリックは、ユーザーが特定の値を指定するプレースホルダ変数を示します。   |
| 固定幅フォント | 固定幅フォントは、段落内のコマンド、URL、サンプル内のコード、画面に表示されるテキスト、または入力するテキストを示します。                 |

## サポートおよびサービス

次の各項に、各サービスに接続するための URL を記載します。

### Oracle サポート・サービス

オラクル製品サポートの購入方法、および Oracle サポート・サービスへの連絡方法の詳細は、次の URL を参照してください。

<http://www.oracle.com/lang/jp/support/index.html>

### 製品マニュアル

製品のマニュアルは、次の URL にあります。

<http://www.oracle.com/technology/global/jp/documentation/index.html>

### 研修およびトレーニング

研修に関する情報とスケジュールは、次の URL で入手できます。

[http://education.oracle.com/pls/web\\_prod-plq-dad/db\\_pages.getpage?page\\_id=3](http://education.oracle.com/pls/web_prod-plq-dad/db_pages.getpage?page_id=3)

### その他の情報

オラクル製品やサービスに関するその他の情報については、次の URL から参照してください。

<http://www.oracle.com/lang/jp/index.html>

<http://www.oracle.com/technology/global/jp/index.html>

---

**注意：** ドキュメント内に記載されている URL や参照ドキュメントには、Oracle Corporation が提供する英語の情報も含まれています。日本語版の情報については、前述の URL を参照してください。

---



---

---

# バックアップおよびリカバリの新機能

この項では、Oracle Database 11g リリース 1 のバックアップおよびリカバリの新機能について説明します。今回のリリースの新機能によって、Oracle Database のバックアップおよびリカバリの管理性が大幅に向上します。特に、データ・リカバリ・アドバイザの導入、Recovery Manager と Data Guard との統合の向上、リカバリ・カタログ機能の拡張およびアーカイブ REDO ログの管理の向上によって、高い管理性が実現されています。

今回のリリースの新機能は、次のとおりです。

- データ・リカバリ・アドバイザ

**データ・リカバリ・アドバイザ**は、自動的にデータ障害を診断し、修復を推奨する組み込みツールです。障害は、手動で修復するか、または自動的に修復されるように要求することができます。データ・リカバリ・アドバイザでは、LIST FAILURE、CHANGE FAILURE、ADVISE FAILURE および REPAIR FAILURE のコマンドがサポートされています。

**参照：** データ・リカバリ・アドバイザのコマンドについては、第 14 章「データ・リカバリ・アドバイザを使用した障害の診断および修復」および『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

- Data Guard との統合の向上

Recovery Manager を TARGET としてデータベースに接続せずに、プライマリ・データベースまたは**フィジカル・スタンバイ・データベース**に対して Recovery Manager の永続構成を設定できるようになりました。Recovery Manager は Data Guard 環境内のすべてのデータベースでシームレスに動作するため、あるデータベースに作成したバックアップをリストア用に、別のデータベースで作成したものをリカバリ用に使用できます。同じ**リカバリ・カタログ**で、すべてのプライマリ・データベースおよびスタンバイ・データベースのメタデータを管理できます。

**参照：** 5-36 ページの「Data Guard 環境での Recovery Manager の構成」および『Oracle Data Guard 概要および管理』を参照してください。

- 長期バックアップの処理の向上

BACKUP ... KEEP コマンドを使用して、長期バックアップを作成するか、またはバックアップに一貫性を持たせるために必要なアーカイブ・ログ・ファイルのみを保存する**アーカイブ・バックアップ**を作成できるようになりました。

**参照：** BACKUP コマンドについては、8-24 ページの「長期格納用のデータベース・バックアップの作成」および『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

- **フラッシュ・リカバリ領域**内のアーカイブ REDO ログのバックアップ・フェイルオーバー  
Recovery Manager は、フラッシュ・リカバリ領域にあるアーカイブ REDO ログ・ファイルのバックアップ時に、リカバリ領域外のアーカイブ先に対してフェイルオーバーを実行できるようになりました。リカバリ領域のログが欠落または破損した場合に、Recovery Manager は、代替格納場所にあるアーカイブ・ログの正常なコピーを使用してバックアップの書込みを続行できます。

**参照：**「[アーカイブ REDO ログ・フェイルオーバー](#)」(8-12 ページ)

- アーカイブ・ログの削除方針の拡張

CONFIGURE コマンドでアーカイブ・ログの削除方針を構成すると、フラッシュ・リカバリ領域を含むすべてのアーカイブ先にその構成が適用されるようになりました。フラッシュ・リカバリ領域と同様に、BACKUP ... DELETE INPUT および DELETE ... ARCHIVELOG のいずれもこの構成に従います。ログがスタンバイ・データベース転送先に適用または転送された後でのみ削除対象となるように、CONFIGURE コマンドで**アーカイブ REDO ログの削除方針**を構成することもできます。方針は、必須のスタンバイ転送先のみに対して設定するか、またはすべてのスタンバイ転送先に対して設定できます。

**参照：**

- 「[アーカイブ REDO ログの削除方針の構成](#)」(5-27 ページ)
  - CONFIGURE ARCHIVELOG DELETION POLICY コマンドについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
  - Data Guard 環境での Recovery Manager の使用方法については、『Oracle Data Guard 概要および管理』を参照してください。
- バックアップを使用しないネットワーク対応データベースの複製  
DUPLICATE コマンドを使用して、既存のデータベース・バックアップを必要とせずに、ネットワーク上に**複製データベース**または**フィジカル・スタンバイ・データベース**を作成できるようになりました。この形式の複製は、**アクティブなデータベースの複製**と呼ばれます。

**参照：** DUPLICATE コマンドについては、[第 23 章「データベースの複製](#)」および『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

- リカバリ・カタログの拡張

リカバリ・カタログの所有者は、同じリカバリ・カタログ・データベース内の他のデータベース・ユーザーに対して、カタログのサブセットへのアクセス権限の付与 (GRANT) または取消し (REVOKE) を実行できるようになりました。このサブセットは、**仮想プライベート・カタログ**と呼ばれます。また、IMPORT CATALOG コマンドを使用して、1 つのリカバリ・カタログ (またはカタログ内の特定のデータベースのメタデータ) を別のリカバリ・カタログにマージすることもできます。

**参照：** リカバリ・カタログのコマンドについては、[第 12 章「リカバリ・カタログの管理](#)」および『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

- マルチセクション・バックアップ

Recovery Manager は、作業を複数のチャンネルに分割することによって、1 つのファイルをパラレルでバックアップできるようになりました。各チャンネルで 1 つの**ファイル・セクション**がバックアップされます。BACKUP コマンドに SECTION SIZE を指定して、**マルチセクション・バックアップ**を作成します。マルチセクション・バックアップは自動的にパラレルでリストアされます。オプションは不要です。

VALIDATE ... SECTION SIZE を使用して、ファイルの検証をパラレル化することができます。

**参照：** BACKUP および VALIDATE コマンドについては、9-3 ページの「セクションへの大規模なデータファイルのバックアップの分割」および『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

- UNDO の最適化

BACKUP コマンドでは、バックアップのリカバリに不要な UNDO はバックアップされなくなりました。すでにコミットされたトランザクションに対して UNDO が生成された場合、その UNDO は不要となります。データベース内の大部分の UNDO はこのような不要な UNDO です。

**参照：** BACKUP コマンドについては、5-23 ページの「バックアップの最適化の概要」および『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

- ブロック・メディア・リカバリのパフォーマンスの向上

Recovery Manager で **ブロック・メディア・リカバリ** を実行すると、バックアップを検索する前に、必要なブロックの **フラッシュバック・ログ** (使用可能な場合) が自動的に検索されるようになりました。フラッシュバック・ログから取得したブロックを使用すると、ブロック・メディア・リカバリのパフォーマンスを大幅に向上できます。

**参照：** 「**ブロック・メディア・リカバリの概要**」(18-2 ページ)

- ブロックの破損の検出の向上

Recovery Manager を含むいくつかのデータベース・コンポーネントおよびユーティリティで、**破損ブロック**を検出して V\$DATABASE\_BLOCK\_CORRUPTION に記録できるようになりました。破損ブロックは、インスタンス・リカバリによって検出されると、このビューに自動的に記録されます。Oracle Database では、ブロックの破損が検出または修復されると、このビューが自動的に更新されます。VALIDATE コマンドは、VALIDATE ... BLOCK、VALIDATE DATABASE などの多くの新しいオプションで拡張されています。

**参照：** VALIDATE コマンドについては、第 15 章「データベース・ファイルおよびバックアップの検証」および『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

- バックアップの圧縮の高速化

Recovery Manager では、バックアップの **バイナリ圧縮**用の既存の BZIP2 アルゴリズムに加えて、ZLIB アルゴリズムもサポートされるようになりました。ZLIB は BZIP2 より高速で実行されますが、作成されるファイルのサイズが大きくなります。ZLIB には、Oracle Advanced Compression オプションが必要です。CONFIGURE COMPRESSION ALGORITHM コマンドを使用して、Recovery Manager のバックアップに BZIP2 (デフォルト) と ZLIB のいずれを使用するかを選択できます。

**参照：** CONFIGURE コマンドについては、6-7 ページの「バックアップ圧縮アルゴリズムの構成」および『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

- スタンバイ・データベースのブロック・チェンジ・トラッキングのサポート

フィジカル・スタンバイ・データベースで **ブロック・チェンジ・トラッキング** を有効にできるようになりました。スタンバイ・データベースのバックアップ時に、Recovery Manager は、**ブロック・チェンジ・トラッキング・ファイル**を使用して、最後の増分バックアップ以降に変更されたブロックを迅速に識別できます。

**参照：**「増分バックアップの作成および更新」(8-15 ページ)

- Recovery Manager の置換変数を使用したスクリプト作成の向上

実行時にユーザーの入力を受け入れる Recovery Manager のコマンド・ファイルおよびストア・スクリプトを作成できるようになりました。これによって、バックアップ・スクリプトで、タグ、ファイル名、リストア・ポイント名などに Recovery Manager の置換変数を使用できます。

**参照：** 4-5 ページの「コマンド・ファイルでの置換変数の使用」および 12-19 ページの「動的ストア・スクリプトの作成および実行」を参照してください。

- VSS 対応アプリケーションとの統合

**Oracle VSS ライター**が、Windows 上で **Volume Shadow Copy Service (VSS)** インフラストラクチャを使用するアプリケーションと統合されました。VSS 対応ソフトウェアおよびストレージ・システムを使用して、**Oracle Database** をバックアップおよびリストアできます。主なメリットとしては、オープン状態のデータベースのシャドウ・コピーを作成できることがあげられます。

**参照：** VSS 対応アプリケー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方法については **Oracle Database** のプラットフォーム・ガイド、BACKUP コマンドについては『**Oracle Database** バックアップおよびリカバリ・リファレンス』を参照してください。

- 消失書込みの検出

DB\_LOST\_WRITE\_PROTECT 初期化パラメータで、スタンバイ・データベースの管理リカバリ中またはプライマリ・データベースのメディア・リカバリ中に**消失書込み**を検出できるようになりました。消失書込みの検出は、デフォルトでは無効になっています。

**参照：**「消失書込みの検出の有効化」(6-13 ページ)

- 読取り専用トランスポータブル表領域のバックアップ

以前のリリースでは、Recovery Manager は、転送先データベースでトランスポータブル表領域が読取り / 書込みに設定されるまで、そのトランスポータブル表領域をバックアップすることはできませんでした。今回のリリースでは、Recovery Manager は、トランスポータブル表領域が読取り / 書込みではない場合でも、そのトランスポータブル表領域をバックアップし、そのバックアップをリストアできるようになりました。

**参照：**「Recovery Manager を使用した表領域およびデータファイルのバックアップ」(8-8 ページ)

- Oracle Enterprise Manager でのバックアップおよびリカバリの拡張

Enterprise Manager に、データ・リカバリ・アドバイザ用のインターフェースが追加されました。

**参照：** バックアップおよびリカバリに関連した Enterprise Manager の拡張の詳細は、『**Oracle Database 2 日でデータベース管理者**』を参照してください。

- Oracle Flashback Transaction

トランザクションを無効にできます。Oracle Database によって、トランザクション間の依存性が判別され、不要な変更を無効にする補正トランザクションが実際に作成されます。データベースは、トランザクション、およびそのトランザクションに依存している他のすべてのトランザクションが発生しなかった場合と同様の状態まで巻き戻されます。

**参照：** この機能の使用方法については、『Oracle Database アドバンスド・アプリケーション開発者ガイド』を参照してください。

■ フラッシュバック・データ・アーカイブ

**フラッシュバック・データ・アーカイブ**によって、表の存続期間中、その表に対するトランザクション関連のすべての変更をデータベースで自動的に追跡して保存できるようになりました。したがって、この機能をデータベース・アプリケーションに組み込む必要はありません。

フラッシュバック・データ・アーカイブは、コンプライアンス、監査レポート、データ分析および DSS（意思決定支援システム）で特に有効です。フラッシュバック・データ・アーカイブとともにいくつかの**論理フラッシュバック機能**を使用すると、非常に古いデータにアクセスできます。

**参照：** フラッシュバック・データ・アーカイブの構成方法および使用方法については、『Oracle Database アドバンスド・アプリケーション開発者ガイド』、フラッシュバック機能の概要は 1-6 ページの「**Oracle フラッシュバック技術**」を参照してください。

■ SMP システム上のデータベースに対するメディア・リカバリのパフォーマンスの向上

対称マルチプロセッシング (SMP) ・コンピュータ上のデータベースのメディア・リカバリが高速化されました。パフォーマンスでの向上された点は、次のとおりです。

- パラレル化がより強化されました。
- REDO の非同期の読取り、解析および適用がより効率的に行われるようになりました。
- パラレル適用アルゴリズム内の同期点が削減されました。
- REDO ログ境界のメディア・リカバリ・**チェックポイント**で、次のログの適用が妨げられなくなりました。

構成は不要ですが、デフォルトの適用率が十分でない場合は、新しい **parallel recovery** 待機イベントを使用してチューニングできます。

**参照：** Oracle 待機イベントについては、『Oracle Database リファレンス』を参照してください。





# 第 I 部

---

## バックアップおよびリカバリの概要

第 I 部では、バックアップおよびリカバリの概要を示し、バックアップおよびリカバリ計画の立案方法について説明します。

- [第 1 章「バックアップおよびリカバリの概要」](#)
- [第 2 章「Recovery Manager の概要」](#)



---

---

# バックアップおよびリカバリの概要

この章では、Oracle Database のバックアップおよびリカバリの目的および Oracle ソリューションの概要を示します。この章の内容は、次のとおりです。

- バックアップおよびリカバリの目的
- Oracle のバックアップおよびリカバリのソリューション
- Oracle フラッシュバック技術
- データ・リカバリ・アドバイザ
- バックアップおよびリカバリのドキュメント・ロードマップ

---

---

**注意：** Recovery Manager (RMAN) をすぐに使用する場合は、[第 2 章「Recovery Manager の概要」](#)に進みます。

---

---

## バックアップおよびリカバリの目的

バックアップ管理者の主な責務は、バックアップおよびリカバリ計画を立案、実装および管理することです。通常、**バックアップおよびリカバリ**計画の目的は、データの消失からデータベースを保護し、データの消失後にデータベースを再構築することです。バックアップ管理者の通常のタスクには、次のものがあります。

- 様々な障害に対する対策の計画およびテスト
- バックアップおよびリカバリ用のデータベース環境の構成
- バックアップ・スケジュールの設定
- バックアップおよびリカバリ環境の監視
- バックアップに関する問題のトラブルシューティング
- 必要に応じたデータの消失からのリカバリ

バックアップ管理者は、バックアップおよびリカバリに関する次のような他の責務を求められることもあります。

- 長期格納用データベース・コピーを作成する場合のデータ保持
- 1つのデータベースまたはホストから別のデータベースまたはホストにデータを移動する場合のデータ送信

このマニュアルでは、前述のタスクの実行方法について説明します。

## データ保護

バックアップ管理者は、主にデータ保護用のバックアップを作成および監視します。**バックアップ**とは、データを再構築するために使用可能なデータベースのデータのコピーのことです。バックアップは、**物理バックアップ**または**論理バックアップ**のいずれかにできます。

物理バックアップとは、データベースを保存およびリカバリするために使用する物理ファイルのコピーのことです。これらのファイルには、データファイル、制御ファイル、アーカイブ REDO ログなどがあります。つまり、データベース情報を別の場所（ディスク、またはテープなどのオフライン・ストレージ・メディア）に保存するファイルのコピーは、すべて物理バックアップになります。

論理バックアップには、表やストアド・プロシージャなどの論理データが含まれています。**Oracle Data Pump**を使用すると、論理データをバイナリ・ファイルにエクスポートし、後でそのバイナリ・ファイルをデータベースにインポートできます。データ・ポンプのコマンドライン・クライアントである `expdp` および `impdp` では、`DBMS_DATAPUMP` および `DBMS_METADATA PL/SQL` パッケージが使用されます。

物理バックアップは、適切なバックアップおよびリカバリ計画の基礎となります。論理バックアップは、様々な状況で物理バックアップを補足するために役立ちますが、物理バックアップがなければデータ消失に十分に対処することはできません。

特に明記されていないかぎり、バックアップおよびリカバリのドキュメントで使用されている**バックアップ**という用語は、物理バックアップを指します。データベースのバックアップとは、物理バックアップを作成する作業のことです。バックアップおよびリカバリのドキュメント・セットでは、ほとんどの場合、物理バックアップについて重点が置かれています。

**Oracle Database** の通常の操作を中断したり、データベースの I/O 操作に影響を及ぼす問題はいくつかありますが、通常、DBA の介入やデータ・リカバリが必要となるのは、**メディア障害**、ユーザー・エラーおよびアプリケーション・エラーのみです。他の障害で、データが消失しない DBA の介入、またはバックアップからのリカバリが不要な DBA の介入が必要になる場合もあります。たとえば、インスタンスの障害が発生した後にデータベースを再起動したり、データファイルが一杯になったことが原因で文の実行が失敗した後にディスク領域をさらに割り当てる必要がある場合があります。

## メディア障害

メディア障害とは、データベースの実行に必要なディスク・ファイルの読取りまたは書込みの障害を発生させるディスクでの物理的な問題のことです。すべてのデータベース・ファイルで、メディア障害が発生する危険性があります。メディア障害が発生した後の適切なリカバリ技術は、影響を受けているファイルおよび使用可能なバックアップのタイプによって異なります。

バックアップおよびリカバリの特に重要な側面は、**障害リカバリ**計画を立案して、データベース・ホスト全体の消失などの重大なデータ消失を回避することです。

## ユーザー・エラー

ユーザー・エラーは、アプリケーション・ロジックのエラーまたは手作業の誤りによって、データベース内のデータが誤って変更されたり、削除された場合に発生します。ユーザー・エラーは、データベースの停止時間を発生させる最も大きな原因であるとみなされています。

ユーザー・エラーによるデータの消失は、局所的な場合または広範囲に及ぶ場合があります。局所的な損害の例としては、従業員表から誤った SMITH を削除してしまった場合などがあります。このタイプの損害には、的確な検出および修復が必要です。広範囲に及ぶ損害の例としては、現在の月の会社の注文を削除するバッチ・ジョブなどがあります。この場合は、データベースの停止時間が長くなるようにするために、思い切った処置が必要となります。

ユーザー・トレーニングや権限の慎重な管理によってほとんどのユーザー・エラーは回避できますが、ユーザー・エラーによってデータが消失した場合に、消失したデータを正常にリカバリできるかどうかは、バックアップ計画にかかっています。

## アプリケーション・エラー

ソフトウェアの障害によってデータ・ブロックが破損する場合があります。**物理的な破損**（メディア破損とも呼ばれます）の場合、ブロックがデータベースで認識されません。**チェックサム**が無効か、ブロック内容がすべて 0（ゼロ）か、またはブロックのヘッダーとフッターが一致していません。破損が広範囲に及んでいない場合は、**ブロック・メディア・リカバリ**を使用して簡単に修復できることがあります。

### 参照：

- [第 8 章「データベースのバックアップ」](#)
- データ・ポンプを使用する方法の詳細は、『Oracle Database ユーティリティ』を参照してください。

## データの保持

データの保持は、データの保護と関連していますが、目的が異なります。たとえば、ビジネスの四半期の終わりに存在していたデータベースのコピーを保持する必要がある場合があります。このバックアップは、障害リカバリ計画の一部ではありません。多くの場合、これらのバックアップが書き込まれるメディアは、バックアップの完了後に使用できません。火災の危険性がない保管場所にテープを送ったり、テスト施設にポータブル・ハード・ドライブを送送する場合などです。Recovery Manager には、バックアップを作成し、**バックアップの保存方針**からそのバックアップを除外する有効な方法が備えられています。このタイプのバックアップは、**アーカイブ・バックアップ**と呼ばれます。

**参照：**「[長期格納用のデータベース・バックアップの作成](#)」(8-24 ページ)

## データ送信

状況によっては、データベースまたはデータベース・コンポーネントのバックアップを作成し、それを別の場所に移動する必要がある場合があります。たとえば、Recovery Manager (RMAN) を使用すると、データベース・コピーの作成、別のデータベースにインポート可能な表領域コピーの作成、またはプラットフォーム間でのデータベース全体の移動を行うことができます。これらのタスクは、厳密にはバックアップおよびリカバリ計画の一部ではありませんが、データベースのバックアップを使用する必要があるため、バックアップ管理者の責務に含まれる場合があります。

**参照:** 第 VII 部「Recovery Manager を使用したデータの送信」の章を参照してください。

## Oracle のバックアップおよびリカバリのソリューション

バックアップおよびリカバリ計画の実装中に、次のソリューションを使用できます。

- **Recovery Manager (RMAN)**

このツールは、Oracle Database で実行されているセッションと統合されて、一連のバックアップおよびリカバリ・アクティビティ（バックアップに関する履歴データの **Recovery Manager リポジトリ** のメンテナンスなど）を実行します。Recovery Manager には、コマンドラインまたは Oracle Enterprise Manager を介してアクセスできます。

- ユーザー管理のバックアップおよびリカバリ

このソリューションでは、ホスト・オペレーティング・システム・コマンドと SQL\*Plus リカバリ・コマンドを組み合わせて、バックアップおよびリカバリを実行します。

Oracle では前述のいずれのソリューションもサポートされており、これらは完全に文書化されていますが、データベースのバックアップおよびリカバリのためのソリューションとしては Recovery Manager を使用することをお勧めします。Recovery Manager は、ユーザー管理の方法で使用可能なタイプと同じタイプのバックアップおよびリカバリをより簡単に実行します。また、Recovery Manager では、様々なホスト・オペレーティング・システム間で共通のバックアップ・タスク用インタフェース、およびユーザー管理の方法では使用できない多くのバックアップの方法が提供されています。

このマニュアルのほとんどの部分では、Recovery Manager ベースのバックアップおよびリカバリについて説明しています。ユーザー管理のバックアップおよびリカバリ方法については、**第 VIII 部「ユーザー管理のバックアップおよびリカバリの実行」**を参照してください。Recovery Manager では、ユーザー管理のバックアップおよびリカバリでは使用できないいくつかのバックアップおよびリカバリの方法および機能を使用することができます。これらの機能の主なものを次に示します。

- 増分バックアップ

**増分バックアップ**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格納されません。このため、より圧縮されたバックアップおよび高速なリカバリが可能になり、**データファイルのメディア・リカバリ**中に REDO を適用する必要が少なくなります。**ブロック・チェンジ・トラッキング**を有効にした場合は、各入力データファイルで全体スキャンが行われないようにすることによって、パフォーマンスを向上させることができます。増分バックアップを実行するには、BACKUP INCREMENTAL コマンドを使用します。

- ブロック・メディア・リカバリ

少数の破損データ・ブロックのみが含まれているデータファイルは、オフラインにしたり、バックアップからリストアせずに修復することができます。**ブロック・メディア・リカバリ**を実行するには、RECOVER コマンドを使用します。

- 未使用ブロックの圧縮

**未使用ブロックの圧縮**では、Recovery Manager は、未使用のデータ・ブロック、および場合によっては現在使用されていない使用済ブロックをスキップできます。

- バイナリ圧縮

Oracle Database に統合されている**バイナリ圧縮**メカニズムによって、バックアップのサイズが削減されます。

- 暗号化バックアップ

Recovery Manager は、Oracle Database に統合されている**バックアップの暗号化**機能を使用して、暗号化された形式でバックアップ・セットを格納します。ディスクに暗号化バックアップを作成するには、データベースで Advanced Security Option を使用している必要があります。暗号化バックアップをテープに直接作成するには、Recovery Manager で Oracle Secure Backup SBT インタフェースを使用している必要があります。ただし、Advanced Security Option は必要ありません。

Recovery Manager またはユーザー管理の方法のいずれを使用しても、データ・ポンプ・エクスポート・ユーティリティを使用して作成したスキーマ・オブジェクトの論理バックアップで物理バックアップを補助できます。リストアおよびリカバリ後に、データ・ポンプ・インポートを使用してデータを再作成できます。論理バックアップの大部分は、バックアップおよびリカバリに関するドキュメントでは説明されていません。

表 1-1 に、様々なバックアップ方法の機能の概要を示します。

表 1-1 バックアップ方法の機能比較

| 機能                         | Recovery Manager   | ユーザー管理  | データ・ポンプ・エクスポート                           |
|----------------------------|--|---|--|
| クローズ状態のデータベース・バックアップ       | サポートされています。インスタンスをマウントする必要があります。   | サポートされています。   | サポートされていません。                             |
| オープン状態のデータベース・バックアップ       | サポートされています。BEGIN/END BACKUP 文を使用する必要はありません。  | サポートされています。BEGIN/END BACKUP 文を使用する必要があります。              | 一貫性バックアップの生成にはロールバックまたは UNDO セグメントが必要です。 |
| 増分バックアップ                   | サポートされています。  | サポートされていません。  | サポートされていません。                             |
| 破損ブロックの検出                  | サポートされています。破損ブロックを識別して V\$DATABASE_BLOCK_CORRUPTION に記録します。  | サポートされていません。  | サポートされています。エクスポート・ログで破損ブロックを識別します。       |
| バックアップに含めるファイルの自動指定        | サポートされています。バックアップするすべてのファイル（データベース全体、表領域、データファイル、制御ファイルなど）の名前および場所を指定します。  | サポートされていません。バックアップするファイルを手動で検索し、コピーする必要があります。           | 該当なし。                                    |
| バックアップ・リポジトリ               | サポートされています。バックアップは、Recovery Manager メタデータのメイン・リポジトリである制御ファイルに記録されます。また、このメタデータを別のデータベースのスキーマである <b>リカバリ・カタログ</b> に格納することもできます。                   | サポートされていません。DBA は、独自のバックアップ・レコードを保持する必要があります。           | サポートされていません。                             |
| メディア・マネージャへのバックアップ         | サポートされています。メディア・マネージャとのインタフェースとして機能します。また、Recovery Manager は、プロキシ・コピーもサポートしています。プロキシ・コピーは、メディア・マネージャでディスクとバックアップ・メディア間のデータの転送を完全に管理できるようにする機能です。 | サポートされています。テープへのバックアップは、手動で行うか、またはメディア・マネージャによって制御されます。 | サポートされていません。                             |
| 初期化パラメータ・ファイルのバックアップ       | サポートされています。  | サポートされています。   | サポートされていません。                             |
| パスワードおよびネットワーク・ファイルのバックアップ | サポートされていません。   | サポートされています。   | サポートされていません。                             |
| プラットフォームに依存しないバックアップ用言語    | サポートされています。  | サポートされていません。  | サポートされています。                              |

## Oracle フラッシュバック技術

『Oracle Database 概要』で説明されているように、**Oracle フラッシュバック技術**は、物理バックアップおよびリカバリ計画を補完します。この機能セットによって、データ保護の追加レイヤーが提供されます。具体的には、フラッシュバック機能を使用すると、バックアップのリストアまたは **Point-in-Time リカバリ** を実行せずに、過去の状態のデータを表示したり、データベースを巻き戻すことができます。通常、フラッシュバック機能が適用されるほとんどの場合は、フラッシュバック機能の方がメディア・リカバリより効率的で簡単です。

### 論理フラッシュバック機能

Oracle のほとんどのフラッシュバック機能は論理レベルで実行されるため、データベース・オブジェクトの表示および操作を行うことができます。Oracle の論理レベルのフラッシュバック機能は、Recovery Manager に依存しないため、バックアップ計画で Recovery Manager を使用するかどうかに関係なく使用できます。フラッシュバック・ドロップを除き、論理フラッシュバック機能は **UNDO データ** に依存しています。UNDO データは、各データベースの更新の影響およびその更新によって上書きされる値に関するレコードです。

Oracle Database には、次の論理フラッシュバック機能が含まれています。

- Oracle Flashback Query

目標時点を指定して、データベースに対する問合せを実行し、目標時点に表示される結果を確認できます。表への更新などの不要な変更からリカバリするには、エラー発生前の目標時点を選択し、消失した行の内容を取得する問合せを実行します。この機能の使用の詳細は、『Oracle Database アドバンスド・アプリケーション開発者ガイド』を参照してください。

- Oracle Flashback Version Query

指定した時間間隔で 1 つ以上の表に存在していたすべての行のすべてのバージョンを表示できます。様々なバージョンの行に関するメタデータ（バージョンを作成したトランザクションの開始時刻、終了時刻、操作、トランザクション ID など）を取得することもできます。この機能を使用すると、消失したデータ値をリカバリし、問い合わせた表に対する変更を監査できます。この機能の使用の詳細は、『Oracle Database アドバンスド・アプリケーション開発者ガイド』を参照してください。

- Oracle Flashback Transaction Query

一定期間 1 つのトランザクションによって行われた変更またはすべてのトランザクションによって行われた変更を表示できます。この機能の使用の詳細は、『Oracle Database アドバンスド・アプリケーション開発者ガイド』を参照してください。

- Oracle Flashback Transaction

トランザクションを無効にできます。Oracle Database によって、トランザクション間の依存性が判別され、不要な変更を無効にする補正トランザクションが実際に作成されます。データベースは、トランザクション、およびそのトランザクションに依存している他のすべてのトランザクションが発生しなかった場合と同様の状態まで巻き戻されます。この機能の使用の詳細は、『Oracle Database アドバンスド・アプリケーション開発者ガイド』を参照してください。

- Oracle Flashback Table

データベースのいずれの部分もオフラインにすることなく、表または表のセットを過去の指定した時点にリカバリできます。ほとんどの場合、フラッシュバック表を使用すると、より複雑な Point-in-Time リカバリ操作を行う必要がなくなります。フラッシュバック表では、表がリストアされ、同時に現在の索引、トリガーおよび制約などの関連する属性が自動的にメンテナンスされます。これによって、ユーザーはデータベース固有のプロパティを検索してリストアすることを回避できます。この機能の使用の詳細は、16-4 ページの「**フラッシュバック表を使用した表の巻戻し**」を参照してください。



## ■ Oracle Flashback Drop

DROP TABLE 文の結果を無効にできます。この機能の使用の詳細は、16-7 ページの「[フラッシュバック・ドロップを使用した DROP TABLE 操作の巻戻し](#)」を参照してください。

**フラッシュバック・データ・アーカイブ**では、一部の論理フラッシュバック機能を使用して、非常に古いデータにアクセスできます。フラッシュバック・データ・アーカイブは、1つ以上の表領域または表領域の一部で構成されています。フラッシュバック・データ・アーカイブを作成する場合は、名前、保存期間および表領域を指定します。デフォルトのフラッシュバック・データ・アーカイブを指定することもできます。保持期間を経過すると、古い履歴データは自動的に消去されます。

フラッシュバック・アーカイブは、個々の表に対して有効または無効にすることができます。デフォルトでは、フラッシュバック・アーカイブはすべての表に対して無効になっています。

### 参照：

- フラッシュバック表およびフラッシュバック・ドロップを実行する方法については、第 16 章「[フラッシュバックおよびデータベースの Point-in-Time リカバリの実行](#)」を参照してください。
- 論理フラッシュバック機能の詳細は、『Oracle Database アドバンスド・アプリケーション開発者ガイド』を参照してください。

## フラッシュバック・データベース

物理レベルでは、**Oracle Flashback Database** を使用すると、**データベースの Point-in-Time リカバリ**より効率的なデータ保護を行うことができます。現行のデータファイルに不要な変更が含まれている場合は、Recovery Manager コマンド FLASHBACK DATABASE を使用して、データファイルを過去の時点の内容に戻すことができます。最終的に生成される結果は DBPITR の結果に類似していますが、バックアップからデータファイルをリストアする必要はなく、メディア・リカバリと比較すると REDO を使用する必要が少ないため、通常、はるかに短時間で戻されます。

フラッシュバック・データベースでは、過去のバージョンのデータ・ブロックへのアクセスに **フラッシュバック・ログ**を使用します。また、アーカイブ REDO ログに含まれている情報も使用します。フラッシュバック・データベースを使用するには、データベース用の **フラッシュ・リカバリ領域**を構成する必要があります。フラッシュバック・ログは、この領域にのみ格納できるためです。デフォルトでは、フラッシュバック・ロギングは有効になっていません。フラッシュバック・ログに使用される領域は、データベースによって自動的に管理され、フラッシュ・リカバリ領域内の他のファイルに必要な領域との間でバランスが取られています。

Oracle Database では、フラッシュバック・データベース、バックアップおよびリカバリと組み合わせてリストア・ポイントも使用できます。**リストア・ポイント**は、**システム変更番号**に対応する別名です。データベースの一部またはすべてを任意の時点の内容に戻す必要が予測される場合は、リストア・ポイントをその時点に作成できます。**保証付きリストア・ポイント**によって、フラッシュバック・データベースを使用してデータベースをリストア・ポイントの時点に戻すことができます。

**参照：** FLASHBACK DATABASE コマンドを使用したフラッシュバック・データベースの実行方法については、16-11 ページの「[フラッシュバック・データベースを使用したデータベースの巻戻し](#)」を参照してください。

## データ・リカバリ・アドバイザー

Oracle Database には、**データ・リカバリ・アドバイザー** ツールが含まれています。このツールは、永続的なデータの障害を診断し、適切な修復オプションを提示して、ユーザーの要求に応じて修復を実行します。データ・リカバリ・アドバイザーによって、すべての Oracle バックアップおよびリカバリのソリューション用のシングル・ポイント・エントリが提供されます。データ・リカバリ・アドバイザーは、Enterprise Manager の Database Control または Grid Control のコンソール、あるいは Recovery Manager のコマンドライン・クライアントを介して使用できます。

通常、データベース障害は、エラー・メッセージ、アラート、トレース・ファイル、トレース・ダンプ、データ整合性チェックの失敗などの様々な症状を示します。データ・リカバリ・アドバイザーは、自動的に診断を行い、これらの障害に関してユーザーに通知します。データ・リカバリ・アドバイザーのコンテキストでは、**障害**とは、一連の修復処置に直接マップできる永続的なデータの破損のことで、各障害には、OPEN または CLOSED のステータスがあります。また、各障害には、CRITICAL、HIGH または LOW の優先順位があります。

障害は、データ整合性チェックによって検出されます。データ整合性チェックとは、データベースまたはそのコンポーネントの状態を評価するために実行する診断手順のことで、**データ整合性チェック**によって障害が検出された場合、データ・リカバリ・アドバイザーは一連の障害による影響を自動的に評価し、一連の修復オプションに結果をマップします。ほとんどの場合、データ・リカバリ・アドバイザーは自動修復オプションおよび手動修復オプションの両方を提示します。

データ・リカバリ・アドバイザーは、最適な自動修復オプションおよびデータベースへのその影響を判断します。**修復オプション**には、データファイルのリストアおよびリカバリ、メディア・リカバリ、フラッシュバック・データベースなどの修復が含まれています。データ・リカバリ・アドバイザーは、自動修復オプションを提示する前に、特定の環境でのそのオプションの妥当性、および提示する修復の実行に必要なメディア・コンポーネントの可用性を検証します。

自動修復オプションを選択すると、Recovery Manager によって Oracle Database でのセッションが調整され、**修復**が自動的に実行されます。データ・リカバリ・アドバイザー ツールは、修復が成功したことを確認し、該当する障害をクローズします。

**参照：** データ・リカバリ・アドバイザーの使用方法については、第 14 章「データ・リカバリ・アドバイザーを使用した障害の診断および修復」を参照してください。

## バックアップおよびリカバリのドキュメント・ロードマップ

図 1-1 に、バックアップおよびリカバリのドキュメントにナビゲートする場合の推奨方法を示します。ロードマップは、Recovery Manager によるバックアップとリカバリ、およびユーザー管理によるバックアップとリカバリの 2 つのメイン・パスに分かれています。それぞれのメイン・パスから分岐し、再度結合するオプションのパスも示されています。

Oracle Database について十分な知識がなく、バックアップ・リカバリについて学習する必要がある場合は、『Oracle Database 2 日でデータベース管理者』を参照してください。バックアップおよびリカバリに関する章で、Oracle Enterprise Manager を使用して基本的な操作を実行する方法が説明されています。また、必要に応じて、『Oracle Database 概要』の関連する章を読むことによって、バックアップおよびリカバリの基本的な原則に関する知識を深めることができます。

図 1-1 バックアップおよびリカバリのドキュメント・ロードマップ

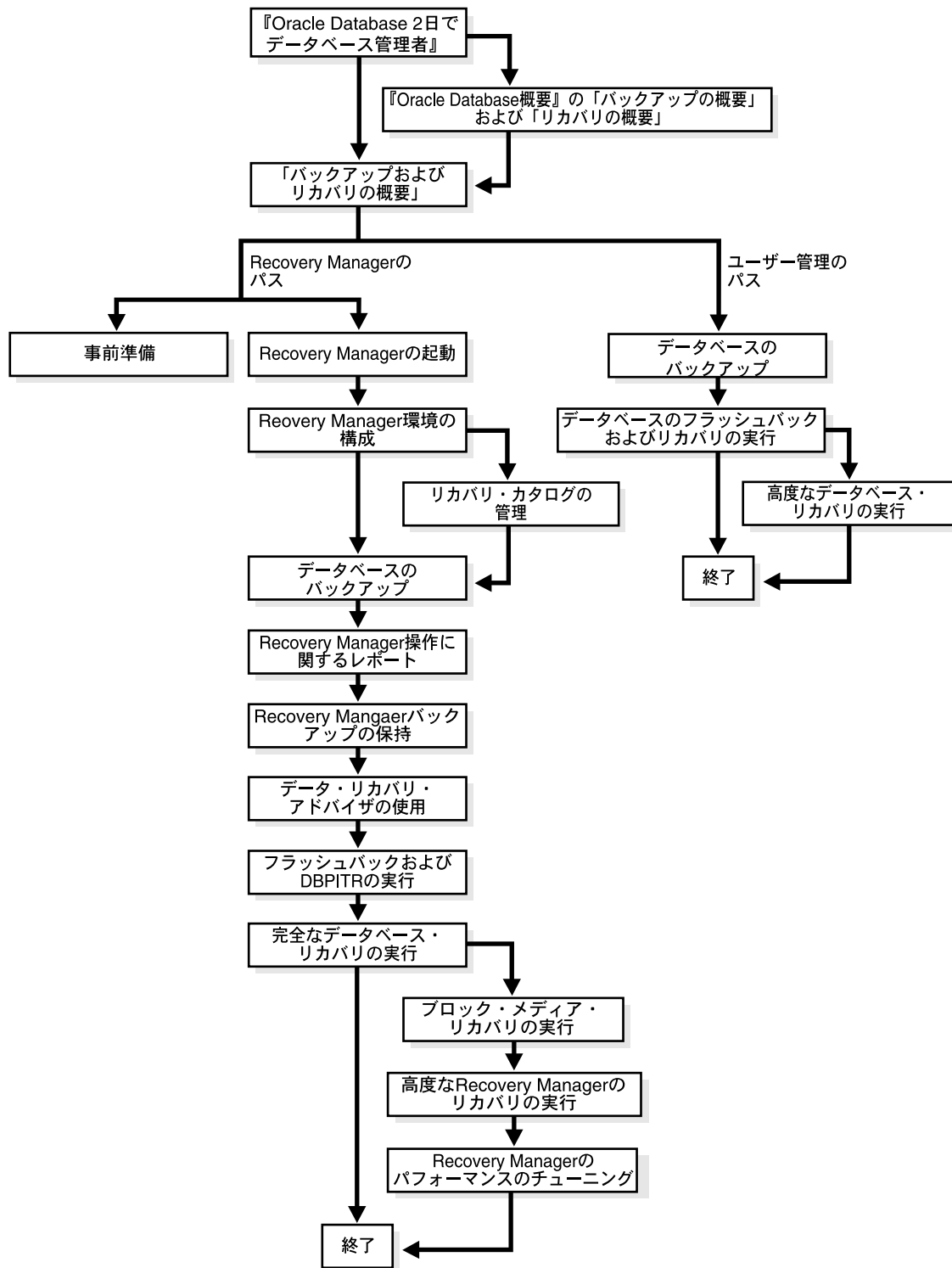


図 1-1 に示すように、バックアップおよびリカバリ計画の実装には、Recovery Manager を使用する方法（推奨）、またはユーザー管理ツールを使用する方法を使用できます。

## Recovery Manager のドキュメント・ロードマップ

Recovery Manager を基本的なバックアップおよびリカバリ・ソリューションとして使用する場合は、最初に 2-1 ページの「[Recovery Manager の概要](#)」を参照してください。この章は、Recovery Manager の最も基本的な方法について説明しており、目的に適しています。Recovery Manager を使用したバックアップおよびリカバリ計画の実装方法の包括的な説明は、次の順序に従って各章を参照してください（オプションの章は示されていません）。

1. [第 4 章「Recovery Manager クライアントの起動および操作」](#) を参照してください。  
この章では、Recovery Manager クライアントを起動してデータベースに接続する方法について説明します。
2. [第 5 章「Recovery Manager 環境の構成」](#) を参照してください。  
この章では、[フラッシュ・リカバリ領域](#)、[バックアップの保存方針](#)および[アーカイブ REDO ログの削除方針](#)の構成などの基本的なタスクの実行方法について説明します。
3. [第 8 章「データベースのバックアップ」](#) を参照してください。  
この章では、基本的なバックアップ計画を実装する方法について説明します。
4. [第 10 章「Recovery Manager 操作に関するレポート」](#) を参照してください。  
この章では、Recovery Manager のバックアップおよびリカバリ操作の監視方法について説明します。具体的には、レポート・コマンド（LIST、REPORT および SHOW）および関連する V\$ ビューとリカバリ・カタログ・ビューの使用方法について説明します。
5. [第 11 章「Recovery Manager バックアップおよびリポジトリ・レコードのメンテナンス」](#) を参照してください。  
この章では、バックアップの存在の確認、バックアップのリポジトリ・ステータスの変更、バックアップの削除、他のメンテナンス・タスクの実行を行う方法について説明します。
6. [第 14 章「データ・リカバリ・アドバイザを使用した障害の診断および修復」](#) を参照してください。  
この章では、データ・リカバリ・アドバイザ・ツールの使用方法について説明します。このツールを使用して、障害の表示、それらの障害への対応に関するアドバイスの入手、および状況によっては障害に対する自動修復を行うことができます。
7. [第 16 章「フラッシュバックおよびデータベースの Point-in-Time リカバリの実行」](#) を参照してください。  
この章では、FLASHBACK DATABASE コマンドの使用法、および RECOVER DATABASE コマンドを使用した Point-in-Time リカバリの実行方法について説明します。
8. [第 17 章「データベースの完全リカバリの実行」](#) を参照してください。  
この章では、個々の表領域またはデータベースのリカバリ方法について説明します。

## ユーザー管理のバックアップおよびリカバリのドキュメント・ロードマップ

Recovery Manager を基本的なバックアップおよびリカバリ・ソリューションとして使用しない場合は、サード・パーティのツールを使用してバックアップを作成し、SQL または SQL\*Plus のコマンドを使用してリカバリを実行する必要があります。次の順序に従って各章を参照してください。

1. [第 27 章「ユーザー管理データベース・バックアップの作成」](#) を参照してください。  
この章では、サード・パーティのツールを使用したバックアップの作成方法について説明します。
2. [第 28 章「ユーザー管理のデータベースのフラッシュバックおよびリカバリの実行」](#) を参照してください。  
この章では、SQL 文 FLASHBACK DATABASE の使用法、および SQL\*Plus の RECOVER コマンドを使用したリカバリの実行方法について説明します。

3. [第29章「ユーザー管理のリカバリの実行: 高度な例」](#)を参照してください。  
この章では、様々なリカバリ・シナリオについて説明します。



---

---

## Recovery Manager の概要

この章は、このマニュアルの各章の詳細を最初に読まずに、Recovery Manager をすぐに使用し始める必要がある新しいユーザーを対象としています。この章では、Recovery Manager の最も重要な概念とタスクについてできるかぎり簡潔に説明しますが、この章を他のバックアップおよびリカバリのドキュメント・セットのかわりにすることはできません。

この章の内容は、次のとおりです。

- [Recovery Manager 環境の概要](#)
- [Recovery Manager の起動およびデータベースへの接続](#)
- [Recovery Manager のデフォルト構成の表示](#)
- [データベースのバックアップ](#)
- [Recovery Manager 操作に関するレポート](#)
- [Recovery Manager バックアップの保持](#)
- [データ・リカバリ・アドバイザを使用した障害の診断および修復](#)
- [フラッシュバック・データベースを使用したデータベースの巻戻し](#)
- [データベース・ファイルのリストアおよびリカバリ](#)

## Recovery Manager 環境の概要

Recovery Manager (RMAN) とは、データベースでバックアップおよびリカバリ・タスクを実行し、バックアップ計画の管理を自動化する Oracle Database クライアントのことです。Recovery Manager によって、データベースのバックアップ、リストアおよびリカバリが大幅に簡単になります。

Recovery Manager 環境は、データのバックアップに使用されるユーティリティとデータベースで構成されます。Recovery Manager の環境には、最低限次のコンポーネントが含まれている必要があります。

- **ターゲット・データベース**

Recovery Manager が TARGET キーワードで接続されている Oracle Database。ターゲット・データベースは、Recovery Manager がバックアップおよびリカバリ操作を実行するデータベースです。Recovery Manager は、常にデータベースの操作に関するメタデータをデータベースの制御ファイルに保持します。Recovery Manager のメタデータは、**Recovery Manager リポジトリ**と呼ばれます。

- **Recovery Manager クライアント**

コマンドを解釈し、それらのコマンドを実行するようにサーバー・セッションに指示し、そのアクティビティをターゲット・データベースの制御ファイルに記録する Oracle Database 実行可能ファイル。この Recovery Manager 実行可能ファイルは、データベースとともに自動的にインストールされ、通常、他のデータベース実行可能ファイルと同じディレクトリに配置されます。たとえば、Linux の場合、Recovery Manager クライアントは \$ORACLE\_HOME/bin に配置されています。

一部の環境では、次のオプションのコンポーネントが使用されます。

- **フラッシュ・リカバリ領域**

バックアップおよびリカバリに関連するファイルをデータベースで格納および管理できるディスクの場所。フラッシュ・リカバリ領域の場所およびサイズは、DB\_RECOVERY\_FILE\_DEST および DB\_RECOVERY\_FILE\_DEST\_SIZE 初期化パラメータを使用して設定します。

- **メディア・マネージャ**

Recovery Manager がテープ・ライブラリなどのシーケンシャル・メディア・デバイスとインタフェースを取るために必要なアプリケーション。メディア・マネージャは、バックアップおよびリカバリ中にこれらのデバイスを制御し、メディアのロード、ラベル付けおよびアンロードを管理します。メディア管理デバイスは **SBT** (テープへのシステム・バックアップ) デバイスと呼ばれる場合があります。

- **リカバリ・カタログ**

1 つ以上のターゲット・データベースに対する Recovery Manager アクティビティを記録するために使用する個別のデータベース・スキーマ。リカバリ・カタログは、制御ファイルが消失した場合に Recovery Manager リポジトリ・メタデータを保持することで、制御ファイル消失後のリストアおよびリカバリを簡単に実行できるようにします。データベースは制御ファイル内の古いレコードを上書きする場合がありますが、Recovery Manager はユーザーが削除しないかぎり、カタログ内にレコードを永久に保持します。

この章では、リカバリ・カタログまたはメディア・マネージャを使用しない最も基本的な構成で Recovery Manager を使用方法について説明します。

**参照：** Recovery Manager 環境の詳細は、第 3 章「Recovery Manager のアーキテクチャ」を参照してください。



## Recovery Manager の起動およびデータベースへの接続

Recovery Manager クライアントは、オペレーティング・システムのコマンド・プロンプトで `rman` コマンドを発行して起動します。Recovery Manager を起動すると、次の例に示すようなコマンド・プロンプトが表示されます。

```
% rman
RMAN>
```

Recovery Manager によるデータベースへの接続は、SQL\*Plus によるデータベースへの接続と同じ方法で指定および認証します。Recovery Manager によるターゲット・データベースまたは補助データベースへの接続には SYSDBA 権限が必要である点のみが異なります。AS SYSDBA キーワードは、暗黙的に指定され、明示的には指定できません。SQL\*Plus を使用する場合は、データベース接続オプションについては、『Oracle Database 管理者ガイド』を参照してください。

---

**注意：** 適切なセキュリティ・プラクティスとして、コマンドラインにパスワードを入力するときは、ブレーン・テキストで入力しないでください。Recovery Manager では、Recovery Manager プロンプトで要求された場合にのみパスワードを入力してください。パスワード保護については、『Oracle Database セキュリティ・ガイド』を参照してください。

---

データベースには、コマンドライン・オプションまたは `CONNECT TARGET` コマンドを使用して接続できます。次の例では、Recovery Manager を起動し、Oracle Net を介してターゲット・データベースに接続します (AS SYSDBA は暗黙的に指定されるため、指定されていないことに注意してください)。Recovery Manager によってパスワードの入力を求められます。

```
% rman
RMAN> CONNECT TARGET SYS@prod

target database Password: password
connected to target database: PROD (DBID=39525561)
```

次の例では、Recovery Manager を起動してから、オペレーティング・システム認証を使用してターゲット・データベースに接続します。

```
% rman
RMAN> CONNECT TARGET /

connected to target database: PROD (DBID=39525561)
```

Recovery Manager クライアントを終了するには、Recovery Manager プロンプトで `EXIT` と入力します。

```
RMAN> EXIT
```

### 一般的な Recovery Manager コマンドライン・オプションの構文

```
RMAN
[ TARGET connectStringSpec
| { CATALOG connectStringSpec }
| LOG ['] filename ['] [ APPEND ]
.
.
.
]...

connectStringSpec ::=
['] [userid] [/ [password]] [@net_service_name] [']
```

次の例では、/tmp/msglog.log にあるテキスト・ファイルに **Recovery Manager セッション** の出力を追加します。

```
% rman TARGET / LOG /tmp/msglog.log APPEND
```

**参照：** Recovery Manager クライアントの起動および操作の詳細は、[第 4 章「Recovery Manager クライアントの起動および操作」](#)を参照してください。

## Recovery Manager のデフォルト構成の表示

Recovery Manager のバックアップおよびリカバリ環境は、ターゲット・データベースごとに事前構成されます。この構成は、Recovery Manager を終了して再起動した場合でも永続し、このターゲット・データベースで行うその後のすべての操作に適用されます。

Recovery Manager の構成済の設定では、バックアップ・デバイスを指定し、バックアップ・デバイスへの接続 (**チャンネル**と呼ばれる)、バックアップ計画に影響する方針などを構成できます。ほとんどの場合は、デフォルトの構成で十分です。

### データベースの現行の構成を表示する手順

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. SHOW ALL コマンドを実行します。

たとえば、Recovery Manager プロンプトでコマンドを次のように入力します。

```
RMAN> SHOW ALL;
```

出力には、この構成を再作成するための CONFIGURE コマンドが表示されます。

**参照：** Recovery Manager 環境の構成方法については、[第 5 章「Recovery Manager 環境の構成」](#)および[第 6 章「Recovery Manager 環境の構成：高度なトピック」](#)を参照してください。

## データベースのバックアップ

ファイルをバックアップするには、BACKUP コマンドを使用します。Recovery Manager は、要求されたバックアップのタイプ用に構成されているデフォルトのデバイスにデータをバックアップします。デフォルトでは、Recovery Manager はディスクにバックアップを作成します。フラッシュ・リカバリ領域が有効になっていて、FORMAT パラメータ ([表 2-1](#)を参照) を指定していない場合、Recovery Manager は、リカバリ領域にバックアップを作成し、自動的に一意の名前を指定します。

デフォルトでは、Recovery Manager はイメージ・コピーではなくバックアップ・セットを作成します。**バックアップ・セット**は、1 つ以上のバックアップ・ピースで構成されています。バックアップ・ピースとは、Recovery Manager のみがアクセス可能な形式で書き込まれた物理ファイルのことです。**多重バックアップ・セット**には、複数の入力ファイルからのブロックが含まれています。Recovery Manager は、バックアップ・セットをディスクまたはテープに書き込むことができます。

BACKUP AS COPY を指定すると、Recovery Manager は各ファイルを**イメージ・コピー**としてコピーします。イメージ・コピーは、ディスクに作成されたデータベース・ファイルのビットごとのコピーです。イメージ・コピーは、Linux の cp や Windows の COPY などのオペレーティング・システム・コマンドで作成したコピーと同じですが、Recovery Manager のリポジトリに記録されるため、Recovery Manager で使用できます。Recovery Manager を使用すると、データベースがオープンしている間にイメージ・コピーを作成できます。

**参照：**

- Recovery Manager バックアップの概念については、第7章「Recovery Manager バックアップの概要」を参照してください。
- Recovery Manager でデータベース・ファイルをバックアップする方法については、第8章「データベースのバックアップ」を参照してください。
- BACKUP コマンドの構文およびセマンティクスについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## ARCHIVELOG モードでのデータベースのバックアップ

データベースは、ARCHIVELOG モードで稼働している場合、オープン状態でバックアップできます。このバックアップは、データベースを一貫性のある状態にするためにリカバリ時に REDO が必要となるため、**非一貫性バックアップ**と呼ばれます。オープン状態のデータベースのバックアップは、バックアップのリカバリに必要なアーカイブ REDO ログが存在するかぎり、データ保護の手段として一貫性バックアップと同様に効率的です。

### データベースをオープンした状態でデータベースおよびアーカイブ REDO ログをバックアップする手順

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. BACKUP DATABASE コマンドを実行します。

たとえば、Recovery Manager プロンプトで次のコマンドを入力して、データベースおよびすべてのアーカイブ REDO ログ・ファイルをデフォルトのバックアップ・デバイスにバックアップします。

```
RMAN> BACKUP DATABASE PLUS ARCHIVELOG;
```

## NOARCHIVELOG モードでのデータベースのバックアップ

データベースが NOARCHIVELOG モードで稼働している場合、有効なデータベースのバックアップは**一貫性バックアップ**のみです。バックアップに一貫性を持たせるには、データベースを一貫性のある状態で停止させた後でマウントする必要があります。バックアップをリストアした後で、リカバリを実行する必要はありません。

### 一貫性のあるデータベースのバックアップを作成する手順

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. データベースを一貫した状態で停止してから、データベースをマウントします。

たとえば、バックアップのためにデータベースを一貫性のある状態にするには、次のコマンドを入力します。

```
RMAN> SHUTDOWN IMMEDIATE;
RMAN> STARTUP FORCE DBA;
RMAN> SHUTDOWN IMMEDIATE;
RMAN> STARTUP MOUNT;
```

3. BACKUP DATABASE コマンドを実行します。

たとえば、Recovery Manager プロンプトで次のコマンドを入力して、データベースをデフォルトのバックアップ・デバイスにバックアップします。

```
RMAN> BACKUP DATABASE;
```

次のコマンドは、データベース内のすべてデータファイルのイメージ・コピー・バックアップを作成します。

```
RMAN> BACKUP AS COPY DATABASE;
```

4. データベースをオープンして、通常の操作を再開します。  
データベースをオープンするには、次のコマンドを入力します。

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

## 一般的なバックアップ・オプション

BACKUP コマンドには、バックアップの出力を制御する多くのオプション、パラメータおよび句が含まれています。次の表に、一般的なバックアップ・オプションの一部を示します。

表 2-1 一般的なバックアップ・オプション

| オプション  | 説明   | 例  |
|--------|--|--|
| FORMAT | バックアップ・ピースおよびコピー用の位置と名前を指定します。一意のファイル名を生成するには、置換変数を使用する必要があります。<br><br>最も一般的な置換変数は、一意の名前を生成する %U です。その他の置換変数には、DB_NAME を示す %d、バックアップ・セットのタイムスタンプを示す %t、バックアップ・セット番号を示す %s およびバックアップ・ピース番号を示す %p があります。 | BACKUP<br>FORMAT 'AL_%d/%t/%s/%p'<br>ARCHIVELOG LIKE '%arc_dest%'; |
| TAG    | バックアップ用のラベルとしてユーザー定義文字列を指定します。 <b>タグ</b> を指定しない場合、Recovery Manager は、日付と時刻を含むデフォルト・タグを割り当てます。Recovery Manager リポジトリでは、タグは常に大文字で格納されることに注意してください。   | BACKUP<br>TAG 'weekly_full_db_bkup'<br>DATABASE MAXSETSIZE 10M;    |

参照: 「バックアップ出力オプションの指定」 (8-3 ページ)

## 増分バックアップの作成

BACKUP INCREMENTAL を指定すると、Recovery Manager はデータベースの**増分バックアップ**を作成します。増分バックアップでは、前回の増分バックアップ後に行われたデータベースへのブロックレベルの変更が取得されます。通常、増分バックアップは、データベースの全体バックアップより小さく、高速です。増分バックアップを使用したリカバリは、REDO ログのみを使用したバックアップより高速です。

増分バックアップ方針の開始点は、**レベル 0 の増分バックアップ** (データベースのすべてのブロックのバックアップ) です。レベル 0 の増分バックアップの内容は**全体バックアップ**と同じです。ただし、レベル 0 のバックアップは、全体バックアップと異なり、増分バックアップ計画の一部とみなされます。

レベル 1 の増分バックアップには、前回の増分バックアップ後に変更されたブロックのみが格納されます。レベル 1 のバックアップの実行時に現行または親のデータベース・**インカネーション**にレベル 0 のバックアップが存在しない場合は、Recovery Manager によってレベル 0 のバックアップが自動的に作成されます。

---

**注意:** 増分バックアップは、一貫性のある状態での停止の後にデータベースをマウントする場合は作成できますが、NOARCHIVELOG データベースをオープンしている場合は作成できません。

---

レベル 1 のバックアップは、**累積増分バックアップ** (最新のレベル 0 のバックアップ後に変更されたブロックをすべて含む) または**差分増分バックアップ** (最新の増分バックアップ後に変更されたブロックのみを含む) のいずれかにできます。増分バックアップは、デフォルトでは差分バックアップです。

増分バックアップからのリストア時には、レベル0のバックアップが開始点として使用され、その後、変更を REDO から1つずつ再適用することを回避できるレベル1のバックアップに基づいて、変更されたブロックが更新されます。増分バックアップを使用したりカバリでは、ユーザーがその他の作業を行う必要はありません。増分バックアップが使用可能な場合、Recovery Manager はリカバリ時に増分バックアップを使用します。

### データベースの増分バックアップを作成する手順

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. BACKUP INCREMENTAL コマンドを実行します。

次の例では、増分バックアップ計画の基礎となるレベル0の増分バックアップを作成します。

```
BACKUP INCREMENTAL LEVEL 0 DATABASE;
```

次の例では、レベル1の累積増分バックアップを作成します。

```
BACKUP INCREMENTAL LEVEL 1 CUMULATIVE DATABASE;
```

次の例では、レベル1の差分増分バックアップを作成します。

```
BACKUP INCREMENTAL LEVEL 1 DATABASE;
```

**参照：** 増分バックアップの概念の詳細は、7-13 ページの「[増分バックアップ](#)」および8-15 ページの「[増分バックアップの作成および更新](#)」を参照してください。

### 増分更新バックアップの作成

Recovery Manager の増分更新バックアップ機能は、効率的な増分バックアップ・ルーチンです。レベル1のバックアップからの変更によって、イメージ・コピーのレベル0の増分バックアップがロールフォワードされます。このため、この変更には、レベル1の増分バックアップが作成された SCN の時点でのすべての変更が含まれます。レベル1の増分バックアップからのすべての変更はすでに適用されているため、更新されたレベル0の増分バックアップのリカバリが高速になります。

BACKUP FOR RECOVER OF COPY コマンドを指定すると、データベースの指定したデータファイル・コピー（レベル0の増分バックアップ）の SCN 以降のすべての変更が増分バックアップに含まれるようになります。次の表に、増分更新バックアップ計画の実装のために、FOR RECOVER OF COPY で使用するオプションを示します。

表 2-2 FOR RECOVER OF COPY のオプション

| BACKUP オプション                                     | 説明   | 例   |
|--|--|---|
| FOR RECOVER OF COPY WITH TAG 'tag_name'          | TAG パラメータを使用して、増分バックアップの基礎となるレベル0の増分バックアップを識別します。指定したタグが付けられたデータファイル・コピーが現行または親のデータベース・ <b>インカネーション</b> に存在しない場合は、指定したタグが付けられたレベル0のデータファイル・コピーが Recovery Manager によって作成されます。 | BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY WITH TAG 'incr_update' DATABASE;               |
| FOR RECOVER OF COPY DATAFILECOPY FORMAT 'format' | 増分バックアップの基礎として使用するデータファイル・コピーを識別します。   | BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY DATAFILECOPY FORMAT '/disk2/df1.cpy' DATABASE; |

### 増分更新バックアップ計画を実装する手順

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. RECOVER COPY および BACKUP INCREMENTAL コマンドを実行します。

増分更新バックアップに基づく計画を実装するには、次のスクリプトを定期的に行う必要があります。

```
RECOVER COPY OF DATABASE
  WITH TAG 'incr_update';
BACKUP
  INCREMENTAL LEVEL 1
  FOR RECOVER OF COPY WITH TAG 'incr_update'
  DATABASE;
```

**参照:** 「増分更新バックアップ」 (8-18 ページ)

## データベース・ファイルおよびバックアップの検証

VALIDATE コマンドを使用すると、すべてのデータベース・ファイルが適切な場所に存在しており、物理的に破損していないかどうかを確認できます。CHECK LOGICAL オプションを指定すると、論理ブロックの破損も確認できます。

### データベース・ファイルを検証する手順

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. 必要なファイルに対して VALIDATE コマンドを実行します。

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・ファイルの物理的および論理的な破損を検証するには、次のコマンドを入力します。

```
BACKUP VALIDATE CHECK LOGICAL
  DATABASE ARCHIVELOG ALL;
```

次の例に示すように、VALIDATE コマンドを使用すると、個々のデータ・ブロックを検証することができます。

```
VALIDATE DATAFILE 4 BLOCK 10 TO 13;
```

次の例に示すように、バックアップ・セットを検証することもできます。

```
VALIDATE BACKUPSET 3;
```

バックアップ・セットの指定には、LIST BACKUP コマンドの出力に表示されるプライマリ・キーを使用します。

**参照:** 第 15 章「データベース・ファイルおよびバックアップの検証」

## Recovery Manager 操作のスクリプト化

Recovery Manager では、週次バックアップなどの繰り返し行うタスクを管理するためのコマンド・ファイルを使用できます。**コマンド・ファイル**とは、Recovery Manager プロンプトで入力する Recovery Manager コマンドと同じコマンドが含まれているクライアント側のテキスト・ファイルのことです。任意のファイル拡張子を指定できます。RUN コマンドを使用すると、スクリプト内の処理のフローをある程度制御できます。

**コマンド・ファイルを作成して実行する手順**

1. テキスト・エディタを使用してコマンド・ファイルを作成します。

たとえば、次の内容でコマンド・ファイルを作成します。

```
# my_command_file.txt
CONNECT TARGET /
BACKUP DATABASE PLUS ARCHIVELOG;
LIST BACKUP;
EXIT;
```

2. Recovery Manager を起動して、Recovery Manager プロンプトで @ コマンドを実行することによってコマンド・ファイルの内容を実行します。

```
% rman
RMAN> @/my_dir/my_command_file.txt # runs specified command file
```

次に示すように、実行するコマンド・ファイルを使用して Recovery Manager を起動することもできます。

```
% rman @/my_dir/my_command_file.txt
```

**参照:** コマンド・ファイルの詳細は 4-4 ページの「[Recovery Manager のコマンド・ファイルの使用](#)」、コマンド・ファイルで置換変数を使用して実行時にパラメータを渡す方法は 4-5 ページの「[コマンド・ファイルでの置換変数の使用](#)」を参照してください。

## Recovery Manager 操作に関するレポート

Recovery Manager の LIST および REPORT コマンドは、Recovery Manager のリポジトリに基づいて、バックアップ・アクティビティに関するレポートを生成します。SHOW ALL コマンドを使用すると、Recovery Manager の現行の構成が表示されます。

### バックアップの表示

リポジトリに含まれているバックアップおよびデータファイル・コピーに関する情報を表示するには、LIST BACKUP および LIST COPY コマンドを実行します。バックアップに対しては、次の表に示すオプションを使用して、LIST の出力形式を制御できます。

**表 2-3 バックアップ用の LIST オプション**

| オプション     | 例                                    | 説明                                      |
|-----------|--------------------------------------|---|
| BY BACKUP | LIST BACKUP OF<br>DATABASE BY BACKUP | バックアップ・セットごとに出力を編成します。これがデフォルトの表示モードです。 |
| BY FILE   | LIST BACKUP BY FILE                  | バックアップされたファイルに従ってバックアップを表示します。          |
| SUMMARY   | LIST BACKUP SUMMARY                  | サマリー出力を表示します。デフォルトでは、出力は VERBOSE になります。 |

バックアップおよびコピーの両方に、次の追加オプションがあります。

表 2-4 追加の LIST オプション

| オプション       | 例                          | 説明   |
|-------------|----------------------------|--|
| EXPIRED     | LIST EXPIRED<br>COPY       | Recovery Manager リポジトリに記録されているが、前回の CROSSCHECK コマンドの実行時にディスクまたはテープ上の適切な場所に存在していなかったバックアップを表示します。<br><b>期限切れのバックアップ</b> は、オペレーティング・システムのユーティリティによって削除された可能性があります。 |
| RECOVERABLE | LIST BACKUP<br>RECOVERABLE | Recovery Manager リポジトリのステータスが AVAILABLE であり、リストアおよびリカバリも行うことができるデータファイルのバックアップまたはコピーを表示します。  |

#### バックアップおよびコピーを表示する手順

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. Recovery Manager プロンプトで、LIST コマンドを実行します。

次の例に示すように、特定のオブジェクトを表示できます。

```
LIST BACKUP OF DATABASE;
LIST COPY OF DATAFILE 1, 2;
LIST BACKUP OF ARCHIVELOG FROM SEQUENCE 10;
LIST BACKUPSET OF DATAFILE 1;
```

**参照：** LIST コマンドの詳細は、10-4 ページの「[バックアップおよびリカバリ関連オブジェクトの表示](#)」を参照してください。

## データベース・ファイルおよびバックアップに関するレポート

REPORT コマンドは、LIST より複雑な分析を実行します。次の表に、主なオプションを示します。

表 2-5 REPORT のオプション

| オプション         | 例                              | 説明  |
|---------------|--------------------------------|---|
| NEED BACKUP   | REPORT NEED<br>BACKUP DATABASE | 現行の保存方針に従ってバックアップする必要があるファイルを表示します。オプションの REDUNDANCY および RECOVERY WINDOW パラメータを使用して、様々な条件を指定します。                |
| OBSOLETE      | REPORT OBSOLETE                | 構成済の <b>バックアップの保存方針</b> に従って不要となっているバックアップを表示します。オプションの REDUNDANCY および RECOVERY WINDOW パラメータを使用して、デフォルトを上書きします。 |
| SCHEMA        | REPORT SCHEMA                  | 現在の時刻（デフォルト）または別の時刻のデータベース内の表領域およびデータファイルを報告します。  |
| UNRECOVERABLE | REPORT<br>UNRECOVERABLE        | データファイルの前のバックアップ以降に、データファイル内のオブジェクトに対してリカバリ不能な操作が実行されているすべてのデータファイルを表示します。                                      |



### データベース・ファイルおよびバックアップのレポートを生成する手順

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. Recovery Manager プロンプトで、REPORT コマンドを実行します。

次の例では、現在構成されているバックアップ保存方針に従って不要となったバックアップをレポートします。

```
REPORT OBSOLETE;
```

次の例では、データベースのデータファイルおよび一時ファイルをレポートします。

```
REPORT SCHEMA;
```

**参照：** Recovery Manager のレポートに REPORT コマンドを使用する方法は、10-11 ページの「[バックアップ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。

## Recovery Manager バックアップの保持

Recovery Manager リポジトリ・メタデータは、ターゲット・データベースの制御ファイルに常に格納されます。このメタデータは、バックアップの管理時に **Recovery Manager のメンテナンス・コマンド** で使用されます。

### バックアップのクロスチェック

CROSSCHECK コマンドは、Recovery Manager バックアップおよびコピーの論理レコードをストレージ・メディアのファイルと同期化します。バックアップがディスク上に存在する場合は、CROSSCHECK によって、そのファイルのヘッダーが有効かどうかを判別されます。バックアップがテープ上に存在する場合は、Recovery Manager によって、バックアップ・ピースの名前と場所の問合せが Recovery Manager リポジトリに対して実行されます。この場合、バックアップおよびコピーは、クロスチェックしてから削除することをお勧めします。

#### ディスク上のすべてのバックアップおよびコピーをクロスチェックする手順

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. 次の例に示すように、CROSSCHECK コマンドを実行します。

```
CROSSCHECK BACKUP;
CROSSCHECK COPY;
```

**参照：** Recovery Manager バックアップをクロスチェックする方法については、11-12 ページの「[Recovery Manager リポジトリのクロスチェック](#)」を参照してください。

### 不要なバックアップの削除

DELETE コマンドは、ディスクおよびテープから Recovery Manager のバックアップおよびコピーを削除し、制御ファイル・リポジトリ内のファイルのステータスを DELETED に更新し、リカバリ・カタログからレコードを削除します (カタログを使用している場合)。Recovery Manager を対話形式で実行し、NOPROMPT オプションを指定しなかった場合は、DELETE によってファイルのリストが表示され、リスト内のファイルを削除する前に確認を求められます。

Recovery Manager リポジトリに記録されているバックアップおよびデータファイルのコピーのうち不要なものが削除されるため、DELETE OBSOLETE コマンドは特に役に立ちます。DELETE コマンドでオプションを使用すると、不要なものを指定したり、構成済みの **バックアップの保存方針** を使用することができます。

**不要なバックアップおよびコピーを削除する手順**

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. 次の例に示すように、DELETE OBSOLETE コマンドを実行します。

```
DELETE OBSOLETE;
```

**参照：** DELETE コマンドの使用方法については、11-20 ページの「[Recovery Manager バックアップおよびアーカイブ REDO ログの削除](#)」を参照してください。

**データ・リカバリ・アドバイザを使用した障害の診断および修復**

データベースに関する問題を診断して修復する最も簡単な方法は、[データ・リカバリ・アドバイザ](#)を使用する方法です。この Oracle Database ツールによって、永続的なデータの障害を診断し、ユーザーに修復オプションを提示して、修復を自動的に実行するインフラストラクチャが提供されます。

**参照：**「[データ・リカバリ・アドバイザの概要](#)」(14-2 ページ)

**障害の表示および修復オプションの決定**

**障害**とは、状態モニターによって検出される永続的なデータの破損のことです。例としては、物理的および論理的なデータ・ブロックの破損やデータファイルの欠落などがあります。各障害には、**障害優先順位**および**障害ステータス**があります。優先順位は、CRITICAL、HIGH または LOW のいずれかになります。ステータスは、OPEN または CLOSED のいずれかになります。

LIST FAILURE コマンドを実行すると、すべての既知の障害を表示できます。障害が存在する場合は、同じセッションで ADVISE FAILURE コマンドを実行し、手動および自動の修復オプションを決定します。次に、これらの 2 つのコマンドの例を示します (出力例も示します)。

**例 2-1 LIST FAILURE および ADVISE FAILURE**

```
RMAN> LIST FAILURE;
```

```
List of Database Failures
=====
```

| Failure ID | Priority | Status | Time Detected | Summary  |
|------------|----------|--------|---------------|--|
| 142        | HIGH     | OPEN   | 23-APR-07     | One or more non-system datafiles are missing                                       |
| 101        | HIGH     | OPEN   | 23-APR-07     | Datafile 1: '/disk1/oradata/prod/system01.dbf' contains one or more corrupt blocks |

```
RMAN> ADVISE FAILURE;
```

```
List of Database Failures
=====
```

| Failure ID | Priority | Status | Time Detected | Summary  |
|------------|----------|--------|---------------|--|
| 142        | HIGH     | OPEN   | 23-APR-07     | One or more non-system datafiles are missing                                       |
| 101        | HIGH     | OPEN   | 23-APR-07     | Datafile 1: '/disk1/oradata/prod/system01.dbf' contains one or more corrupt blocks |

```
analyzing automatic repair options; this may take some time
using channel ORA_DISK_1
analyzing automatic repair options complete
```

```

Mandatory Manual Actions
=====
no manual actions available

Optional Manual Actions
=====
1. If file /disk1/oradata/prod/users01.dbf was unintentionally renamed or moved, restore it

Automated Repair Options
=====
Option Repair Description
-----
1      Restore and recover datafile 28; Perform block media recovery of
      block 56416 in file 1
      Strategy: The repair includes complete media recovery with no data loss
      Repair script: /disk1/oracle/log/diag/rdbms/prod/prod/hm/reco_660500184.hm

```

ADVISE FAILURE の出力には、手動および自動の両方の修復オプションが示されています。まず、手動での問題の解決を試行します。問題を手動で解決できない場合は、自動修復に関するセクションを参照してください。

自動**修復オプション**には、1 つ以上の障害に関するサーバー管理の修復が記載されています。修復は、可能な場合は統合されているため、1 回の修復で複数の障害を解決できます。修復オプションには、実行する修復およびその修復の実行によってデータが失われるかどうかを示されます。

例 2-1 の出力には、Recovery Manager コマンドが含まれている修復スクリプトのファイル名が示されています。データ・リカバリ・アドバイザーを使用して障害を自動的に修復しない場合は、このスクリプトをリカバリ計画の基礎として使用します。

**参照：** 14-7 ページの「[障害の表示](#)」および 14-10 ページの「[修復オプションの決定](#)」を参照してください。

## 障害の修復

**Recovery Manager セッション**で LIST FAILURE および ADVISE FAILURE を実行した後、REPAIR FAILURE を実行して修復オプションを実行できます。他のコマンド・オプションを指定せずに REPAIR FAILURE を実行すると、Recovery Manager は、現在のセッションで最新の ADVISE FAILURE コマンドの最初の修復オプションを使用します。最新の ADVISE FAILURE コマンドで取得した修復オプション番号を指定することもできます。例 2-2 に、例 2-1 で検出された障害の修復方法を示します。

### 例 2-2 REPAIR FAILURE

```
RMAN> REPAIR FAILURE;
```

デフォルトでは、REPAIR FAILURE の実行を開始する前に確認を求められます。修復を実行した後、データ・リカバリ・アドバイザーは、障害が修復されている可能性に関してすべての既存の障害を再評価します。データ・リカバリ・アドバイザーは、障害が存在しているかどうかを常に確認し、修復されている障害を自動的にクローズします。エラーのため修復を完了できなかった場合は、そのエラーが原因で新しい評価が実行され、既存の障害および修復が再評価されます。

**参照：** 「[障害の修復](#)」 (14-13 ページ)

## フラッシュバック・データベースを使用したデータベースの巻戻し

**Oracle Flashback Database** を使用すると、データベース全体を過去の時点に巻き戻すことができます。メディア・リカバリとは異なり、データファイルをリストアしてデータベースを過去の状態に戻す必要はありません。

Recovery Manager の FLASHBACK DATABASE コマンドを使用するには、**フラッシュバック・ログ**が生成されるように、データベースを事前に構成しておく必要があります。この構成タスクの詳細は、5-28 ページの「**Oracle Flashback Database およびリストア・ポイントの構成**」を参照してください。フラッシュバック・データベースによって、コマンドの実行時に存在しているデータファイルに対する変更が巻き戻されます。このコマンドを使用してメディア障害またはデータファイルの欠落を修復することはできません。

FLASHBACK DATABASE を発行する場合は、データベースがマウントされている必要があります。以前に作成した**リストア・ポイント**が**フラッシュバック・データベース・ウィンドウ**内にある場合は、そのリストア・ポイントにフラッシュバックできます。

### フラッシュバック・データベースを使用してデータベースを巻き戻す手順

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. データベースがマウント状態であることを確認します。

次のコマンドを実行して、データベースを停止してからマウントします。

```
SHUTDOWN IMMEDIATE;  
STARTUP MOUNT;
```

3. FLASHBACK DATABASE コマンドを実行します。

次の例は、このコマンドの様々な形式を示しています。

```
FLASHBACK DATABASE TO SCN 861150;  
  
FLASHBACK DATABASE  
  TO RESTORE POINT BEFORE_CHANGES;  
  
FLASHBACK DATABASE TO TIME  
  "TO_DATE('06/20/07','MM/DD/YY')";
```

4. フラッシュバック・データベースを実行した後、SQL\*Plus を使用してデータベースを読取り専用でオープンし、いくつかの問合せを実行して、データベースの内容を検証します。

次のように入力して、データベースを読取り専用でオープンします。

```
SQL "ALTER DATABASE OPEN READ ONLY";
```

5. 結果が適切である場合は、次の一連のコマンドを実行してデータベースを停止してからオープンできます。

```
SHUTDOWN IMMEDIATE;  
STARTUP MOUNT;  
ALTER DATABASE OPEN RESETLOGS;
```

**参照:** 「**フラッシュバック・データベースを使用したデータベースの巻戻し**」 (16-11 ページ)

## データベース・ファイルのリストアおよびリカバリ

物理データベース・ファイルを Recovery Manager でリストアおよびリカバリするには、RESTORE および RECOVER コマンドを使用します。データファイルのリストアでは、リカバリ操作に必要なバックアップからデータファイルを取得します。メディア・リカバリは、リストアしたデータファイルに REDO ログおよび増分バックアップに含まれている変更を適用して、データファイルを目的の SCN または目標時点の状態にします。

**参照:** 第 17 章「データベースの完全リカバリの実行」

## データベース・ファイルのリストアおよびリカバリの準備

メディア障害によってデータベース・ファイルが破損したため、データベースをリカバリする必要がある場合は、必要なバックアップがあることを最初に確認する必要があります。RESTORE ... PREVIEW コマンドを使用すると、指定した時点にリストアするために Recovery Manager で使用可能なバックアップをレポートすることはできますが、リストアすることはできません。Recovery Manager は、メタデータは問い合わせますが、実際にバックアップ・ファイルは読み取りません。このコマンドは、データベースをオープンしているときにも実行できます。

### データベースのリストアおよびリカバリを確認する手順

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. 必要に応じて、次のコマンドで示すように、現行の表領域およびデータファイルを表示します。

```
RMAN> REPORT SCHEMA;
```

3. PREVIEW オプションを指定して RESTORE DATABASE コマンドを実行します。

次のコマンドでは、バックアップのメタデータが冗長モードで表示されないように SUMMARY を指定します（出力例も示します）。

```
RMAN> RESTORE DATABASE PREVIEW SUMMARY;
```

```
Starting restore at 21-MAY-07
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=80 device type=DISK
```

```
List of Backups
```

```
=====
Key          TY LV S Device Type Completion Time #Pieces #Copies Compressed Tag
-----
11          B F A DISK          18-MAY-07          1          2          NO
TAG20070518T181114
13          B F A DISK          18-MAY-07          1          2          NO
TAG20070518T181114
using channel ORA_DISK_1
```

```
List of Archived Log Copies for database with db_unique_name PROD
```

```
=====
Key          Thrd Seq          S Low Time
-----
47           1    18          A 18-MAY-07
Name: /disk1/oracle/dbs/db1r_60ffa882_1_18_0622902157.arc
```

```
Media recovery start SCN is 586534
Recovery must be done beyond SCN 587194 to clear datafile fuzziness
validation succeeded for backup piece
Finished restore at 21-MAY-07
```

## データベース全体のリカバリ

データベース全体をリカバリするには、RESTORE DATABASE および RECOVER DATABASE コマンドを使用します。必要なすべてのファイルのバックアップを事前に作成しておく必要があります。このシナリオでは、すべてのデータファイルを元の場所にリストアできることを想定しています。元の場所にアクセスできない場合は、17-11 ページの「[デフォルト以外の場所へのデータファイルのリストア](#)」の説明に従って SET NEWNAME コマンドを使用します。

### データベース全体をリカバリする手順

1. 2-15 ページの「[データベース・ファイルのリストアおよびリカバリの準備](#)」の説明に従ってリカバリを準備します。

2. データベースをマウント状態にします。

次の例では、(起動されている場合は) データベース・インスタンスを終了し、データベースをマウントします。

```
RMAN> STARTUP FORCE MOUNT;
```

3. データベースをリストアします。

次の例では、事前構成されたディスク・チャンネルを使用して、データベースをリストアします。

```
RMAN> RESTORE DATABASE;
```

4. 次の例に示すように、データベースをリカバリします。

```
RMAN> RECOVER DATABASE;
```

5. 次の例に示すように、データベースをオープンします。

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

## 表領域のリカバリ

データベースがオープンされている場合は、個々の表領域に対して RESTORE TABLESPACE および RECOVER TABLESPACE コマンドを使用します。この場合、リカバリが必要な表領域をオフラインにしてリストアし、表領域をリカバリしてからオンラインに戻す必要があります。

データファイルを新しい場所にリストアできない場合は、Recovery Manager の SET NEWNAME コマンドを RUN コマンド内で使用して、新しいファイル名を指定します。その後、SQL 文の ALTER DATABASE RENAME FILE と同等の SWITCH DATAFILE ALL コマンドを使用して制御ファイルを更新し、RUN コマンドで SET NEWNAME が発行されたすべてのデータファイルの新しい名前を反映します。

ユーザー管理のメディア・リカバリとは異なり、オンラインの表領域を [バックアップ・モード](#) する必要はありません。ユーザー管理ツールとは異なり、Recovery Manager では、データ・ブロックの形式が認識されるため、追加のロギングまたはバックアップが必要ありません。

### データベースがオープンされている場合に個々の表領域をリカバリする手順

1. 2-15 ページの「[データベース・ファイルのリストアおよびリカバリの準備](#)」の説明に従ってリカバリを準備します。

2. リカバリする表領域をオフラインにします。

users 表領域をオフラインにするには、次のように入力します。

```
RMAN> SQL 'ALTER TABLESPACE users OFFLINE';
```

3. 表領域をリストアおよびリカバリします。

Recovery Manager プロンプトで次の RUN コマンドを実行すると、users 表領域にデータファイルの新しい名前が設定されます。

```
RUN
{
  SET NEWNAME FOR DATAFILE '/disk1/oradata/prod/users01.dbf'
  TO '/disk2/users01.dbf';
  RESTORE TABLESPACE users;
  SWITCH DATAFILE ALL; # update control file with new filenames
  RECOVER TABLESPACE users;
}
```

4. 次の例に示すように、データベースをオンラインにします。

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

データファイル・レベルのリカバリには、RESTORE DATAFILE および RECOVER DATAFILE を使用することもできます。

**参照：**

- 「表領域の完全リカバリの実行」 (17-14 ページ)
- 「オンライン・バックアップおよびバックアップ・モード」 (7-3 ページ)

## 個々のデータ・ブロックのリカバリ

Recovery Manager は、破損した個々のデータファイル・ブロックをリカバリできます。Recovery Manager がバックアップ用のファイルの完全なスキャンを実行する場合、破損したすべてのブロックが V\$DATABASE\_BLOCK\_CORRUPTION に表示されます。通常、破損は、アラート・ログ、トレース・ファイルまたは SQL 問合せの結果に記録されます。

### データ・ブロックをリカバリする手順

1. 破損したブロックのブロック番号がわからない場合は、これらの番号を取得します。

トレース・ファイルおよびアラート・ログを検索する最も簡単な方法は、SQL\*Plus をターゲット・データベースに接続し、次の問合せを実行する方法です。

```
SQL> SELECT NAME, VALUE
2 FROM V$DIAG_INFO;
```

2. Recovery Manager を起動し、ターゲット・データベースに接続します。
3. RECOVER コマンドを実行してブロックを修復します。

次の Recovery Manager コマンドによって、破損したすべてのブロックがリカバリされます。

```
RMAN> RECOVER CORRUPTION LIST;
```

次の例に示すように、個々のブロックをリカバリすることもできます。

```
RMAN> RECOVER DATAFILE 1 BLOCK 233, 235 DATAFILE 2 BLOCK 100 TO 200;
```

**参照：** 第 18 章「ブロック・メディア・リカバリの実行」





# 第 II 部

---

## Recovery Manager の起動および構成

第 II 部では、Recovery Manager 環境の基本的な構成要素およびその構成方法について説明します。第 II 部に含まれる章は次のとおりです。

- 第 3 章「Recovery Manager のアーキテクチャ」
- 第 4 章「Recovery Manager クライアントの起動および操作」
- 第 5 章「Recovery Manager 環境の構成」
- 第 6 章「Recovery Manager 環境の構成 : 高度なトピック」



---

## Recovery Manager のアーキテクチャ

この章では、Recovery Manager のインタフェースおよび Recovery Manager 環境の基本的な構成要素について説明します。この章の内容は、次のとおりです。

- Recovery Manager 環境
- Recovery Manager コマンドライン・クライアント
- Recovery Manager チャンネル
- Recovery Manager リポジトリ
- メディア管理
- フラッシュ・リカバリ領域
- Data Guard 環境での Recovery Manager

## Recovery Manager 環境

Recovery Manager 環境は、バックアップおよびリカバリ計画で使用される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ます。

表 3-1 に、通常の Recovery Manager 環境の構成要素の一部を示します。

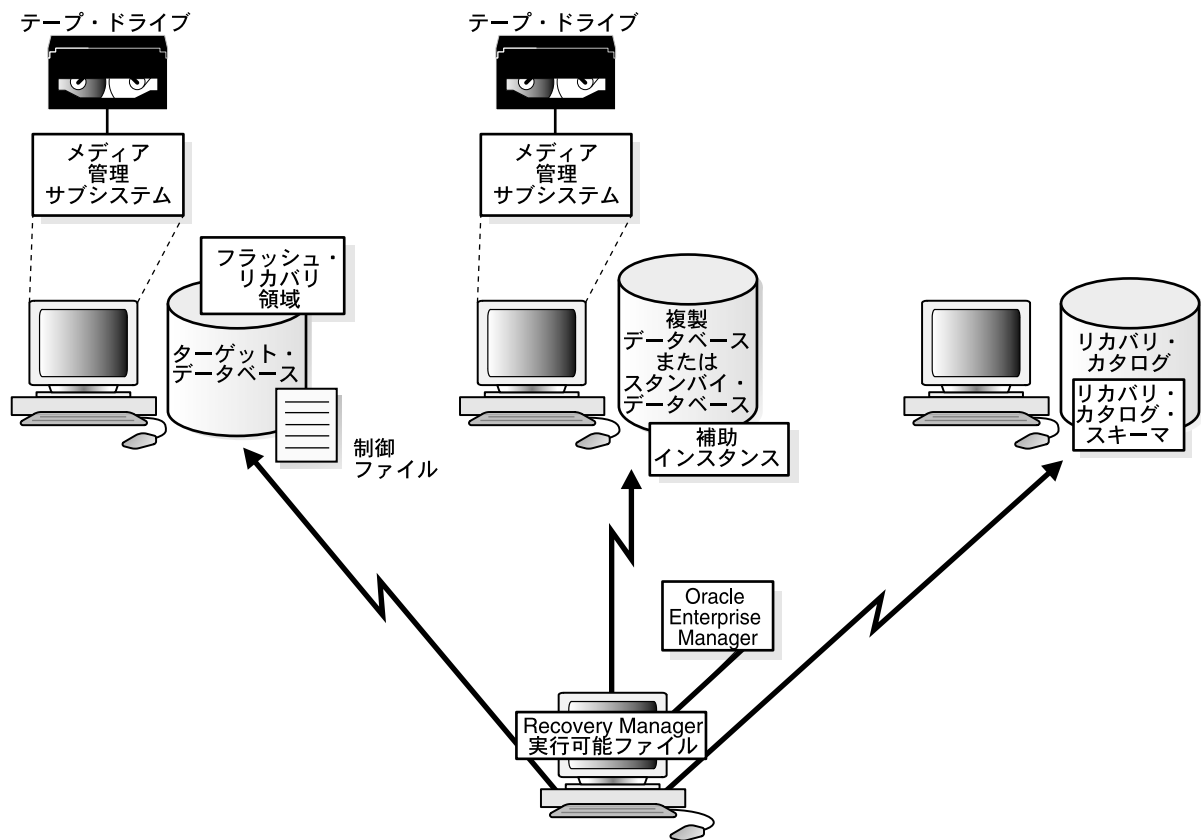
表 3-1 Recovery Manager 環境の構成要素

| 構成要素                      | 説明  |
|---------------------------|---|
| Recovery Manager クライアント   | ターゲット・データベースに対する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できるため、Oracle Net を介してターゲット・ホストに接続された任意のホスト上に配置できます。   |
| ターゲット・データベース              | Recovery Manager によってバックアップまたはリストアされる制御ファイル、データファイルおよびオプションのアーカイブ REDO ログが含まれているデータベース。Recovery Manager は、ターゲット・データベースの制御ファイルを使用して、ターゲット・データベースに関するメタデータを収集します。また、Recovery Manager の操作に関する情報は、制御ファイルに格納されます。バックアップおよびリカバリ操作は、ターゲット・データベース上で動作するサーバー・セッションによって実行されません。   |
| リカバリ・カタログ・データベース          | リカバリ・カタログが含まれているデータベース。バックアップおよびリカバリを実行するために Recovery Manager で使用されるメタデータが含まれています。複数のターゲット・データベースの Recovery Manager メタデータが含まれている 1 つのリカバリ・カタログを作成することができます。Recovery Manager を <b>フィジカル・スタンバイ・データベース</b> で使用しないかぎり、Recovery Manager 使用時のリカバリ・カタログはオプションです。Recovery Manager では、メタデータが各ターゲット・データベースの制御ファイルに格納されるためです。  |
| リカバリ・カタログ・スキーマ            | リカバリ・カタログ・データベース内のユーザー。Recovery Manager によってメンテナンスされるメタデータ表を所有します。Recovery Manager は、定期的に、ターゲット・データベースの制御ファイルからリカバリ・カタログにメタデータを伝播します。   |
| フィジカル・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・データベースによって作成されたアーカイブ REDO ログを使用して更新されます。フィジカル・スタンバイ・データベースには、プライマリ・データベースと同じ DBID 値および DB_NAME 値が含まれていますが、DB_UNIQUE_NAME 値は異なります。プライマリ・データベースにアクセスできなくなった場合、スタンバイ・データベースにフェイルオーバーできます。<br><br>スタンバイ・データベースは、Recovery Manager を使用して作成、バックアップまたはリカバリできます。スタンバイ・データベースで作成したバックアップは、同じ本番データベースのプライマリ・データベースまたは別のスタンバイ・データベースでも使用できます。Data Guard 環境で Recovery Manager を使用する場合は、リカバリ・カタログが必要です。<br><br><b>注意:</b> Recovery Manager では、ロジカル・スタンバイ・データベースは別のデータベースとして処理されます。プライマリ・データベースとは DBID が異なるためです。Data Guard 環境での Recovery Manager の使用方法については、『Oracle 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                  | テスト目的で使用するプライマリ・データベースのコピー。コピー元のデータベースとは DBID が異なります。   |
| フラッシュ・リカバリ領域              | 制御ファイルのコピー、オンライン REDO ログのコピー、アーカイブ REDO ログ、 <b>フラッシュバック・ログ</b> 、Recovery Manager バックアップなどのリカバリ関連ファイルの格納のために使用可能なディスクの場所。フラッシュ・リカバリ領域内のファイルは、Oracle および Recovery Manager によって自動的に管理されます。   |
| メディア・マネージャ                | Recovery Manager でストレージ・システム（テープなど）へのバックアップを実行するための、ベンダー固有のアプリケーション。  |
| メディア管理カタログ                | メディア管理アプリケーションについてのメタデータを格納するベンダー固有のリポジトリ。  |
| Oracle Enterprise Manager | データベースに対するブラウザベースのインタフェース。Recovery Manager によるバックアップおよびリカバリにも使用できます。  |

Recovery Manager 環境に必須の構成要素は、ターゲット・データベースおよび Recovery Manager クライアントのみですが、実際の構成はより複雑です。たとえば、複数のメディア・マネージャに接続している Recovery Manager クライアント、複数のターゲット・データベースおよび複数の補助データベースが存在し、それらがすべて Enterprise Manager を介してアクセスされる環境を構成する場合があります。

図 3-1 に、Recovery Manager 環境での構成要素の例を示します。この例では、プライマリ・データベース、スタンバイ・データベースおよびリカバリ・カタログ・データベースがすべて別々のコンピュータ上に配置されています。プライマリ・データベースおよびスタンバイ・データベースのホストでは、ローカル接続されたテープ・ドライブが使用されます。Recovery Manager クライアントおよび Enterprise Manager コンソールは、別のコンピュータ上で実行されます。

図 3-1 Recovery Manager 環境の例



参照： Oracle Net については、『Oracle Database Net Services 管理者ガイド』を参照してください。

## Recovery Manager コマンドライン・クライアント

Recovery Manager コマンドライン・クライアントを使用すると、バックアップおよびリカバリ操作を詳細に管理するためのコマンドを入力できます。Recovery Manager は、コマンドを対話モードまたはバッチ・モードで実行できるコマンド言語インタプリタを使用します。Recovery Manager の上位に構築された Enterprise Manager でバックアップおよびリカバリ機能を使用している場合でも、Recovery Manager クライアントがバックグラウンドで実行されています。

## Recovery Manager チャンネル

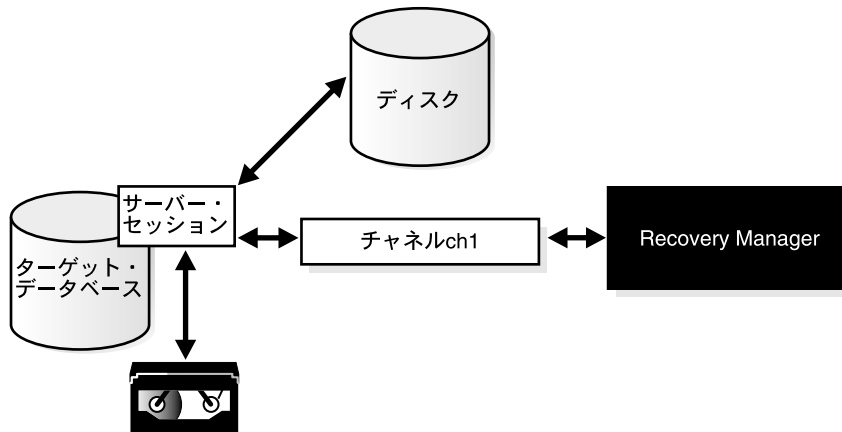
Recovery Manager クライアントは、データベース・サーバー・セッションに、すべてのバックアップおよびリカバリ作業を実行するように指示します。セッションの構成は、オペレーティング・システムによって異なります。たとえば、Linux では、サーバー・セッションはサーバー・プロセスに対応しますが、Windows では、データベース・サービス内のスレッドに対応します。

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操作を実行しません。Recovery Manager クライアントをターゲット・データベースに接続すると、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、操作を実行するようにサーバー・セッションに指示します。

**チャンネル**は、デバイス・タイプに対する 1 つのデータ・ストリームであり、1 つのデータベース・サーバー・セッションに対応します。チャンネルは、データを PGA メモリーに読み取り、処理して出力デバイスに書き込みます。チャンネルの基本的な動作については、21-2 ページの「[Recovery Manager のパフォーマンスのチューニングの基本的な概念](#)」を参照してください。

大部分の Recovery Manager コマンドはチャンネルによって実行されます。チャンネルは、Recovery Manager セッション間にわたって保持されるように構成するか、または手動で各 **Recovery Manager セッション** に割り当てる必要があります。図 3-2 に示すように、チャンネルは、ターゲット・データベースまたは補助データベースのインスタンス上でサーバー・セッションを開始することによって、Recovery Manager クライアントからそれらのインスタンスへの接続を確立します。

図 3-2 チャンネルの割当て



## チャンネルおよびデバイス

Recovery Manager でサポートされているデバイス・タイプは、ディスクおよび **SBT** (テープへのシステム・バックアップ) です。SBT デバイスは、サード・パーティの **メディア・マネージャ** によって制御されます。通常、SBT デバイスはテープ・ライブラリおよびテープ・ドライブです。

バックアップにディスク・チャンネルを使用すると、チャンネルによって、ディスク上の、バックアップを作成しているターゲット・データベース・インスタンスのファイル名領域にバックアップが作成されます。データファイルを格納できる任意のデバイスに、バックアップを実行できます。Recovery Manager は、ディスク・バックアップの作成時にメディア・マネージャをコールしません。

ディスク以外のメディアにバックアップを作成するには、Oracle Secure Backup などのメディア管理ソフトウェアを使用して、このソフトウェアでサポートされるチャンネルを割り当てる必要があります。Recovery Manager は、割り当てられたチャンネル・タイプがディスクでない場合は常にメディア・マネージャと通信します。SBT チャンネルを使用した場合にメディア・マネージャがリソースを割り当てる方法およびタイミングは、ベンダーによって異なります。コマンドが発行された時点でリソースを割り当てるメディア・マネージャや、読取り用または書込み用にファイルをオープンしてからリソースを割り当てるメディア・マネージャがあります。

**参照:** 「バックアップ用のデフォルト・デバイスの構成: ディスクまたは SBT」 (5-3 ページ)

## 自動チャンネルおよび手動チャンネル

CONFIGURE CHANNEL コマンドを使用すると、Recovery Manager セッション間にわたってディスクまたはテープとともに使用できるようにチャンネルを構成できます。この方法は、**自動チャンネル割当て**と呼ばれます。Recovery Manager では、ディスクへのバックアップに使用できる DISK チャンネルが 1 つ、事前に構成されています。

自動チャンネルを使用できるコマンドを実行すると、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ンネルを割り当てます。BACKUP コマンドを実行すると、Recovery Manager は、指定したメディアへのバックアップに必要なチャンネル・タイプのみを割り当てます。RESTORE コマンドおよび **Recovery Manager のメンテナンス・コマンド**を実行すると、Recovery Manager は、コマンドの実行に必要なすべてのデバイス・タイプ用のチャンネルを割り当てます。自動チャンネルの名前は Recovery Manager によって決定されます。

また、チャンネルの手動割当てを選択することもできます。手動で割り当てられた各チャンネルは、データベースに個別に接続します。手動でチャンネルを割り当てる場合は、dev1、ch2 などのユーザー定義の名前を指定します。

コマンドの実行時にデバイスで使用できるチャンネルの数によって、コマンドの実行中に Recovery Manager がこのデバイスに対する読取りまたは書込みをパラレルに行うかどうかが決まります。パラレル化では、ファイルのバックアップが複数のチャンネルで実行されます。各チャンネルで複数のファイルをバックアップできますが、**マルチセクション・バックアップ**を実行しないかぎり、ファイルは複数のチャンネルでバックアップされません。

**参照:**

- ALLOCATE CHANNEL 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- ALLOCATE CHANNEL FOR MAINTENANCE については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- 5-5 ページの「ディスクのチャンネルの構成」および 5-13 ページの「メディア・マネージャで使用する SBT チャンネルの構成」を参照してください。

## Recovery Manager リポジトリ

**Recovery Manager リポジトリ**は、Recovery Manager がバックアップ、リカバリおよびメンテナンスに使用する、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です。Recovery Manager は常にそのメタデータを制御ファイルに格納します。制御ファイル内のこのメタデータは、データベースの Recovery Manager バックアップに関する正式なレコードです。このため、制御ファイルの保護は、バックアップ計画において重要です。Recovery Manager は、Recovery Manager リポジトリ情報を格納した制御ファイルのみを使用して、必要なすべてのバックアップおよびリカバリ操作を実行することができ、構成された保存方針を満たすために必要なすべてのレコードをメンテナンスします。

また、**リカバリ・カタログ**も作成できます。リカバリ・カタログは、Oracle Database スキーマに格納される Recovery Manager メタデータのリポジトリです。制御ファイルにバックアップ・アクティビティを記録する場合は領域に制限がありますが、リカバリ・カタログにはより長期の履歴を格納できます。すべてのデータベースの Recovery Manager メタデータが含まれているリカバリ・カタログを1つ作成することによって、バックアップおよびリカバリを簡単に管理できます。

リカバリ・カタログの所有者は、他のデータベース・ユーザーに対してカタログへの制限つきアクセス権の付与および取消しを行うことができます。制限付きユーザーは、それぞれ**仮想プライベート・カタログ**と呼ばれる独自のメタデータへの完全な読取り / 書込み権限を持っています。データベースに1つ以上の仮想プライベート・カタログが存在する場合、そのデータベースには1セットのカタログ表が含まれています。これらの表は、ベース・リカバリ・カタログの所有者によって所有されています。ベース・リカバリ・カタログの所有者は、各仮想プライベート・カタログ・ユーザーがアクセスできるデータベースを制御します。

一部の Recovery Manager 機能は、リカバリ・カタログを使用している場合にのみ機能します。たとえば、リカバリ・カタログに**ストアド・スクリプト**を作成し、このスクリプトで Recovery Manager ジョブを実行できます。それ以外の Recovery Manager コマンドが、特にリカバリ・カタログの管理に関連している場合もあります。これらのコマンドは、Recovery Manager がリカバリ・カタログに接続されていない場合は使用できません（また、使用する必要はありません）。

リカバリ・カタログは、Recovery Manager によってのみメンテナンスされます。ターゲット・データベース・インスタンスがカタログに直接アクセスすることはありません。Recovery Manager は、リポジトリを更新するなんらかの操作が終了した後、および特定の操作が行われる前に、データベース構造についての情報、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・カタログに伝播します。

**参照：** 第11章「Recovery Manager バックアップおよびリポジトリ・レコードのメンテナンス」および第12章「リカバリ・カタログの管理」

## メディア管理

オラクル社提供のメディア管理レイヤー（MML）APIを使用すると、サード・パーティ・ベンダーは**メディア・マネージャ**を構築できます。メディア・マネージャとは、Recovery Manager とともにベンダーのハードウェアで動作し、シーケンシャル・メディア・デバイス（テープ・ドライブなど）へのバックアップを可能にするソフトウェアです。メディア・マネージャは、テープなどのシーケンシャル・メディアのロード、アンロードおよびラベル付けを処理します。シーケンシャル・メディア・デバイスで Recovery Manager を使用するには、メディア・マネージャ・ソフトウェアをインストールする必要があります。

バックアップまたはリストア時、Recovery Manager クライアントは、ターゲット・データベース・インスタンスに接続して、メディア・マネージャに要求を送信するようにインスタンスに指示します。Recovery Manager クライアントとメディア・マネージャが直接通信することはありません。

## Recovery Manager とメディア・マネージャの相互作用

メディア・マネージャにバックアップまたはリストアを実行する前に、メディア・マネージャとの通信を処理する1つ以上のチャンネルを割り当てておく必要があります。メディア・マネージャで使用するデフォルトのチャンネルも構成できます。このチャンネルは、チャンネルを明示的に割り当てていないメディア・マネージャを使用するすべてのバックアップおよびリカバリ作業で使用されます。



Recovery Manager は、テープのロード、ラベル付けまたはアンロードを行う固有のコマンドを発行しません。バックアップ時、Recovery Manager は、メディア・マネージャにバイト・ストリームを渡し、そのストリームに一意の名前を関連付けます。バックアップをリストアする必要がある場合、Recovery Manager は、メディア・マネージャにバイト・ストリームの取得を要求します。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・マネージャによって管理されます。たとえば、メディア・マネージャは、テープおよび各テープ上のファイルの名前をラベル付けして管理し、テープを自動的にロードおよびアンロードしたり、テープをロードおよびアンロードするようにオペレータに通知します。

メディア・マネージャの中には、**プロキシ・コピー**機能をサポートしているものがあります。この機能を使用すると、メディア・マネージャが、データファイルとバックアップ・デバイス間のデータの移動全体を処理します。このような製品では、ストレージ・サブシステムとメディア・サブシステム間で高速接続などの技術を使用して、プライマリ・データベース・サーバーの負荷を軽減している場合があります。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、メディア・マネージャは、データを移動する方法およびタイミングに関するすべての決定を行います。

**参照:** 「メディア・マネージャで使用する SBT チャンネルの構成」  
(5-13 ページ)

## Oracle Secure Backup

**Oracle Secure Backup** は、ファイル・システムをテープにバックアップすることによって信頼性の高い安全なデータ保護を提供するメディア・マネージャです。SAN、ギガビット・イーサネットおよび SCSI 環境のすべての主要なテープ・ドライブおよびライブラリがサポートされています。

Oracle Secure Backup は、データベースのバックアップおよびリカバリのアルゴリズムに関する専用の機能は備えていませんが、SBT インタフェースを介して Recovery Manager のメディア管理レイヤーとして機能できます。この機能で、Oracle Secure Backup は、サポートされている他のサード・パーティの SBT ライブラリと同じサービスを Recovery Manager に提供します。ただし、Oracle Secure Backup は、他のメディア・マネージャにはない機能も備えています。

**参照:** Oracle Secure Backup の使用方法については、『Oracle Secure Backup 管理者ガイド』を参照してください。

## Backup Solutions Program

Oracle Partner Program の一部である Oracle Backup Solutions Program (BSP) には、オラクル社の MML 仕様に準拠した製品を生産している、メディア・マネージャのベンダーが参加しています。ベンダーのメディア管理製品を、ご使用のプラットフォームで使用できる場合があります。詳細は、Oracle サポート・サービスに連絡して使用可能な製品のリストを入手するか、または各ベンダーに連絡して BSP への参加を確認してください。あるいは、次に示す Backup Solutions Program の Web サイトにアクセスしてください。

<http://www.oracle.com/technology/deploy/availability>

オラクル社は、Recovery Manager との互換性に関してメディア・マネージャ・ベンダーを認証しているわけではありません。可用性、バージョンの互換性および機能については、オラクル社ではなく、メディア・マネージャ・ベンダーに問い合せてください。

## フラッシュ・リカバリ領域

バックアップおよびリカバリ関連の様々なファイルを作成するコンポーネントは、互いを認識しません。また、それぞれのデータが格納されるファイル・システムのサイズも認識しません。ディスクベースの自動バックアップおよびリカバリでは、**フラッシュ・リカバリ領域** (単にリカバリ領域とも呼ばれる) を作成できます。バックアップ関連のファイルは、フラッシュ・リカバリ領域によって自動的に管理されます。

フラッシュ・リカバリ領域を使用すると、バックアップ関連のファイル用のディスク領域の管理および異なるタイプのファイル間での領域使用の均衡化を手動で行う必要性が最小限に抑えられます。このような点から、フラッシュ・リカバリ領域を使用すると、データベースの継続的な管理が簡単になります。リカバリ領域を有効にしてバックアップ管理を簡単にすることをお勧めします。

リカバリ領域の作成時には、ディスク上の場所および記憶領域の上限を選択します。また、バックアップ・ファイルがリカバリに必要となる期間を制御する**バックアップの保存方針**も設定します。データベースは、この領域内で、データベースのバックアップ、アーカイブ REDO ログおよびその他のリカバリ関連ファイルに使用されるストレージを管理します。不要になったファイルは、Recovery Manager が新しいファイル用の領域を要求したときに削除されます。

**参照：** フラッシュ・リカバリ領域およびその構成方法については、5-14 ページの「[フラッシュ・リカバリ領域の構成](#)」を参照してください。

## Data Guard 環境での Recovery Manager

Data Guard 環境で Recovery Manager を使用する場合は、リカバリ・カタログが必要です。リカバリ・カタログには、すべてのプライマリ・データベースおよびスタンバイ・データベースのメタデータを格納できます。

Data Guard 環境のデータベースは、初期化パラメータ・ファイルの DB\_UNIQUE\_NAME パラメータで一意に識別されます。Data Guard 環境で Recovery Manager を正常に動作させるには、同じ DBID を持つすべてのデータベース間で DB\_UNIQUE\_NAME が一意である必要があります。

**参照：** Data Guard 環境での Recovery Manager の使用方法については、『Oracle Data Guard 概要および管理』を参照してください。

## Data Guard 環境での Recovery Manager の構成

バックアップおよびリカバリでの Recovery Manager の継続的な使用を簡単にするには、プライマリ・データベースおよび**フィジカル・スタンバイ・データベース**のそれぞれに多くの永続構成を設定します。これらの設定によって、Recovery Manager の動作の様々な点が制御されます。たとえば、**バックアップの保存方針**、テープまたはディスクへのバックアップのデフォルトの格納先、デフォルトのバックアップ・デバイス・タイプなどを構成できます。

FOR DB\_UNIQUE\_NAME 句を指定して CONFIGURE コマンドを使用すると、TARGET としてスタンバイ・データベースまたはプライマリ・データベースに接続せずに、Data Guard 環境でデータベースの永続構成を作成できます。たとえば、Recovery Manager をリカバリ・カタログに接続し、SET DBID コマンドを実行してから、データベース作成時に Recovery Manager の構成が適用されるように、フィジカル・スタンバイ・データベースを作成する前にその構成を作成できます。

Recovery Manager は、TARGET としてデータベースに接続されている場合、リカバリ・カタログの**再同期化**中にデータベースの制御ファイルを更新します。ただし、データベースに TARGET として接続されていない場合、このデータベースに対して FOR DB\_UNIQUE\_NAME を使用すると、Recovery Manager はリカバリ・カタログ内の構成のみを変更します。

**参照：** 「[Data Guard 環境での Recovery Manager の構成](#)」 (5-36 ページ)

## Data Guard 環境での Recovery Manager によるファイル管理

Recovery Manager は、リカバリ・カタログを使用して、Data Guard 環境内のすべてのデータベース・ファイルのファイル名を追跡します。また、カタログには、オンライン REDO ログ・ファイル、スタンバイ REDO ログ・ファイル、一時ファイル、アーカイブ REDO ログ・ファイル、バックアップ・セットおよびイメージ・コピーが作成される場所も記録されます。

## Data Guard 環境でのバックアップの互換性

Recovery Manager コマンドは、リカバリ・カタログのメタデータを使用して、Data Guard 環境内の異なる物理データベース間で透過的に動作します。たとえば、表領域をフィジカル・スタンバイ・データベースにバックアップして、プライマリ・データベースにリストアおよびリカバリすることができます。また、表領域をプライマリ・データベースにバックアップして、フィジカル・スタンバイ・データベースにリストアおよびリカバリすることもできます。

---

**注意：** ロジカル・スタンバイ・データベースのバックアップは、プライマリ・データベースでは使用できません。

---

スタンバイ制御ファイルのバックアップと非スタンバイ制御ファイルのバックアップには互換性があります。たとえば、スタンバイ制御ファイルをプライマリ・データベースにリストアし、プライマリ制御ファイルをフィジカル・スタンバイ・データベースにリストアできます。この互換性は、制御ファイルのバックアップを Data Guard 環境の 1 つのデータベースにオフロードできることを意味します。データベース・ファイルのファイル名は、データベースのリストアおよびリカバリ中に Recovery Manager によって自動的に更新されます。

## Data Guard 環境でのバックアップの関連付け

リカバリ・カタログは、すべてのデータベース・ファイルまたはバックアップ・ファイルを DB\_UNIQUE\_NAME に関連付けることによって、Data Guard 環境内のファイルを追跡します。ファイルを作成するデータベースは、そのファイルに関連付けられます。たとえば、Recovery Manager で standby1 という一意の名前を持つデータベースをバックアップすると、standby1 がこのバックアップに関連付けられます。CHANGE ... RESET DB\_UNIQUE\_NAME を使用してバックアップを別のデータベースに関連付けないかぎり、バックアップは作成したデータベースに対応付けられたままです。

## Data Guard 環境でのバックアップのアクセシビリティについて

バックアップのアクセシビリティは、その関連付けとは異なります。Data Guard 環境では、関連付けられたデータベースのみがディスク・バックアップにアクセスできるとリカバリ・カタログによってみなされますが、1 つのデータベースに作成されたテープ・バックアップにはすべてのデータベースがアクセスできます。バックアップ・ファイルがいずれのデータベースにも関連付けられていない場合、リカバリ・カタログ・ビュー内のバックアップ・ファイルに関する行の SITE\_KEY 列に null が表示されます。デフォルトでは、Recovery Manager は、TARGET として接続されているデータベースに SITE\_KEY が null のファイルに関連付けます。

BACKUP、RESTORE、CROSSCHECK などの Recovery Manager コマンドは、すべてのアクセス可能なバックアップで動作します。たとえば、RECOVER COPY 操作の場合、Recovery Manager では、データベースに関連付けられているイメージ・コピーのみがリカバリの対象とみなされます。Recovery Manager では、ディスクおよびテープ上の増分バックアップが、イメージ・コピーのリカバリ対象とみなされます。データベース・リカバリでは、Recovery Manager によって、データベースおよびテープ上のすべてのファイルに対応付けられたディスク・バックアップのみがリストアの対象とみなされます。

バックアップのアクセシビリティの相違点を説明するために、データベース prod と standby1 が異なるホストに存在していると想定します。Recovery Manager は、prod 上のデータファイル 1 を本番ホスト上の /prhhost/disk1/df1.dbf およびテープにバックアップします。Recovery Manager は、standby1 上のデータファイル 1 をスタンバイ・ホスト上の /sbyhost/disk2/df1.dbf およびテープにバックアップします。Recovery Manager が prod データベースに接続している場合、Recovery Manager コマンドを使用しても、スタンバイ・ホストにある /sbyhost/disk2/df1.dbf バックアップでは操作を実行できません。ただし、Recovery Manager では、standby1 で作成されたテープ・バックアップはリストアの対象とみなされます。

---

---

**注意：** スタンバイ・ホストからプライマリ・ホスト（またはその逆）にバックアップを FTP し、このホスト上のデータベースに TARGET として接続した後で、バックアップを CATALOG に追加できます。ターゲット・データベースによってファイルがカタログに追加されると、そのファイルはターゲット・データベースに関連付けられます。

---

---

**参照：**

- Data Guard 環境での Recovery Manager バックアップおよびリカバリの実行方法については、『Oracle Data Guard 概要および管理』を参照してください。
- 「[Data Guard 環境でのメンテナンス・コマンド](#)」(11-3 ページ)
- Data Guard 環境でのリカバリ・カタログの管理方法については、[第 12 章「リカバリ・カタログの管理」](#)を参照してください。
- CONFIGURE 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

---

## Recovery Manager クライアントの 起動および操作

この章では、Recovery Manager コマンドライン・インタフェースを起動する方法およびデータベースに接続する方法について説明します。この章の内容は、次のとおりです。

- Recovery Manager の起動および終了
- Recovery Manager 出力の場所の指定
- Recovery Manager のグローバリゼーション・サポート環境変数の設定
- Recovery Manager コマンドの入力
- Recovery Manager によるデータベース接続の確立
- Recovery Manager のパイプ・インタフェースの使用

## Recovery Manager の起動および終了

この Recovery Manager 実行可能ファイルは、データベースとともに自動的にインストールされ、通常、他のデータベース実行可能ファイルと同じディレクトリに配置されます。たとえば、Linux の場合、Recovery Manager クライアントは \$ORACLE\_HOME/bin に配置されています。Recovery Manager を起動する場合、次の基本的なオプションがあります。

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

```
% rman
```

- **ターゲット・データベース** および (場合によっては) **リカバリ・カタログ** に接続している間に、オペレーティング・システムのコマンドラインで Recovery Manager 実行可能ファイルを起動します。次に例を示します。

```
% rman TARGET / # operating system authentication
% rman TARGET SYS@prod NOCATALOG # RMAN prompts for SYS password
% rman TARGET / CATALOG rco@catdb # RMAN prompts for rco password
```

---

**注意：** ほとんどの Recovery Manager コマンドでは、有効な作業を実行するために Recovery Manager が 1 つ以上のターゲット・データベースに接続する必要があります。Recovery Manager を様々なタイプのデータベースに接続する方法の詳細は、4-8 ページの「[Recovery Manager によるデータベース接続の確立](#)」を参照してください。

---

Recovery Manager やプログラムを終了するには、Recovery Manager プロンプトで EXIT または QUIT と入力します。

```
RMAN> EXIT
```

**参照：** Recovery Manager コマンドラインの構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## Recovery Manager 出力の場所の指定

デフォルトでは、Recovery Manager はコマンド出力を標準出力に書き込みます。出力をログ・ファイルにリダイレクトするには、次の例に示すように、Recovery Manager の起動時にコマンドラインで LOG パラメータを入力します。

```
% rman LOG /tmp/rman.log
```

この場合、Recovery Manager によってコマンド入力は表示されますが、Recovery Manager 出力は表示されません。Recovery Manager 出力をログ・ファイルと標準出力の両方に送信する場合は、Linux の tee コマンドまたは同等のコマンドを使用する方法が最も簡単です。たとえば、次の方法で、入力と出力の両方を Recovery Manager コマンドライン・インタフェース内に表示できます。

```
% rman | tee rman.log
RMAN>
```

**参照：** Recovery Manager コマンドライン・オプションについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## Recovery Manager のグローバル化・サポート環境変数の設定

Recovery Manager を起動する前に、環境変数 `NLS_DATE_FORMAT` および `NLS_LANG` を設定しておくに役立つ場合があります。これらの変数によって、`RESTORE`、`RECOVER`、`REPORT` などの Recovery Manager コマンドの時間パラメータで使用される書式が決まります。

次に、一般的な言語および日付書式の設定例を示します。

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

Recovery Manager を使用してマウントされていないデータベースに接続し、その後、Recovery Manager が接続された状態でそのデータベースをマウントする場合は、`NLS_LANG` 環境変数にデータベースが使用するキャラクタ・セットを指定してください。

マウントされていないデータベースのキャラクタ・セットは、デフォルトの `US7ASCII` とみなされます。キャラクタ・セットがデフォルトと異なる場合、Recovery Manager は、データベースのマウント後にエラーを戻します。たとえば、キャラクタ・セットが `WE8DEC` の場合は、`NLS_LANG` 変数を次のように設定できます。

```
NLS_LANG=american_america.we8dec
```

`NLS_LANG` および `NLS_DATE_FORMAT` は、使用される `NLS_DATE_FORMAT` に対して設定する必要があります。

### 参照：

- `NLS_LANG` および `NLS_DATE_FORMAT` パラメータの詳細は、『Oracle Database リファレンス』を参照してください。
- 『Oracle Database グローバリゼーション・サポート・ガイド』

## Recovery Manager コマンドの入力

Recovery Manager コマンドは、Recovery Manager プロンプトから直接入力するか、またはテキスト・ファイルから読み取ることができます。

この項の内容は、次のとおりです。

- [Recovery Manager プロンプトでの Recovery Manager コマンドの入力](#)
- [Recovery Manager でのコマンド・ファイルの使用](#)
- [Recovery Manager コマンド・ファイルへのコメントの入力](#)
- [コマンド・ファイルでの置換変数の使用](#)
- [Recovery Manager 構文の確認](#)

## Recovery Manager プロンプトでの Recovery Manager コマンドの入力

Recovery Manager クライアントでコマンドを受け入れる準備が整うと、次の例に示すようなコマンド・プロンプトが表示されます。

```
RMAN>
```

Recovery Manager で実行するコマンドを入力します。たとえば、次のように入力します。

```
RMAN> CONNECT TARGET
RMAN> BACKUP DATABASE;
```

ほとんどの Recovery Manager コマンドでは、多くのパラメータを指定し、セミコロンで終了する必要があります。STARTUP、SHUTDOWN、CONNECT などの一部のコマンドは、セミコロンで終了しているかどうかに関係なく使用できます。

完全なコマンドではないテキスト行を入力すると、Recovery Manager は、行番号を表示して入力の継続を求めます。たとえば、次のように入力します。

```
RMAN> BACKUP DATABASE
2> INCLUDE CURRENT
3> CONTROLFILE
4> ;
```

## Recovery Manager でのコマンド・ファイルの使用

繰り返し作業の場合は、Recovery Manager コマンドを含むテキスト・ファイルを作成し、@ 引数の後にファイル名を続けて Recovery Manager クライアントを起動します。たとえば、現行のディレクトリに次のテキスト行を含むテキスト・ファイル `cmdfile1` を作成します。

```
BACKUP DATABASE PLUS ARCHIVELOG;
```

このコマンド・ファイルをコマンドラインから実行すると、コマンド・ファイルに含まれているコマンドが実行されます。次に例を示します。

```
% rman TARGET / @cmdfile1
```

コマンドの完了後、Recovery Manager は終了します。

Recovery Manager コマンド・プロンプトで @ を使用して、**Recovery Manager セッション**中にコマンド・ファイルの内容を実行することもできます。Recovery Manager は、ファイルを読み取り、ファイル内のコマンドを実行します。たとえば、次のように入力します。

```
RMAN> @cmdfile1
```

Recovery Manager は、コマンド・ファイル内のコマンドの実行後、次のメッセージを表示します。

```
RMAN> **end-of-file**
```

オペレーティング・システムのコマンドラインからコマンド・ファイルを実行した場合と異なり、Recovery Manager は終了しません。

**参照：** Recovery Manager コマンドラインの構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## Recovery Manager コマンド・ファイルへのコメントの入力

Recovery Manager でのコメント文字は、シャープ記号 (#) です。行のシャープ記号から最後までテキストはすべて無視されます。たとえば、次のコマンド・ファイルの内容は、データベースおよびアーカイブ REDO ログをバックアップし、コメントを追加します。

```
# Command file name: mybackup.rman
# The following command backs up the database
BACKUP DATABASE;
# The following command backs up the archived redo logs
BACKUP ARCHIVELOG ALL;
```

次の例では、単一の Recovery Manager コマンドを複数の行に分割できることを示します。

```
RMAN> BACKUP # this is a comment
2> SPFILE;
```



## コマンド・ファイルでの置換変数の使用

コマンド・ファイルの実行時に、コマンド・ファイル内の置換変数で使用する1つ以上の値を USING 句に指定できます。この方法で、コマンド・ファイルを動的にすることができます。

SQL\*Plus と同様に、&1 は最初の値を配置する場所を示し、&2 は2番目の値を配置する場所を示します。それ以降も同様です。置換変数の構文では、&integer の後にオプションのピリオドが続きます。たとえば、&1.3 のようになります。オプションのピリオドは変数の一部であり、値と置換されます。したがって、置換テキストの直後に別の整数を続けることができます。たとえば、値 mybackup を変数 &1.3 を含むコマンド・ファイルに渡すと、置換の結果は mybackup3 になります。

次の手順では、置換変数を含むコマンド・ファイルをコールする動的シェル・スクリプトを作成および使用する方法について説明します。

### 動的シェル・スクリプトを作成および使用する手順

1. 置換変数を使用する Recovery Manager コマンド・ファイルを作成します。

次に、四半期ごとに実行される quarterly\_backup.cmd という名前のコマンド・ファイルの内容の例を示します。このスクリプトでは、テープ・セットの名前、FORMAT 指定内の文字列および作成するリストア・ポイントの名前の置換変数が使用されています。

```
# quarterly_backup.cmd
CONNECT TARGET /
RUN
{
  ALLOCATE CHANNEL c1
  DEVICE TYPE sbt
  PARMS 'ENV=(OB_MEDIA_FAMILY=&1)';
  BACKUP DATABASE
  TAG &2
  FORMAT '/disk2/bck/&1%U.bck'
  KEEP FOREVER
  RESTORE POINT &3;
}
EXIT;
```

2. 前の手順で作成した Recovery Manager コマンド・ファイルを実行するために使用できるシェル・スクリプトを作成します。

次の例では、runbackup.sh というシェル・スクリプトを作成します。この例では、書式およびリストア・ポイント名のシェル変数を作成し、これらの変数の値をスクリプトへのコマンドライン引数として受け入れます。

```
#!/bin/tcsh
# name: runbackup.sh
# usage: use the tag name and number of copies as arguments
set media_family = $argv[1]
set format = $argv[2]
set restore_point = $argv[3]
man @'/disk1/scripts/whole_db.cmd' USING $media_family $format $restore_point
```

3. 前の手順で作成したシェル・スクリプトを実行します。コマンドラインで適切な引数を指定します。

次の例では、runbackup.sh シェル・スクリプトを実行し、メディア・ファミリー名として archival\_backup、書式文字列として bck0906、リストア・ポイント名として FY06Q3 を渡します。

```
% runbackup.sh archival_backup bck0906 FY06Q3
```

**参照：** @ 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## Recovery Manager 構文の確認

Recovery Manager コマンドを実行せずに、その構文の正しさをテストする必要がある場合があります。コマンドライン引数 CHECKSYNTAX を使用して、Recovery Manager クライアントを起動します。これによって、入力したコマンドの解析のみが行われ、有効な Recovery Manager 構文ではないコマンドに対して RMAN-00558 エラーが戻されます。

**参照：** CHECKSYNTAX コマンドライン・オプションについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

### コマンドラインでの Recovery Manager 構文の確認

コマンドを実際には実行せずに、Recovery Manager コマンドの構文を対話形式で確認できます。

#### Recovery Manager 構文をコマンドラインで確認する手順

1. CHECKSYNTAX パラメータを使用して、Recovery Manager を起動します。

たとえば、次のコマンドを入力します。

```
% rman CHECKSYNTAX
```

2. 確認する Recovery Manager コマンドを入力します。

次に、対話形式のセッションの例を示します。ユーザーが入力するテキストは太字になっています。

```
RMAN> run [ backup database; ]
```

```
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-00558: error encountered while parsing input commands
RMAN-01006: error signalled during parse
```

```
      RMAN-02001: unrecognized punctuation symbol "["
```

```
RMAN> run { backup database; }
```

```
The command has no syntax errors
```

```
RMAN>
```

### コマンド・ファイル内の Recovery Manager 構文の確認

コマンド・ファイル内のコマンドをテストするには、CHECKSYNTAX パラメータを使用して Recovery Manager を起動し、渡すコマンド・ファイルを @ コマンドを使用して指定します。

#### コマンド・ファイル内のコマンドをテストする手順

1. テキスト・エディタを使用してコマンド・ファイルを作成します。

次の内容の /tmp/goodcmdfile を作成するとします。

```
# command file with legal syntax
RESTORE DATABASE;
RECOVER DATABASE;
```

次の内容の別のコマンド・ファイル /tmp/badcmdfile を作成するとします。

```
# command file with bad syntax commands
RESTORE DATABASE
RECOVER DATABASE
```

2. Recovery Manager プロンプトから次の形式でコマンド・ファイルを実行します。ここで、*filename* はコマンド・ファイルの名前です。

```
% rman CHECKSYNTAX @filename
```

次の例は、CHECKSYNTAX を使用して /tmp/goodcmdfile を実行した場合の出力を示しています。

```

RMAN> # command file with legal syntax
2> restore database;
3> recover database;
4>
The cmdfile has no syntax errors

```

Recovery Manager complete.

これに対して、次の例は、CHECKSYNTAX を使用して /tmp/badcmdfile を実行した場合の出力を示しています。

```

RMAN> #command file with syntax error
2> restore database
3> recover
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS=====
RMAN-00571: =====
RMAN-00558: error encountered while parsing input commands
RMAN-01005: syntax error: found "recover": expecting one of: "archivelog,
             channel, check, controlfile, clone, database, datafile, device,
             from, force, high, (, preview, ;, skip, spfile, standby, tablespace,
             until, validate"
RMAN-01007: at line 3 column 1 file: /tmp/badcmdfile

```

4-5 ページの「コマンド・ファイルでの置換変数の使用」の説明に従って、置換変数を使用してコマンド・ファイルを動的にします。置換変数を含むコマンド・ファイルの構文を確認する場合は、Recovery Manager によって値を入力するように求められます。例 4-1 は、動的コマンド・ファイルの構文を確認する際に無効な値を入力した場合を示しています。太字のテキストは、プロンプトとして入力されたテキストを示しています。

#### 例 4-1 無効な構文が含まれているコマンド・ファイルの構文の確認

```

RMAN> CONNECT TARGET *
2> BACKUP TAG
Enter value for 1: mybackup
abc COPIES
Enter value for 2: mybackup
abc
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-00558: error encountered while parsing input commands
RMAN-01009: syntax error: found "identifier": expecting one of: "integer"
RMAN-01008: the bad identifier was: mybackup
RMAN-01007: at line 2 column 25 file: /tmp/whole_db.cmd

```

文字列 **mybackup** は COPIES の有効な引数ではないため、Recovery Manager は構文エラーを示します。

## Recovery Manager によるデータベース接続の確立

この項では、ターゲット・データベースに Recovery Manager クライアントを接続する方法について説明します。この項では次の項目を取り上げます。

- Recovery Manager によるデータベース接続
- Recovery Manager による補助データベースへの接続
- コマンド・ファイル内での Recovery Manager によるデータベース接続の確立
- Recovery Manager による接続で発生した問題の診断

## Recovery Manager によるデータベース接続

適切な作業を行うために、Recovery Manager クライアントをデータベースに接続する必要があります。次の表に、Recovery Manager で確立できるデータベース接続のタイプを示します。

表 4-1 Recovery Manager によるデータベース接続の概要

| データベース接続のタイプ                | キーワード     | 説明   |
|-----------------------------|-----------|--|
| ターゲット・データベース                | TARGET    | Recovery Manager によってバックアップまたはリストアされるデータベース  |
| リカバリ・カタログ・データベース            | CATALOG   | 制御ファイルとは別の、Recovery Manager リポジトリのオプションのバックアップ先となるデータベース   |
| 補助インスタンス<br>または<br>補助データベース | AUXILIARY | フィジカル・スタンバイ・データベース、または複製データベースの作成、表領域のトランスポート、表領域の Point-in-Time リカバリの実行などの特定の作業を実行するために作成されたデータベース・インスタンス<br><br>補助データベースを使用する多くのタスクとして、Recovery Manager は、タスク実行中に使用する自動補助インスタンスを作成し、そのインスタンスに接続してタスクを実行し、タスクの完了時にそのインスタンスを破棄します。自動補助インスタンスに接続するために明示的なコマンドは指定しません。 |

## Recovery Manager によるデータベース接続の認証

Recovery Manager によるデータベースへの接続は、SQL\*Plus によるデータベースへの接続と同じ方法で指定および認証します。Recovery Manager によるターゲット・データベースまたは補助データベースへの接続には SYSDBA 権限が必要である点のみが異なります。AS SYSDBA キーワードは、ターゲット接続および補助接続に対して暗黙的に指定され、明示的には指定できません。

リカバリ・カタログへの接続では、SYSDBA 権限は必要ありません。カタログ・スキーマの所有者には、RECOVERY\_CATALOG\_OWNER ロールを付与する必要があります。

**参照：** SQL\*Plus を使用する場合のデータベース接続オプションについては、『Oracle Database 管理者ガイド』を参照してください。

### オペレーティング・システムを使用した、Recovery Manager によるデータベース接続の認証

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には、Oracle SID を指定する環境変数を設定する必要があります。たとえば、UNIX シェルで SID を prod に設定するには、次のように入力します。

```
% ORACLE_SID=prod; export ORACLE_SID
```

オペレーティング・システム認証を使用すると、特別なオペレーティング・システム・グループによって SYSDBA 接続が制御されます。通常、このグループは OSDBA と呼ばれます。このグループは、データベースのインストール・プロセスの一環として作成され、特定の名前が割り当てられます。この特定の名前は、オペレーティング・システムによって異なります。

現行のオペレーティング・システム・ユーザーが OSDBA グループのメンバーで、Oracle SID が設定されている場合は、次のように入力すると、Recovery Manager が SYSDBA 権限でこのデータベースに接続できます。

```
% rman TARGET /
```

**パスワード・ファイルを使用した、Recovery Manager によるデータベース接続の認証** データベースで **パスワード・ファイル** が使用されている場合、Recovery Manager はパスワードを使用してこのデータベースに接続できます。パスワード・ファイルは、ローカルまたはリモート・アクセスに使用します。ネット・サービス名を指定して SYSDBA でリモート接続する場合は、パスワード・ファイルを使用する必要があります。

---

**注意：** 適切なセキュリティ・プラクティスとして、コマンドラインにパスワードを入力するときは、プレーン・テキストで入力しないでください。Recovery Manager では、Recovery Manager プロンプトで要求された場合にのみパスワードを入力してください。パスワード保護については、『Oracle Database セキュリティ・ガイド』を参照してください。

---

Recovery Manager は、次の例のように、接続文字列にパスワードを含めずに起動できます。

```
% rman TARGET SYS@prod

target database Password: password
connected to target database: PROD1 (DBID=39525561)
```

Recovery Manager によってパスワードの入力が求められますが、文字はエコーされません。

## オペレーティング・システムのコマンドラインからの Recovery Manager によるデータベース接続の確立

オペレーティング・システムのコマンドラインからターゲット・データベースに接続するには、rman コマンドの後に接続情報を続けて入力します。Recovery Manager プロンプトが表示されたら、コマンドの実行を開始できます。

表 4-2 に、この章の例で使用される一般的な値の意味を示します。

**表 4-2 例の値**

| 例で使用される値 | 意味   |
|----------|--|
| SYS      | SYSDBA 権限を持つユーザー   |
| prod     | ターゲット・データベースのネット・サービス名。  |
| rco      | リカバリ・カタログのスキーマを所有するユーザー。このユーザーは、RECOVERY_CATALOG_OWNER ロールが付与されているリカバリ・カタログ・データベースで定義されています。 |
| 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      | 補助インスタンスのネット・サービス名。  |

例 4-2 は、オペレーティング・システム認証を使用するターゲット・データベースへの接続を示しています。NOCATALOG オプションは、セッションでリカバリ・カタログが使用されないことを示します。

#### 例 4-2 システム・プロンプトからのターゲット・データベースへの接続

```
% rman TARGET / NOCATALOG

connected to target database: PROD (DBID=39525561)
using target database control file instead of recovery catalog

RMAN>
```

例 4-3 は、Oracle Net 認証を使用するターゲット・データベースへの接続を示しています。Recovery Manager によってパスワードの入力を求められます。

#### 例 4-3 システム・プロンプトからのターゲット・データベースへの接続

```
% rman TARGET SYS@prod NOCATALOG

target database Password: password
connected to target database: PROD (DBID=39525561)

RMAN>
```

CATALOG キーワードを使用してリカバリ・カタログに接続します。例 4-4 は、ターゲット・データベースおよびリカバリ・カタログ・データベースに対して Oracle Net 認証を使用する接続を示しています。いずれの場合も、Recovery Manager によってパスワードの入力を求められます。

#### 例 4-4 システム・プロンプトからのターゲット・データベースおよびカタログ・データベースへの接続

```
% rman TARGET SYS@prod CATALOG rco@catdb

target database Password: password
connected to target database: PROD (DBID=39525561)
recovery catalog database Password: password
connected to recovery catalog database

RMAN>
```

NOCATALOG または CATALOG を指定せずに Recovery Manager を起動することもできます。コマンドラインで NOCATALOG を指定しなかった場合および Recovery Manager の起動後に CONNECT CATALOG を指定しなかった場合、Recovery Manager リポジトリを使用する必要があるコマンドを初めて実行すると、Recovery Manager はデフォルトで NOCATALOG モードに設定されます。

---

---

**注意：** Recovery Manager リポジトリを使用するコマンドを NOCATALOG モードで実行した後で、リカバリ・カタログに接続するには、Recovery Manager を終了して再起動する必要があります。

---

---

オペレーティング・システムのコマンドラインでターゲット・データベースに接続する場合は、Recovery Manager プロンプトが表示された後で、コマンドの実行を開始できます。

## Recovery Manager プロンプトからのデータベース接続の確立

ターゲット・データベースに接続せずに Recovery Manager を起動する場合は、Recovery Manager プロンプトで `CONNECT TARGET` コマンドを発行し、ターゲット・データベースに接続してから有効な作業の実行を開始する必要があります。

### Recovery Manager プロンプトからデータベース接続を確立する手順

1. オペレーティング・システムのコマンドラインで、データベース接続を確立せずに Recovery Manager クライアントを起動します。たとえば、`rman` を次のように入力します。

```
% rman
RMAN>
```

2. Recovery Manager プロンプトで、1 つ以上の `CONNECT` コマンドを入力します。

次の例では、オペレーティング・システム認証を使用してターゲット・データベースに接続します。

```
RMAN> CONNECT TARGET /
```

次の例では、ターゲット・データベースに接続した後、リカバリ・カタログに接続します。ターゲット接続ではオペレーティング・システム認証が使用され、カタログ・データベース接続では Oracle Net 認証が使用されます。Recovery Manager によって、リカバリ・カタログ・ユーザーのパスワードの入力を求められます。

```
RMAN> CONNECT TARGET /
RMAN> CONNECT CATALOG rco@catdb
```

```
recovery catalog database Password: password
connected to recovery catalog database
```

次の例では、データベース・レベルの資格証明でターゲット・データベースに接続します。Recovery Manager によって SYS パスワードの入力を求められます。

```
% rman
RMAN> CONNECT TARGET SYS@prod

target database Password: password
connected to target database: PROD (DBID=39525561)
```

**参照：** `CONNECT` コマンドについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## Recovery Manager による補助データベースへの接続

`DUPLICATE` コマンドを使用するには、**補助インスタンス**に接続する必要があります。Recovery Manager 表領域の **Point-in-Time リカバリ** を実行する場合にも、補助インスタンスに接続する必要がある場合があります。

---

**注意：** `DUPLICATE ... FROM ACTIVE DATABASE` コマンドを使用する場合は、ネット・サービス名が必要です。詳細は、23-8 ページの「**手順 1: 補助インスタンス用の Oracle パスワード・ファイルの作成**」を参照してください。

---

補助接続の形式はターゲット・データベース接続と同じですが、`TARGET` キーワードのかわりに `AUXILIARY` キーワードを使用します。例 4-5 では、Recovery Manager プロンプトからターゲット・データベースおよび補助インスタンスに接続しています。

**例 4-5 Recovery Manager プロンプトからのターゲット・データベースおよびカタログ・データベースへの接続**

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

auxiliary database Password: password
connected to auxiliary database: PROD (DBID=30472568)
```

**参照：**

- DUPLICATE コマンドの使用方法の詳細は、第 23 章「データベースの複製」を参照してください。
- TSPITR の実行方法の詳細は、第 20 章「Recovery Manager の表領域の Point-in-Time リカバリ (TSPITR) の実行」を参照してください。

**コマンド・ファイル内での Recovery Manager によるデータベース接続の確立**

データベース・レベルの接続情報（ユーザー名およびパスワード）とともに CONNECT コマンドを記述して Recovery Manager コマンド・ファイルを作成した場合、このファイルへの読取り権限を持っているすべてのユーザーがこのパスワードを参照できます。パスワードを含めた CONNECT 文字列を、コマンド・ファイルに安全に組み込む方法はありません。

CONNECT コマンドを使用する Recovery Manager コマンド・ファイルを作成した場合、このコマンド・ファイルを @ コマンドで実行すると接続文字列はエコーされません。この動作によって、Recovery Manager 出力を含むログ・ファイルに接続文字列が表示されなくなります。たとえば、次のように listbkup.rman コマンド・ファイルを作成するとします。

```
cat > listbkup.rman << EOF
CONNECT TARGET /
LIST BACKUP;
EOF
```

Recovery Manager を @ コマンドライン・オプションを指定して実行し、このスクリプトを実行します。

```
% rman @listbkup.rman
```

このコマンド・ファイルを実行すると、次の出力に示すように、接続文字列がアスタリスクに置き換えられます。

```
RMAN> CONNECT TARGET *
2> LIST BACKUP;
3>
connected to target database: RDBMS (DBID=771530996)

using target database control file instead of recovery catalog

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



## Recovery Manager による接続で発生した問題の診断

ターゲット・データベース、カタログ・データベースおよび補助データベースへの接続時に Recovery Manager で発生したエラーを診断する際、SQL\*Plus を使用してこれらのデータベースに直接接続すると、接続情報またはデータベースに関する根本的な問題を検出できます。

### ターゲット・データベースおよび補助データベースへの接続時に発生した問題の診断

Recovery Manager は、常に SYSDBA 権限を使用してターゲット・データベースおよび補助データベースに接続します。このため、SQL\*Plus を使用してターゲット・データベースおよび補助データベースへの接続時に発生した問題を診断する場合は、SYSDBA 接続を要求して、Recovery Manager の動作を再現します。

たとえば、次の Recovery Manager コマンドで接続エラーが発生したとします。

```
RMAN> CONNECT TARGET /
```

この場合、次の SQL\*Plus コマンドを使用して前の接続操作を再現できます。

```
SQL> CONNECT / AS SYSDBA
```

### リカバリ・カタログへの接続時に発生した問題の診断

Recovery Manager でリカバリ・カタログ・データベースに接続する場合、SYSDBA 権限は使用されません。そのため、SQL\*Plus を使用してリカバリ・カタログ・データベースへの接続時に発生した問題を診断する場合は、Recovery Manager に入力したものと同一データベース接続文字列を入力する必要があります。AS SYSDBA は指定しないでください。

## Recovery Manager のパイプ・インタフェースの使用

Recovery Manager にコマンドを発行してそれらのコマンドから出力を受信するかわりに、Recovery Manager のパイプ・インタフェースを使用できます。このインタフェースを使用すると、Recovery Manager は、オペレーティング・システム・シェルではなく DBMS\_PIPE PL/SQL パッケージによってコマンドを取得し、出力を送信します。このインタフェースを使用することによって、Recovery Manager へのポータブル・プログラム・インタフェースを作成できます。

パイプ・インタフェースを起動するには、Recovery Manager クライアントの PIPE コマンドライン・パラメータを使用します。Recovery Manager は 2 つのプライベート・パイプを使用します。1 つはコマンドの受信用、もう 1 つは出力の送信用です。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、次のコマンドで Recovery Manager を起動できます。

```
% rman PIPE abc TARGET /
```

Recovery Manager は、ターゲット・データベースで 2 つのパイプをオープンします。ORA\$RMAN\_ABC\_IN はユーザー・コマンドの受信に使用するパイプ、ORA\$RMAN\_ABC\_OUT はすべての出力を Recovery Manager に戻すために使用するパイプです。入力と出力の両方のパイプでは、すべてのメッセージは VARCHAR2 型です。

セキュリティの問題が発生する可能性があるため、Recovery Manager では、パイプ・インタフェースをパブリック・パイプで使用することはできません。パブリック・パイプの場合、パイプ名を知っているすべてのユーザーが、Recovery Manager にコマンドを送信してその出力を取得できます。

パイプは、初期化されていない場合はプライベート・パイプとして作成されます。Recovery Manager の起動前に入力パイプにコマンドを入力する場合は、DBMS\_PIPE.CREATE\_PIPE をコールして最初にパイプを作成しておく必要があります。パイプは、プライベート・パイプとして明示的に作成しないと、そのパイプへの最初のアクセス時に自動的にパブリック・パイプとして作成されます。Recovery Manager でパブリック・パイプを使用しようとすると、エラーが戻されます。

---

**注意：** 複数の Recovery Manager セッションをターゲット・データベースに対して実行する場合は、各 **Recovery Manager セッション** に一意のパイプ名を使用する必要があります。一意のパイプ名を生成する方法の1つとして、DBMS\_PIPE.UNIQUE\_SESSION\_NAME ファンクションを使用できます。

---

## パイプを介した複数の Recovery Manager コマンドの連続実行の例

この例では、Recovery Manager を制御するアプリケーションで複数のコマンドを連続実行します。パイプに各コマンドが送信および実行されて出力が戻されると、Recovery Manager は一時停止して次のコマンドを待機します。

### パイプを介して Recovery Manager コマンドを実行する手順

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

```
% rman PIPE abc TARGET /
```

TIMEOUT オプションを指定することもできます。このオプションを指定すると、指定した時間内（秒単位）に入力パイプから入力を受信しなかった場合、Recovery Manager が強制終了されます。たとえば、次のように入力します。

```
% rman PIPE abc TARGET / 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 を使用して目的のコマンドを入力パイプに入力します。パイプ・モードでは、Recovery Manager で入力の受入れ準備が完了すると、標準 Recovery Manager プロンプトではなく RMAN-00572 メッセージが表示されます。
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 出力を読み取ります。
4. 手順2と3を繰り返し、手順1で起動した Recovery Manager インスタンスで他のコマンドを実行します。
5. Recovery Manager の起動時に TIMEOUT オプションを使用した場合は、指定した時間内に入力を受信しなければ Recovery Manager が自動的に終了します。すぐに Recovery Manager を終了するには、EXIT コマンドを送信します。

## パイプを介した Recovery Manager コマンドの単一ジョブでの実行の例

この例では、Recovery Manager を制御するアプリケーションで1つ以上のコマンドを単一ジョブとして実行します。パイプに入力されたコマンドの実行後、Recovery Manager が終了します。

### パイプを介して単一ジョブで Recovery Manager コマンドを実行する手順

1. ターゲット・データベースへの接続後、パイプを作成します（ORA\$RMAN\_pipe\_IN という名前のパイプが存在しない場合）。
2. 目的のコマンドを入力パイプに入力します。パイプ・モードでは、Recovery Manager で入力の受入れ準備が完了すると、標準 Recovery Manager プロンプトではなく RMAN-00572 メッセージが表示されます。
3. PIPE オプションを使用して Recovery Manager を起動し、TIMEOUT 0 を指定します。たとえば、次のように入力します。

```
% rman PIPE abc TARGET / TIMEOUT 0
```

4. `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取り、実行します。入力パイプのすべてのコマンドを実行すると、Recovery Manager はすぐに終了します。
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 出力を読み取ります。

**参照：** `DBMS_PIPE` パッケージのドキュメントについては、『Oracle Database 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 環境の構成

この章では、Recovery Manager を構成する場合の基本的なタスクについて説明します。この章の内容は、次のとおりです。

- Recovery Manager バックアップの環境の構成
- メディア・マネージャにバックアップするための Recovery Manager の構成
- フラッシュ・リカバリ領域の構成
- バックアップの保存方針の構成
- バックアップの最適化の構成
- アーカイブ REDO ログの削除方針の構成
- Oracle Flashback Database およびリストア・ポイントの構成
- Data Guard 環境での Recovery Manager の構成

**参照：** この章で説明していない構成オプション（バックアップの圧縮やバックアップの暗号化など）については、第 6 章「Recovery Manager 環境の構成:高度なトピック」を参照してください。

## Recovery Manager バックアップの環境の構成

Recovery Manager には、バックアップに必要なほとんどのパラメータに対して、基本的なバックアップおよびリカバリを実行できる適切なデフォルトが備えられています。Recovery Manager ベースのバックアップ計画を実装する場合、一般的な構成を理解すると、Recovery Manager を効果的に使用できます。

Recovery Manager の継続的な使用を簡単にするには、各ターゲット・データベースに多くの永続構成を設定します。これらの設定によって、Recovery Manager の動作の様々な点が制御されます。たとえば、バックアップの保存方針、バックアップのデフォルトの格納先、デフォルトのバックアップ・デバイス・タイプなどを構成できます。SHOW および CONFIGURE コマンドを使用すると、Recovery Manager 構成の表示および変更を行うことができます。

この項では、Recovery Manager 構成の概要および CONFIGURE コマンドを使用して Recovery Manager のデフォルトの動作をバックアップとリカバリ環境用に変更する方法について説明します。また、使用可能な主な設定および一般的な設定値についても説明します。

---

**注意：** テープにバックアップする場合に Recovery Manager をメディア・マネージャとともに使用するように設定する方法については、5-9 ページの「メディア・マネージャにバックアップするための Recovery Manager の構成」を参照してください。

---

この項の内容は、次のとおりです。

- Recovery Manager の永続的な構成の表示およびクリア
- バックアップ用のデフォルト・デバイスの構成: ディスクまたは SBT
- バックアップ用のデフォルト・タイプの構成: バックアップ・セットまたはコピー
- チャンネルの構成
- 制御ファイルおよびサーバー・パラメータ・ファイルの自動バックアップの構成

**参照：** CONFIGURE 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## Recovery Manager の永続的な構成の表示およびクリア

SHOW コマンドを使用すると、ターゲット・データベース用に設定された Recovery Manager の現在の値を表示できます。これらのコマンドが、現在デフォルト値に設定されているかどうかを表示することもできます。

### CONFIGURE コマンド設定を表示または変更する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. Recovery Manager の SHOW コマンドを実行します。

たとえば、例 5-1 に示すように、SHOW ALL を実行します（出力例も示します）。出力には、変更したパラメータおよびデフォルトに設定されたパラメータが含まれています。構成は、構成の再作成に必要な一連の Recovery Manager コマンドとして表示されます。出力をテキスト・ファイルに保存したり、このコマンド・ファイルを使用して同じまたは別のデータベースに構成を再作成することができます。

### 例 5-1 SHOW ALL コマンド

```
SHOW ALL;
```

```
RMAN configuration parameters for database with db_unique_name PROD1 are:  
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 3 DAYS;  
CONFIGURE BACKUP OPTIMIZATION ON;  
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
```

```

CONFIGURE CONTROLFILE AUTOBACKUP ON;
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE SBT_TAPE TO '%F'; # default
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F'; # default
CONFIGURE DEVICE TYPE 'SBT_TAPE' PARALLELISM 2 BACKUP TYPE TO COMPRESSED BACKUPSET;
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE SBT_TAPE TO 1; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE SBT_TAPE TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMS 'ENW=(OB_DEVICE=tape1)';
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE OFF; # default
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/disk1/oracle/dbs/snapcf_ev.f'; # default

```

SHOW コマンドは、特定の構成の名前を指定して使用することもできます。たとえば、次のように入力すると、保存方針およびデフォルトのデバイス・タイプを表示できます。

```

SHOW RETENTION POLICY;
SHOW DEFAULT DEVICE TYPE;

```

- 必要に応じて、CONFIGURE ... CLEAR コマンドを使用して構成をデフォルト値に戻します。次に例を示します。

```

CONFIGURE BACKUP OPTIMIZATION CLEAR;
CONFIGURE RETENTION POLICY CLEAR;
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK CLEAR;

```

**参照：** SHOW 構文の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## バックアップ用のデフォルト・デバイスの構成：ディスクまたは SBT

バックアップ先のデバイス・タイプが指定されていないバックアップは、構成済のデフォルト・デバイスに格納されます。Recovery Manager は、デフォルトのデバイス・タイプとしてディスクを使用するように事前構成されています。追加の構成は必要ありません。

デフォルトのデバイス・タイプをディスクからテープに変更したり、テープからディスクに戻す必要がある場合があります。表 5-1 に、デフォルトのデバイスを構成するコマンドを示します。

表 5-1 デフォルトのデバイス・タイプを構成するコマンド

| コマンド                                  | 説明  |
|---------------------------------------|---|
| CONFIGURE DEFAULT DEVICE TYPE TO DISK | <p>バックアップをデフォルトでディスクに格納するように指定します。</p> <p>リカバリ領域が有効になっている場合、バックアップ場所はデフォルトで<b>フラッシュ・リカバリ領域</b>になります。リカバリ領域が有効になっていない場合、バックアップ場所はデフォルトでディスク上のオペレーティング・システム固有のディレクトリになります。</p> <p>ディスクにバックアップする場合、データベース・ファイルの論理ブロック・サイズは、バックアップ先のデバイスの物理ブロック・サイズの倍数である必要があります。たとえば、ブロック・サイズが 2KB の DISK タイプのデバイスは、論理ブロック・サイズが 2KB、4KB、6KB などのデータベース・ファイルのバックアップ先としてのみ使用できます。ほとんどのディスク・ドライブの物理ブロック・サイズは 512 バイトであるため、この制限が、ディスク・ドライブへのバックアップに影響を及ぼすことはほとんどありません。ただし、書込み可能な DVD または物理ブロック・サイズが大きいデバイスにバックアップする場合は、この制限の影響を受ける可能性があります。</p> |

表 5-1 デフォルトのデバイス・タイプを構成するコマンド (続き)

| コマンド                                    | 説明  |
|---|---|
| CONFIGURE DEFAULT<br>DEVICE TYPE TO sbt | バックアップをデフォルトでテープに格納するように指定します。<br>メディア・マネージャで使用する Recovery Manager の設定方法については、5-9 ページの「 <a href="#">メディア・マネージャにバックアップするための Recovery Manager の構成</a> 」を参照してください。<br>Recovery Manager がメディア・マネージャと通信できるようになった後、テープにバックアップが作成されるように Recovery Manager を構成し、デフォルトのデバイス・タイプとして <b>SBT</b> を指定します。 |

デフォルトのデバイスは、BACKUP コマンドの DEVICE TYPE 句を使用して、常に上書きできます。

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

#### 構成済みのデフォルトのデバイス・タイプを変更する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. SHOW ALL コマンドを実行して、現在構成されているデフォルトのデバイスを表示します。
3. TO DISK または TO sbt のいずれかを指定して、CONFIGURE DEFAULT DEVICE TYPE コマンドを実行します。

**参照：** DEVICE TYPE 句を指定した BACKUP コマンドを使用する方法の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## バックアップ用のデフォルト・タイプの構成: バックアップ・セットまたはコピー

BACKUP コマンドによって、バックアップ・セットまたはイメージ・コピーを作成できます。ディスクの場合、CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO コマンドを使用すると、デフォルトのバックアップ・タイプとしてバックアップ・セットまたはイメージ・コピーを作成するように Recovery Manager を構成できます。ディスクのデフォルトのバックアップ・タイプは、圧縮されていない[バックアップ・セット](#)です。

---

**注意：** Recovery Manager はディスクにのみ[イメージ・コピー](#)を書き込むことができるため、テープのバックアップ・タイプはバックアップ・セットのみになります。

---

Recovery Manager は[バイナリ圧縮](#)を使用してバックアップ・セットを作成できます。BACKUP TYPE TO ... BACKUPSET 句に COMPRESSED オプションを指定すると、デバイス・タイプに対してデフォルトで圧縮バックアップ・セットを使用するように Recovery Manager を構成できます。圧縮を無効にするには、他の必要な設定を示す引数を指定して CONFIGURE DEVICE TYPE コマンドを使用しますが、COMPRESSED キーワードは省略します。

#### バックアップのデフォルト・タイプを構成する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。



- バックアップ・セットまたはコピーをデフォルトのバックアップ・タイプとして構成します。

次の例では、ディスク・バックアップのバックアップ・タイプをコピーおよびバックアップ・セットに構成しています。

```
CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO COPY; # image copies
CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO BACKUPSET; # uncompressed
```

次の例では、バックアップ・セットの圧縮を構成しています。

```
CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO COMPRESSED BACKUPSET;
CONFIGURE DEVICE TYPE sbt BACKUP TYPE TO COMPRESSED BACKUPSET;
```

**参照：**

- 「バックアップ・セット」 (7-3 ページ)
- 「圧縮バックアップの作成」 (8-6 ページ)

## チャネルの構成

3-4 ページの「[Recovery Manager チャネル](#)」で説明されているように、Recovery Manager [チャネル](#)とは、データベース・サーバー・セッションへの接続のことです。Recovery Manager は、ほぼすべての Recovery Manager タスクの実行にチャネルを使用します。

### チャネルの構成

ディスクまたは [SBT](#) チャネルの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 では、ALLOCATE CHANNEL コマンドで 1 回だけのオプションを指定する場合と同じオプションを使用します。デバイス・タイプに汎用チャネル設定を構成できます。汎用チャネル設定は、デバイスの構成済設定に基づいて作成されるチャネルで使用されるテンプレートです。

---

**注意：** この項では、ディスク・チャネルの構成について説明します。テープのチャネルを構成する方法については、5-13 ページの「[メディア・マネージャで使用する SBT チャネルの構成](#)」を参照してください。

---

CONFIGURE CHANNEL を使用してデバイスの汎用チャネル設定を指定する場合は、設定が競合していても、以前の設定は廃棄されることに注意してください。たとえば、構成済ディスク・チャネルの FORMAT のみを指定する 2 つ目の CONFIGURE CHANNEL コマンドを実行すると、ディスク・チャネルの MAXPIECESIZE がデフォルト値に戻ります。

```
CONFIGURE CHANNEL DEVICE TYPE DISK MAXPIECESIZE 2G;
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT /tmp/%U;
```

### ディスクのチャネルの構成

デフォルトでは、Recovery Manager は、1 つのディスク・チャネルをすべての操作に割り当てます。このチャネルに対して、バックアップ用の新しいデフォルトの場所などの他のオプションを指定する必要がある場合があります。例 5-2 では、ディスク・バックアップを /disk1 ディレクトリに書き込むように Recovery Manager を構成し、相対ファイル名にデフォルト以外の書式を指定します。

#### 例 5-2 デフォルト以外のバックアップ場所の構成

```
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '/disk1/ora_df%t_s%s_s%p';
```

例 5-2 では、Recovery Manager は、書式指定子 %t を 4 バイトのタイムスタンプに、%s をバックアップ・セット番号に、%p をバックアップ・ピース番号に自動的に置換します。

---

**注意：** ディスク・チャンネルに明示的な書式を構成すると、Recovery Manager は、デフォルトでは**フラッシュ・リカバリ領域**にバックアップを作成しません。この場合、フラッシュ・リカバリ領域のディスク領域管理機能は失われます。

---

次の例のように、ASM ディスクの場所を指定することもできます。

```
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '+dgroup1';
```

**参照：** バックアップを作成する方法については、8-7 ページの「[Recovery Manager を使用したデータベース・ファイルのバックアップ](#)」を参照してください。

## ディスクおよび SBT デバイス用のチャンネルの平行化の構成

コマンドの実行時にデバイス・タイプで使用できるチャンネルの数によって、Recovery Manager が読取りまたは書込みを平行で行うかどうかが決まります。通常、コマンドの実行に使用するチャンネルの数は、アクセスされるデバイスの数と一致する必要があります。したがって、テープ・バックアップの場合は、テープ・ドライブごとに1つのチャンネルを割り当てます。ディスク・バックアップの場合は、物理ディスクごとに1つのチャンネルを割り当てます。ただし、複数のチャンネルを使用して、ディスク・サブシステム・アーキテクチャ用にバックアップを最適化できる場合は除きます。適切な数のチャンネルが割り当てられない場合、I/O 操作中の Recovery Manager のパフォーマンスが低下します。

CONFIGURE DEVICE TYPE sbt を使用すると、**SBT** デバイスに対して**チャンネルの平行化**設定、バックアップ・セットの**バイナリ圧縮**などのオプションを構成できます。デバイス・タイプの構成は、チャンネルの構成とは別に設定できます。

**例 5-3** では、Recovery Manager で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できるように SBT デバイスの平行化を変更します（出力例も示します）。構成済の各 SBT チャンネルでは、データの合計の約半分ずつがバックアップされます。

### 例 5-3 SBT デバイスの平行化の構成

```
RMAN> CONFIGURE DEVICE TYPE sbt PARALLELISM 2;
```

```
old RMAN configuration parameters:
CONFIGURE DEVICE TYPE 'SBT_TAPE' BACKUP TYPE TO COMPRESSED BACKUPSET PARALLELISM 1;
new RMAN configuration parameters:
CONFIGURE DEVICE TYPE 'SBT_TAPE' PARALLELISM 2 BACKUP TYPE TO COMPRESSED BACKUPSET;
new RMAN configuration parameters are successfully stored
```

**例 5-4** では、SBT デバイスのデフォルトのバックアップ・タイプを、圧縮されていないバックアップ・セットに変更します（出力例も示します）。

### 例 5-4 SBT デバイスのバックアップ・タイプの構成

```
RMAN> CONFIGURE DEVICE TYPE sbt BACKUP TYPE TO BACKUPSET;
```

```
old RMAN configuration parameters:
CONFIGURE DEVICE TYPE 'SBT_TAPE' PARALLELISM 2 BACKUP TYPE TO COMPRESSED BACKUPSET;
new RMAN configuration parameters:
CONFIGURE DEVICE TYPE 'SBT_TAPE' BACKUP TYPE TO BACKUPSET PARALLELISM 2;
new RMAN configuration parameters are successfully stored
```

平行化およびバックアップ・タイプを設定するためにこの例で使用されている CONFIGURE DEVICE TYPE コマンドは、指定されていない設定値には影響を及ぼしません。**例 5-3** では、平行化を変更しても、圧縮済バックアップ・セットのデフォルトのバックアップ・タイプは変更されていません。**例 5-4** では、デフォルトのバックアップ・タイプを変更しても、平行化は変更されていません。

**参照：**

- ディスク・バックアップをパラレル化する方法については、8-5 ページの「[ディスク・バックアップの複数のフォーマットの指定](#)」を参照してください。
- BACKUP コマンドの CHANNEL パラメータの詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- Oracle Real Application Clusters (RAC) 構成でのパラレル化の詳細は、『Oracle Real Application Clusters 管理およびデプロイメント・ガイド』を参照してください。

**構成済チャネルの手動での変更**

ジョブの実行中にチャネルを手動で割り当てると、構成済チャネルのすべての設定が無視されます。たとえば、デフォルトのデバイス・タイプが SBT に構成された状態で、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 は、RUN コマンド内で手動で割り当てられたディスク・チャネルのみを使用して、CONFIGURE DEVICE TYPE、CONFIGURE DEFAULT DEVICE または CONFIGURE CHANNEL 設定を使用して設定されたすべてのデフォルトを上書きします。

**参照：**

- 構成済チャネルおよび割り当て済チャネルについては、3-4 ページの「[Recovery Manager チャネル](#)」を参照してください。
- ALLOCATE 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- CONFIGURE 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

**制御ファイルおよびサーバー・パラメータ・ファイルの自動バックアップの構成**

7-12 ページの「[制御ファイルおよびサーバー・パラメータ・ファイルの自動バックアップ](#)」で説明されているように、Recovery Manager は、制御ファイルおよびサーバー・パラメータ・ファイルを自動的にバックアップするように構成することができます。自動バックアップは、バックアップ・レコードが追加されるたびに実行されます。また、データベースが ARCHIVELOG モードで実行されている場合も、自動バックアップは、制御ファイル内のデータベース構造メタデータが変更されるたびに実行されます。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・ファイルが消失した場合でも、[制御ファイルの自動バックアップ](#)を使用して、Recovery Manager はデータベースをリカバリできます。

自動バックアップのファイル名は標準書式に従っているため、Recovery Manager は、リポジトリにアクセスせずにこのファイルを検索し、サーバー・パラメータ・ファイルをリストアできます。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すると、Recovery Manager は、自動バックアップから制御ファイルをリストアできます。制御ファイルをマウントすると、Recovery Manager リポジトリが使用可能になり、Recovery Manager は、データファイルをリストアし、アーカイブ REDO ログが検索できるようになります。

次のコマンドを実行すると、自動バックアップ機能を有効にできます。

```
CONFIGURE CONTROLFILE AUTOBACKUP ON;
```

次のコマンドを実行すると、自動バックアップ機能を無効にできます。

```
CONFIGURE CONTROLFILE AUTOBACKUP OFF;
```

**参照：** CONFIGURE 構文については、『Oracle Database バックアップおよびリカバリ・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書式の構成

デフォルトでは、すべての構成済デバイスの自動バックアップ・ファイルの書式は FORMAT 句の置換変数 %F です。この変数の書式は、c-*IIIIIIIIII*-*YYYYMMDD*-*QQ* に変換され、プレースホルダは次のように定義されます。

- *IIIIIIIIII* は、DBID を表します。
- *YYYYMMDD* は、バックアップが生成された日のタイム・スタンプです。
- *QQ* は、00 から始まる 16 進の順序です。最大値は FF です。

次のコマンドを使用すると、デフォルトの書式を変更できます。ここで、*deviceSpecifier* は有効なデバイス・タイプです。'string' には置換変数 %F (他の置換変数は除く) が含まれている必要があります。また、'string' は指定したデバイスの有効なハンドルとなります。

```
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT
FOR DEVICE TYPE deviceSpecifier TO 'string';
```

たとえば、次のコマンドを実行すると、制御ファイルの自動バックアップにデフォルト以外のファイル名を指定できます。ファイル名で、? は ORACLE\_HOME を表します。

```
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT
FOR DEVICE TYPE DISK TO '?/oradata/cf_%F';
```

次の例では、自動ストレージ管理ディスク・グループに書き込まれるように自動バックアップを構成します。

```
CONFIGURE CONTROLFILE AUTOBACKUP
FOR DEVICE TYPE DISK TO '+dgroup1/%F';
```

デバイスに対する制御ファイルの自動バックアップの書式をクリアするには、次のコマンドを使用します。

```
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK CLEAR;
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE sbt CLEAR;
```

データベースに **フラッシュ・リカバリ領域** が設定されている場合、ディスクに対する制御ファイルの自動バックアップ書式をクリアすると、制御ファイルの自動バックアップをフラッシュ・リカバリ領域に格納することができます。

## 構成済の制御ファイルの自動バックアップ書式の上書き

RUN コマンド内または Recovery Manager プロンプトのいずれかで指定できる SET CONTROLFILE AUTOBACKUP FORMAT コマンドを指定すると、現行のセッションのみで構成済の自動バックアップ書式が上書きされます。優先順位は次のとおりです。

1. SET CONTROLFILE AUTOBACKUP FORMAT (RUN ブロック内)
2. SET CONTROLFILE AUTOBACKUP FORMAT (Recovery Manager プロンプト)
3. CONFIGURE CONTROLFILE AUTOBACKUP FORMAT

次の例では、2つの形式の SET CONTROLFILE AUTOBACKUP FORMAT の相互作用を示します。

```
SET CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO 'controlfile_%F';
BACKUP AS COPY DATABASE;
RUN
{
    SET CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '/tmp/%F.bck';
    BACKUP AS BACKUPSET
```

```

DEVICE TYPE DISK
DATABASE;
}

```

最初の SET CONTROLFILE AUTOBACKUP FORMAT は、制御ファイルのすべての構成済自動バックアップ書式より優先され、Recovery Manager クライアントが終了するまで制御ファイルの自動バックアップの名前を制御します。RUN ブロック内の SET CONTROLFILE AUTOBACKUP FORMAT では、RUN ブロックが有効なかぎり、RUN ブロックの外にある SET CONTROLFILE AUTOBACKUP FORMAT は上書きされます。

## メディア・マネージャにバックアップするための Recovery Manager の構成

ほとんどのプラットフォームでは、テープなどのシーケンシャル・メディアにバックアップしたり、これらのメディアからリストアしたりする場合には、Oracle Database と **メディア・マネージャ** を統合する必要があります。テープに対してデータベース・システムとファイル・システムの両方のバックアップをサポートしている **Oracle Secure Backup** をメディア・マネージャとして使用できます。特に Oracle Secure Backup とともに使用するために Recovery Manager を設定する方法の詳細は、『Oracle Secure Backup 管理者ガイド』を参照してください。

Oracle Secure Backup を使用しない場合は、サード・パーティのメディア・マネージャを使用できます。この項では、サード・パーティのメディア・マネージャとともに使用するために Recovery Manager を構成する一般的な手順について説明します。実際の手順は、インストールしたメディア管理製品およびデータベースを実行しているプラットフォームによって異なります。Oracle Secure Backup 以外のメディア・マネージャとともに Recovery Manager を使用する場合は、製品固有のすべての情報をベンダーから入手する必要があります。

メディア・マネージャを構成する際、次の項を順に参照してください。

1. **Recovery Manager** をメディア・マネージャとともに使用するための前提条件
2. **メディア管理ライブラリの場所の確認**
3. **Recovery Manager** のバックアップに対するメディア管理ソフトウェアの構成
4. **メディア・マネージャ・ライブラリが正常に統合されたかどうかのテスト**
5. **メディア・マネージャで使用する SBT チャンネルの構成**

**参照：** メディア管理ソフトウェアおよび Recovery Manager が受ける影響の概要は、3-6 ページの「**メディア管理**」を参照してください。

## Recovery Manager をメディア・マネージャとともに使用するための前提条件

Recovery Manager をサード・パーティのメディア・マネージャとともに使用するには、メディア・マネージャをインストールし、Recovery Manager と通信できるようにする必要があります。詳細は、メディア管理ソフトウェアのベンダーのドキュメントを参照してください。

通常、まず、メディア管理ソフトウェアをターゲット・ホストまたは本番ネットワークにインストールして構成します。オペレーティング・システム・ファイルの Recovery Manager 以外のバックアップをターゲット・データベース・ホスト上に作成できることを確認します。この手順に従って、メディア・マネージャとターゲット・ホストの基本的な統合が正常に完了したことを確認すると、後で行うトラブルシューティングが簡単になります。Recovery Manager を使用しないでメディア・マネージャにファイルをバックアップする方法については、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。

次に、サード・パーティのメディア管理モジュールを入手してインストールし、データベース・サーバーと統合します。このモジュールには、Oracle Database がメディア・マネージャへのアクセス時にロードおよび使用する **メディア管理ライブラリ** が含まれています。このモジュールは通常、別ライセンスのサード・パーティ製品です。詳細は、メディア管理モジュールのベンダーに問い合せてください。

## メディア管理ライブラリの場所の確認

メディア・マネージャとともに Recovery Manager を使用する前に、メディア管理ライブラリの場所を指定します。Recovery Manager でメディア・マネージャとの通信に使用する **チャンネル** の割当てまたは構成を行う場合は、ALLOCATE CHANNEL または CONFIGURE CHANNEL コマンドに SBT\_LIBRARY パラメータを指定する必要があります。SBT\_LIBRARY パラメータには、ライブラリへのパスを指定します。

次に、チャンネルの構文の例を示します。ここで、*pathname* は、ライブラリの絶対ファイル名です。

```
CONFIGURE CHANNEL DEVICE TYPE sbt
  PARMS 'SBT_LIBRARY=pathname';
```

Recovery Manager は、メディア・マネージャと通信するためのチャンネルを割り当てる場合、SBT\_LIBRARY パラメータで指定されたライブラリをロードします。

割当て済または事前構成済のチャンネルで SBT\_LIBRARY パラメータの値を指定しない場合、Recovery Manager は、プラットフォーム固有のデフォルトの場所を検索します。Linux および UNIX では、デフォルトのライブラリ・ファイル名は \$ORACLE\_HOME/lib/libobk.so です。拡張子名はプラットフォームに応じて異なり、.so、.sl (HP-UX の場合)、.a (AIX の場合) などが使用されます。Windows では、デフォルトのライブラリの場所は %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、標準のデータベース・インストールには含まれません。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た場合にのみ含まれます。

---

**参照：** プラットフォーム上でメディア・マネージャを統合する方法については、オペレーティング・システム固有のデータベース・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。

## Recovery Manager のバックアップに対するメディア管理ソフトウェアの構成

メディア管理ソフトウェアのインストール後、Recovery Manager でバックアップを受け入れることができるように、ベンダーの要件に従ってソフトウェアを構成します。インストールしたメディア管理ソフトウェアのタイプによっては、メディア・プールの定義、ユーザーおよびクラスの構成などが必要になる場合があります。

Recovery Manager の適切な設定については、メディア管理ベンダーのドキュメントを参照してください。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。ALLOCATE CHANNEL または CONFIGURE CHANNEL コマンドに PARMS 値が必要な場合、あるいは BACKUP または CONFIGURE コマンドに FORMAT 文字列が推奨されている場合の詳細は、ベンダーのドキュメントを参照してください。

例 5-5 に、Oracle Secure Backup の PARMS 設定を示します。この PARMS 設定は、datafile\_mf という一連のテープにバックアップを行うようにメディア・マネージャに指示します。PARMS 設定は、常にベンダー固有です。

### 例 5-5 Oracle Secure Backup の PARMS 設定

```
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE'
  PARMS 'ENV=(OB_MEDIA_FAMILY=datafile_mf)';
```



**参照:**

- ALLOCATE CHANNEL 構文については、『Oracle Database バックアップ およびリカバリ・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、『Oracle Database バックアップ およびリカバリ・リファレンス』を参照してください。
- Oracle Secure Backup に対する Recovery Manager 固有のパラメータ設定については、『Oracle Secure Backup リファレンス』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たかどうかのテスト**

データベース・サーバーで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 でメディア・マネージャにバックアップできるかどうかをテストします。

**メディア・マネージャでの ALLOCATE CHANNEL のテスト**

次の手順では、ALLOCATE CHANNEL コマンドを使用して、Recovery Manager がメディア・マネージャと通信できるかどうかを確認する基本的なテストを実行します。

**チャネルの割当てをテストする手順**

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. メディア管理ソフトウェアの要件に従って、PARMS を指定して ALLOCATE CHANNEL コマンドを実行します。

次の RUN コマンドは、ベンダー固有の PARMS 設定の例を示しています。

```
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt
  PARMS 'SBT_LIBRARY=/mydir/lib/libobk.so,
  ENV=(OB_DEVICE=drive1,OB_MEDIA_FAMILY=datafile_mf)';
}
```

3. Recovery Manager 出力を確認します。

エラー・メッセージが表示されない場合、データベースはメディア管理ライブラリを正常にロードしています。ORA-27211 エラーが発生した場合、メディア管理ライブラリはロードされていません。

```
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03009: failure of allocate command on c1 channel at 11/30/2007 13:57:18
ORA-19554: error allocating device, device type: SBT_TAPE, device name:
ORA-27211: Failed to load Media Management Library
Additional information: 25
```

この場合、メディア管理ソフトウェアの使用環境にメディア管理ライブラリが正常にインストールされているかどうかを確認し、SBT\_LIBRARY パラメータの値（5-10 ページの「メディア管理ライブラリの場所の確認」を参照）を再確認する必要があります。

SBT\_LIBRARY パラメータに指定された場所またはデフォルトの場所でデータベースがメディア管理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。

チャンネルの割当てに失敗すると、データベースは、常に[自動診断リポジトリ](#)・ホーム・ディレクトリの TRACE サブディレクトリにトレース・ファイルを書き込みます。出力例を次に示します。

```
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
```

**参照：** 自動診断リポジトリを使用してデータベース操作を監視する方法については、『Oracle Database 管理者ガイド』を参照してください。

## メディア・マネージャでのバックアップおよびリストア操作のテスト

メディア・マネージャでチャンネルの割当てをテストした後、テスト・バックアップを作成し、リストアします。たとえば、[例 5-6](#) の（メディア管理ベンダーで要求されているチャンネル設定に置き換える）コマンドを使用すると、メディア・マネージャにバックアップを作成できるかどうかをテストできます。データベースでサーバー・パラメータ・ファイルが使用されていない場合は、かわりに、現行の制御ファイルをバックアップします。

### 例 5-6 テープへのサーバー・パラメータ・ファイルのバックアップ

```
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt
  PARMS 'SBT_LIBRARY=/mydir/lib/libobk.so,
  ENV=(OB_DEVICE=drive1,OB_MEDIA_FAMILY=datafile_mf)';
  BACKUP SPFILE;
  # If your database does not use a server parameter file, use:
  # BACKUP CURRENT CONTROLFILE;
}
```

バックアップが正常に完了した後、サーバー・パラメータ・ファイルを初期化パラメータ・ファイルとしてリストアします。[例 5-7](#) では、[例 5-6](#) で作成したバックアップを一時ディレクトリにリストアします。

### 例 5-7 テープからのサーバー・パラメータ・ファイルのリストア

```
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt
  PARMS 'SBT_LIBRARY=/mydir/lib/libobk.so,
  ENV=(OB_DEVICE=drive1,OB_MEDIA_FAMILY=datafile_mf)';
  RESTORE SPFILE TO PFILE '/tmp/test_restore.f';
  # If your database does not use a server parameter file, use:
  # RESTORE CURRENT CONTROLFILE;
}
```

バックアップおよびリストア操作が正常に完了すると、Recovery Manager でメディア・マネージャを使用できます。バックアップの失敗には、次の状況が考えられます。

- バックアップが中断する。
 

バックアップが中断した場合、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のメディア・マネージャ・ジョブの有無を確認し、問題を修正します。5-9 ページの「[メディア・マネージャにバックアップするための Recovery Manager の構成](#)」の手順を正しく実行していることを確認します。
- バックアップが「ORA-27211: メディア管理ライブラリのロードに失敗しました」を戻して失敗する。
 

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。5-9 ページの「[メディア・マネージャにバックアップするための Recovery Manager の構成](#)」の手順を正しく実行していることを確認します。また、メディア管理ソフトウェアの要件に従って、PARMS および FORMAT 文字列を正しく設定していることも確認します。



**参照:** 22-11 ページの「Media Management API のテスト」および第 22 章「Recovery Manager 操作のトラブルシューティング」を参照してください。

## メディア・マネージャで使用する SBT チャンネルの構成

この項では、メディア・マネージャで使用するチャンネルを構成する方法について説明します。構成済チャンネルの概要およびその使用方法については、6-2 ページの「高度なチャンネル・オプションの構成」を参照してください。

### メディア・マネージャのバックアップ・ピース名

**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 コマンドに指定する FORMAT 文字列によって決まります。メディア・マネージャでは、バックアップ・ピース名がバックアップ・ファイル名として認識されるため、**メディア管理カタログ**内で一意である必要があります。

FORMAT パラメータ内の置換変数を使用すると、一意のバックアップ・ピース名を生成できます。たとえば、%d にはデータベース名を指定し、%t にはバックアップ・タイムスタンプを指定します。ほとんどの用途に対して %U を使用できます。この場合、Recovery Manager は一意なファイル名を自動的に生成します。バックアップ・ピース名の例としては、12i1nk47\_1\_1 があります。FORMAT パラメータを指定しない場合、Recovery Manager は %U 置換変数を含む一意のファイル名を自動的に生成します。

使用しているメディア・マネージャで、ファイルの名前およびサイズに制限がある場合があります。この場合、バックアップ・ピース名がメディア・マネージャの制限に従うように、バックアップ・ピース名の指定により詳細な制御を行う必要があります。たとえば、14 文字のバックアップ・ピース名のみをサポートするメディア・マネージャ、特別な FORMAT 文字列を必要とするメディア・マネージャなどが存在します。%U 置換変数によって生成される一意の名前が 14 文字を超えないように注意してください。

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可能なファイルの最大サイズが制限されている場合があります。Recovery Manager がこの制限を超えるバックアップ・セットを生成しないようにする必要があります。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 コマンドに設定可能なパラメータ MAXPIECESIZE を使用します。

#### 参照:

- MAXPIECESIZE を設定する方法については、『Oracle Database バックアップおよびリカバリ・リファレンス』および 7-6 ページの「バックアップ・ピースの数およびサイズ」を参照してください。
- BACKUP コマンドの FORMAT 文字列に使用可能な変数の完全なリストは、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- メディア・マネージャでの文字列の文字制限については、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。

### 自動 SBT チャンネルの構成

メディア・マネージャをバックアップする最も簡単な方法は、**SBT** チャンネルを構成する方法です。5-3 ページの「バックアップ用のデフォルト・デバイスの構成: ディスクまたは SBT」で説明されているように、テープ・デバイスをデフォルトのバックアップ先として使用できます。

**メディア・マネージャで使用するチャンネルを構成する手順**

1. 一般的な SBT チャンネルを構成します。

この構成では、5-12 ページの「[メディア・マネージャでのバックアップおよびリストア操作のテスト](#)」でテストしたすべてのパラメータを入力します。次の例では、ベンダー固有のチャンネル・パラメータを構成し、デフォルト・デバイスを設定します。

```
CONFIGURE CHANNEL DEVICE TYPE sbt
PARMS 'ENV=(OB_RESOURCE_WAIT_TIME=1minute,OB_DEVICE=tape1)';
```

2. 次のコマンドを実行して、デフォルトのデバイス・タイプを SBT に設定します。

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
```

複数のテープ・デバイスを使用する場合は、チャンネルの平行化を指定する必要があります (5-6 ページの「[ディスクおよび SBT デバイス用のチャンネルの平行化の構成](#)」を参照)。次のように構成すると、2 つのテープ・ドライブに平行にバックアップできます。

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 2;
```

必要に応じて、次のコマンドを実行してチャンネルの構成を確認します。

```
SHOW CHANNEL FOR DEVICE TYPE sbt;
```

3. テープへのテスト・バックアップを実行します。

次のコマンドを実行すると、サーバー・パラメータ・ファイルがテープにバックアップされます。

```
BACKUP SPFILE;
```

4. バックアップを表示して、テスト・バックアップがメディア・マネージャに送信されたことを確認します。

```
LIST BACKUP OF SPFILE;
```

## フラッシュ・リカバリ領域の構成

3-7 ページの「[フラッシュ・リカバリ領域](#)」で説明されているように、[フラッシュ・リカバリ領域](#)機能を使用すると、バックアップおよびリカバリに関連する様々なファイルの作成と管理をデータベースで実行可能なディスク領域を設定できます。フラッシュ・リカバリ領域を使用することをお勧めします。バックアップ計画を実装する場合の最初の手順の1つとして、フラッシュ・リカバリ領域を構成することを検討してください。

この項では、フラッシュ・リカバリ領域の機能の概要、この領域に格納されているファイルの識別方法、ファイル管理の規則および最も重要な構成オプションの概要について説明します。この項の内容は、次のとおりです。

- [フラッシュ・リカバリ領域の概要](#)
- [フラッシュ・リカバリ領域の有効化](#)
- [制御ファイルおよび REDO ログの場所の構成](#)
- [フラッシュ・リカバリ領域内での Recovery Manager によるファイル作成の構成](#)

**参照:** 「[フラッシュ・リカバリ領域のメンテナンス](#)」 (11-7 ページ)

## フラッシュ・リカバリ領域の概要

フラッシュ・リカバリ領域には、制御ファイル、オンライン REDO ログ、アーカイブ REDO ログ、フラッシュバック・ログおよび Recovery Manager バックアップを含めることができます。フラッシュ・リカバリ領域内のファイルは、永続的なファイルまたは一時的なファイルに分類されます。永続的なファイルとは、データベース・インスタンスで使用されるアクティブなファイルのことです。永続的でないすべてのファイルは一時的なファイルとなります。通常、Oracle Database では、バックアップの保存方針に従って不要になった一時的なファイルまたはテープにバックアップされた一時的なファイルは最終的に削除されます。

表 5-2 に、リカバリ領域内のファイル、各ファイルの分類（永続的または一時的）およびデータベースの可用性への影響を示します。

表 5-2 フラッシュ・リカバリ領域内のファイル

| ファイル                      | タイプ | リカバリ領域にアクセスできない場合のデータベースの動作  |
|---------------------------|-----|--|
| 現行の制御ファイルの多重コピー           | 永続的 | データベースでフラッシュ・リカバリ領域に格納されている制御ファイルの多重コピーに書き込むことができない場合は、インスタンスで障害が発生します。アクセス可能な多重コピーがリカバリ領域外にある場合でも障害が発生します。<br><b>参照:</b> フラッシュ・リカバリ領域で制御ファイルを構成する方法については、5-20 ページの「 <a href="#">制御ファイルの場所の構成</a> 」を参照してください。                                    |
| オンライン REDO ログ・ファイル        | 永続的 | フラッシュ・リカバリ領域外のアクセス可能な場所にオンライン REDO ログのミラー化されたコピーが存在している場合、インスタンスの可用性は影響を受けません。そうでない場合、インスタンスで障害が発生します。<br><b>参照:</b> フラッシュ・リカバリ領域でオンライン REDO ログ・ファイルを構成する方法については、5-20 ページの「 <a href="#">オンライン REDO ログの場所の構成</a> 」を参照してください。                      |
| アーカイブ REDO ログ・ファイル        | 一時的 | フラッシュ・リカバリ領域外のアクセス可能な場所にログがアーカイブされる場合、インスタンスの可用性は影響を受けません。そうでない場合、データベースは、オンライン REDO ログをアーカイブできないため最終的に停止します。<br><b>参照:</b> フラッシュ・リカバリ領域でアーカイブ REDO ログ・ファイルを構成する方法については、5-21 ページの「 <a href="#">アーカイブ REDO ログの場所の構成</a> 」を参照してください。               |
| 外部のアーカイブ REDO ログ・ファイル     | 一時的 | インスタンスの可用性は影響を受けません。<br><b>注意:</b> 外部のアーカイブ REDO ログ・ファイルは、LogMiner セッション用のロジカル・スタンバイ・データベースによって受信されます。 <a href="#">外部のアーカイブ REDO ログ</a> の DBID は、通常のアーカイブ・ログの DBID とは異なっています。このため、外部のアーカイブ REDO ログ・ファイルは、ロジカル・スタンバイ・データベース上ではバックアップまたはリストアできません。 |
| データファイルおよび制御ファイルのイメージ・コピー | 一時的 | インスタンスの可用性は影響を受けません。   |
| バックアップ・ピース                | 一時的 | インスタンスの可用性は影響を受けません。   |

表 5-2 フラッシュ・リカバリ領域内のファイル (続き)

| ファイル        | タイプ | リカバリ領域にアクセスできない場合のデータベースの動作  |
|-------------|-----|--|
| フラッシュバック・ログ | 一時的 | <p>保証付きリストア・ポイントが定義されていない場合、インスタンスの可用性は影響を受けません。この場合、データベースによって、フラッシュバック・データベースが自動的に無効となり、アラート・ログにメッセージが書き込まれてデータベース処理が続行されます。保証付きリストア・ポイントが構成されている場合、インスタンスはフラッシュバック・ログとの相互依存関係が原因で失敗します。</p> <p><b>フラッシュバック・ログ</b>は、<b>データベースの Point-in-Time リカバリ</b>にかわる有効な機能を提供する <b>Oracle Flashback Database</b> 機能によって生成されます。これらのログは一時的なファイルであり、フラッシュ・リカバリ領域に格納する必要があります。他の一時的なファイルとは異なり、フラッシュバック・ログは他のメディアにバックアップできません。フラッシュ・リカバリ領域にフラッシュバック・ログを格納できる十分な領域がなく、リカバリ領域が他のバックアップ保存要件を満たしている場合、そのリカバリ領域では、フラッシュバック・ログを削除することができます。</p> <p><b>参照：</b>フラッシュバック・ロギングを有効にする方法については、5-34 ページの「<b>フラッシュバック・データベースの有効化</b>」を参照してください。</p> |

Windows プラットフォームを使用している場合は、**Volume Shadow Copy Service (VSS)** を **Oracle VSS ライター** とともに使用できます。この場合、フラッシュ・リカバリ領域によって、VSS スナップショットにすでにバックアップされているファイルの管理が自動化され、必要に応じてそれらのファイルが削除されます。

**参照：**

- [第 16 章「フラッシュバックおよびデータベースの Point-in-Time リカバリの実行」](#)
- VSS 環境でのバックアップの作成方法の詳細は、Oracle Database の [プラットフォーム・ガイド](#)を参照してください。

### Oracle Managed Files および自動ストレージ管理

フラッシュ・リカバリ領域は、**Oracle Managed Files (OMF)** および**自動ストレージ管理**とともに使用できます。フラッシュ・リカバリ領域は、OMF の上に構築されるため、OMF を格納できるすべての場所に格納できます。リカバリ領域は、ASM で使用することもできます。

ASM ストレージにフラッシュ・リカバリ領域を設定しない場合でも、Oracle Managed Files を使用して ASM ディスク・グループでバックアップ・ファイルを管理することができます。新しいバックアップで領域が必要となった場合にリカバリ可能目標を達成するためにファイルを自動削除する必要がなくなるため、フラッシュ・リカバリ領域の主要なメリットのうちの 1 つ (ファイルの自動削除) が失われます。ただし、OMF の他の自動機能は動作します。

バックアップを格納する場合、フラッシュ・リカバリ領域を使用せずに ASM で OMF を使用することはできますが、お薦めしません。ASM 下のファイルは直接操作しにくいからです。

### Oracle によるフラッシュ・リカバリ領域でのディスク領域の管理方法

フラッシュ・リカバリ領域内の領域は、保存方針に従って保持する必要があるバックアップおよびアーカイブ・ログと、削除される可能性がある他のファイルの間でバランスが取られています。Oracle Database では、他の目的のために領域の再利用が必要になるまで、対象のファイルはフラッシュ・リカバリ領域から削除されません。したがって、多くの場合、最近テープに移されたファイルはディスクでリカバリに使用できます。リカバリ領域は、テープに対するキャッシュとして使用できます。Oracle Database では、フラッシュ・リカバリ領域が一杯になると、フラッシュ・リカバリ領域内の領域を再利用するために必要に応じて対象のファイルが自動的に削除されます。

- 参照：** 11-7 ページの「[フラッシュ・リカバリ領域の規則的削除](#)」および 11-8 ページの「[フラッシュ・リカバリ領域が一杯になった場合の対応](#)」を参照してください。

## フラッシュ・リカバリ領域の有効化

リカバリ領域を有効にするには、表 5-3 に必須と示されている初期化パラメータを設定する必要があります。Oracle Real Application Clusters (Oracle RAC) データベースでは、すべてのインスタンスでこれらの初期化パラメータの値が同じである必要があります。この場所は、クラスタ・ファイル・システム、ASM または共有ディレクトリ上である必要があります。

表 5-3 フラッシュ・リカバリ領域の初期化パラメータ

| 初期化パラメータ                      | 必須   | 説明  |
|-------------------------------|------|---|
| DB_RECOVERY_FILE_DEST_SIZE    | 該当   | <p><b>ディスク割当て制限</b>を指定します。ディスク割当て制限は、このデータベースのリカバリ領域で使用されるデータの最大格納量をバイト単位で示したものです。このパラメータは、DB_RECOVERY_FILE_DEST の前に設定する必要があります。</p> <p>DB_RECOVERY_FILE_DEST_SIZE 設定には、次のディスク・オーバーヘッドは含まれていません。</p> <ul style="list-style-type: none"> <li>■ ブロック 0、または各 Oracle ファイルのオペレーティング・システムのブロック・ヘッダーは含まれていません。</li> </ul> <p>フラッシュ・リカバリ領域に必要な実際のディスク使用量を計算する場合は、このデータに対して 10% の追加を見込みます。</p> <ul style="list-style-type: none"> <li>■ 基礎となるファイル・システムがミラー化または圧縮されている場合、あるいは Oracle Database では認識されないオーバーヘッドの影響を受けている場合、DB_RECOVERY_FILE_DEST_SIZE は、ディスクを占有している実際のサイズを示しません。</li> </ul> <p>たとえば、フラッシュ・リカバリ領域が 2 方向ミラー化 ASM ディスク・グループ上に存在する場合、x バイトの各ファイルは、ASM ディスク・グループの 2x バイトを占有します。この場合、DB_RECOVERY_FILE_DEST_SIZE は、ASM ディスク・グループのディスクのサイズの半分以下に設定します。同様に、3 方向ミラー化 ASM ディスク・グループを使用する場合、DB_RECOVERY_FILE_DEST_SIZE は、ディスク・グループ内のディスクのサイズの 3 分 1 以下にする必要があります。</p> |
| DB_RECOVERY_FILE_DEST         | 該当   | <p>リカバリ領域の場所を指定します。ファイル・システム上のディレクトリまたは ASM ディスク・グループを指定することができますが、RAW ディスクは指定できません。この場所は、ディスク割当て制限に対応できる十分な大きさである必要があります。</p>  |
| DB_FLASHBACK_RETENTION_TARGET | 該当せず | <p>データベースをフラッシュバックできる時点の上限 (分) を指定します。このパラメータは、フラッシュバック・データベースでのみ必要です。</p> <p>このパラメータによって、リカバリ領域で保持されるフラッシュバック・ログ・データのサイズが間接的に決定されます。ただし、データベースによって生成されるフラッシュバック・ログのサイズは、データベース・ワークロードによって大幅に異なる可能性があります。一定期間に行われるデータベース更新によって影響を受けるブロックが増えると、この期間に生成されるフラッシュバック・ログ・データによって使用されるディスク領域も増えます。</p>  |

**参照：** ALTER SYSTEM 構文については『Oracle Database SQL 言語リファレンス』、データベースの初期化パラメータの設定および変更については『Oracle Database 管理者ガイド』を参照してください。



## フラッシュ・リカバリ領域のサイズを設定する場合の考慮事項

フラッシュ・リカバリ領域は、大きいほど便利になります。フラッシュ・リカバリ領域は、5-15 ページの表 5-2 に示されているファイルを格納できる十分なサイズにすることを勧めます。リカバリ領域には、データベース内のすべてのデータファイルのコピー、および選択したバックアップ計画で使用される増分バックアップを格納する必要があります。

前述のサイズの領域を保持することが難しい場合は、最も重要な表領域のバックアップ、およびテープにまだコピーされていないすべてのアーカイブ・ログを保持できるサイズの領域を作成することをお勧めします。最低限、フラッシュ・リカバリ領域は、テープにコピーされていないアーカイブ REDO ログを格納できるサイズにする必要があります。リカバリ領域にフラッシュバック・ログを格納できる十分な領域がなく、リカバリ領域が他のバックアップ保存要件を満たしている場合、そのリカバリ領域では、フラッシュバック・ログを削除して領域を確保することができます。

有効なフラッシュ・リカバリ領域のサイズを判断する方法は、次の状況によって異なります。

- データベースに、頻繁に変更されるデータ・ブロックが少数含まれているか、多数含まれているか
- バックアップをディスクにのみ格納するか、ディスクおよびテープに格納するか
- 冗長性に基づいた**バックアップの保存方針**を使用するか、リカバリ期間に基づいた保存方針を使用するか
- 論理エラーに対応する場合の **Point-in-Time リカバリ**の代替方法として、フラッシュバック・データベースの使用を計画するか、**保証付きリストア・ポイント**の使用を計画するか

フラッシュバック・ロギングを有効にする場合、生成されるフラッシュバック・ログのサイズは、生成される REDO ログのサイズと同程度になります。たとえば、DB\_FLASHBACK\_RETENTION\_TARGET を 24 時間に設定し、1 日に 20GB の REDO がデータベースで生成される場合は、目安として 20 ~ 30GB のディスク領域をフラッシュバック・ログに使用できるようにします。フラッシュバック・ロギングが有効になっている場合、同じ目安が保証付きリストア・ポイントに適用されます。たとえば、毎日 20GB の REDO がデータベースで生成され、保証付きリストア・ポイントが 1 日保持される場合は、20GB ~ 30GB のディスク領域を割り当てます。

バックアップ保存方針が REDUNDANCY 1 に設定されている場合にフラッシュ・リカバリ領域のサイズを決定し、**増分更新バックアップ**を使用して Oracle 推奨の計画に従うとします。この例では、次の式を使用して、ディスク割当て制限を見積もります。 $n$  は、増分更新の間隔 (日) です。 $y$  は、ロジカル・スタンバイ・データベースへの外部のアーカイブ REDO ログの適用の遅延です。

```
Disk Quota =
Size of a copy of database +
Size of an incremental backup +
Size of (n+1) days of archived redo logs +
Size of (y+1) days of foreign archived redo logs (for logical standby) +
Size of control file +
Size of an online redo log member * number of log groups +
Size of flashback logs (based on DB_FLASHBACK_RETENTION_TARGET value)
```

## フラッシュ・リカバリ領域の場所を設定する場合の考慮事項

フラッシュ・リカバリ領域は、データファイル、制御ファイル、オンライン REDO ログなどのアクティブなデータベース・ファイルがデータベースによって保持されている**データベース領域**とは別のディスクに配置します。データベース領域と同じディスクにフラッシュ・リカバリ領域を設定している場合、メディアで障害が発生すると、アクティブなデータベース・ファイルとバックアップの両方を失う可能性があります。

DB\_RECOVERY\_FILE\_DEST は、DB\_CREATE\_FILE\_DEST 初期化パラメータまたは DB\_CREATE\_ONLINE\_LOG\_DEST\_n 初期化パラメータのいずれとも異なる値に設定することをお勧めします。DB\_RECOVERY\_FILE\_DEST がこれらのパラメータと同じ場合は、データベースによってアラート・ログに警告が書き込まれます。

次のいずれかの条件を満たす場合、複数のデータベースで DB\_RECOVERY\_FILE\_DEST に同じ値を指定できます。

- DB\_UNIQUE\_NAME 初期化パラメータが指定されている 2 つのデータベースで、DB\_UNIQUE\_NAME の値が同じではない。
- DB\_UNIQUE\_NAME が指定されていないデータベースで、DB\_NAME に同じ値が指定されているデータベースがない。

このように、複数のデータベースで 1 つのフラッシュ・リカバリ領域を共有する場合、フラッシュ・リカバリ領域の場所は、すべてのデータベースのリカバリ・ファイルを保持できるサイズにする必要があります。データベースの DB\_RECOVERY\_FILE\_DEST\_SIZE の値を追加し、ミラー化または圧縮などのオーバーヘッドを見込みます。

## フラッシュ・リカバリ領域の場所および初期サイズの設定

表 5-3 は、フラッシュ・リカバリ領域を有効にするために設定する必要がある初期化パラメータを示しています。この項では、リカバリ領域の場所の指定方法およびその初期サイズの設定方法について説明します。

### フラッシュ・リカバリ領域の最適なサイズを決定する手順

1. フラッシュバック・ロギングまたは保証付きリストア・ポイントを使用する場合は、V\$ARCHIVED\_LOG を問い合わせ、DB\_FLASHBACK\_RETENTION\_TARGET に設定する時間にデータベースで生成される REDO のサイズを決定します。
2. リカバリ領域のサイズを設定します。

フラッシュバック・ロギングまたは保証付きリストア・ポイントを使用する場合は、手順 1 で取得したサイズの値が設定に組み込まれていることを確認します。次のいずれかの方法で、DB\_RECOVERY\_FILE\_DEST\_SIZE 初期化パラメータを設定します。

- データベースを停止して、データベースの初期化パラメータ・ファイルに DB\_RECOVERY\_FILE\_DEST\_SIZE パラメータを設定します。次に例を示します。  
DB\_RECOVERY\_FILE\_DEST\_SIZE = 10G
- データベースがオープン状態のときに、SQL 文 ALTER SYSTEM SET を使用してリカバリ領域のサイズを指定します。次に例を示します。

```
ALTER SYSTEM SET
  DB_RECOVERY_FILE_DEST_SIZE = 10G
  SCOPE=BOTH SID='*';
```

- Database Configuration Assistant を使用してサイズを設定します。

3. リカバリ領域の場所を設定します。

次のいずれかの方法で、初期化パラメータを設定します。

- データベースの初期化パラメータ・ファイルに DB\_RECOVERY\_FILE\_DEST を設定します。次に例を示します。

```
DB_RECOVERY_FILE_DEST = '/u01/oradata/rcv_area'
```

- データベースがオープン状態のときに、SQL 文 ALTER SYSTEM SET を使用して DB\_RECOVERY\_FILE\_DEST を指定します。次に例を示します。

```
ALTER SYSTEM SET
  DB_RECOVERY_FILE_DEST = '+disk1'
  SCOPE=BOTH SID='*';
```

- Database Configuration Assistant を使用して場所を設定します。

フラッシュバック・ロギングを使用しない場合は、データベースをオープンし（クローズしている場合）、残りの手順は実行しないでください。

- フラッシュバック・ロギングが有効になっている場合は、`DB_FLASHBACK_RETENTION_TARGET` で指定された期間、通常のワークロードでデータベースを実行します。

これによって、データベースでフラッシュバック・ログの代表的なサンプルを生成できます。

- `V$FLASHBACK_DATABASE_LOG` ビューを次のように問い合わせます。

```
SELECT ESTIMATED_FLASHBACK_SIZE
FROM   V$FLASHBACK_DATABASE_LOG;
```

フラッシュバック・データベースが有効になっているため、データベース・ワークロードに基づいて、現在のフラッシュバック保存目標を達成するために必要なディスク領域の見積りが結果として表示されます。

- 必要に応じて、`DB_FLASHBACK_RETENTION_TARGET` で指定された期間に生成された実際のフラッシュバック・ログのサイズに基づいて、フラッシュバック・ログ領域要件を調整します。

**参照:** 「フラッシュ・リカバリ領域でのフラッシュバック・ログの領域の管理」 (11-8 ページ)

## 制御ファイルおよび REDO ログの場所の構成

5-15 ページの「フラッシュ・リカバリ領域の概要」で説明されているように、現行の制御ファイルおよびオンライン REDO ログの多重コピーのみが永続的なファイルです。この項では、これらのファイルおよびアーカイブ・ログの場所の設定方法について説明します。

### オンライン REDO ログの場所の構成

オンライン REDO ログ・ファイルが作成される場所を決定する初期化パラメータは、`DB_CREATE_ONLINE_LOG_DEST_n`、`DB_RECOVERY_FILE_DEST` および `DB_CREATE_FILE_DEST` です。これらのパラメータの組合せによるオンライン REDO ログの作成への影響の詳細は、『Oracle Database SQL 言語リファレンス』の `CREATE DATABASE` 文の `LOGFILE` 句に関する事項を参照してください。

次の SQL 文を使用すると、フラッシュ・リカバリ領域にオンライン REDO ログを作成できます。

- `CREATE DATABASE`
- `ALTER DATABASE ADD LOGFILE`
- `ALTER DATABASE ADD STANDBY LOGFILE`
- `ALTER DATABASE OPEN RESETLOGS`

フラッシュ・リカバリ領域に作成されるオンライン・ログのデフォルトのサイズは 100MB です。ログ・メンバーのファイル名は、データベースによって自動的に生成されます。

### 制御ファイルの場所の構成

初期化パラメータ `CONTROL_FILES`、`DB_CREATE_ONLINE_LOG_DEST_n`、`DB_RECOVERY_FILE_DEST` および `DB_CREATE_FILE_DEST` が相互に作用して、データベース制御ファイルが作成される場所が決まります。これらのパラメータの相互作用の詳細は、『Oracle Database SQL 言語リファレンス』の `CREATE CONTROLFILE` のセマンティクスに関する事項を参照してください。

データベースによって Oracle 管理の制御ファイルが作成され、そのデータベースでサーバー・パラメータ・ファイルが使用されている場合は、そのサーバー・パラメータ・ファイルに `CONTROL_FILES` 初期化パラメータが設定されます。このデータベースでクライアント側の初期化パラメータ・ファイルが使用されている場合は、初期化パラメータ・ファイルに `CONTROL_FILES` 初期化パラメータを手動で設定する必要があります。



## アーカイブ REDO ログの場所の構成

アーカイブ・ログはデータベースによって自動的に管理されるため、アーカイブの場所としてフラッシュ・リカバリ領域を使用することをお勧めします。フラッシュ・リカバリ領域内のアーカイブ・ログに対して生成されるファイル名は Oracle 管理ファイル用のファイル名であり、LOG\_ARCHIVE\_FORMAT では決定されません。どのアーカイブ・スキームを選択した場合も、常にアーカイブ REDO ログの複数のコピーを作成することをお勧めします。

次に、アーカイブ REDO ログ用の基本的なオプションを推奨順に示します。

1. フラッシュ・リカバリ領域に対してのみアーカイブを有効にし、ディスクをミラー化してアーカイブ REDO ログの保護に必要な冗長性を作成します。

DB\_RECOVERY\_FILE\_DEST を指定し、LOG\_ARCHIVE\_DEST\_n を指定しない場合は、LOG\_ARCHIVE\_DEST\_10 が暗黙的にリカバリ領域に設定されます。LOG\_ARCHIVE\_DEST\_10 を空の文字列に設定すると、この動作を無効にできます。

2. フラッシュ・リカバリ領域に対してアーカイブを有効にし、他の LOG\_ARCHIVE\_DEST\_n 初期化パラメータをフラッシュ・リカバリ領域外の場所に設定します。

フラッシュ・リカバリ領域が構成されている場合は、LOG\_ARCHIVE\_DEST\_n パラメータを LOCATION=USE\_DB\_RECOVERY\_FILE\_DEST に設定すると、フラッシュ・リカバリ領域をアーカイブ先として追加できます。

3. フラッシュ・リカバリ領域外の場所にのみアーカイブされるように、LOG\_ARCHIVE\_DEST\_n 初期化パラメータを設定します。

フラッシュ・リカバリ領域を使用する場合、LOG\_ARCHIVE\_DEST および LOG\_ARCHIVE\_DUPLEX\_DEST 初期化パラメータは使用できません。使用すると、インスタンスを起動できなくなります。かわりに、LOG\_ARCHIVE\_DEST\_n パラメータを設定します。データベースで LOG\_ARCHIVE\_DEST\_n が使用されている場合は、リカバリ領域を構成できません。

また、アーカイブを有効にしており、LOG\_ARCHIVE\_DEST、LOG\_ARCHIVE\_DEST\_n または DB\_RECOVERY\_FILE\_DEST のいずれの値も設定していない場合、REDO ログはプラットフォーム固有のデフォルトの場所にアーカイブされることにも注意してください。たとえば、Solaris でのデフォルトの場所は ?/dbs です。

**参照：** LOG\_ARCHIVE\_DEST\_n パラメータのセマンティクスの詳細は、『Oracle Database リファレンス』を参照してください。

## フラッシュ・リカバリ領域内での Recovery Manager によるファイル作成の構成

この項では、フラッシュ・リカバリ領域でファイルを作成するための Recovery Manager コマンドまたは暗黙的な処理（制御ファイルの自動バックアップなど）について説明します。また、コマンドによってフラッシュ・リカバリ領域または別の場所のいずれかにファイルを作成するかどうかを制御する方法について説明します。コマンドを次に示します。

### ■ BACKUP

ディスク・バックアップの目的で FORMAT 句を指定していない場合は、Oracle Managed Files の名前形式の名前が付いたバックアップ・ピースおよびイメージ・コピーがフラッシュ・リカバリ領域に作成されます。フラッシュ・リカバリ領域が有効になっていて、BACKUP またはチャンネルで FORMAT を指定している場合、Recovery Manager は、リカバリ領域ではなくプラットフォーム固有の場所にバックアップを作成します。

### ■ 制御ファイルの自動バックアップ

Recovery Manager は、制御ファイルの自動バックアップをフラッシュ・リカバリ領域に作成できます。Recovery Manager コマンド CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK CLEAR を使用すると、ディスク上での制御ファイルの自動バックアップ場所を指定したすべての構成済 FORMAT オプションをクリアできます。他の格納場所が構成されていない場合、Recovery Manager はフラッシュ・リカバリ領域に制御ファイルの自動バックアップを作成します。

- RESTORE ARCHIVELOG  
LOG\_ARCHIVE\_DEST\_n パラメータのいずれかを  
LOCATION=USE\_DB\_RECOVERY\_FILE\_DEST に明示的または暗黙的に設定します。SET ARCHIVELOG DESTINATION を指定してこの動作を無効にしない場合、Recovery Manager はアーカイブ REDO ログ・ファイルをフラッシュ・リカバリ領域にアーカイブします。
- RECOVER DATABASE または RECOVER TABLESPACE、RECOVER ... BLOCK、および FLASHBACK DATABASE  
これらのコマンドを使用すると、コマンドからの要求に応じて、メディア・リカバリ中に使用されるアーカイブ REDO ログ・ファイルをバックアップからリストアできます。Recovery Manager は、これらの操作中に必要なすべての REDO ログ・ファイルをフラッシュ・リカバリ領域にリストアし、メディア・リカバリ中に適用した後で削除します。  
リストアされたアーカイブ・ログをフラッシュ・リカバリ領域に格納するには、LOG\_ARCHIVE\_DEST\_n パラメータのいずれかを  
LOCATION = USE\_DB\_RECOVERY\_FILE\_DEST に設定します。リストアされたログを他の場所に格納するための SET ARCHIVELOG DESTINATION は使用しないでください。

## バックアップの保存方針の構成

7-18 ページの「バックアップの保存方針」で説明されているように、**バックアップの保存方針**では、データのリカバリ要件に従って保存する必要があるバックアップを指定します。保存方針は、**リカバリ期間**または**冗長性**のいずれに基づいていてもかまいません。保存方針を指定するには、CONFIGURE RETENTION POLICY コマンドを使用します。

**参照：** CONFIGURE 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## 冗長性に基づく保存方針の構成

CONFIGURE RETENTION POLICY コマンドの REDUNDANCY パラメータでは、Recovery Manager が保持する必要がある各データファイルおよび制御ファイルの全体またはレベル 0 のバックアップの数を指定します。つまり、特定のデータファイルまたは制御ファイルの全体またはレベル 0 のバックアップの数が REDUNDANCY の設定を超えると、Recovery Manager は余分なバックアップを不要とみなします。デフォルトの保存方針は、REDUNDANCY 1 です。

作成するバックアップが増えるに従い、Recovery Manager は保存するバックアップおよび不要なバックアップを記録します。Recovery Manager は、不要でないバックアップのリカバリに必要なすべてのアーカイブ・ログおよび増分バックアップを保持します。

月曜日、火曜日、水曜日および木曜日にデータファイル 7 の全体バックアップを作成するとします。このデータファイルの全体バックアップは 4 つになります。REDUNDANCY が 2 の場合、月曜日と火曜日のバックアップは不要になります。金曜日にもう 1 つのバックアップを作成すると、データファイル 7 の水曜日のバックアップが不要となります。

REDUNDANCY が 1 であるという別の場合を想定します。月曜日の正午にレベル 0 のデータベース・バックアップを実行し、火曜日と水曜日の正午にレベル 1 の累積バックアップを、木曜日の正午にレベル 0 のバックアップを実行します。それぞれの日次バックアップの直後に DELETE OBSOLETE を実行します。水曜日に DELETE コマンドを実行しても、火曜日のレベル 1 のバックアップは冗長でないため、削除されません。火曜日のレベル 1 のバックアップは、火曜日の正午から水曜日の正午までの時点に月曜日のレベル 0 のバックアップをリカバリするために使用できます。ただし、木曜日の DELETE コマンドでは、以前のレベル 0 およびレベル 1 のバックアップが削除されます。

次の例のように、Recovery Manager プロンプトで CONFIGURE RETENTION POLICY コマンドを実行します。

```
CONFIGURE RETENTION POLICY TO REDUNDANCY 3;
```

**参照：** 「保存方針に基づく不要な Recovery Manager バックアップの削除」 (11-24 ページ)

## リカバリ期間に基づく保存方針の構成

CONFIGURE コマンドの RECOVERY WINDOW パラメータでは、現時点と最初のリカバリ可能ポイント間の日数を指定します。Recovery Manager は、全体バックアップまたは **レベル 0 の増分バックアップ** がリカバリ期間内にある場合はいずれのバックアップも不要とみなしません。また、Recovery Manager は、リカバリ期間内のランダムな時点までリカバリする必要があるすべてのアーカイブ・ログおよびレベル 1 増分バックアップを保持します。

Recovery Manager プロンプトで CONFIGURE RETENTION POLICY コマンドを実行します。次の例では、先週の任意の時点までデータベースをリカバリできます。

```
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;
```

Recovery Manager は、リカバリ期間によって不要とみなされたバックアップを自動的に削除しません。かわりに、REPORT OBSOLETE の出力や V\$OBSOLETE\_BACKUP\_FILES の OBSOLETE 列に、OBSOLETE として表示します。DELETE OBSOLETE コマンドを実行すると、Recovery Manager は不要なファイルを削除します。

**参照:** 「保存方針に基づく不要な Recovery Manager バックアップの削除」 (11-24 ページ)

## 保存方針の無効化

保存方針を無効にすると、Recovery Manager はすべてのバックアップを必要とみなします。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;
```

保存方針を NONE に構成することは、保存方針を消去することとは異なります。保存方針を消去すると、デフォルトの REDUNDANCY 1 の設定に戻りますが、NONE では保存方針が完全に無効になります。

保存方針を無効にし、保存方針オプションをコマンドに渡さずに REPORT OBSOLETE または DELETE OBSOLETE を実行すると、不要なバックアップを判断する保存方針が存在しないため、Recovery Manager はエラーを発行します。

---

**注意:** フラッシュ・リカバリ領域を使用している場合は、保存方針が無効になっている状態でデータベースを実行しないでください。ファイルが不要とみなされないと、ファイルをフラッシュ・リカバリ領域から削除できるのは、そのファイルが他のディスクの場所またはテープなどの 3 次ストレージ・デバイスにバックアップされた場合のみになります。リカバリ領域内のすべての領域が使用され、データベースの通常操作に影響を及ぼす可能性があります。詳細は、5-16 ページの「[Oracle によるフラッシュ・リカバリ領域でのディスク領域の管理方法](#)」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または無効にするには、CONFIGURE コマンドを実行します。バックアップを最適化すると、同じファイルまたは同じバージョンのファイルがすでにバックアップされている場合などの特定の状況においてファイルのバックアップがスキップされます。

## バックアップの最適化の概要

**バックアップの最適化**が有効な場合、指定されたデバイス・タイプにすでにファイルがバックアップされていると、BACKUP コマンドで同一ファイルのバックアップがスキップされます。表 5-4 に、ファイルがすでにバックアップされているファイルと同一かどうかを判断する際の、Recovery Manager の基準を示します。

表 5-4 同一ファイルかどうかを判断する基準

| ファイル・タイプ   | 同一ファイルかどうかを判断する基準  |
|------------|--|
| データファイル    | すでにバックアップに含まれているデータファイルと同じ <b>DBID</b> 、 <b>チェックポイント SCN</b> 、作成 SCN、および RESETLOGS の SCN と時刻。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、読取り専用か、または正常にクローズされている必要があります。 |
| アーカイブ・ログ   | 同じ DBID、スレッド、順序番号、および RESETLOGS の SCN と時刻。   |
| バックアップ・セット | 同じ DBID、バックアップ・セット・レコード ID およびスタンプ。  |

ファイルが同一でありすでにバックアップされていると Recovery Manager が判断した場合、そのファイルはスキップ対象になります。ただし、Recovery Manager は、ファイルをスキップするかどうかについて、さらに詳細な確認を行う必要があります。これは、Recovery Manager が指定されたデバイス・タイプに十分なバックアップを保持しているかどうかを判断するアルゴリズムに、保存方針とバックアップ多重化機能の両方の要素があるためです。

Recovery Manager は、次の条件を満たしている場合にバックアップの最適化を使用します。

- CONFIGURE BACKUP OPTIMIZATION ON コマンドを実行し、バックアップの最適化を有効にしている。
- BACKUP DATABASE、ALL または LIKE オプションを指定した BACKUP ARCHIVELOG、BACKUP BACKUPSET ALL、BACKUP RECOVERY AREA、BACKUP RECOVERY FILES あるいは BACKUP DATAFILECOPY を実行する。
- 1 つのタイプのチャンネルのみを割り当てている (1 つの BACKUP コマンドでディスクと SBT チャンネルを併用していない)。

---

**注意：** **UNDO のバックアップの最適化**では、Recovery Manager は、すでにコミットされたトランザクションについて、バックアップのリカバリに不要となった UNDO を除外します。バックアップの最適化を有効または無効にすることができますが、UNDO のバックアップの最適化は組込み動作です。

---

たとえば、バックアップの最適化を構成したと想定します。これらのコマンドは、データベース、すべてのアーカイブ・ログおよび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 TYPE sbt DATABASE PLUS ARCHIVELOG;
BACKUP DEVICE TYPE sbt BACKUPSET ALL;
```

バックアップされたすべてのファイルが最後のバックアップ以降に変更されていない場合、Recovery Manager はこれらのファイルを再度バックアップしません。また、Recovery Manager は、ファイルがすでにバックアップされているため、コマンドで指定されたすべてのファイルをスキップする場合、エラーを通知しません。

BACKUP コマンドに FORCE オプションを指定すると、いつでも最適化を無効にできます。たとえば、次のコマンドを実行します。

```
BACKUP DATABASE FORCE;
BACKUP ARCHIVELOG ALL FORCE;
```

**参照：** バックアップの最適化の規則の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』の CONFIGURE エントリに関する項を参照してください。

## SBT バックアップのバックアップの最適化への保存方針の影響

バックアップの最適化は、**SBT** デバイスへのバックアップ時に常に適用されるわけではありません。リカバリ期間および冗長性に基づく保存方針の通常のバックアップの最適化の動作の例外については、以降の項を参照してください。

---

**注意：** メディア・マネージャで独自の内部期限切れ方針を使用している場合、バックアップの最適化を有効にするには、注意が必要です。定期的に CROSSCHECK コマンドを実行して、Recovery Manager リポジトリをメディア・マネージャと同期化してください。同期化を実行しないと、Recovery Manager は、テープ上に格納されたバックアップがメディア・マネージャによって廃棄されたことを認識できず、最適化によってバックアップをスキップする場合があります。

---

### リカバリ期間に基づく保存方針による SBT バックアップのバックアップの最適化

バックアップの最適化が有効になっていて、リカバリ期間に基づく**バックアップの保存方針**が構成されているとします。この場合、SBT バックアップを実行すると、Recovery Manager は、最新のバックアップが**リカバリ期間**の前に作成されているデータファイルを常にバックアップします。たとえば、次の例を想定します。

- 現在の時点は 2 月 21 日です。
- リカバリ期間は 7 日です。
- 表領域 tools のテープへの最新のバックアップは 1 月 3 日に作成されています。
- 表領域 tools は読取り専用です。

2 月 21 日に、表領域 tools をテープにバックアップするコマンドを発行します。この表領域は読取り専用であるため、1 月 3 日のバックアップ以降も変更されていませんが、Recovery Manager はバックアップを実行します。これは、7 日間のリカバリ期間内にこの表領域のバックアップが存在しないためです。

この動作によって、メディア・マネージャは古いテープを期限切れにすることができます。この動作が行われない場合、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持し続けることとなります。2 月 21 日に表領域 tools のより新しいバックアップを作成することによって、メディア・マネージャは、1 月 3 日のバックアップを含むテープを期限切れとして処理できます。

### 冗長性に基づく保存方針による SBT バックアップのバックアップの最適化

冗長性に基づく保存方針を構成するとします。この場合、Recovery Manager は、SBT への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が  $r+1$  個存在する場合にのみ、これらのファイルのバックアップをスキップします。ここで、 $r$  は CONFIGURE RETENTION POLICY TO REDUNDANCY  $r$  で設定された値です。

たとえば、バックアップの最適化を有効にして、次の保存方針を設定するとします。

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE BACKUP OPTIMIZATION ON;
CONFIGURE RETENTION POLICY TO REDUNDANCY 2;
```

この例では、Recovery Manager は、3 つの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のみ、バックアップをスキップします。ここで、これまでにバックアップしたことのない読取り / 書込み表領域 users に対して、1 週間の間に表 5-5 に示す処理を実行するとします。

表 5-5 バックアップの最適化への冗長性設定の影響

| 曜日  | 処理                            | 結果                     | 冗長バックアップ   |
|-----|-------------------------------|------------------------|------------|
| 月曜日 | users を NORMAL モードでオフラインにします。 |                        |            |
| 火曜日 | BACKUP DATABASE               | users 表領域がバックアップされます。  |            |
| 水曜日 | BACKUP DATABASE               | users 表領域がバックアップされます。  |            |
| 木曜日 | BACKUP DATABASE               | users 表領域がバックアップされます。  | 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ません。 | 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ません。 | 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ます。     |            |
| 月曜日 | BACKUP DATABASE               | users 表領域がバックアップされます。  | 水曜日のバックアップ |

火曜日、水曜日および木曜日のバックアップによって、オフライン表領域 users がバックアップされ、3つ（冗長性設定 +1）のバックアップが存在する必要があるという条件を満たします。金曜日および土曜日のバックアップでは、バックアップの最適化のため、users 表領域のバックアップは行われません。users の火曜日のバックアップは、木曜日以降に不要となります。

日曜日にすべての不要なバックアップを削除すると、users の火曜日のバックアップも削除されます。火曜日のバックアップは、保存方針設定によって不要となります。月曜日のデータベース全体のバックアップによって、users 表領域がバックアップされ、3つ（冗長性設定 +1）のバックアップが存在する必要があるという条件を満たします。この方法で、時間の経過とともにテープを再利用できます。

**参照：**「バックアップの最適化の構成」(5-23 ページ)

## バックアップの最適化の構成

デフォルトでは、バックアップの最適化は OFF に構成されています。SHOW BACKUP OPTIMIZATION コマンドを使用すると、バックアップの最適化の現行の設定を表示できます。

### バックアップの最適化を構成する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. SHOW BACKUP OPTIMIZATION コマンドを実行して、最適化が現在有効になっているかどうかを確認します。

たとえば、次のコマンドを入力します。

```
SHOW BACKUP OPTIMIZATION;
```

SHOW BACKUP OPTIMIZATION の出力例を次に示します。

```
RMAN configuration parameters for database with db_unique_name PROD1 are:
CONFIGURE BACKUP OPTIMIZATION ON;
```

3. 次のコマンドを実行して、バックアップの最適化を有効にします。

```
CONFIGURE BACKUP OPTIMIZATION ON;
```

**参照：** Recovery Manager バックアップを最適化する方法の例は、9-4 ページの「バックアップの最適化を使用したファイルのスキップ」を参照してください。



## アーカイブ REDO ログの削除方針の構成

Recovery Manager を使用すると、アーカイブ REDO ログがディスクからの削除対象となる場合を制御する永続的な構成を作成できます。

### アーカイブ REDO ログの削除方針

CONFIGURE ARCHIVELOG DELETION POLICY コマンドを使用すると、アーカイブ REDO ログが削除対象になる場合を指定できます。この削除方針は、**フラッシュ・リカバリ領域**を含むすべてのアーカイブ先に適用されます。

アーカイブ REDO ログは、データベースによって自動的に削除されるか、またはユーザーが実行した Recovery Manager コマンドによって削除されます。フラッシュ・リカバリ領域内のログのみが、データベースによって自動的に削除されます。フラッシュ・リカバリ領域内のアーカイブ REDO ログ・ファイルの場合は、データベースによってできるかぎり保存され、追加のディスク領域が必要になると対象となるログが削除されます。フラッシュ・リカバリ領域の内部にあるか外部にあるかに関係なく、BACKUP ... DELETE INPUT または DELETE ARCHIVELOG を発行すると、すべての場所から対象のログを手動で削除できます。

#### アーカイブ REDO ログの削除方針が無効になっている場合

デフォルトでは、**アーカイブ REDO ログの削除方針**は NONE に構成されています。この場合、Recovery Manager は、次の両方の条件を満たしている場合にリカバリ領域内のアーカイブ REDO ログ・ファイルを削除対象であるとみなします。

- アーカイブ REDO ログが、フラッシュ・リカバリ領域の内部にあるか外部にあるかに関係なく、LOG\_ARCHIVE\_DEST\_n で指定された必須のリモート転送先に転送されている。
- アーカイブ REDO ログがディスクまたは SBT に 1 回以上バックアップされているか、またはアーカイブ REDO ログが**バックアップの保存方針**に従って不要となっている。

バックアップ保存方針では、ログは**保証付きリストア・ポイント**で不要である場合および **Oracle Flashback Database** で不要である場合のみ不要とみなされます。アーカイブ REDO ログは、ログが SYSDATE-'DB\_FLASHBACK\_RETENTION\_TARGET' より後に作成された場合にフラッシュバック・データベースで必要とされます。

#### 参照：

- 方針のオプションの詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』の CONFIGURE ARCHIVELOG DELETION POLICY エントリを参照してください。
- Data Guard 環境でのアーカイブ・ログの削除方針の構成方法については、『Oracle Data Guard 概要および管理』を参照してください。

#### アーカイブ REDO ログの削除方針が有効になっている場合

CONFIGURE ARCHIVELOG DELETION POLICY BACKED UP *integer* TIMES TO DEVICE TYPE コマンドを使用すると、アーカイブ・ログの削除方針を有効化できます。この構成では、指定した数のアーカイブ・ログのバックアップが指定したデバイス・タイプに存在する場合のみアーカイブ・ログが削除対象になるように指定されます。

削除方針が BACKED UP *integer* TIMES 句で構成されている場合、指定したデバイス・タイプ上に *integer* 個のバックアップがすでに存在していないかぎり、BACKUP ARCHIVELOG コマンドはログをコピーします。ログの *integer* 個のバックアップが存在している場合、BACKUP ARCHIVELOG コマンドはログをスキップします。これによって、アーカイブ・ログの削除方針は、デフォルトの BACKUP ARCHIVELOG コマンドの NOT BACKED UP *integer* TIMES 句として機能します。BACKUP コマンドに FORCE オプションを指定すると、削除方針を無効にできます。

アーカイブ・ログの削除方針には、Data Guard 環境固有のオプションもあります。たとえば、APPLIED ON STANDBY 句を指定すると、Recovery Manager は、すべての必須のリモート転送先でログが適用された後、ログを削除します。たとえば、SHIPPED TO STANDBY を指定すると、Recovery Manager は、すべての必須のスタンバイ転送先にログが転送された後、ログを削除します。

**参照：**

- 方針のオプションの詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』の CONFIGURE ARCHIVELOG DELETION POLICY エントリを参照してください。
- Data Guard 環境でのアーカイブ・ログの削除方針の構成方法については、『Oracle Data Guard 概要および管理』を参照してください。

## アーカイブ REDO ログの削除方針の有効化

この項では、アーカイブ REDO ログの削除方針の構成方法について説明します。デフォルトでは、削除方針は NONE に設定されています。

### アーカイブ REDO ログの削除方針を有効にする手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. 必要なオプションを指定して、CONFIGURE ARCHIVELOG DELETION POLICY コマンドを実行します。

次の例では、アーカイブ REDO ログが 2 回以上テープにバックアップされた場合、アーカイブ REDO ログがフラッシュ・リカバリ領域およびすべてのローカルのアーカイブ先からの削除の対象となるように指定しています。

```
CONFIGURE ARCHIVELOG DELETION POLICY
  TO BACKED UP 2 TIMES TO SBT;
```

**参照：**

- 「バックアップ後のアーカイブ REDO ログの削除」 (8-14 ページ)
- Data Guard 環境でのアーカイブ REDO ログの管理方法については、『Oracle Data Guard 概要および管理』を参照してください。
- CONFIGURE ARCHIVELOG DELETION POLICY コマンドの詳細およびアーカイブ・ログが削除対象となる条件については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## Oracle Flashback Database およびリストア・ポイントの構成

この項では、Oracle Flashback Database の環境を計画および構成する方法について説明します。この項では、リストア・ポイントの作成方法についても説明します。この項の内容は、次のとおりです。

- リストア・ポイントおよびフラッシュバック・データベース
- フラッシュバック・データベースおよび保証付きリストア・ポイントの前提条件
- フラッシュバック・データベースの有効化
- フラッシュ・リカバリ領域の場所および初期サイズの設定
- 通常のリストア・ポイントと保証付きリストア・ポイントの作成
- 最適なフラッシュバック・データベースのパフォーマンスのための環境の構成



## リストア・ポイントおよびフラッシュバック・データベース

**Oracle Flashback Database** およびリストア・ポイントは、関連データ保護機能です。これらの機能は、データベースの不要な変更を無効にする場合に **Point-in-Time** リカバリのかわりに使用できるより効率的な機能です。フラッシュバック・データベースを使用すると、データベース全体の時間を巻き戻し、時間枠内でデータベース変更の結果を無効にできます。この機能は、**データベースの Point-in-Time リカバリ** に類似しています。

リストア・ポイントには、フラッシュバック・データベースおよびその他のメディア・リカバリ操作に関連する機能が備えられています。特に、任意の SCN で作成された、保証付きリストア・ポイントによって、フラッシュバック・データベースを使用してデータベースをこの SCN に巻き戻すことができます。リストア・ポイントおよびフラッシュバック・データベースは、別々またはともに使用することができます。

### フラッシュバック・データベース

Oracle Flashback Database には、Recovery Manager コマンド **FLASHBACK DATABASE** または SQL 文 **FLASHBACK DATABASE** を使用してアクセスできます。いずれかのコマンドを使用して、論理データの破損やユーザー・エラーから、データベースを迅速にリカバリできます。

フラッシュバック・データベースは、従来の **Point-in-Time** リカバリと類似しています。フラッシュバック・データベースを使用すると、データベースを直前の状態に戻すことができます。ただし、フラッシュバック・データベースは、**Point-in-Time** リカバリよりはるかに高速です。フラッシュバック・データベースでは、バックアップからデータファイルをリストアする必要がなく、アーカイブ REDO ログから変更をほとんど適用する必要がないためです。

データファイルが影響を受けていないかぎり、フラッシュバック・データベースを使用して、データベースに対する最も不要な変更を無効にすることができます。データベースを前の **インカネーション** の状態に戻すこともできます。つまり、ALTER DATABASE OPEN RESETLOGS 文による影響を元に戻すことができます。FLASHBACK DATABASE コマンドを使用して、データベースの変更を無効にする方法については、16-11 ページの「**フラッシュバック・データベースを使用したデータベースの巻戻し**」を参照してください。

フラッシュバック・データベースでは、独自のロギング・メカニズムによって **フラッシュバック・ログ** が作成され、**フラッシュ・リカバリ領域** に格納されます。フラッシュバック・データベースは、フラッシュバック・ログが使用可能な場合にのみ使用できます。したがって、フラッシュバック・データベースを使用するには、フラッシュバック・ログが作成されるようにデータベースを事前に設定しておく必要があります。

フラッシュバック・データベースを有効にするには、フラッシュ・リカバリ領域を構成し、**フラッシュバック保存目標** を設定します。この保存目標によって、フラッシュバック・データベースを使用してデータベースを巻き戻す時点が指定されます。この目標時点以降、データベースでは、データファイル内の変更されたすべてのブロックのイメージがフラッシュバック・ログに定期的にコピーされます。

フラッシュバック・データベースを使用してデータベースを過去の目標時点に巻き戻すと、コマンドによって、目標時点以降に変更されたブロックが判別され、フラッシュバック・ログからリストアされます。データベースでは、目標時点の直前の各ブロックのバージョンがリストアされます。次に、データベースでは、REDO ログを使用して、これらのブロックがフラッシュバック・ログに書き込まれた後に行われた変更が再適用されます。

ディスクまたはテープ上の REDO ログは、フラッシュバック・ログに記録されている期間中、使用可能である必要があります。たとえば、フラッシュバック保存目標が 1 週間の場合は、過去 1 週間のすべての変更が含まれているオンライン REDO ログおよびアーカイブ REDO ログに、アクセスできるようにしておく必要があります。実際、**Point-in-Time** リカバリをサポートするには、通常、REDO ログはフラッシュバック保存目標より長い期間必要とされることに注意してください。

**フラッシュバック・データベース・ウィンドウ** FLASHBACK DATABASE コマンドをサポートするための十分なフラッシュバック・ログ・データが現在存在する SCN の範囲を **フラッシュバック・データベース・ウィンドウ** と呼びます。フラッシュバック・データベース・ウィンドウは、使用可能なフラッシュバック・ログの最も古い SCN より前に延長することはできません。

---

**注意：** 表領域の削除やデータファイルの縮小などの一部のデータベース操作は、フラッシュバック・データベースで無効にできません。この場合、フラッシュバック・データベース・ウィンドウは、行った操作の直後から始まります。

---

フラッシュバック・ログは、フラッシュ・リカバリ領域外の場合にはバックアップできません。フラッシュバック・データベース・ウィンドウを満たすための十分なフラッシュバック・ログが保存される可能性を高くするために、フラッシュ・リカバリ領域を増加することができます (5-19 ページの「[フラッシュ・リカバリ領域の場所および初期サイズの設定](#)」を参照)。

フラッシュ・リカバリ領域が、保存方針に必要なフラッシュバック・ログおよびアーカイブ REDO ログや他のバックアップなどのファイルを保持できるサイズでない場合、データベースでは、最も古い SCN からフラッシュバック・ログを削除して他のファイルに領域を確保することができます。この結果、フラッシュバック・データベース・ウィンドウは、フラッシュ・リカバリ領域のサイズ、保存する必要がある他のバックアップおよび必要なフラッシュバック・ロギング・データの量に応じてフラッシュバック保存目標より短くなる可能性があります。フラッシュバック保存目標は目標であり、フラッシュバック・データベースが使用可能になることを保証するものではありません。

フラッシュバック・データベース・ウィンドウの長さが十分でないため FLASHBACK DATABASE を使用できないときは、ほとんどの場合、同様の目的でデータベースの Point-in-Time リカバリ (DBPITR) を使用できます。フラッシュバック・データベースを使用して特定の時点に戻すか、またはフラッシュバック・ウィンドウのサイズを確保できるようにする場合は、保証付きリストア・ポイントが唯一の方法となります。

**参照：**

- フラッシュバック・データベースについては、16-11 ページの「[フラッシュバック・データベースを使用したデータベースの巻戻し](#)」を参照してください。
- DBPITR については、16-15 ページの「[データベースの Point-in-Time リカバリの実行](#)」を参照してください。
- 保証付きリストア・ポイントおよびフラッシュバック・データベースについては、5-31 ページの「[保証付きリストア・ポイント](#)」を参照してください。

## 通常のリストア・ポイント

**通常のリストア・ポイント**では、リストア・ポイント名が SCN または特定の時点に割り当てられます。これによって、リストア・ポイントはこの SCN のブックマークまたは別名として機能します。通常のリストア・ポイントは、無効にする必要がある操作を実行する前に作成できます。制御ファイルに、リストア・ポイント名および SCN が格納されます。

フラッシュバック機能または Point-in-Time リカバリを使用する必要がある場合は、時刻または SCN のかわりに、リストア・ポイント名を使用します。次のコマンドでは、リストア・ポイントをこの方法で使用できます。

- Recovery Manager の RECOVER DATABASE および FLASHBACK DATABASE コマンド
- SQL の FLASHBACK TABLE 文

通常のリストア・ポイントでは、事前に手動で SCN を記録したり、問題が発生した後でフラッシュバック問合せなどの機能を使用して適切な SCN を判別する必要がありません。

通常のリストア・ポイントは軽量です。制御ファイルには、データベースのパフォーマンスに大きな影響を与えることなく、数千もの通常のリストア・ポイントの記録を保持できます。通常のリストア・ポイントは、手動で削除しない場合でも、最終的には制御ファイルからエージアウトされるため、継続的なメンテナンスは不要です。

**参照：** フラッシュバック問合せの使用方法については、『Oracle Database アドバンスド・アプリケーション開発者ガイド』を参照してください。

## 保証付きリストア・ポイント

通常のリストア・ポイントと同様に、**保証付きリストア・ポイント**は、リカバリ操作で SCN の別名として機能します。保証付きリストア・ポイントは、制御ファイルからエージ・アウトされず、明示的に削除する必要がある点が主な違いとなります。通常、保証付きリストア・ポイントは、通常のリストア・ポイントで動作するすべてのコマンドで、SCN の別名として使用できます。特に明記されていないかぎり、通常のリストア・ポイントの使用箇所および使用方法に関する情報は、保証付きリストア・ポイントにも適用されます。

保証付きリストア・ポイントによって、**フラッシュバック・ログ**が無効になっている場合でも、フラッシュバック・データベースを使用してデータベースをリストア・ポイント SCN の状態に巻き戻すことができます。フラッシュバック・ロギングが有効になっている場合は、保証付きリストア・ポイントによって、最も古い保証付きリストア・ポイント以降の任意の SCN まで、フラッシュバック・データベースに必要なフラッシュバック・ログが保存されます。したがって、フラッシュバック・ロギングが有効になっている場合は、データベースを 1 つの SCN にのみ巻き戻すのではなく、連続したいずれかの SCN に巻き戻すことができます。

---

**注意：** フラッシュバック・ロギングが無効になっている場合は、FLASHBACK DATABASE を使用して、保証付きリストア・ポイントと現時点の間にある SCN にデータベースを巻き戻すことはできません。この場合、データベースを中間の SCN に戻す方法は、DBPITR を使用する方法のみです。

---

必要なログを格納できる十分なディスク領域がリカバリ領域にある場合は、保証付きリストア・ポイントを使用して、数日前または数週間前の良好な状態にデータベース全体を巻き戻すことができます。フラッシュバック・データベースの場合と同様に、ダイレクト・ロード・インサートなどの NOLOGGING 操作の影響も、保証付きリストア・ポイントを使用して無効にすることができます。

---

**注意：** フラッシュバック・データベースに適用される制限事項は、保証付きリストア・ポイントにも適用されます。たとえば、データファイルの縮小または表領域の削除を行うと、その影響を受けるデータファイルは、保証付きリストア・ポイントまでフラッシュバックされません。コマンドの前提条件および使用方法の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』の FLASHBACK DATABASE エントリに関する項を参照してください。

---

**保証付きリストア・ポイントおよびストレージ・スナップショット** 実際、保証付きリストア・ポイントは、ストレージ・スナップショットにかわる有効な機能です。多くの場合、ストレージ・スナップショットは、大規模なデータベースの更新、アプリケーションのバッチの適用やアップグレードなどの危険を伴う操作の前に、データベースを保護するために使用します。操作のテスト用にスナップショットまたは**複製データベース**を作成するかわりに、プライマリ・データベースまたは**フィジカル・スタンバイ・データベース**に保証付きリストア・ポイントを作成できます。これによって、必要なフラッシュバック・ログが確実に保存される状態で、危険を伴う操作を実行できます。

## フラッシュバック・データベースおよび保証付きリストア・ポイントのロギング

フラッシュバック・データベースおよび保証付きリストア・ポイントのロギングでは、変更が適用される前のデータファイル・ブロックのイメージ取得する必要があります。FLASHBACK DATABASE コマンドでは、これらのイメージを使用して、データファイルを前の状態に戻すことができます。

通常のフラッシュバックのロギングと保証付きリストア・ポイントのロギングの主な違いは、ブロックをロギングするタイミング、およびフラッシュ・リカバリ領域で領域圧迫に応じてログを削除できるかどうかに関連しています。これらの違いは、ログに対する領域の使用状況およびデータベースのパフォーマンスに影響します。

リカバリ可能目標によって、フラッシュバック・データベースのロギングを有効にするかどうか、または保証付きリストア・ポイントを使用するかどうか、あるいはその両方を行うかが部分的に決定されます。これらの機能を個別に使用した場合および同時に使用した場合のパフォーマンスおよび領域の使用状況への影響も、決定を行う場合の要因として考慮する必要があります。

**保証付きリストア・ポイントおよびフラッシュ・リカバリ領域の領域使用状況** 次の規則によって、フラッシュ・リカバリ領域でのフラッシュバック・ログの作成、保存、上書きおよび削除が制御されます。

- フラッシュバック・ログは、フラッシュ・リカバリ領域に十分な領域がある場合、フラッシュバック保存目標の達成に必要なになると常に作成されます。
- フラッシュバック・ログは、フラッシュバック保存目標の達成に不要になるほど古くなると、再利用されます。
- データベースでフラッシュバック・ログを作成する必要がある場合に、フラッシュ・リカバリ領域が一杯になるか、またはディスク領域がなくなると、かわりに最も古いフラッシュバック・ログが再利用されます。

---

**注意：** 最も古いフラッシュバック・ログを再利用すると、フラッシュバック・データベース・ウィンドウは短くなります。ディスク領域の不足が原因で、多くのフラッシュバック・ログが再利用されると、フラッシュバック保存目標が達成されない場合があります。

---

- フラッシュ・リカバリ領域が一杯になると、他のファイル用の領域を確保するために、バックアップ保存方針を満たすのに必要とされないアーカイブ REDO ログがフラッシュ・リカバリ領域によって自動的に削除される場合があります。この場合、FLASHBACK DATABASE を使用するために、その REDO ログ・ファイルを使用する必要があるフラッシュバック・ログも削除されます。
- フラッシュ・リカバリ領域内のファイルは、保証付きリストア・ポイントを満たすために必要な場合は削除対象になりません。したがって、バックアップ保存方針を満たすために必要なファイルに加えて、保証付きリストア・ポイントを満たすために必要なフラッシュバック・ログなどのファイルを保存すると、フラッシュ・リカバリ領域が完全に一杯になる場合があります。フラッシュ・リカバリ領域が一杯になった場合は、「[フラッシュ・リカバリ領域が一杯になった場合の対応](#)」を参照してください。

保証付きリストア・ポイントを作成する場合は、フラッシュバック・データベースの完全なロギングを有効にしているかどうかに関係なく、フラッシュ・リカバリ領域で使用可能な領域を監視する必要があります。フラッシュ・リカバリ領域のディスク領域の使用状況を監視する方法については、11-8 ページの「[フラッシュ・リカバリ領域でのフラッシュバック・ログの領域の管理](#)」を参照してください。

---

**注意：** 保存方針の要件および保証付きリストア・ポイントのため、フラッシュ・リカバリ領域からの削除対象となるファイルがない場合、データベースはディスクが一杯になった場合と同様に動作します。多くの場合、これによってデータベースは停止します。詳細は、11-8 ページの「[フラッシュ・リカバリ領域が一杯になった場合の対応](#)」を参照してください。

---

**フラッシュバック・ロギングが無効になっている状態での保証付きリストア・ポイントの**

**ロギング** フラッシュバック・データベースのロギングが無効になっているときに保証付きリストア・ポイントを作成するとします。この場合、保証付きリストア・ポイントの時点以降にデータファイル・ブロックが初めて変更されると、データベースによって、変更前のブロックのイメージがフラッシュバック・ログに格納されます。このため、フラッシュバック・ログには、保証付きリストア・ポイントが作成された時点で変更されているすべてのデータ・ブロックの内容が保持されます。同じブロックに対するこれ以降の変更では、そのブロックが最後に変更されてから別の保証付きリストア・ポイントが作成されないかぎり、その内容が再度ロギングされることはありません。

このロギング方法には、次の重要な効果があります。

- FLASHBACK DATABASE を使用すると、ブロック・イメージによって、保証付きリストア・ポイントの時点でのデータファイルの内容を再作成できます。
- 同じデータを繰り返し変更するワークロードでは、ディスク領域の使用量が、通常のフラッシュバック・ロギングより少なくなる可能性があります。変更された各ブロックは 1 回のみロギングされるため、必要な領域が少なくなります。このディスク領域の節約は、大きいボリュームまたは大きいバッチが挿入されたアプリケーションでは大きなメリットはありませんが、小さいボリュームが挿入されたアプリケーションでは有効な場合があります。また、フラッシュバック・データベース・ロギングを行わずに保証付きリストア・ポイントをロギングした場合のパフォーマンスのオーバーヘッドも小さくなる可能性があります。

保証付きリストア・ポイントが作成された時点でデータベースを戻すことが第一の目標であるとしてします。この場合は、通常、フラッシュバック・ロギングを無効にして、保証付きリストア・ポイントのみを使用する方がより効率的です。たとえば、週末にデータベース・ホストに対してアプリケーション・アップグレードを実行するとします。アップグレードの開始時に、保証付きリストア・ポイントを作成します。アップグレードに失敗した場合は、FLASHBACK DATABASE を使用して変更を無効にします。

**保証付きリストア・ポイントが定義された状態でのフラッシュバック・データベースの**

**ロギング** フラッシュバック・データベースを有効にし、1 つ以上の保証付きリストア・ポイントを定義すると、データベースで通常のフラッシュバック・ロギングが実行されます。この場合、現時点と現在定義されている最も古い保証付きリストア・ポイントの間の任意の時点にフラッシュバックするために必要なフラッシュバック・ログが、リカバリ領域に保存されます。フラッシュバック・ログは、保証を実現するために必要な場合、領域圧迫に応じて削除されることはありません。

フラッシュバック・ロギングでは、パフォーマンスのオーバーヘッドが発生します。また、データベースでのアクティビティのパターンに応じて、フラッシュ・リカバリ領域で著しい領域圧迫が発生する場合があります。したがって、フラッシュ・リカバリ領域で使用される領域を監視する必要があります。

## フラッシュバック・データベースおよび保証付きリストア・ポイントの前提条件

次に、フラッシュバック・データベースを有効にする前提条件を示します。

- Oracle Flashback Database の操作ではアーカイブ・ログが使用されるため、データベースを ARCHIVELOG モードで実行している必要があります。
- フラッシュバック・ログはフラッシュ・リカバリ領域にのみ格納可能なため、フラッシュ・リカバリ領域を有効にする必要があります。
- Real Application Clusters データベースの場合、フラッシュ・リカバリ領域は、クラスタ化されたファイル・システムまたは ASM に格納する必要があります。

保証付きリストア・ポイントを使用できるようにするには、データベースが次の前提条件を満たしている必要があります。

- COMPATIBLE 初期化パラメータを 10.2 以上に設定する必要があります。



- データベースは ARCHIVELOG モードで実行されている必要があります。  
保証付きリストア・ポイントまでデータベースを巻き戻すには、FLASHBACK DATABASE コマンドで、リストア・ポイントの時点の前後から始まるアーカイブ REDO ログを使用する必要があります。
- フラッシュ・リカバリ領域を構成する必要があります (5-14 ページの「[フラッシュ・リカバリ領域の構成](#)」を参照)。  
保証付きリストア・ポイントでは、フラッシュバック・ロギングに類似したメカニズムが使用されます。フラッシュバック・ロギングの場合と同様に、Oracle Database によって、必要なログがフラッシュ・リカバリ領域に格納されている必要があります。
- フラッシュバック・データベースが有効になっていない場合 (または以前に作成したすべての保証付きリストア・ポイントが削除されている場合) は、保証付きリストア・ポイントを初めて作成する際に、データベースがマウントされているが、オープンされていない必要があります

---

**注意:** 通常のリストア・ポイントを使用するための特別な要件はありません。

---

## フラッシュバック・データベースの有効化

Oracle Flashback Database のロギングを有効にするには、DB\_FLASHBACK\_RETENTION\_TARGET 初期化パラメータを設定し、ALTER DATABASE FLASHBACK ON 文を発行します。

### フラッシュバック・ロギングを有効にする手順

1. SQL\*Plus を起動し、データベースがマウントされているが、オープンされていないことを確認します。たとえば、次のように入力します。

```
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
```

2. オプションで、DB\_FLASHBACK\_RETENTION\_TARGET に、フラッシュバックの期間の長さを分単位で設定します。

```
ALTER SYSTEM SET DB_FLASHBACK_RETENTION_TARGET=4320; # 3 days
```

デフォルトでは、DB\_FLASHBACK\_RETENTION\_TARGET は 1 日 (1440 分) に設定されます。

3. データベース全体でフラッシュバック・データベース機能を有効にします。

```
ALTER DATABASE FLASHBACK ON;
```

4. 必要に応じて、特定の表領域のフラッシュバック・ロギングを無効にします。

デフォルトでは、すべての永続表領域に対してフラッシュバック・ログが生成されます。必要に応じて、特定の表領域のフラッシュバック・ロギングを無効にして、オーバーヘッドを軽減できます。次に例を示します。

```
ALTER TABLESPACE tbs_3 FLASHBACK OFF;
```

表領域のフラッシュバック・ロギングは、次のコマンドを使用して、後で再度有効にできます。

```
ALTER TABLESPACE tbs_3 FLASHBACK ON;
```

表領域に対して Oracle Flashback Database を無効にする場合は、FLASHBACK DATABASE を実行する前に、データファイルをオフラインにする必要があります。

次のコマンドを使用すると、データベース全体のフラッシュバック・ロギングを無効にできません。

```
ALTER DATABASE FLASHBACK OFF;
```

**フィジカル・スタンバイ・データベース**でフラッシュバック・データベースを有効にすると、スタンバイ・データベースをフラッシュバックできます。スタンバイ・データベースのフラッシュバック・データベースには、Data Guard 環境で使用可能な多くのアプリケーションがあります。詳細は、『Oracle Data Guard 概要および管理』を参照してください。

## 通常のリストア・ポイントと保証付きリストア・ポイントの作成

通常のリストア・ポイントまたは保証付きリストア・ポイントを作成するには、リストア・ポイントの名前を指定し、保証付きリストア・ポイントであるか通常のリストア・ポイント（デフォルト）であるかを指定して、SQL 文 CREATE RESTORE POINT を使用します。

### リストア・ポイントを作成する手順

1. SQL\*Plus をターゲット・データベースに接続します。
2. データベースがオープンまたはマウントされていることを確認します。データベースがマウントされている場合、(フィジカル・スタンバイ・データベースでないかぎり) 正しく停止されている必要があります。
3. CREATE RESTORE POINT 文を実行します。

次に、SQL\*Plus で通常のリストア・ポイントを作成する方法の例を示します。

```
SQL> CREATE RESTORE POINT before_upgrade;
```

次に、保証付きリストア・ポイントの作成方法の例を示します。

```
SQL> CREATE RESTORE POINT before_upgrade GUARANTEE FLASHBACK DATABASE;
```

### 参照:

- SQL の CREATE RESTORE POINT 文の詳細は、『Oracle Database SQL 言語リファレンス』を参照してください。
- リストア・ポイントを表示する方法については、10-10 ページの「[リストア・ポイントの表示](#)」を参照してください。
- リストア・ポイントを削除する方法については、11-9 ページの「[リストア・ポイントの削除](#)」を参照してください。

## 最適なフラッシュバック・データベースのパフォーマンスのための環境の構成

フラッシュバック・ログをメンテナンスすると、Oracle Database インスタンスで発生するオーバーヘッドが比較的制限されます。変更されたブロックがメモリーからフラッシュバック・ログに比較的低い頻度で定期的書き込まれ、プロセスおよび I/O オーバーヘッドが制限されます。

フラッシュバック・データベースが有効になっている大規模な本番データベースのパフォーマンスを向上させるには、次のガイドラインに従うことをお勧めします。

- フラッシュ・リカバリ領域には、可能な場合、オペレーティング・システムのファイル・キャッシュを使用せずに高速なファイル・システムを使用します。

通常、データベースによってフラッシュ・リカバリ領域に作成されるファイル（フラッシュバック・ログなど）は、サイズが大きくなります。オペレーティング・システムのファイル・キャッシュは、通常、これらのファイルに対しては有効ではなく、実際には、このファイル・キャッシュによって、これらのファイルに対する読取り / 書込みを行った場合の CPU オーバーヘッドが増加する場合があります。したがって、オペレーティング・システムのファイル・キャッシュが行われない ASM などのファイル・システムを使用することをお勧めします。

- フラッシュ・リカバリ領域を保持するファイル・システムに、十分なディスク・スピンドルを構成します。  
大規模な本番データベースでは、データベースでフラッシュバック・ログを効果的に書き込むために必要となるディスク・スループットをサポートするために、複数のディスク・スピンドルが必要な場合があります。
- フラッシュ・リカバリ領域の保持に使用されるストレージ・システムに非揮発性 RAM がない場合は、ストライプ化されたストレージ・ボリュームにファイル・システムを構成しません。  
比較的小さなストライプ・サイズ（128KB など）を使用します。この方法では、フラッシュバック・ログへのそれぞれの書込みを複数のスピンドルに分散することによって、パフォーマンスを向上できます。
- 大規模なデータベースでは、初期化パラメータ LOG\_BUFFER を 8MB 以上に設定します。  
この設定では、データベースによって、フラッシュバック・データベース・ログの書込みに最大メモリー（通常は 16MB）が割り当てられます。

フラッシュバック・データベースのロギングで発生するオーバーヘッドは、データベース・ワークロードでの読取りと書込みの組合せによって異なります。ワークロードが書込み集中型の場合、フラッシュバック・データベースのロギングを有効にするとオーバーヘッドが増加します。問合せによってデータは変更されないため、フラッシュバック・データベースのロギング・アクティビティには影響しません。

## Data Guard 環境での Recovery Manager の構成

Data Guard 環境で Recovery Manager を使用する場合、CONFIGURE コマンドを使用すると、この環境での物理データベースの設定の登録および構成を行うことができます。Recovery Manager は、DB\_UNIQUE\_NAME 初期化パラメータを使用して、データベースを別のデータベースと区別します。したがって、Data Guard 環境で DB\_UNIQUE\_NAME の一意性を維持することが重要です。

Data Guard 環境内のデータベースに構成を作成または変更する場合、Recovery Manager はリカバリ・カタログに接続している必要があります。SET DBID コマンドを使用して、**Recovery Manager セッション**で DBID を設定する場合は、Data Guard 環境で Recovery Manager が TARGET としてデータベースに接続されていなくても、スタンバイ・データベースを構成できます。この場合は、作成されていないスタンバイ・データベースに構成を作成することもできます。

次の形式の CONFIGURE コマンドを使用できます。

- CONFIGURE DB\_UNIQUE\_NAME を使用すると、**フィジカル・スタンバイ・データベース**への接続が定義され、新しいデータベースが暗黙的に登録されます。  
Recovery Manager が TARGET として初めてスタンバイ・データベースに接続する場合は、新しいスタンバイ・データベースも自動的に登録されます。
- CONFIGURE FOR DB\_UNIQUE\_NAME を使用すると、Data Guard 環境でのデータベースの設定が構成されます。  
たとえば、環境内の指定したデータベースまたはすべてのデータベースに対して、チャンネルやデフォルトのデバイスなどを構成できます。SHOW ALL FOR DB\_UNIQUE\_NAME を使用すると、特定のスタンバイ・データベースの構成を表示でき、SHOW ALL FOR DB\_UNIQUE\_NAME ALL を使用すると、認識されるすべてのデータベースの構成を表示できます。

Data Guard 環境には、Data Guard にのみ関連する多くの考慮事項があります。たとえば、アーカイブ・ログがスタンバイ・データベースに転送されたか、適用されたかに基づいて、**アーカイブ REDO ログの削除方針**を構成できます。



**参照：**

- スタンバイ・データベースで使用するために Recovery Manager 環境を構成する方法については、『Oracle Data Guard 概要および管理』を参照してください。
- CONFIGURE ARCHIVELOG DELETION POLICY コマンドの詳細およびアーカイブ・ログが削除対象となる条件については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。



---

## Recovery Manager 環境の構成 : 高度なトピック

この章では、設定および構成作業を実行する方法について説明します。この章の内容は、次のとおりです。

- 高度なチャンネル・オプションの構成
- 高度なバックアップ・オプションの構成
- 補助インスタンスのデータファイル名の構成
- スナップショット制御ファイルの場所の構成
- 共有サーバーで使用するための Recovery Manager の構成
- 消失書込みの検出の有効化

## 高度なチャンネル・オプションの構成

5-5 ページの「チャンネルの構成」ではチャンネルの構成の基本が説明されていますが、この項では、より高度なチャンネルのトピックについて説明します。この項の内容は、次のとおりです。

- チャンネル制御オプション
- 固有のチャンネル・パラメータの構成

**参照：** チャンネルの構成および割当ての概要については 3-4 ページの「Recovery Manager チャンネル」、CONFIGURE 構文については『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

### チャンネル制御オプション

チャンネルを手動で割り当てる場合も自動チャンネル割当てを使用する場合も、チャンネル・コマンドおよびオプションを使用して、チャンネルの動作を制御できます。表 6-1 に、チャンネルの動作を制御する方法の概要を示します。特に明記されていないかぎり、CONFIGURE CHANNEL と ALLOCATE CHANNEL コマンドの両方ですべてのチャンネル・パラメータがサポートされています。

表 6-1 チャンネル制御オプション

| チャンネル制御のタイプ                   | コマンド  |
|-------------------------------|---|
| I/O 帯域幅消費に対する制限               | バックアップの制限メカニズムとして機能する RATE チャンネル・パラメータを使用できます。  |
| バックアップ・セットおよびバックアップ・ピースに対する制限 | MAXPIECESIZE チャンネル・パラメータを使用して、バックアップ・ピースのサイズに制限を設定できます。また、BACKUP および CONFIGURE コマンドで MAXSETSIZE パラメータを使用して、バックアップ・セットのサイズの制限を設定することもできます。 |
| ベンダー固有の手順                     | PARMS チャンネル・パラメータを使用して、メディア・マネージャに関するベンダー固有の情報を指定できます。SEND コマンドを使用して、ベンダー固有のコマンドをメディア・マネージャに送信することもできます。                                  |
| バックアップおよびリストア操作でのチャンネルの平行化    | チャンネルは、CONFIGURE DEVICE TYPE ... PARALLELISM を使用して永続的に平行化するか、または複数の ALLOCATE CHANNEL コマンドを使用してジョブ・レベルで平行化することができます。                       |
| データベース・インスタンスの接続設定            | CONNECT チャンネル・パラメータを使用して、操作を実行するインスタンスを指定できます。  |

**参照：** ALLOCATE CHANNEL 構文については『Oracle Database バックアップおよびリカバリ・リファレンス』、CONFIGURE 構文については『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

### 固有のチャンネル・パラメータの構成

特定のタイプのすべてのチャンネルに適用されるパラメータの構成に加えて、特定のチャンネルに適用されるパラメータを構成することもできます。特定のチャンネルを構成するには、CONFIGURE CHANNEL *n* (*n* は 255 未満の正の整数) コマンドを実行します。

チャンネルに番号を手動で割り当てる場合、各チャンネルに 1 つ以上のチャンネル・オプション (MAXPIECESIZE、FORMAT など) を指定する必要があります。固有の番号を割り当てたチャンネルをバックアップで使用すると、構成済の汎用チャンネル設定ではなく、そのチャンネルの構成設定が使用されます。

各チャンネルに設定されたパラメータを個別に制御する必要がある場合は、固有のチャンネルを番号ごとに構成します。この方法は、次のような場合に実行する必要があります。

- 個々のノードがバックアップの完全なセットにアクセスできない Oracle Real Application Clusters (Oracle RAC) 構成を実行している場合。すべてのバックアップに 1 つ以上のチャンネルでアクセスできるように、ノード固有の接続文字列を使用して各チャンネルを構成する必要があります。
- 各チャンネルで異なる PARMS 設定が必要なメディア・マネージャを使用している場合。

**参照：** Oracle RAC 環境での Recovery Manager バックアップについては、『Oracle Real Application Clusters 管理およびデプロイメント・ガイド』を参照してください。

## 固有のチャンネルの構成の例

次の例では、ディスク・バックアップを 2 つのディスクに送信します。ディスク・チャンネルを次のように構成します。

```
CONFIGURE DEFAULT DEVICE TYPE TO disk;          # backup goes to disk
CONFIGURE DEVICE TYPE disk PARALLELISM 2;      # two channels used in parallel
CONFIGURE CHANNEL 1 DEVICE TYPE DISK FORMAT '/disk1/%U' # 1st channel to disk1
CONFIGURE CHANNEL 2 DEVICE TYPE DISK FORMAT '/disk2/%U' # 2nd channel to disk2
BACKUP DATABASE; # backup - first channel goes to disk1 and second to disk2
```

2 つのテープ・ドライブがあり、各テープ・ドライブで異なるテープ・メディア・ファミリのテープを使用するという別の場合を想定します。デフォルトの出力デバイスおよびデフォルトのテープ・チャンネルを次の例のように構成して、データベース・バックアップを平行化します。

### 例 6-1 テープ・デバイスのチャンネルの平行化の構成

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;          # backup goes to sbt
CONFIGURE DEVICE TYPE sbt PARALLELISM 2;      # two sbt channels allocated by default
# Configure channel 1 to pool named first_pool
CONFIGURE CHANNEL 1 DEVICE TYPE sbt
    PARMS 'ENV=(OB_MEDIA_FAMILY=first_pool)';
# configure channel 2 to pool named second_pool
CONFIGURE CHANNEL 2 DEVICE TYPE sbt
    PARMS 'ENV=(OB_MEDIA_FAMILY=second_pool)';
BACKUP DATABASE; # first stream goes to 'first_pool' and second to 'second_pool'
```

例 6-1 では、バックアップ・データは 2 つのテープ・デバイス間で分割されます。構成済の各チャンネルは、データの合計の約半分ずつをバックアップします。

## CONFIGURE CHANNEL と平行化設定の関係

PARALLELISM 設定は、固有に構成したチャンネルの数に制約を受けません。たとえば、20 の異なるテープ・デバイスにバックアップを実行する場合、20 の異なる SBT チャンネルを構成できません。各チャンネルには、手動で番号 (1 から 20 まで) を割り当て、個別の一連のチャンネル・オプションを設定します。このような場合、PARALLELISM にデバイス数 (この例では 20) 以下の任意の値を設定できます。

平行化・チャンネルには、常に、Recovery Manager によって 1 から順に PARALLELISM 設定の値までの番号が付けられます。たとえば、デフォルト・デバイスが SBT で、平行化が 3 に設定されている場合、チャンネルの名前は次のようになります。

```
ORA_SBT_TAPE_1
ORA_SBT_TAPE_2
ORA_SBT_TAPE_3
```

DEVICE TYPE sbt (シノニム sbt\_tape ではない) を構成した場合でも、Recovery Manager では常に ORA\_SBT\_TAPE\_n という名前が使用されます。また、チャンネルを個別に構成した場合はそのチャンネルを使用し、構成していない場合は汎用チャンネルを使用して、常に、PARALLELISM に指定したチャンネルの番号が割り当てられます。パラレル化設定より大きい数で特定のチャンネルを構成した場合、Recovery Manager ではこれらのチャンネルが使用されないことに注意してください。

**参照：** チャンネルの詳細は、3-4 ページの「Recovery Manager チャンネル」を参照してください。

## 高度なバックアップ・オプションの構成

バックアップを作成するように Recovery Manager を構成する方法の基本については、5-2 ページの「Recovery Manager バックアップの環境の構成」を参照してください。この項では、より高度な構成オプションについて説明します。この項の内容は、次のとおりです。

- バックアップ・セットの最大サイズの構成
- バックアップ・ピースの最大サイズの構成
- バックアップの多重化の構成
- 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
- バックアップ圧縮アルゴリズムの構成
- バックアップの暗号化の構成

## バックアップ・セットの最大サイズの構成

テープ・バックアップでは、複数のテープにまたがって**多重バックアップ・セット**を使用できます。つまり、バックアップ・セットの各データファイルにあるブロックは、複数のテープに書き込まれます。マルチボリュームのバックアップ・セットのいずれかのテープで障害が発生すると、1つのテープのみでなく、すべてのテープ上のデータが失われます。バックアップが**マルチセクション・バックアップ**でない場合、バックアップ・セットには、データファイルの一部ではなくデータファイル全体が常に含まれます。MAXSETSIZE を使用すると、各バックアップ・セットが複数のテープにまたがるのではなく、1つのテープに収まるように指定できます。

CONFIGURE MAXSETSIZE コマンドを使用すると、チャンネルに作成されるバックアップ・セットの最大サイズを制限できます。この CONFIGURE 設定は、チャンネルが手動で割り当てられたか構成されたかにかかわらず、BACKUP コマンドを使用してバックアップ・セットを作成する際にすべてのチャンネルに適用されます。デフォルト値はバイトで指定され、KB の単位に切り捨てられます。

CONFIGURE MAXSETSIZE コマンドで設定した値は、指定したチャンネルのデフォルトになります。構成済の MAXSETSIZE 値は、個別の BACKUP コマンドに MAXSETSIZE オプションを指定して上書きできます。

Recovery Manager プロンプトで次のコマンドを発行するとします。

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;  
CONFIGURE CHANNEL DEVICE TYPE sbt PARMS 'ENV=(OB_MEDIA_FAMILY=first_pool)';  
CONFIGURE MAXSETSIZE TO 7500K;  
BACKUP TABLESPACE users;  
BACKUP TABLESPACE tools MAXSETSIZE 5G;
```

結果は次のようになります。

- users 表領域のバックアップには、構成済チャンネル **SBT** および構成済の MAXSETSIZE のデフォルト設定である 7500K が使用されます。
- tools 表領域のバックアップには、BACKUP コマンドの MAXSETSIZE に設定した 5G が使用されます。

**参照：**

- 「[BACKUP ... MAXSETSIZE を使用したバックアップ・セットのサイズの制限](#)」 (9-2 ページ)
- BACKUP 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## バックアップ・ピースの最大サイズの構成

バックアップ・ピースのサイズがファイル・システムまたはメディア管理ソフトウェアで許容される最大ファイル・サイズを超えると、問題が発生します。CONFIGURE CHANNEL または ALLOCATE CHANNEL コマンドの MAXPIECESIZE パラメータを使用すると、バックアップ・ピースのサイズを制限できます。

たとえば、バックアップ・ピースのサイズを常に 2GB 以下に制限するには、自動 DISK チャネルを次のように構成して、BACKUP DATABASE を実行します。

```
CONFIGURE CHANNEL DEVICE TYPE DISK MAXPIECESIZE 2G;
BACKUP DATABASE;
```

---

**注意：** バージョン 2.0 の Media Management API では、メディア・マネージャに書込み可能なバックアップ・ピースの最大サイズを指定できるのはメディア管理ベンダーです。Recovery Manager は、MAXPIECESIZE の設定に関係なく、この制限を優先します。

---

**参照：** CONFIGURE CHANNEL ... MAXPIECESIZE コマンドについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## バックアップの多重化の構成

CONFIGURE ... BACKUP COPIES コマンドを使用すると、指定したファイル・タイプの指定したデバイス・タイプ上に作成する各バックアップ・ピースのコピー数を指定できます。このタイプのバックアップは、**多重バックアップ・セット**と呼ばれます。多重化を行うための CONFIGURE 設定は、バックアップ・セットへのデータファイル、制御ファイルおよびアーカイブ・ログのバックアップにのみ影響し、イメージ・コピーには影響しません。

---

**注意：** **制御ファイルの自動バックアップ**は、多重化されません。

---

Recovery Manager では、ディスクまたはテープにバックアップを多重化できますが、テープとディスクにバックアップを同時に多重化することはできません。テープへのバックアップ時に、コピーの数が、使用可能なテープ・デバイスの数を超えないようにしてください。多重化の構成の例を次に示します。

```
# Makes 2 disk copies of each datafile and control file backup set
# (autobackups excluded)
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;
```

BACKUP COPIES 構成をデフォルト値に戻すには、同じ CONFIGURE コマンドに CLEAR オプションを指定して実行します。次に例を示します。

```
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE sbt CLEAR;
```

デフォルトでは、各デバイス・タイプの CONFIGURE ... BACKUP COPIES が 1 に設定されています。

---

---

**注意：** 永続的なコピー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してください。

---

---

**参照：**

- バックアップの多重化の概要は、7-10 ページの「[Recovery Manager を使用したバックアップの複数のコピー](#)」を参照してください。
- 多重バックアップの作成方法については、9-7 ページの「[バックアップ・セットの多重化](#)」を参照してください。
- BACKUP 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- CONFIGURE 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- SET 構文については、『Oracle Database バックアップおよびリカバリ・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

次の場合のように、指定した表領域を通常のバックアップ・スケジュールから除外することもできます。

- 表領域の再構築が簡単であるため、毎日バックアップするより再構築の方がコスト効率がよい場合。
- 表領域に、バックアップする必要がない一時データまたはテスト・データが含まれている場合。
- 表領域が頻繁に変更されないため、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必要がある場合。

CONFIGURE EXCLUDE FOR TABLESPACE を実行すると、指定した表領域を BACKUP DATABASE コマンドから除外できます。この除外条件は、これ以降にこの表領域に追加されるすべてのデータファイルに適用されます。

たとえば、テスト表領域 cwm\_lite および example をデータベース全体のバックアップから除外するには、次のコマンドを実行します。

```
CONFIGURE EXCLUDE FOR TABLESPACE cwm_lite;  
CONFIGURE EXCLUDE FOR TABLESPACE example;
```

次のコマンドを実行すると、cwm\_lite および example 以外のデータベース内のすべての表領域がバックアップされます。

```
BACKUP DATABASE;
```

除外されるように構成した表領域は、BACKUP コマンドでそれらの表領域を明示的に指定するか、または BACKUP DATABASE コマンドで NOEXCLUDE オプションを指定して、バックアップすることができます。たとえば、次のコマンドのいずれかを入力します。

```
# backs up the whole database, including cwm_lite and example  
BACKUP DATABASE NOEXCLUDE;  
BACKUP TABLESPACE cwm_lite, example; # backs up only cwm_lite and example
```

cwm\_lite および example に対する除外機能を無効にするには、次のコマンドを実行します。

```
CONFIGURE EXCLUDE FOR TABLESPACE cwm_lite CLEAR;  
CONFIGURE EXCLUDE FOR TABLESPACE example CLEAR;
```

これらの表領域は、これ以降に実行されるデータベース全体のバックアップでバックアップされます。



**参照：**

- BACKUP 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- CONFIGURE 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## バックアップ圧縮アルゴリズムの構成

Recovery Manager では、バックアップ・セットの**バイナリ圧縮**がサポートされています。サポートされているアルゴリズムは、BZIP2（デフォルト）および ZLIB です。BZIP2 アルゴリズムは最大の圧縮用に最適化され、ZLIB アルゴリズムは CPU の効率用に最適化されます。BZIP2 は、ZLIB より多くの CPU リソースを消費しますが、通常はより圧縮されたバックアップを作成します。ZLIB 圧縮の場合、Oracle Advanced Compression オプションが必要なため、COMPATIBLE 初期化パラメータを 11.0.0 以上に設定する必要があります。

次の構文で圧縮アルゴリズムを構成できます。ここで、*alg\_name* は、BZIP2 または ZLIB を指定するプレースホルダです。

**例 6-2 バックアップ圧縮アルゴリズムの構成**

```
CONFIGURE COMPRESSION ALGORITHM TO 'alg_name';
```

## バックアップの暗号化の構成

セキュリティを向上させるために、Recovery Manager バックアップ・セットに対して**バックアップの暗号化**を構成できます。暗号化バックアップは、不正なユーザーが取得しても読み取ることができません。この機能を使用するには、Enterprise Edition のデータベースが必要です。

### バックアップの暗号化

V\$RMAN\_ENCRYPTION\_ALGORITHMS ビューには、Recovery Manager でサポートされている暗号化アルゴリズムのリストが含まれています。暗号化アルゴリズムが指定されていない場合、デフォルトの暗号化アルゴリズムは 128 ビット Advanced Encryption Standard (AES) です。Recovery Manager の暗号化では、ターゲット・データベースで COMPATIBLE 初期化パラメータが 10.2.0 以上に設定されている必要があります。

Recovery Manager には、次の暗号化モードがあります。

- **バックアップの透過的暗号化**  
これがデフォルトのモードで、Oracle ウォレットが使用されます。ウォレットは、認証および署名資格証明（秘密鍵、証明書、SSL で必要な信頼できる証明書など）の格納に使用されるパスワード保護されたコンテナです。
- **バックアップのパスワード暗号化**  
このモードでは、パスワード保護のみが使用されます。暗号化バックアップを作成およびリストアする場合、パスワードを入力する必要があります。
- **バックアップのデュアル・モード暗号化**  
このモードでは、ウォレットまたはパスワードが必要です。

---

**注意：** ウォレットベースの暗号化は、パスワードが必要ないため、パスワードベースの暗号化より安全です。パスワードベースの暗号化は、バックアップをトランスポータブルにする必要があるため絶対に必要な場合にのみ使用してください。

---

暗号化バックアップは、必要な復号化キーが使用可能な場合にかぎり、リストアおよびリカバリ中に自動的に復号化されます。バックアップ・セットごとに別々のキーが取得されます。キーは、暗号化形式でバックアップ・ピースに格納されます。バックアップは、ユーザーが指定するパスワードまたは Oracle ウォレットによって取得されたキーを使用して復号化されます。

Recovery Manager を使用してディスクに暗号化バックアップを作成するには、データベースで Advanced Security Option を使用している必要があります。Oracle Secure Backup SBT は、暗号化 Recovery Manager バックアップをテープに直接作成するためにサポートされている唯一のインタフェースです。Oracle Secure Backup 以外の SBT ライブラリを使用して暗号化 Recovery Manager バックアップを作成しようとすると、Recovery Manager は ORA-19916 エラーを発行します。Advanced Security Option は、Oracle Secure Backup SBT を使用して暗号化バックアップを作成する場合は必要ありません。

暗号化バックアップ・セットで BACKUP BACKUPSET コマンドを使用すると、バックアップ・セットは暗号化形式でバックアップされます。BACKUP BACKUPSET ではすでに暗号化されたバックアップ・セットがディスクまたはテープにコピーされるのみのため、BACKUP BACKUPSET 中に復号化キーは必要とされません。操作中にデータが復号化されることはありません。BACKUP BACKUPSET コマンドを実行しても、バックアップ・セットを暗号化または復号化することはできません。

**参照：** Oracle ウォレットの構成の詳細は、『Oracle Database Advanced Security 管理者ガイド』を参照してください。

**バックアップの透過的暗号化** 透過的暗号化では、必要な Oracle キー管理インフラストラクチャが使用可能な場合にかぎり、DBA の介入なしで暗号化バックアップを作成およびリストアできます。透過的暗号化は、日次バックアップ操作（バックアップを作成元と同じデータベースにリストア）に最適です。透過的暗号化は、Recovery Manager の暗号化のデフォルトです。

透過的暗号化を使用する場合は、まず、『Oracle Database Advanced Security 管理者ガイド』の説明に従って、各データベースに Oracle ウォレットを構成する必要があります。バックアップの透過的暗号化では、暗号化形式および自動ログイン形式の Oracle ウォレットがサポートされています。Oracle ウォレットを使用する場合は、バックアップの暗号化を実行する前にウォレットがオープンされている必要があります。自動ログイン・ウォレットを使用する場合は、暗号化バックアップの操作をいつでも行うことができます。自動ログイン・ウォレットは常にオープンしているためです。

---

**注意：** 自動ログイン・ウォレットを使用する場合は、暗号化バックアップ・データとともにバックアップしないようにしてください。バックアップと自動ログイン・ウォレットの両方を取得すると、暗号化バックアップの読取りが可能になるためです。Oracle ウォレットはウォレット・パスワードがないと使用できないため、この形式のウォレットはバックアップを作成しても安全です。

---

Oracle ウォレットを構成した後は、DBA の介入なしで暗号化バックアップを作成およびリストアできます。透過的データ暗号化を使用して暗号化されているデータベース内のいくつかの列をバックアップの暗号化を使用してバックアップすると、バックアップ中にそれらの列に対して 2 度目の暗号化が行われます。バックアップ・セットがリストア中に復号化されると、暗号化された列は、元の暗号化された形式に戻ります。

Oracle キー管理インフラストラクチャによって以前のすべてのマスター・キーが Oracle ウォレットにアーカイブされるため、現行のデータベース・マスター・キーを変更または再設定しても、以前のマスター・キーを使用して実行された暗号化バックアップは引き続きリストアできます。データベース・マスター・キーはいつでも再設定できます。Recovery Manager は、このデータベースによって作成されたすべての暗号化バックアップを常にリストアできます。

---

**注意：** Oracle ウォレットを失うと、透過的に暗号化されたバックアップをリストアすることができなくなります。

---

**バックアップのパスワード暗号化** パスワード暗号化では、暗号化バックアップを作成およびリストアする場合に、DBA がパスワードを入力する必要があります。パスワード暗号化バックアップをリストアするには、バックアップを作成する場合に使用したパスワードと同じパスワードが必要となります。

パスワード暗号化は、リモートの場所でリストアし、送信中は保護されている必要があるバックアップに有効です。パスワード暗号化は、永続的には構成できません。パスワード暗号化を排他的に使用する場合は、Oracle ウォレットを構成する必要はありません。

---

**注意：** パスワード暗号化バックアップを暗号化するときに使用したパスワードを忘れた（または失った）場合は、バックアップをリストアできなくなります。

---

パスワード暗号化を使用するには、Recovery Manager スクリプトで `SET ENCRYPTION ON IDENTIFIED BY password ONLY` コマンドを使用します。

**バックアップのデュアル・モード暗号化** デュアル・モード暗号化バックアップでは、透過的なリストアまたはパスワードを指定したリストアのいずれかを実行できます。デュアル・モード暗号化バックアップは、通常は Oracle ウォレットを使用してオンサイトでリストアされるが、Oracle ウォレットを使用できないオフサイトでリストアする必要がある場合もあるバックアップを作成する場合に有効です。

デュアル・モード暗号化バックアップをリストアする場合は、Oracle ウォレットまたは復号化用のパスワードのいずれかを使用できます。

---

**注意：** デュアル・モード暗号化バックアップを暗号化するときに使用したパスワードを忘れ（または失い）、Oracle ウォレットも失った場合は、バックアップをリストアできなくなります。

---

デュアル・モード暗号化バックアップ・セットを作成するには、Recovery Manager スクリプトで `SET ENCRYPTION ON IDENTIFIED BY password` コマンドを指定する必要があります。

## Recovery Manager バックアップの暗号化モードの構成

`CONFIGURE` コマンドを使用すると、バックアップの透過的暗号化を永続的に構成できます。このコマンドを使用して、次の内容を指定できます。

- すべてのデータベース・ファイルのバックアップに透過的暗号化を使用するかどうか
- 特定の表領域のバックアップに透過的暗号化を使用するかどうか
- バックアップの暗号化にどのアルゴリズムを使用するか

`SET ENCRYPTION` コマンドを使用して、次の処理を実行することもできます。

- `CONFIGURE ENCRYPTION` コマンドで指定された暗号化設定を無効にします。たとえば、`SET ENCRYPTION OFF` を使用すると、暗号化バックアップを作成するようにデータベースが構成されている場合でも、暗号化されていないバックアップが作成されます。
- Recovery Manager クライアントが終了するまで保持される、バックアップの暗号化のパスワードを設定します。パスワードは機密情報であるため、Recovery Manager の複数のセッションにわたってパスワードが保持される構成は、Recovery Manager では許可されていません。

アーカイブ REDO ログのバックアップを暗号化するかどうかを制御する永続的な構成はありません。アーカイブ REDO ログ・ファイルを含むバックアップ・セットは、次のいずれかの条件が満たされている場合に暗号化されます。

- アーカイブ・ログのバックアップの作成時に `SET ENCRYPTION ON` が有効である。
- データベース全体または 1 つ以上の表領域のバックアップに対して暗号化が構成されている。

この動作によって、データファイルの暗号化バックアップに関連付けられている REDO も暗号化されます。

### すべての Recovery Manager バックアップが暗号化されるように環境を構成する手順

- 『Oracle Database Advanced Security 管理者ガイド』の説明に従って、暗号化ウォレットを設定します。
- 次の Recovery Manager コマンドを発行します。

```
CONFIGURE ENCRYPTION FOR DATABASE ON;
```

この段階では、デフォルトで、このデータベースによって作成されるすべての Recovery Manager バックアップ・セットで透過的暗号化が使用されます。

次のコマンドを使用すると、**Recovery Manager セッション**の永続的な暗号化構成を明示的に上書きできます。

```
SET ENCRYPTION ON;
```

暗号化設定は、Recovery Manager セッション中に SET ENCRYPTION OFF コマンドを発行するか、または次のコマンドを使用して永続的な設定を再度変更するまで有効です。

```
CONFIGURE ENCRYPTION FOR DATABASE OFF;
```

### バックアップ暗号化アルゴリズムの構成

CONFIGURE コマンドを使用すると、バックアップ・セットの書き込み時に、暗号化に使用するデフォルト・アルゴリズムを永続的に構成できます。指定可能な値は、V\$RMAN\_ENCRYPTION\_ALGORITHMS に表示されます。デフォルト・アルゴリズムは、128 ビット AES です。

### デフォルトのバックアップ暗号化アルゴリズムを構成する手順

- Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
- ターゲット・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。
- V\$RMAN\_ENCRYPTION\_ALGORITHMS.ALGORITHM\_NAME から有効な値を指定して、CONFIGURE ENCRYPTION ALGORITHM コマンドを実行します。

次の例では、アルゴリズムを 256 ビット AES 暗号化に構成します。

```
CONFIGURE ENCRYPTION ALGORITHM TO 'AES256';
```

## 補助インスタンスのデータファイル名の構成

**表領域の Point-in-Time リカバリ**を実行するか、または Recovery Manager を使用してデータ転送を実行するとします。この場合、TSPITR またはデータベースの複製を開始する前に、**補助インスタンス**のデータファイルの名前を設定する必要がある場合があります。補助インスタンスのデータファイル名を設定するには、次のコマンドを実行します。ここで、*datafileSpec* にはデータファイルの元の名前または番号を指定し、*filename* には指定したファイルの新しいパスを指定します。

```
CONFIGURE AUXNAME FOR datafileSpec TO 'filename';
```

たとえば、datafile 2 の新しい補助ファイル名を次のように構成するとします。

```
CONFIGURE AUXNAME FOR DATAFILE 2 TO '/newdisk/datafiles/df2.df';
```

この CONFIGURE コマンド設定は、他の設定と同様に、次の例に示すように CONFIGURE ... CLEAR を使用してクリアしないかぎり、Recovery Manager セッション間で永続的に適用されます。

```
CONFIGURE AUXNAME FOR DATAFILE 2 CLEAR;
```

TSPITR または DUPLICATE コマンドを実行する場合、CONFIGURE AUXNAME を使用すると、補助データベースで使用するファイル名を事前に構成でき、これらの操作の実行中に補助ファイル名を手動で指定する必要がなくなります。

DUPLICATE コマンドでファイルの名前を変更する場合は、SET NEWNAME コマンドのかわりに CONFIGURE AUXNAME を使用できます。AUXNAME は一度設定すると、その後 DUPLICATE コマンドを発行するときにファイル名を再設定する必要がない点で SET NEWNAME と異なります。AUXNAME 設定は、CONFIGURE AUXNAME ... CLEAR を発行しないかぎり適用されたままになります。一方、SET NEWNAME コマンドは、ファイル名を変更するたびに発行する必要があります。

TSPITR と CONFIGURE AUXNAME を併用する方法の詳細は第 20 章「[Recovery Manager の表領域の Point-in-Time リカバリ \(TSPITR\) の実行](#)」、データベース複製の実行時に CONFIGURE AUXNAME の使用方法の詳細は第 23 章「[データベースの複製](#)」を参照してください。

## スナップショット制御ファイルの場所の構成

Recovery Manager では、リカバリ・カタログを読み取り一貫性バージョンの制御ファイルと再同期化する必要がある場合、一時[スナップショット制御ファイル](#)が作成されます。リカバリ・カタログとの再同期化または現行の制御ファイルのバックアップを実行する場合、スナップショット制御ファイルが必要になります。

スナップショット制御ファイルのデフォルトの場所はプラットフォーム固有であり、各ターゲット・データベースの Oracle ホームによって異なります。たとえば、一部の Linux プラットフォームでのデフォルトのファイル名は、\$ORACLE\_HOME/dbs/snapcf\_@.f となります。ターゲット・データベースに対して[フラッシュ・リカバリ領域](#)が構成されている場合、スナップショット制御ファイルのデフォルトの場所はそのフラッシュ・リカバリ領域内ではありません。

## スナップショット制御ファイルの構成場所の表示

SHOW コマンドを実行すると、スナップショットの現行の場所を表示できます。次の例では、デフォルトのルールによって決定されたスナップショットの場所を表示します。

```
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を示します。

```
RMAN> SHOW SNAPSHOT CONTROLFILE NAME;  
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の場所の設定

CONFIGURE SNAPSHOT CONTROLFILE NAME TO '*filename*' コマンドを使用すると、スナップショット制御ファイルの名前を変更できます。これ以降に Recovery Manager によって作成されるスナップショット制御ファイルには、指定したファイル名が使用されます。

たとえば、Recovery Manager を起動して次のコマンドを入力するとします。

```
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/oracle/oradata/trgt/snap_trgt.ctl';
```

また、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。

スナップショット制御ファイルの場所をデフォルトにリセットするには、CONFIGURE SNAPSHOT CONTROLFILE LOCATION CLEAR コマンドを実行します。

### 参照：

- 「[リカバリ・カタログの再同期化](#)」 (12-22 ページ)
- Oracle RAC 構成でのスナップショット制御ファイルの処理については、『[Oracle Real Application Clusters 管理およびデプロイメント・ガイド](#)』を参照してください。

## 共有サーバーで使用するための Recovery Manager の構成

Recovery Manager は、共有サーバー・ディスパッチャを介しては、ターゲット・データベースに接続できません。Recovery Manager には専用サーバー・プロセスが必要です。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、Oracle Net 構成を変更し、Recovery Manager 接続専用のサーバー・プロセスを指定する必要があります。

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 を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・サービス名の接続文字列の CONNECT\_DATA 属性に (SERVER=DEDICATED) を含める必要があります。

Oracle Net 構成は、システムによって大幅に異なります。次に、多くの構成方法の一例を示します。この例では、共有サーバー・アーキテクチャを使用して、tnsnames.ora 内の次のサービス名がターゲット・データベースに接続すると想定しています。ここで、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) (port=1521))
    (CONNECT_DATA= (SERVICE_NAME=inst1) (SERVER=shared))
  )
```

### Recovery Manager を共有サーバーで使用する手順

1. tnsnames.ora ファイルに、共有されない SID に接続するネット・サービス名を作成します。たとえば、次のように入力します。

```
inst1_ded =
  (DESCRIPTION=
    (ADDRESS= (PROTOCOL=tcp) (HOST=inst1_host) (port=1521))
    (CONNECT_DATA= (SERVICE_NAME=inst1) (SERVER=dedicated))
  )
```

2. SQL\*Plus を起動し、共有サーバー・サービス名と専用サーバー・サービス名の両方を使用して接続することで、各セッションのモードを確定します。

たとえば、SYSDBA 権限で inst1\_ded に接続してから、次の SELECT 文を実行します (出力例も示します)。

```
SQL> SELECT SERVER
       2 FROM   V$SESSION
       3 WHERE  SID = (SELECT DISTINCT SID FROM V$MYSTAT);
```

```
SERVER
-----
DEDICATED
1 row selected.
```

共有サーバー・セッションに接続するには、SYSDBA 権限で inst1\_shs に接続してから、次の SELECT 文を実行します (出力例も示します)。

```
SQL> SELECT SERVER
       2 FROM   V$SESSION
       3 WHERE  SID = (SELECT DISTINCT SID FROM V$MYSTAT);
```

```
SERVER
-----
SHARED
1 row selected.
```



3. Recovery Manager を起動し、専用サービス名を使用してターゲット・データベースに接続します。必要に応じて、リカバリ・カタログに接続します。たとえば、次のように入力します。

```
% rman
RMAN> CONNECT TARGET SYS@inst1_ded

target database Password: password
connected to target database: INST1 (DBID=39525561)

RMAN> CONNECT CATALOG rman@catdb
```

**参照：** Oracle Net の接続文字列構文の詳細は、プラットフォーム固有の Oracle マニュアルおよび『Oracle Database Net Services リファレンス』を参照してください。

## 消失書込みの検出の有効化

データ・ブロックの**消失書込み**は、実際には永続ストレージで書込みが行われなかったにもかかわらず、I/O サブシステムでブロック書込みの完了が確認された場合に発生します。この後のブロック読取りでは、失効したデータ・ブロックが I/O サブシステムによって戻されます。このデータ・ブロックを使用してデータベースの他のブロックを更新すると、ブロックが破損する場合があります。

バッファ・キャッシュ・ブロック読取りがデータベースによって REDO ログに記録されるように、DB\_LOST\_WRITE\_PROTECT 初期化パラメータを TYPICAL または FULL に設定することができます。デフォルト設定は NONE です。このパラメータを TYPICAL に設定すると、読取り/書込み表領域のバッファ・キャッシュ読取りはインスタンスによって REDO ログに記録されますが、読取り専用表領域は記録されません。FULL に設定すると、読取り専用表領域の読取りもインスタンスによって記録されます。TYPICAL モードでのパフォーマンス・オーバーヘッドは約 5～10% です。Oracle RAC では、FULL モードでのオーバーヘッドが 20% まで増加する可能性があります。

消失書込みの検出は、Data Guard とともに使用すると最も効果的です。この場合、プライマリ・データベースとスタンバイ・データベースの両方に DB\_LOST\_WRITE\_PROTECT を設定します。スタンバイ・データベースは、管理リカバリ中に REDO を適用する際、対応するブロックを読み取って SCN を REDO ログ内の SCN と比較します。プライマリ・データベースのブロック SCN がスタンバイ・データベースのブロック SCN より小さい場合は、プライマリ・データベース上の消失書込みを検出し、外部エラー (ORA-752) をスローします。SCN が大きい場合は、スタンバイ・データベース上の消失書込みを検出し、内部エラー (ORA-600 [3020]) をスローします。いずれの場合も、スタンバイ・データベースは障害の理由をアラート・ログおよびトレース・ファイルに書き込みます。

消失書込みをプライマリ・データベースで修復するには、スタンバイ・データベースへのフェイルオーバーを開始する必要があります。消失書込みをスタンバイ・データベースで修復するには、スタンバイ・データベース全体を再作成するか、または影響を受けたファイルのみのバックアップをリストアする必要があります。

Data Guard を使用しない場合も、消失書込みの検出を有効にすると役に立ちます。この場合、消失書込みは、通常データベース操作中またはメディア・リカバリ中に発生する可能性があります。通常データベース操作中に発生した場合は、エラーを検出する決定的な方法はありません。表に一貫性がないなどの間接的な兆候を、消失書込みに明確に結び付けることはできません。ただし、消失書込みが発生した可能性がある時点以前に作成したバックアップを保持している場合は、そのバックアップを代替の場所にリストアしてリカバリできます。問題を診断するには、失効したブロック読取りの SCN までデータベースまたは表領域をリカバリします。これによって、消失書込みエラー (ORA-752) が生成されます。

メディア・リカバリ中に消失書込みエラーが発生した場合は、データベースを RESETLOGS オプションでオープンする対応のみが可能です。データベースは一貫性のある状態ですが、RESETLOGS SCN より後のすべてのデータは消失しています。データベース作成後に作成したバックアップを使用してリカバリする場合は、他の失効したブロックによってデータベースが破損していない保証はないことに注意してください。これは、リストアするバックアップが、消失書込みが発生した後に作成された可能性があるためです。消失書込みによってデータベースが破損していないことを保証するには、データベース作成からのメディア・リカバリを実行する必要があります。ただし、この方法は、ほとんどのデータベース環境で現実的ではありません。

**参照：**

- 消失書込みの検出および修復のためにスタンバイ・データベースを使用する方法については、『Oracle Data Guard 概要および管理』を参照してください。
- DB\_LOST\_WRITE\_PROTECT 初期化パラメータについては、『Oracle Database リファレンス』を参照してください。



# 第 III 部

---

## データのバックアップおよびアーカイブ

第 III 部では、Recovery Manager ユーティリティを使用して拡張バックアップおよびリカバリ操作を実行する方法について説明します。また、Recovery Manager のパフォーマンス・チューニングおよびトラブルシューティングについて説明します。

第 III 部に含まれる章は次のとおりです。

- [第 7 章「Recovery Manager バックアップの概要」](#)
- [第 8 章「データベースのバックアップ」](#)
- [第 9 章「データベースのバックアップ: 高度なトピック」](#)



---

## Recovery Manager バックアップの概要

この章では、すべてのタイプの Recovery Manager バックアップを作成するために理解しておく必要がある一般的な概念について説明します。この章の内容は、次のとおりです。

- Recovery Manager の一貫性バックアップおよび非一貫性バックアップ
- オンライン・バックアップおよびバックアップ・モード
- バックアップ・セット
- イメージ・コピー
- Recovery Manager を使用したバックアップの複数のコピー
- 制御ファイルおよびサーバー・パラメータ・ファイルの自動バックアップ
- 増分バックアップ
- バックアップの保存方針

## Recovery Manager の一貫性バックアップおよび非一貫性バックアップ

バックアップを作成する Recovery Manager コマンドは、BACKUP です。Recovery Manager の BACKUP コマンドでは、次のタイプのファイルのバックアップがサポートされています。

- データファイルおよび制御ファイル
- サーバー・パラメータ・ファイル
- アーカイブ REDO ログ
- Recovery Manager バックアップ

データベースは、ネットワーク構成ファイル、パスワード・ファイル、Oracle ホームの内容などの他のタイプのファイルに依存していますが、これらのファイルは Recovery Manager ではバックアップできません。同様に、Oracle の一部の機能（外部表など）には、データファイル、制御ファイルおよび REDO ログ以外のファイルが必要な場合があります。Recovery Manager では、これらのファイルをバックアップできません。リストにないファイルについては、Recovery Manager 以外の方法でバックアップします。

Recovery Manager で BACKUP コマンドを実行すると、出力は常に 1 つ以上のバックアップ・セットまたは 1 つ以上のイメージ・コピーになります。**バックアップ・セット**は Recovery Manager に固有の独自の形式ですが、**イメージ・コピー**はファイルのビットごとのコピーです。デフォルトでは、Recovery Manager はバックアップ・セットを作成します。

### 一貫性バックアップ

BACKUP コマンドを使用すると、データベースの一貫性バックアップおよび非一貫性バックアップを作成できます。**一貫性バックアップ**は、データベースが一貫性のある状態のときに実行します。SHUTDOWN NORMAL、SHUTDOWN IMMEDIATE または SHUTDOWN TRANSACTIONAL コマンドを使用して停止すると、データベースは一貫性のある状態になります。一貫性のある状態での停止では、すべての REDO がデータファイルに適用されたことが保証されます。データベースをマウントし、この時点でバックアップを作成すると、後でデータベース・バックアップをリストアし、メディア・リカバリを実行せずにオープンすることができます。

### 非一貫性バックアップ

一貫性のないデータベース・バックアップを**非一貫性バックアップ**といいます。インスタンスでの障害の発生後または SHUTDOWN ABORT コマンドの実行後に作成されたバックアップと同様に、データベースがオープンされているときに作成されたバックアップには一貫性がありません。非一貫性バックアップからデータベースをリストアする場合、Oracle では、データベースをオープンする前に、メディア・リカバリを実行し、REDO ログ内の保留中の更新情報を適用する必要があります。

---

**注意：** データベースが NOARCHIVELOG モードの場合、Recovery Manager で非一貫性バックアップを作成することはできません。NOARCHIVELOG モードのデータベースに対してユーザー管理のバックアップ方法を使用する場合、このデータベースの非一貫性バックアップは作成しないでください。

---

データベースが ARCHIVELOG モードで実行され、アーカイブ REDO ログおよびデータファイルをバックアップするかぎり、非一貫性バックアップを適切なバックアップおよびリカバリ計画の基礎にすることができます。非一貫性バックアップでは、データベースを完全に保護するバックアップを作成するためにデータベースを停止する必要がないため、優れた可用性が実現されます。

## オンライン・バックアップおよびバックアップ・モード

オンラインの表領域またはデータベースの**ユーザー管理バックアップ**を実行すると、データベース・ライターによるファイルの更新と同時に、オペレーティング・システム・ユーティリティによってデータファイルのバックアップが作成される場合があります。ユーティリティは、更新途中の状態のブロックを読み取ることができるため、バックアップ・メディアにコピーされるブロックの前半は更新されていても、後半には古いデータが含まれていることがあります。このタイプの**論理的な破損**は、**分裂ブロック**と呼ばれます。つまり、SCNに関して一貫性のないブロックです。このバックアップを後でリストアし、ブロックをリカバリする必要がある場合は、ブロックを使用できないため、リカバリは失敗します。

ユーザー管理のオンライン・バックアップを実行する場合は、BEGIN BACKUP 句を指定して ALTER DATABASE または ALTER TABLESPACE 文を使用して、データファイルを**バックアップ・モード**にする必要があります。表領域がバックアップ・モードに設定されている場合、データベースは、ブロックを変更する前に、ブロック全体の変更前のイメージを REDO ストリームに書き込みます。データベースは、ブロックに対する変更をオンライン REDO ログに記録もします。また、バックアップ・モードでは、ファイルがバックアップ・モードから削除されるまで、**データファイル・チェックポイント**がフリーズされます。Oracle Database では、サード・パーティのバックアップ・ツールでデータ・ブロックのコピーの前にファイル・ヘッダーがコピーされることを保証できないため、このようなセーフガードが実行されます。

ユーザー管理ツールとは異なり、Recovery Manager では、データ・ブロックの形式が認識されるため、追加のロギングまたはバックアップが必要ありません。Recovery Manager では、分裂ブロックがバックアップされないことが保証されています。Recovery Manager によるバックアップ中、データベース・サーバー・セッションは各データ・ブロックを読み取り、ブロック・ヘッダーとフッターを比較することによって、分裂していないかどうかを確認します。ブロックが分裂している場合、セッションはそのブロックを再度読み取ります。同じ分裂が検出された場合、ブロックは永続的に破損しているとみなされます。また、Recovery Manager では、ブロックが読み取られる順序が認識されているため、データファイルのヘッダーのチェックポイントをフリーズする必要はありません。このため、そのファイルに最適なチェックポイントを取得できます。

**参照：** Recovery Manager を使用しない場合にオンラインの表領域をバックアップする方法については、27-5 ページの「**オンラインの表領域およびデータファイルのユーザー管理バックアップの作成**」を参照してください。

## バックアップ・セット

Recovery Manager で BACKUP コマンドを実行すると、1 つ以上のバックアップ・セットまたはイメージ・コピーを作成できます。デフォルトでは、出力先がディスクの場合もメディア・マネージャの場合も、バックアップ・セットが作成されます。

この項の内容は、次のとおりです。

- [バックアップ・セットおよびバックアップ・ピース](#)
- [バックアップ・セットの圧縮](#)
- [バックアップ・セットの暗号化](#)
- [バックアップ・ピースのファイル名](#)
- [バックアップ・ピースの数およびサイズ](#)
- [バックアップ・セットの数およびサイズ](#)
- [多重バックアップ・セット](#)
- [プロキシ・コピー](#)

## バックアップ・セットおよびバックアップ・ピース

Recovery Manager は、バックアップ・データを**バックアップ・セット** (Recovery Manager バックアップの最小単位) と呼ばれる論理構造に格納できます。バックアップ・セットには、1 つ以上のデータファイル、アーカイブ REDO ログ、制御ファイルまたはサーバー・パラメータ・ファイルのデータが含まれます。バックアップ・セットの作成およびアクセスは、Recovery Manager によってのみ行われます。バックアップ・セットは、Recovery Manager によってテープ・ドライブやテープ・ライブラリなどのメディア・マネージャにバックアップを書き込むことができる唯一の形式です。

バックアップ・セットには、1 つ以上のバイナリ・ファイルが Recovery Manager 固有の形式で含まれます。このファイルは、**バックアップ・ピース** と呼ばれます。1 つのバックアップ・セットに、複数のデータファイルを含めることができます。たとえば、1 つのバックアップ・ピースで構成される 1 つのバックアップ・セットに、10 個のデータファイルをバックアップできます。この場合、Recovery Manager では、1 つのバックアップ・ピースが出力として作成されます。バックアップ・セットには、このバックアップ・ピースのみが含まれます。

BACKUP コマンドに SECTION SIZE パラメータを指定すると、Recovery Manager によって**マルチセクション・バックアップ**が作成されます。これは、複数のチャネルでパラレルに作成される単一の大規模なファイルのバックアップです。各チャネルによって、1 つのバックアップ・ピースが作成されます。各バックアップ・ピースには、バックアップされるファイルの**ファイル・セクション**が 1 つ含まれます。

マルチセクション・バックアップ以外のバックアップでは、Recovery Manager は、正常に完了したバックアップ・セットのみをリポジトリに記録します。部分バックアップ・セットのようなものではありません。これは、Recovery Manager メタデータに部分的なバックアップ・セットのレコードが含まれる可能性がある失敗したマルチセクション・バックアップとは異なります。後者の場合は、DELETE コマンドを使用して、部分的なバックアップ・セットを削除する必要があります。

---

**注意:** Recovery Manager は、部分的なバックアップをリストアおよびリカバリの候補とみなしません。

---

**参照:** データベースをバックアップする方法については、第 8 章「データベースのバックアップ」を参照してください。

## バックアップ・セットの圧縮

Recovery Manager は、データファイルをバックアップ・セットにバックアップする際に、**未使用ブロックの圧縮**を使用してデータファイル・ブロックをスキップできます。Recovery Manager は、一度も使用されていないブロックを常にスキップします。『Oracle Database バックアップおよびリカバリ・リファレンス』BACKUP AS BACKUPSET エントリに示されている特定の条件では、現在使用されていないブロックもスキップします。このため、通常、データファイルのバックアップ・セットはデータファイルのコピーよりサイズが小さく、書込みにかかる時間が短くなります。未使用ブロックの圧縮は、Recovery Manager でバックアップ・ピースにデータファイルを書き込む方法の基本であり、無効にはできません。

Recovery Manager では、バックアップ・セットの**バイナリ圧縮**もサポートされています。サポートされているアルゴリズムは、BZIP2 (デフォルト) および ZLIB です。BZIP2 アルゴリズムは最大の圧縮用に最適化され、ZLIB アルゴリズムは CPU の効率用に最適化されます。BZIP2 は、ZLIB より多くの CPU リソースを消費しますが、通常はより圧縮されたバックアップを作成します。ZLIB 圧縮の場合、Oracle Advanced Compression オプションが必要なため、COMPATIBLE 初期化パラメータを 11.0.0 以上に設定する必要があります。

Recovery Manager は、バックアップ・セットの内容を圧縮してからディスクに書き込みます。Recovery Manager 圧縮を使用すると、リカバリ中に特別な解凍手順を実行する必要がありません。

**UNDO のバックアップの最適化**では、Recovery Manager は、トランザクションがすでにコミットされているためバックアップのリカバリに不要となった UNDO を除外します。バックアップの最適化を有効または無効にすることができますが、UNDO のバックアップの最適化は組込み動作です。

**参照：**

- 「バックアップ圧縮アルゴリズムの構成」 (6-7 ページ)
- 「バックアップの最適化の概要」 (5-23 ページ)
- BACKUP AS BACKUPSET については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## バックアップ・セットの暗号化

Recovery Manager では、バックアップ・セットの**バックアップの暗号化**がサポートされています。ウォレットベースの透過的暗号化またはパスワードベースの暗号化（あるいはその両方）を使用できます。CONFIGURE ENCRYPTION コマンドを使用すると、永続的な透過的暗号化を構成できます。パスワードベースの暗号化を指定するには、**Recovery Manager セッション** レベルで SET ENCRYPTION コマンドを使用します。

---

**注意：** ウォレットベースの暗号化は、パスワードが必要ないため、パスワードベースの暗号化より安全です。パスワードベースの暗号化は、バックアップをトランスポートする必要があるため絶対に必要な場合にのみ使用してください。

---

Recovery Manager を使用してディスクに暗号化バックアップを作成するには、データベースで Advanced Security Option を使用している必要があります。**Oracle Secure Backup SBT** は、暗号化 Recovery Manager バックアップをテープに直接作成するためにサポートされている唯一のインタフェースです。Advanced Security Option は、Oracle Secure Backup SBT に暗号化バックアップを作成する場合は必要ありません。

**参照：**

- 「バックアップの暗号化の構成」 (6-7 ページ)
- 「Recovery Manager バックアップの暗号化」 (9-11 ページ)

## バックアップ・ピースのファイル名

FORMAT 句を使用して名前を指定することができます。FORMAT パラメータを指定しない場合、Recovery Manager は、%U 置換変数を含む一意のファイル名をデフォルトのバックアップ場所に自動的に生成します。%U によって生成される SBT バックアップ・ピース名の例は 12i1nk47\_1\_1 です。ディスク上のバックアップ・ピースの例は、次のとおりです。

```
/d1/orcva/TEST/backupset/2007_12_12/o1_mf_nnndf_TAG20071212T162825_2qy199jm_.bkp
```

FORMAT 句では、一意のファイル名を生成するために %U 以外の置換変数がサポートされています。たとえば、データベースの名前を生成するために %d、DBID 用に %I、タイムスタンプ用に %t などを使用できます。

最大 4 つの FORMAT パラメータを指定できます。複数の FORMAT パラメータを指定すると、複数のコピーを指定する場合にのみ Recovery Manager で複数の FORMAT パラメータを使用できます。BACKUP ... COPIES、SET BACKUP COPIES または CONFIGURE ... BACKUP COPIES コマンドを使用すると、複数のコピーを作成できます。

---

**注意：** メディア・マネージャを使用する場合の FORMAT の制限（名前のサイズ、ネーミング規則など）については、ベンダーのドキュメントを参照してください。

---

**参照：**

- 「[Recovery Manager バックアップのフォーマットの指定](#)」 (8-4 ページ)
- FORMAT 句およびその置換変数については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## バックアップ・ピースの数およびサイズ

デフォルトでは、バックアップ・セットは、1つのバックアップ・ピースで構成されます。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 コマンドの MAXPIECESIZE オプションを指定します。このオプションによって、バックアップ・ピースのサイズが指定したバイト数に制限されます。バックアップ・セットの合計サイズが指定したバックアップ・ピースのサイズを超えた場合、Recovery Manager は、バックアップ・セットの内容を保持する複数の物理ピースを作成します。

このオプションは、複数のテープにまたがるバックアップ・ピースを管理できないメディア・マネージャに使用できます。たとえば、最大容量が 10GB のテープに、80GB のデータを保持するバックアップ・セットを作成する必要がある場合、メディア・マネージャで使用するテープに格納できる 10GB のバックアップ・ピースを作成するように、Recovery Manager に指示する必要があります。この場合、バックアップ・セット・メディアは 8 つのテープから構成されず、SBT 2.0 をサポートしているメディア・マネージャの場合、サポートするバックアップ・ピースの最大サイズを示す値を Recovery Manager に戻すことができるため、Recovery Manager はその値をバックアップ・アクティビティの計画に使用できます。

BACKUP コマンドに SECTION SIZE パラメータを指定すると、Recovery Manager によって [マルチセクション・バックアップ](#) が作成されます。この場合、1つのバックアップ・セットに複数のバックアップ・ピースを含めることができます。各バックアップ・ピースには [ファイル・セクション](#) が含まれます。マルチセクション・バックアップの目的は、複数のチャンネルで大規模なファイルをパラレルでバックアップできるようにすることです。

**参照：**

- 「[バックアップ・ピースの最大サイズの構成](#)」 (6-5 ページ)
- ALLOCATE CHANNEL 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- CONFIGURE 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## バックアップ・セットの数およびサイズ

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できます。*backupSpec* 句を 1 つ指定するごとに、1 つ以上のバックアップ・セットが作成されません。

バックアップ・セットの合計数および合計サイズは、ほとんどの場合、Recovery Manager の内部アルゴリズムに基づいています。ただし、CONFIGURE または BACKUP コマンドの MAXSETSIZE パラメータを使用して、Recovery Manager の動作に影響を及ぼすことができます。このパラメータを使用してバックアップ・セットのサイズを制限することによって、セット内のファイル数を間接的に制限でき、場合によっては、Recovery Manager に追加のバックアップ・セットを作成させることができます。また、BACKUP ... FILESPERSET を指定して、各バックアップ・セット内のファイルの最大数を指定することもできます。

**参照：**

- 「[バックアップ・セットのサイズ](#)」 (9-2 ページ)
- Recovery Manager のバッファ管理については、[第 21 章「Recovery Manager のパフォーマンスのチューニング」](#)を参照してください。
- *backupSpec* 句の構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。



## 多重バックアップ・セット

バックアップ・セットの作成時に、Recovery Manager は、ディスクから複数のファイルを同時に読み取って、同じバックアップ・セットにそれらのブロックを書き込むことができます。たとえば、Recovery Manager は、2つのデータファイルから同時に読取りを行って、それらのデータファイルのブロックを単一のバックアップ・ピースに組み合わせることができます。複数のファイルからのブロックの組合せは、バックアップの**多重化**と呼ばれます。これに対して、イメージ・コピーは多重化されません。

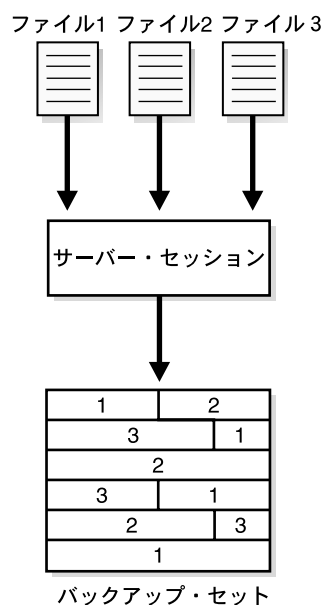
---

**注意：** Recovery Manager によってデータファイルの**マルチセクション・バックアップ**が作成された場合、そのデータファイルは他のすべてのデータファイルまたは**ファイル・セクション**とは多重化されません。

---

図 7-1 に示すように、Recovery Manager は、1つのバックアップ・ピースのみを含むバックアップ・セットに3つのデータファイルをバックアップできます。このバックアップ・ピースには、3つの入力データファイルのデータ・ブロックが組み合わされて格納されます。

図 7-1 データファイルの多重化



Recovery Manager による多重化は、複数の要因によって決定されます。たとえば、各バックアップ・セットに格納されるデータファイルの数は、BACKUP コマンドの FILESPERSET パラメータによって決定されます。Recovery Manager が同時に読み取ることができるデータファイルの数は、ALLOCATE CHANNEL または CONFIGURE CHANNEL の MAXOPENFILES パラメータによって定義されます。基本的な多重化アルゴリズムは、次のとおりです。

- 各バックアップ・セット内のファイルの数

この数は、FILESPERSET 設定および各チャンネルによって読み込まれるファイル数の最小値です。FILESPERSET のデフォルトは 64 です。

- **多重化レベル**

これは、同時に読み取られ、同じバックアップ・ピースに書き込まれる入力ファイルの数です。多重化のレベルは、MAXOPENFILES および各バックアップ・セット内のファイル数の最小値です。MAXOPENFILES のデフォルトは 8 です。

1つのチャンネルで12のデータファイルをバックアップするとします。各バックアップ・セット内のファイル数は4です。多重化のレベルは、この数と8の小さいほうになります。したがって、ブロックは、チャンネルによって4つのデータファイルから各バックアップ・ピースに同時に書き込まれます。

次に、1つのチャンネルで50のデータファイルをバックアップするとします。各バックアップ・セット内のファイル数は50です。多重化のレベルは、この数と8の小さいほうです。したがって、ブロックは、チャンネルによって8つのデータファイルから各バックアップ・ピースに同時に書き込まれます。

Recovery Managerによるバックアップ・セットの多重化は、**メディア・マネージャによる多重化**とは異なります。メディア・マネージャによる多重化のタイプの1つとして、メディア・マネージャが、複数のRecovery Managerチャンネルからの同時出力を単一のシーケンシャル・デバイスに書き込んだ場合に発生するものがあります。また、別のタイプとして、バックアップによって同じテープ上にデータベース・ファイルとデータベース以外のファイルが混在した場合に発生するものもあります。

---

**注意：** Recovery Manager バックアップには、メディア管理による多重化を使用しないことをお勧めします。

---

**参照：**

- バックアップ時に多重化がディスク・バッファの割当てに及ぼす影響については、21-4 ページの「**入力ディスク・バッファの割当て**」を参照してください。
- BACKUP 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## プロキシ・コピー

**プロキシ・コピー**では、Recovery Manager は、プロキシ・コピー機能をサポートしているメディア・マネージャに、データ送信の制御を移します。プロキシ・コピーはこの機能をサポートしているメディア・マネージャでのみ使用でき、タイプがDISKのチャンネルでは使用できません。BACKUP コマンドのPROXY オプションを使用して、バックアップでプロキシ・コピーを実行することを指定します。

BACKUP PROXY コマンドでバックアップするファイルごとに、Recovery Manager は、プロキシ・コピーを実行できるかどうかをメディア・マネージャに問い合わせて確認します。メディア・マネージャがファイルのプロキシ・コピーを実行できない場合、Recovery Manager は、PROXY オプションを使用しなかった場合と同様にファイルをバックアップします。(PROXY ONLY オプションを使用している場合に、プロキシ・コピーが実行できない場合は、Recovery Manager はバックアップを行いません。)

制御ファイルはプロキシ・コピーではバックアップされません。制御ファイルのバックアップ操作でPROXY オプションを指定した場合、このオプションは自動的に無視され、制御ファイルのバックアップが行われます。

**参照：**

- V\$PROXY\_DATAFILE および V\$PROXY\_ARCHIVEDLOG の詳細は、『Oracle Database リファレンス』を参照してください。
- BACKUP コマンドおよび PROXY オプションについては、『Oracle Database バックアップおよびリカバリ・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**は、1つのデータファイル、アーカイブ REDO ログ・ファイルまたは制御ファイルの正確なコピーです。イメージ・コピーはRecovery Manager 固有の形式では格納されません。これらは、オペレーティング・システム・コマンドでファイルをコピーした場合と同じ形式で格納されます。イメージ・コピーは、Recovery Manager のリストアおよびリカバリ操作中に使用できます。また、Recovery Manager 以外のリストアおよびリカバリ方法でもイメージ・コピーを使用できます。

## Recovery Manager で作成したイメージ・コピー

イメージ・コピーを作成して Recovery Manager リポジトリに記録するには、Recovery Manager の BACKUP AS COPY コマンドを実行します。また、ディスクのデフォルトのバックアップ・タイプをイメージ・コピーとして構成することもできます。コピーの作成には、データベース・サーバー・セッションが使用されます。このサーバー・セッションが、ファイル内のブロックの検証や Recovery Manager リポジトリへのイメージ・コピーの記録などの処理を実行します。

バックアップ・ピースと同様に、FORMAT 変数は、イメージ・コピーの名前の指定にも使用されます。イメージ・コピーの場合、デフォルトの書式 %U (7-5 ページの「バックアップ・ピースのファイル名」を参照) の定義が異なります。次に、%U によって生成されるデータファイル 2 の名前の例を示します。

```
/dl/oracle/work/orcva/RDBMS/datafile/o1_mf_sysaux_2qylnqm3_.dbf
```

イメージ・コピーを作成する場合は、BACKUP コマンドの DB\_FILE\_NAME\_CONVERT パラメータで、出力コピーに名前を指定することもできます。このパラメータは、初期化パラメータ DB\_FILE\_NAME\_CONVERT と同様に機能します。ファイル名の接頭辞のペアを指定すると、出力ファイルの名前が変更されます。ファイルがペアのいずれによっても変換されない場合、FORMAT 仕様で使用されます。FORMAT が指定されていない場合、デフォルトの書式 %U が使用されます。

例 7-1 では、ファイル名の先頭に /maindisk/oradata/users が付くデータファイルを、先頭に /backups/users\_ts が付くファイル名に変更してコピーします。

### 例 7-1 DB\_FILE\_NAME\_CONVERT を使用したファイル名の指定

```
BACKUP AS COPY
  TABLESPACE users
  DB_FILE_NAME_CONVERT ('/maindisk/oradata/users',
                        '/backups/users_ts');
```

RESTORE コマンドを実行すると、デフォルトでは、イメージ・コピー・バックアップをコピーすることによって、データファイルまたは制御ファイルが元の場所にリストアされます。バックアップ・セットではなく、イメージ・コピーが選択されます。バックアップ・セットを使用すると、リストアするファイルを検索するためにバックアップ・セット全体が読み取られ、オーバーヘッドが増加するためです。

現行のデータファイルをリストアおよびリカバリする必要があり、ディスク上に使用可能なイメージ・コピーが存在する場合は、Recovery Manager でイメージ・コピーを元の場所にコピーする必要はありません。リストア対象のデータファイルのかわりに、イメージ・コピーを使用できます。このタスクの実行方法については、17-16 ページの「コピーへの切替え後の完全リカバリの実行」を参照してください。

#### 参照：

- バックアップ・セットまたはイメージ・コピーのいずれかを Recovery Manager バックアップのデフォルト・タイプにする方法については、5-4 ページの「バックアップ用のデフォルト・タイプの構成: バックアップ・セットまたはコピー」を参照してください。
- 「ディスクへの Recovery Manager バックアップ用のバックアップ・セットまたはバックアップ・コピーの指定」(8-3 ページ)
- イメージ・コピーでの %U の意味については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## ユーザー管理イメージ・コピー

Recovery Manager では、ディスク上にファイルのイメージ・コピーを作成するオペレーティング・システム固有のファイル・コピー・コマンドや、サード・パーティ・ユーティリティなど、Recovery Manager 以外のメカニズムで作成したイメージ・コピーを使用できます。このタイプのコピーは、[ユーザー管理バックアップ](#)または[オペレーティング・システムのバックアップ](#)と呼ばれます。

CATALOG コマンドを実行すると、既存のイメージ・コピーを検査して、そのメタデータを Recovery Manager リポジトリに入力できます。ただし、CATALOG コマンドでは次の処理は行われません。

- 破損がないことを保証するためのデータファイルのコピー内のすべてのブロックの読取り
- イメージ・コピーが[バックアップ・モード](#)で正常に作成されたことの保証

これらのファイルは、カタログへの追加後、Recovery Manager によって生成されたイメージ・コピーの場合と同様に RESTORE または SWITCH コマンドで使用できます。

ミラー化されたディスク・ボリューム上にデータファイルを格納しているサイトでは、[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で、新しいユーザー管理コピーが存在することを Recovery Manager に通知して、そのコピーをバックアップの対象にします。CHANGE ... UNCATALOG コマンドの実行によってコピーが使用できなくなった場合は、Recovery Manager に通知する必要があります。

### 参照：

- [第 27 章「ユーザー管理データベース・バックアップの作成」](#)
- データファイルおよびアーカイブ・ログのイメージ・コピーをカタログに追加する方法については、11-16 ページの「[Recovery Manager リポジトリへのバックアップ・レコードの追加](#)」を参照してください。
- 「[Recovery Manager を使用したミラーの分割によるバックアップの実行](#)」(9-9 ページ)
- CHANGE 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## Recovery Manager を使用したバックアップの複数のコピー

Recovery Manager では、次の 2 つの方法を使用して、バックアップの同じコピーを複数作成できます。

- BACKUP ... COPIES コマンドでバックアップを多重化します。この場合、Recovery Manager は各バックアップ・セットの複数のコピーを作成します。
- ファイルをバックアップ・セットまたはイメージ・コピーとしてバックアップしてから、Recovery Manager の BACKUP BACKUPSET または BACKUP COPY OF コマンドで、そのバックアップ・セットまたはイメージ・コピーをバックアップします。

## 多重バックアップ・セット

データファイル、アーカイブ REDO ログ・ファイル、サーバー・パラメータ・ファイルおよび制御ファイルをバックアップ・ピースにバックアップする場合、Recovery Manager は、[多重バックアップ・セット](#)を作成して、1 つの BACKUP コマンドで、異なるバックアップ先にバックアップ・セットの各バックアップ・ピースの同じコピーを最大 4 つ作成できます。イメージ・コピーを作成するバックアップ操作では、多重化はサポートされていません。

BACKUP コマンドの使用時に、CONFIGURE、SET または BACKUP コマンドで COPIES パラメータを使用して、バックアップ・セットの多重化を指定できます。Recovery Manager では、ディスクまたはテープにバックアップを多重化できますが、テープとディスクにバックアップを同時に多重化することはできません。テープへのバックアップ時に、コピーの数が、使用可能なテープ・デバイスの数を超えないようにしてください。

BACKUP コマンドの FORMAT パラメータでは、多重バックアップの出力先を指定します。次の例では、データファイル 7 のバックアップのコピーを 3 つ作成します。

```
BACKUP DEVICE TYPE DISK COPIES 3 DATAFILE 7
  FORMAT '/disk1/%U', '?/oradata/%U', '?/%U';
```

この場合、Recovery Manager は、各バックアップ・ピースの最初のコピーを /disk1 に、2 番目のコピーを ?/oradata に、3 番目のコピーを Oracle ホームに格納します。Recovery Manager は、それぞれ異なる一意のバックアップ・セット・キーを持つ 3 つのバックアップ・セットを作成するわけではありません。Recovery Manager は、一意のキーを持つ 1 つのバックアップ・セットを作成して、そのバックアップ・セット内の各バックアップ・ピースの同じコピーを 3 つ生成します。

**参照：**

- 「バックアップの多重化の構成」 (6-5 ページ)
- 「バックアップ・セットの多重化」 (9-7 ページ)
- CONFIGURE 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- SET 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## バックアップのバックアップ

BACKUP コマンドを使用すると、既存のバックアップ・セットおよびイメージ・コピーをバックアップできます。

### バックアップ・セットのバックアップ

Recovery Manager の BACKUP BACKUPSET コマンドを使用すると、ディスク上に作成したバックアップ・セットをバックアップできます。このコマンドは、複数のメディア間でバックアップを実行する場合に有効です。

バックアップ・セットのいずれかのコピーが破損または欠落していることが検出された場合、Recovery Manager は、同じバックアップ・セットの他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存在するアーカイブ REDO ログを Recovery Manager がバックアップする際の動作に似ています。

例 7-2 は、本番バックアップ・スケジュールの一部として、週に 1 回 BACKUP コマンドを実行する方法を示しています。この方法では、すべてのバックアップがディスクとテープの両方に作成されます。

#### 例 7-2 テープへのバックアップ・セットのバックアップ

```
BACKUP DEVICE TYPE DISK AS BACKUPSET
  DATABASE PLUS ARCHIVELOG;
BACKUP
  DEVICE TYPE sbt
  BACKUPSET ALL; # copies backup sets on disk to tape
```

---

**注意：** 自動チャンネルを使用した sbt へのバックアップでは、最初に CONFIGURE DEVICE TYPE sbt コマンドを実行する必要があります。

---

BACKUP BACKUPSET を使用して、バックアップ領域の割当てを管理することもできます。  
例 7-3 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て、それらをディスクから削除しています。

### 例 7-3 領域割当ての管理

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

ここで使用している DELETE INPUT は、DELETE ALL INPUT と同等であり、Recovery Manager は、バックアップ・セットのすべての既存のコピーを削除します。バックアップを 4 つの場所に多重化している場合、Recovery Manager は、バックアップ・セットのピースのコピーを 4 つともすべて削除します。

参照: 「Recovery Manager バックアップのバックアップ」(8-28 ページ)

### イメージ・コピーのバックアップ

BACKUP COPY OF コマンドを使用すると、データベース・ファイルの既存のイメージ・コピーをバックアップ・セットまたはイメージ・コピーのいずれかとしてバックアップできます。このコマンドを使用する際は、コマンドで指定するすべてのデータファイルのイメージ・コピーがすでに存在している必要があります。1 つのデータファイルに複数のコピーが存在する場合は、最新のコピーが使用されます。表領域またはデータベース全体を指定すると、データベースまたは表領域にデータファイルがあり、対応するイメージ・コピーのバックアップがない場合、Recovery Manager はエラーを発行します。

## 制御ファイルおよびサーバー・パラメータ・ファイルの自動バックアップ

制御ファイルおよびサーバー・パラメータ・ファイルの最新のバックアップを作成しておく、多くのリカバリ状況で非常に役立ちます。このようなバックアップの作成をサポートするために、データベースには、制御ファイルおよびサーバー・パラメータ・ファイルの自動バックアップ機能が備わっています。自動バックアップは、BACKUP コマンドで明示的に要求された現行の制御ファイルのバックアップとは関係なく行われます。

制御ファイルの自動バックアップによって、Recovery Manager 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、データベースをリカバリできます。自動バックアップの格納に使用されるパスは標準的な書式に準拠しているため、Recovery Manager は、その自動バックアップからサーバー・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すると、Recovery Manager によって自動バックアップから制御ファイルがリストアされます。制御ファイルをマウントした後、マウントされた制御ファイル内の 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。

### Recovery Manager が制御ファイルの自動バックアップを実行する場合

CONFIGURE CONTROLFILE AUTOBACKUP が ON に設定されている場合、Recovery Manager は、正常な BACKUP コマンドの最後に、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（データベースの起動に使用された場合）を自動的にバックアップします。データベースが ARCHIVELOG モードで実行されている場合、データベースの構造上の変更によって制御ファイルの内容が影響を受けると、Recovery Manager は、制御ファイルの自動バックアップを作成します。



## Recovery Manager による制御ファイルの自動バックアップの実行方法

自動バックアップは、バックアップ・ジョブ中に割り当てられた最初のチャンネルによって作成され、独自のバックアップ・セットに格納されます。データベースの構造変更後の自動バックアップの場合は、構造変更に関連付けられたサーバー・プロセスによってバックアップが作成されます。

サーバー・パラメータ・ファイルがデータベースによって使用されている場合は、Recovery Manager によって、制御ファイルの自動バックアップと同じバックアップ・セットにバックアップされます。自動バックアップが完了すると、バックアップ・ピースの完全パスおよびデバイス・タイプを含むメッセージが、データベースによって **自動診断リポジトリ** にあるアラート・ログに書き込まれます。

---

**注意：** 制御ファイルの自動バックアップは多重化されません。

---

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・タイプのデフォルトの書式 %F が使用されるため、Recovery Manager は、リポジトリを使用しなくても、ファイルの場所を判断してそのファイルをリストアできます。CONFIGURE CONTROLFILE AUTOBACKUP FORMAT コマンドを使用すると、別の書式を指定できます。ただし、すべての自動バックアップの書式に %F 変数を含める必要があります。デフォルトの書式を使用しない場合は、障害リカバリ中に、自動バックアップの生成に使用された書式を指定する必要があります。そうしない場合、Recovery Manager は自動バックアップをリストアできません。

### 参照：

- 「[制御ファイルおよびサーバー・パラメータ・ファイルの自動バックアップの構成](#)」 (5-7 ページ)
- CONFIGURE 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- BACKUP 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- 置換変数 %F については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## 増分バックアップ

デフォルトでは、Recovery Manager は全体バックアップを作成します。データファイルの **全体バックアップ** には、バックアップするファイルのすべての割当て済ブロックが含まれます。データファイルの全体バックアップは、イメージ・コピーである場合があります。この場合は、すべてのデータ・ブロックがバックアップされます。また、バックアップ・セットに格納される場合もあります。この場合は、使用されていないデータファイル・ブロックはスキップされます。

全体バックアップは、Recovery Manager バックアップのデフォルト・タイプです。全体バックアップは、後続の増分バックアップには影響せず、**増分バックアップ** 計画の一部とはみなされません。イメージ・コピーにはデータファイル内のすべてのデータ・ブロックが含まれるため、イメージ・コピーは常に全体バックアップとなります。バックアップ・セットにはデータファイル内のすべてのデータ・ブロックが含まれる可能性があるため、バックアップ・セットはデフォルトでは全体バックアップとなります。ただし、**未使用ブロックの圧縮** は、未使用のブロックは除外され、現在使用されていないブロックが除外される場合もあることを意味します (7-4 ページの「[バックアップ・セットの圧縮](#)」を参照)。

全体バックアップに対し、増分バックアップでは、以前のバックアップ以降に変更されたデータ・ブロックのみがコピーされます。Recovery Manager を使用して、データファイル、表領域またはデータベース全体の増分バックアップを作成できます。全体バックアップを増分バックアップ計画に含めて、後続の増分バックアップの親として指定することはできません。

## 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**を作成できます。各増分レベルは、0 または 1 の値で示されます。**レベル 0 の増分バックアップ**は、後続の増分バックアップの基本となるバックアップであり、データが含まれるすべてのブロックをコピーします。レベル 0 のデータベース・バックアップは、バックアップ・セットまたはイメージ・コピーとして作成できます。

レベル 0 の増分バックアップと全体バックアップの違いは、全体バックアップは増分計画には含まれないという点のみです。したがって、全体バックアップは、レベルが 0 より大きい増分バックアップの親となります。

レベル 1 の増分バックアップは、次のいずれかのタイプです。

- **差分増分バックアップ**: レベル 1 またはレベル 0 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をバックアップします。
- **累積増分バックアップ**: レベル 0 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をバックアップします。

デフォルトでは、増分バックアップは、差分バックアップに設定されています。

---

---

**注意:** リカバリ時間がディスク領域より重要である場合は、差分バックアップより累積バックアップをお勧めします。これは、リカバリ中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。

---

---

バックアップ・ファイルのサイズは、変更されたブロック数、増分バックアップ・レベルおよび増分のタイプ（差分または累積）によって異なります。

### 差分増分バックアップ

レベル 1 の差分バックアップでは、Recovery Manager は、レベル 1（累積または差分）またはレベル 0 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をバックアップします。たとえば、レベル 1 の差分バックアップでは、Recovery Manager はレベル 1 の最新のバックアップを確認して、そのバックアップ以降に変更されたすべてのブロックをバックアップします。レベル 1 のバックアップが使用できない場合、Recovery Manager は、レベル 0 の基本バックアップ以降に変更されたすべてのブロックをコピーします。

現行または親の**インカネーション**でレベル 0 のバックアップが使用できない場合の動作は、互換性モード設定によって異なります。互換性が 10.0.0 以上の場合、Recovery Manager は、ファイルの作成以降に変更されたすべてのブロックをコピーします。それ以外の場合、Recovery Manager は、以前のリリースと同様に、レベル 0 のバックアップを生成します。



図 7-2 差分増分バックアップ

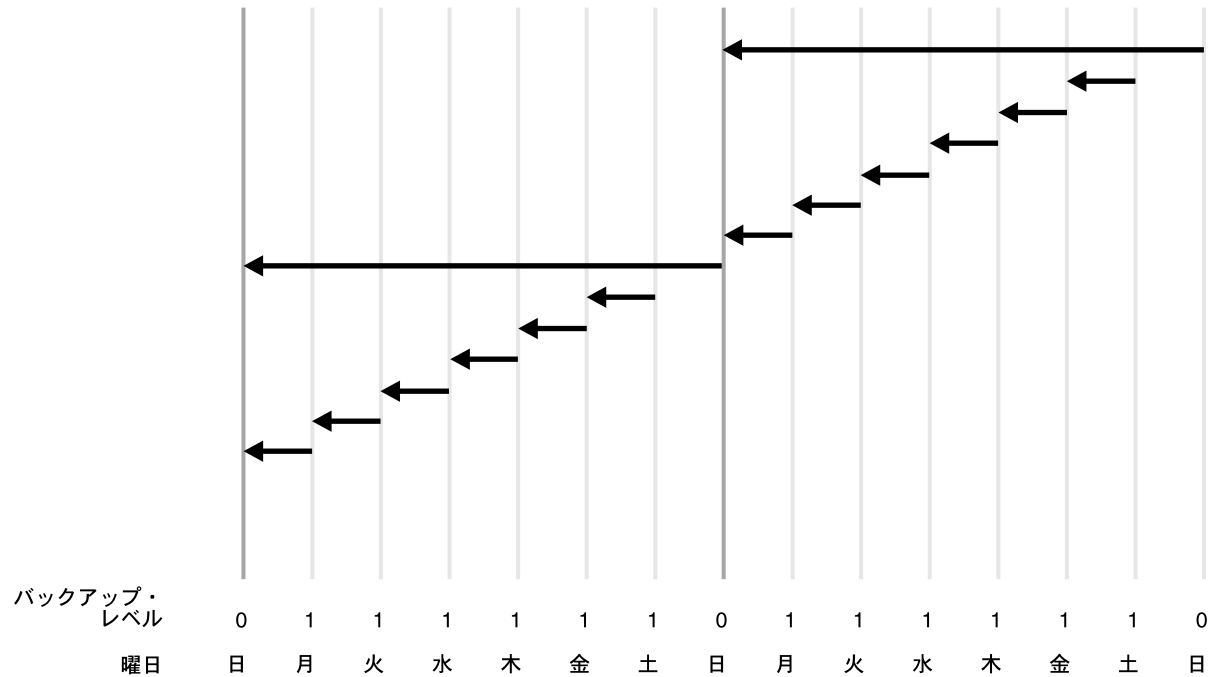


図 7-2 に示す例では、毎週、次の処理が実行されます。

- 日曜日

レベル 0 の増分バックアップによって、このデータベースでこれまでに使用されたすべてのブロックがバックアップされます。

- 月曜日から土曜日

月曜日から土曜日には、レベル 1 の差分増分バックアップによって、レベル 1 またはレベル 0 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毎日バックアップされます。つまり、月曜日のバックアップでは、日曜日のレベル 0 のバックアップ以降に変更されたブロックがコピーされ、火曜日のバックアップでは、月曜日のレベル 1 のバックアップ以降に変更されたブロックがコピーされる、というように実行されます。

### 累積増分バックアップ

レベル 1 の累積バックアップでは、Recovery Manager は、現行または親のインカネーションでレベル 0 の最新の増分バックアップ以降に使用されたすべてのブロックをバックアップします。累積増分バックアップでは、特定のレベルの 1 つの増分バックアップのみが必要となるため、リストアに必要な作業が軽減されます。累積バックアップでは、同じレベルの以前のバックアップによって実行された作業が繰り返されるため、差分バックアップよりも多くの領域および時間が必要です。

図 7-3 累積増分バックアップ

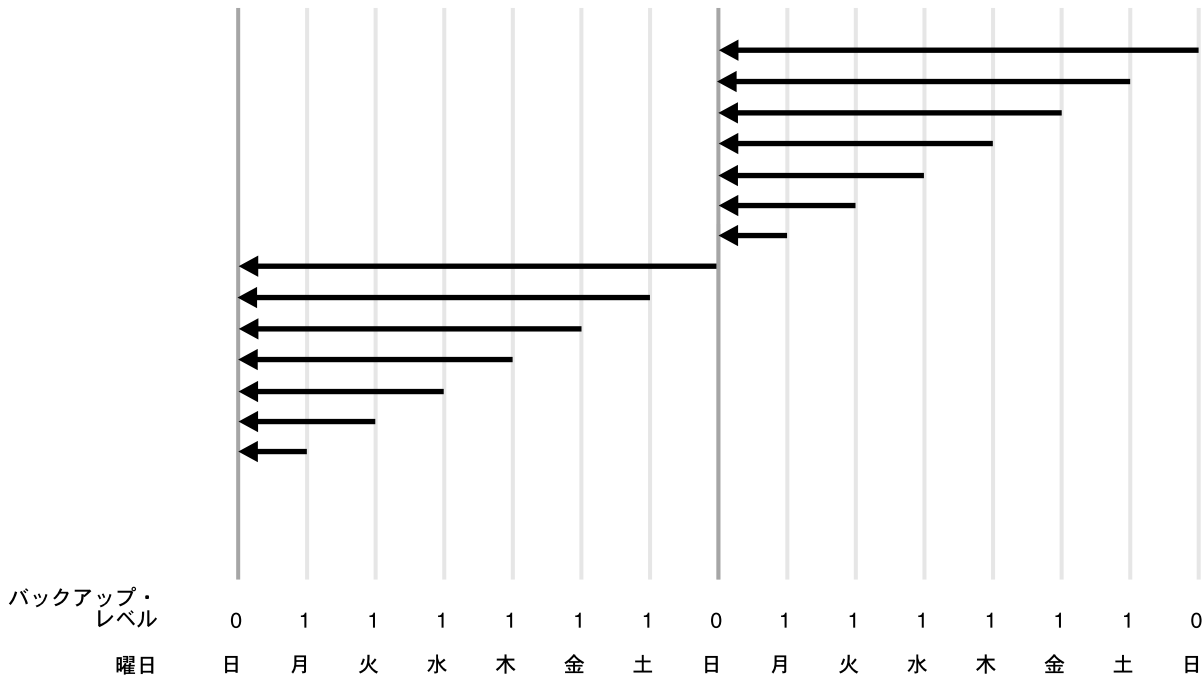


図 7-3 に示す例では、毎週、次の処理が実行されます。

- 日曜日  
レベル 0 の増分バックアップによって、このデータベースでこれまでに使用されたすべてのブロックがバックアップされます。
- 月曜日から土曜日  
レベル 1 の累積増分バックアップによって、レベル 0 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がコピーされます。レベル 0 の最新のバックアップは日曜日に作成されているため、月曜日から土曜日まで毎日行われるレベル 1 のバックアップでは、日曜日の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。

参照：「増分バックアップの作成および更新」(8-15 ページ)

## ブロック・チェンジ・トラッキング

増分バックアップの**ブロック・チェンジ・トラッキング**機能を使用すると、各データファイル内の変更されたブロックを**ブロック・チェンジ・トラッキング・ファイル**に記録することによって増分バックアップのパフォーマンスが向上します。このファイルは、**データベース領域**に格納される小さなビットマップ・ファイルです。REDO の生成時に、Recovery Manager は、変更されたブロックを追跡します。

ブロック・チェンジ・トラッキングを有効にした場合、Recovery Manager は、チェンジ・トラッキング・ファイルを使用して、増分バックアップ用の変更されたブロックを識別します。これによって、データファイル内のすべてのブロックをスキャンする必要がなくなります。**レベル 0 の増分バックアップ**にはすべてのブロックが含まれるため、増分レベルが 0 (ゼロ) より大きい場合にのみ、Recovery Manager はブロック・チェンジ・トラッキングを使用します。

参照：「ブロック・チェンジ・トラッキングを使用した、増分バックアップのパフォーマンスの向上」(8-21 ページ)

## 増分バックアップのアルゴリズム

Recovery Manager で増分バックアップの作成に使用されるアルゴリズムを理解するには、次の概念を理解しておく必要があります。

- チェックポイント SCN

すべてのデータファイルに**データファイル・チェックポイント SCN**があります。この SCN は、`V$DATAFILE.CHECKPOINT_CHANGE#` で参照できます。この SCN より小さい SCN を持つすべての変更がそのファイル内に存在することが保証されます。レベル 0 の増分バックアップがリストアされると、リストアされたデータファイルには、レベル 0 が作成されたときのチェックポイント SCN が含まれます。レベル 1 の増分バックアップがファイルに適用されると、そのファイルのチェックポイント SCN は、増分レベル 1 が作成されたときにファイルに含まれていたチェックポイント SCN に進みます。

- 増分開始 SCN

この SCN は、レベル 1 の増分バックアップにのみ適用されます。バックアップには、SCN が増分開始 SCN 以上のすべてのブロックが含まれます。SCN が増分開始 SCN より小さいブロックは、バックアップには含まれません。増分開始 SCN は、多くの場合、レベル 1 のバックアップの親のチェックポイント SCN です。

- ブロック SCN

データファイルの各データ・ブロックには、ブロックに対する最新の変更が行われた時点の SCN が記録されます。

Recovery Manager は、ファイルのレベル 1 の増分バックアップの作成時に、ファイルを読み取り、すべてのブロックの SCN を確認し、SCN がこのバックアップの増分開始 SCN 以上であるブロックをバックアップします。差分バックアップの場合、増分開始 SCN は最新のレベル 1 のバックアップのチェックポイント SCN です。累積バックアップの場合、増分開始 SCN は最新のレベル 0 のバックアップのチェックポイント SCN です。

ブロック・チェンジ・トラッキングが有効になっている場合、Recovery Manager は、ビットマップを使用して、増分開始 SCN からチェックポイント SCN までの範囲内で変更されていないブロックの読取りを回避します。Recovery Manager は、読み取るすべてのブロックを確認し、ブロック内の SCN を使用してバックアップに含めるブロックを決定します。

増分バックアップ・アルゴリズムによって、Recovery Manager は、リカバリ中、変更されたデータが含まれるすべてのブロックを適用できます。これには、NOLOGGING オプションで作成されたオブジェクトに対する変更も含まれます。したがって、NOLOGGING の変更を行う前に作成されたバックアップをリストアする場合、それらのバックアップをリカバリする方法は、増分バックアップのみとなります。

**参照：** NOLOGGING モードの詳細は、『Oracle Database 概要』を参照してください。

## 増分バックアップを使用したリカバリ

**メディア・リカバリ**中、Recovery Manager は、リストアされるファイルを確認し、増分バックアップを使用してそれらをリカバリできるかどうかを判断します。増分バックアップとアーカイブ REDO ログのいずれかを選択できる場合、Recovery Manager は常に増分バックアップを選択します。ブロック・レベルで変更を適用するほうが、REDO を適用するより高速なためです。

Recovery Manager では、リカバリ中、増分バックアップをデータファイルに適用するためにデータファイルの基本増分バックアップをリストアする必要はありません。たとえば、データファイルのイメージ・コピーをリストアし、増分バックアップを使用してそれらをリカバリできます。

**参照：** 「増分バックアップおよびアーカイブ REDO ログの選択」  
(13-5 ページ)

## バックアップの保存方針

CONFIGURE RETENTION POLICY コマンドを使用すると、永続的かつ自動的な**バックアップの保存方針**を作成できます。バックアップの保存方針が有効な場合、Recovery Manager は、CONFIGURE コマンドで指定された基準に従って、データファイルまたは制御ファイルのバックアップを**不要なバックアップ**（リカバリに必要ではなくなった）とみなします。REPORT OBSOLETE コマンドを使用して不要なファイルを表示し、DELETE OBSOLETE コマンドを使用してそれらのファイルを削除できます。

長期にわたってバックアップを作成していると、古いバックアップは保存方針を満たすために必要ではなくなります。Recovery Manager は、不要になったファイルを識別できますが、自動的に削除しません。DELETE OBSOLETE コマンドを使用して、保存方針を満たすために必要ではなくなったファイルを削除する必要があります。

フラッシュ・リカバリ領域が構成されている場合、新しいファイル用にリカバリ領域がさらに必要になると、不要になったファイルまたはすでにテープにバックアップされたファイルがデータベースによって自動的に削除されます。ディスク割当て制限の規則は、保存方針の規則とは異なりますが、ディスク割当て制限を満たすために、保存方針に違反してファイルが削除されることはありません。詳細は、11-8 ページの「**フラッシュ・リカバリ領域が一杯になった場合の対応**」を参照してください。

バックアップが不要になるのは、REPORT OBSOLETE または DELETE OBSOLETE で、ユーザー定義の保存方針に基づいて、バックアップがリカバリに必要ないと判断された場合です。バックアップが**期限切れのバックアップ**とみなされるのは、Recovery Manager がクロスチェックを実行してファイルを検出できない場合のみです。したがって、不要とはファイルが必要ないことを意味し、期限切れとはファイルが検出されないことを意味します。

バックアップ保存方針は、全体またはレベル 0 のデータファイルおよび制御ファイルのバックアップにのみ適用されます。データファイルのコピーおよびプロキシ・コピーの場合、Recovery Manager がコピーまたはプロキシ・コピーを不要と判断すると、それらのコピーを削除できます。データファイルのバックアップ・セットの場合、バックアップ・セット内のすべてのデータファイルのバックアップが不要になると、Recovery Manager はそのバックアップ・セットを削除できます。

保存方針は、アーカイブ REDO ログおよびレベル 1 の増分バックアップに直接は影響しません。これらのファイルは、それらを必要とする全体バックアップが存在しなくなった場合に不要となります。保存方針は、データファイルおよび制御ファイルの全体バックアップまたはレベル 0 のバックアップのみでなく、アーカイブ REDO ログおよびレベル 1 の増分バックアップにも影響します。まず、Recovery Manager は不要なデータファイルおよび制御ファイルのバックアップを判断します。次に、Recovery Manager は、保持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をリカバリする際に必要ないすべてのアーカイブ・ログおよびレベル 1 の増分バックアップを、不要とみなします。

---

---

**注意：** Recovery Manager を使用せずに（メディア・マネージャのテープの保存方針などによって）バックアップを削除すると、Recovery Manager は自動保存方針を実装できません。テープ上のすべての Recovery Manager 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、メディア・マネージャがそのテープを期限切れにしないように設定することをお勧めします。

---

---

保存方針を実装する場合、**冗長性**と**リカバリ期間**という相互に排他的な 2 つのオプションがあります。

## リカバリ期間

リカバリ期間とは、現在の時点から**リカバリ可能ポイント**までの期間です。リカバリ可能ポイントは、想定される Point-in-Time リカバリの最も早い時点（メディア障害後にリカバリ可能な最も早い時点）です。たとえば、1週間のリカバリ期間を実装した場合、Recovery Manager は、データベースを最大7日前までリカバリできるように、全体バックアップおよび必要な増分バックアップとアーカイブ・ログを保持します。この保存方針は、次のように構成します。

```
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;
```

このコマンドを実行すると、各データファイルに対して、リカバリ可能ポイントより前のバックアップが1つ保持されます。たとえば、リカバリ期間が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1つ存在する必要があります。

```
SYSDATE - BACKUP CHECKPOINT TIME >= 7
```

この条件を満たす最新のバックアップよりも古いすべてのバックアップは不要となります。

図 7-4 に示す保存方針を設定するとします。保存方針の内容は、次のとおりです。

- リカバリ期間は7日です。
- データベースのバックアップは2週間ごとにスケジュールされ、次の日付に実行されます。
  - 1月1日
  - 1月15日
  - 1月29日
  - 2月12日
- データベースは ARCHIVELOG モードで実行され、アーカイブ・ログは、保存方針で必要とされる間のみ、ディスク上に保存されます。

図 7-4 リカバリ期間 1

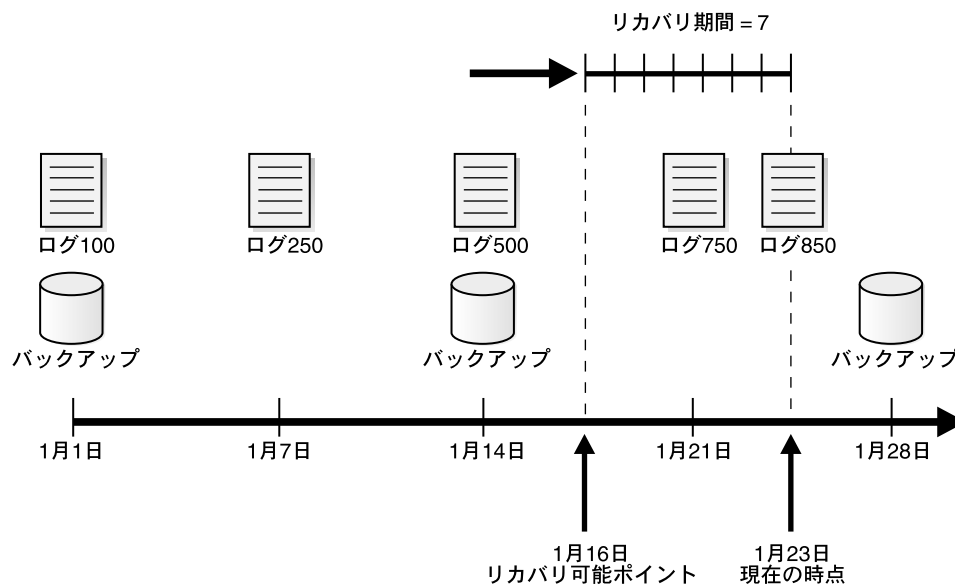
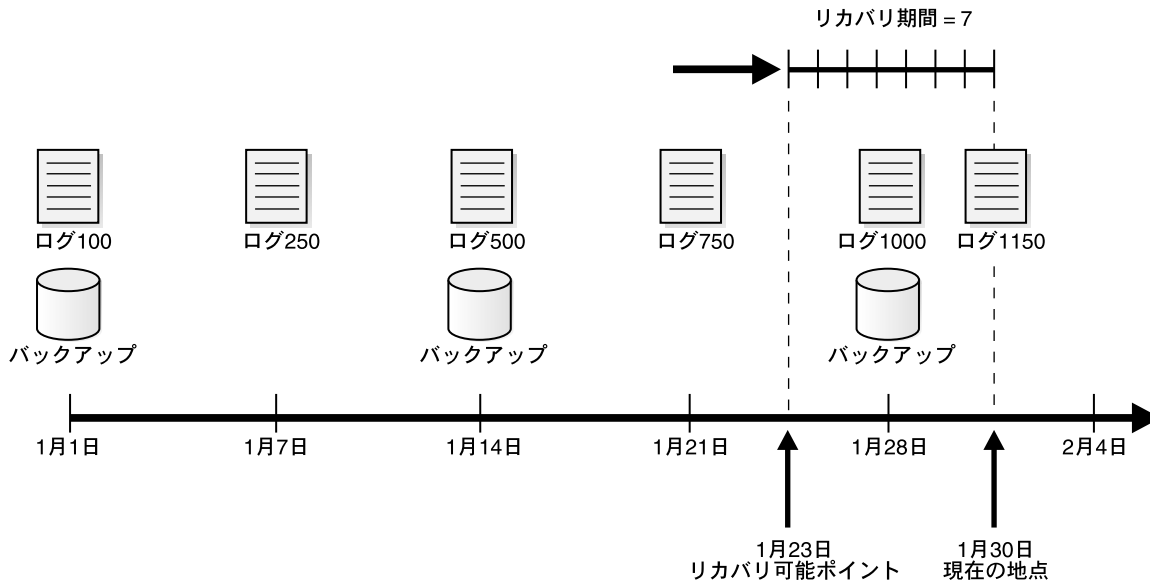


図 7-4 に示すように、現在の時点は1月23日、リカバリ可能ポイントは1月16日です。このため、リカバリには、1月14日のバックアップおよびログ順序500から850のアーカイブ・ログが必要です。500より前のログおよび1月1日のバックアップは、期間内の時点へのリカバリには必要ありません。

図 7-5 に、1 週間後の同じ例を示します。

図 7-5 リカバリ期間 2



この例では、現在の時点は1月30日、リカバリ可能ポイントは1月23日です。最新（1月28日）のバックアップがリカバリ期間に存在していますが、この場合でも1月14日のバックアップは必要です。これは、1月28日のバックアップをリストアしても、期間の最初の時点（1月23日）にはリカバリできないためです。期間内のいずれの時点にもリカバリできるようにするために、1月14日のバックアップおよび順序500から1150のすべてのアーカイブ・ログを保存する必要があります。

参照：「リカバリ期間に基づく保存方針の構成」（5-23 ページ）

## バックアップ冗長性

リカバリ期間を使用すると、保持する必要があるバックアップの数が一定ではなく、バックアップ・スケジュールによって異なるため、ディスク領域の計画が複雑になる場合があります。これに対して、冗長性に基づく保存方針では、データファイルごとに保持する必要があるバックアップ数を指定します。たとえば、次のように入力して冗長性を2に構成できます。

```
CONFIGURE RETENTION POLICY TO REDUNDANCY 2;
```

デフォルトの保存方針は、REDUNDANCY 1 に構成されています。

参照：「冗長性に基づく保存方針の構成」（5-22 ページ）

## 不要なバックアップのバッチ削除

REPORT OBSOLETE コマンドを実行すると、保存方針に従って、現在不要なバックアップを確認できます。関連コマンド DELETE OBSOLETE を実行すると、保存方針に従って、不要なすべてのファイルを削除できます。DELETE OBSOLETE を定期的に行うことで、不要なバックアップの格納によって消費される領域を最小限に抑えることができます。たとえば、週に1回実行するスクリプトで DELETE OBSOLETE を実行できます。

また、REPORT または DELETE コマンドで REDUNDANCY または RECOVERY WINDOW オプションを指定して、構成されている保存方針を無効にすることもできます。構成したリカバリ期間より短いリカバリ期間で DELETE OBSOLETE を使用すると、リカバリ可能な期間が事実上短縮されることに注意してください。たとえば、構成した期間が14日にもかかわらず、DELETE OBSOLETE RECOVERY WINDOW OF 7 DAYS を実行した場合は、7日から14日間の間の時間にはリカバリできなくなります。

**参照：**

- レポートの生成およびバックアップの削除については、[第 10 章「Recovery Manager 操作に関するレポート」](#)を参照してください。
- DELETE 構文の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- REPORT 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## バックアップ保存方針およびフラッシュ・リカバリ領域の削除規則

Recovery Manager のステータス OBSOLETE は、常に保存方針に応じて決定されます。たとえば、Recovery Manager リポジトリ内のデータベースのバックアップが OBSOLETE とみなされる理由は、そのバックアップがリカバリ期間内の時点へのリカバリに必要なか、または冗長であるためです。

[フラッシュ・リカバリ領域](#)を構成している場合、内部アルゴリズムを使用して、フラッシュ・リカバリ領域のファイルから、構成されている保存方針を満たす必要がなくなったファイルが選択されます。ステータスが OBSOLETE のバックアップは、[ディスク割当て制限](#)の規則によって削除対象となっています。ディスク割当て制限の規則に従ってフラッシュ・リカバリ領域から削除するファイルが決定される際に、保存方針が違反されることはありません。

フラッシュ・リカバリ領域の OBSOLETE ステータスの規則と、ディスク割当て制限で削除対象とする規則には、1 つの重要な違いがあります。1000 から 2000 のアーカイブ・ログがディスク上に存在し、現行のリカバリ期間に必要であるとします。これらのログをテープにバックアップする場合、保存方針では、ディスク上のログは必要であるとみなされます。一方、フラッシュ・リカバリ領域のディスク割当て制限のアルゴリズムでは、ディスク上のログはテープにバックアップされているため、削除対象とみなされます。したがって、ディスク上のログはリポジトリでは OBSOLETE ステータスを持ちませんが、フラッシュ・リカバリ領域では削除対象になります。





---

---

## データベースのバックアップ

この章では、最も基本的なバックアップ作業の実行方法および Recovery Manager を使用したバックアップ計画の実装方法について説明します。この章の内容は、次のとおりです。

- [Recovery Manager バックアップの概要](#)
- [バックアップ出力オプションの指定](#)
- [Recovery Manager を使用したデータベース・ファイルのバックアップ](#)
- [Recovery Manager を使用したアーカイブ REDO ログのバックアップ](#)
- [増分バックアップの作成および更新](#)
- [長期格納用のデータベース・バックアップの作成](#)
- [Recovery Manager バックアップのバックアップ](#)

**参照：**

- バックアップの最適化、多重化、バックアップの暗号化、再開可能なバックアップなどの高度なトピックについては、[第9章「データベースのバックアップ:高度なトピック」](#)を参照してください。
- Data Guard 環境での Recovery Manager バックアップおよびリカバリの実行方法については、『Oracle Data Guard 概要および管理』を参照してください。

## Recovery Manager バックアップの概要

この項では、Recovery Manager バックアップの概要について説明します。

### Recovery Manager バックアップの目的

Recovery Manager バックアップの主要な目的は、データを保護することです。[メディア障害](#)が発生した場合は、バックアップをリストアして、消失した変更をリカバリできます。

また、バックアップを作成すると、8-24 ページの「[長期格納用のデータベース・バックアップの作成](#)」の説明に従って長期アーカイブ用のデータを保存したり、[第 VII 部「Recovery Manager を使用したデータの送信」](#)の章の説明に従ってデータを送信することができます。

### Recovery Manager バックアップの基本的な概念

[第 7 章「Recovery Manager バックアップの概要」](#)で説明されているように、Recovery Manager クライアント内から BACKUP コマンドを使用して、データベース全体または一部をバックアップできます。この章で説明する多くの方法は、Oracle Enterprise Manager で提供されるオラクル社推奨のバックアップ計画でも使用されます（『Oracle Database 2 日でデータベース管理者』を参照）。

多くの場合、バックアップ計画に従ってデータベースを構成しておく、Recovery Manager プロンプトで次のコマンドを入力してデータベースをバックアップできます。

```
RMAN> BACKUP DATABASE;
```

Recovery Manager は、構成された設定、以前のバックアップのレコードおよびデータベース構造の制御ファイル・レコードを使用して、効率的な一連のバックアップ手順を決定します。その後、それらの手順を実行します。

3-8 ページの「[Data Guard 環境での Recovery Manager によるファイル管理](#)」で説明されているように、Data Guard 環境の任意のデータベースで Recovery Manager バックアップを実行できます。バックアップにアクセスできるかぎり、環境内の任意のデータベースのすべてのバックアップを他のデータベースのリカバリに使用できます。データベース・ファイルのすべてのバックアップ（制御ファイルのバックアップを含む）を[フィジカル・スタンバイ・データベース](#)にオフロードすると、プライマリ・データベース上のリソースの消費を回避できます。

#### 参照：

- BACKUP コマンドについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- Recovery Manager でスタンバイ・データベースをバックアップする方法については、『Oracle Data Guard 概要および管理』を参照してください。

## バックアップ出力オプションの指定

BACKUP DATABASE などの Recovery Manager コマンドに最低限必要なオプションのみを指定した場合、Recovery Manager は、構成された環境および Recovery Manager の組込みデフォルトに基づいて、バックアップ先デバイス、バックアップの出力場所およびバックアップ・タグを自動的に決定します。

また、BACKUP に引数を指定して、これらのデフォルトを上書きすることもできます。次の項では、最も一般的なオプションについて説明します。

- Recovery Manager バックアップ用のデバイス・タイプの指定
- ディスクへの Recovery Manager バックアップ用のバックアップ・セットまたはバックアップ・コピーの指定
- Recovery Manager バックアップのフォーマットの指定
- Recovery Manager バックアップのタグの指定
- 圧縮バックアップの作成

**参照：** バックアップの多重化や再開などの高度なバックアップ・オプションについては、第9章「データベースのバックアップ:高度なトピック」を参照してください。

## Recovery Manager バックアップ用のデバイス・タイプの指定

BACKUP コマンドでは、ディスクまたはテープ・デバイスのいずれにバックアップするかを指定する DEVICE TYPE 句を使用します。次に、ディスクにバックアップする例を示します。

### 例 8-1 デバイス・タイプ DISK の指定

```
BACKUP DATABASE  
    DEVICE TYPE DISK;
```

DEVICE TYPE 句を指定しないで BACKUP を実行すると、Recovery Manager は、構成されているデフォルトのデバイス（ディスクまたは SBT）にバックアップを格納します。デフォルトのデバイスは、5-3 ページの「バックアップ用のデフォルト・デバイスの構成: ディスクまたは SBT」で説明されている CONFIGURE DEFAULT DEVICE TYPE コマンドを使用して設定します。

## ディスクへの Recovery Manager バックアップ用のバックアップ・セットまたはバックアップ・コピーの指定

Recovery Manager は、イメージ・コピーまたはバックアップ・セットとしてバックアップをディスク上に作成できます。デフォルトのディスク・デバイスの構成方法については、5-4 ページの「バックアップ用のデフォルト・タイプの構成: バックアップ・セットまたはコピー」を参照してください。このデフォルトは、AS COPY 句または AS BACKUPSET 句を使用して上書きできます。イメージ・コピーとしてディスクにバックアップするには、BACKUP AS COPY を使用します。

### 例 8-2 イメージ・コピーの作成

```
BACKUP AS COPY  
    DEVICE TYPE DISK  
    DATABASE;
```

バックアップ・セットにデータをバックアップするには、AS BACKUPSET 句を使用します。次の例に示すように、バックアップ・セットは、構成されているデフォルト・デバイスに作成したり、ディスクまたはテープに明示的に格納できます。

**例 8-3 バックアップ・セットの作成**

```
BACKUP AS BACKUPSET
  DATABASE;
```

```
BACKUP AS BACKUPSET
  DEVICE TYPE DISK
  DATABASE;
```

```
BACKUP AS BACKUPSET
  DEVICE TYPE SBT
  DATABASE;
```

**Recovery Manager バックアップのフォーマットの指定**

Recovery Manager には、BACKUP コマンドで生成されるファイルに名前を指定するための様々なオプションが用意されています。Recovery Manager は、優先順位に従って示されている次のルール・セットを使用して、出力ファイル形式を決定します。

1. BACKUP コマンドに FORMAT パラメータを指定した場合、生成されるファイル名はこの設定で制御されます。

たとえば、次のコマンドに示すように、出力を特定の場所に格納することができます。

```
BACKUP DATABASE
  FORMAT "/disk1/backup_%U"; # specifies a location on the file system
```

この場合、バックアップは、生成された一意のファイル名（/disk1/backup\_ という接頭辞付き）で格納されます。置換変数 %U が必要であることに注意してください。この置換変数は、ファイル名のその部分に一意の文字列を生成するために使用されます。

また、次の例に示すように、FORMAT パラメータを使用して、バックアップ先として ASM ディスク・グループを指定することもできます。

```
BACKUP DATABASE
  FORMAT '+dgroup1'; # specifies an ASM disk group
```

この場合、必要に応じて **自動ストレージ管理** で一意のファイル名が生成されるため、%U は不要です。ただし、必要な場合は、%U を指定できます。

---

**注意：** フラッシュ・リカバリ領域が有効になっている場合に FORMAT を指定すると、Recovery Manager は FORMAT の設定に従います。FORMAT 句で場所を指定しなかった場合、Recovery Manager はプラットフォーム固有の場所にバックアップを作成します。

---

2. FORMAT 設定がバックアップで使用される特定のチャンネル用に構成されている場合、生成されるファイル名の制御はこの設定によって行われます。
3. FORMAT 設定がバックアップで使用されるデバイス・タイプ用に構成されている場合、生成されるファイル名の制御はこの設定によって行われます。
4. ディスク・バックアップ中に **フラッシュ・リカバリ領域** が有効になっていて、FORMAT が指定されていない場合は、自動生成された名前でもフラッシュ・リカバリ領域にバックアップが作成されます。
5. このリストの他のいずれの条件も当てはまらない場合は、バックアップのデフォルトの場所およびファイル名の書式はプラットフォーム固有になります。

**参照：** FORMAT 句については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。ご使用のプラットフォームのデフォルトのファイルの場所については、インストレーション・ガイドを参照してください。

## ディスク・バックアップの複数のフォーマットの指定

通常、テープにバックアップする場合には、デフォルトの %U 変数によってテープ・バックアップに一意のファイル名が生成されるため、フォーマットを指定する必要はありません。ただし、ディスクへのバックアップで、パフォーマンスを向上するために複数のドライブにバックアップを分散させる必要がある場合には、フォーマットを指定できます。この場合、各ディスク・ドライブに1つの DISK チャンネルを割り当て、ALLOCATE CHANNEL コマンドで FORMAT 文字列を指定して、それぞれのディスクにファイル名が生成されるようにします。たとえば、次のコマンドを発行します。

```
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 AS COPY DATABASE;
}
```

チャンネルを次のように構成すると、将来、デフォルトでこのようにバックアップを分散できるようになります。

```
CONFIGURE DEVICE TYPE DISK PARALLELISM 3;
CONFIGURE DEFAULT DEVICE TYPE TO DISK;
CONFIGURE CHANNEL 1 DEVICE TYPE DISK FORMAT '/disk1/%d_backups/%U';
CONFIGURE CHANNEL 2 DEVICE TYPE DISK FORMAT '/disk2/%d_backups/%U';
CONFIGURE CHANNEL 3 DEVICE TYPE DISK FORMAT '/disk3/%d_backups/%U';
BACKUP AS COPY DATABASE;
```

## Recovery Manager バックアップのタグの指定

Recovery Manager は、バックアップを識別する方法として、作成するすべてのバックアップに **タグ** と呼ばれる文字列を追加します。デフォルトのタグを受け入れるか、または BACKUP コマンドの TAG パラメータで独自のタグを指定できます。

### バックアップ・タグ

ユーザー指定のタグは、バックアップまたはコピーの様々な目的や使用方法を示す場合に有効です。タグは、バックアップ・セット、プロキシ・コピー、データファイル・コピーまたは制御ファイル・コピーに指定できます。たとえば、SWITCH コマンドで使用するデータファイルのコピーには for\_switch\_only、RESTORE コマンドでのみ使用するファイルのコピーには for\_restore\_only というタグを指定することができます。

タグは一意である必要はないため、複数のバックアップ・セットまたはイメージ・コピーに同じタグ (weekly\_backup など) を付けることができます。特定のタグが含まれているバックアップからデータファイルをリストアするように指定するとします。要求したファイルの複数のバックアップにそのタグが含まれている場合、Recovery Manager は、RESTORE コマンドの制約内で、指定したタグが含まれている最新のバックアップをリストアします。

実際には、タグは、多くの場合、増分バックアップ計画などの1つの計画の一環として作成された一連のバックアップを区別する場合に使用されます。たとえば、BACKUP TAG weekly\_incremental などのタグを付けて、週次増分バックアップを作成できます。BACKUP コマンドの様々な形式を使用して、タグとバックアップを関連付けることができます。また、多くの RESTORE および RECOVER コマンドでは、タグを指定して RESTORE または RECOVER 操作で使用するバックアップを制限できます。

BACKUP コマンドの TAG パラメータを使用して明示的にタグを指定しない場合は、Recovery Manager によって、バックアップ (制御ファイルの自動バックアップ以外) のデフォルト・タグが暗黙的に作成されます。タグの形式は、TAGYYYYMMDDTHHMMSS です。ここで、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 のバックアップには、タグ TAG20070208T133437 が割り当てられる場合があります。日時は、バックアップを実行するインスタンスのタイムゾーンで、Recovery Manager がバックアップを開始した日時です。1つの BACKUP コマンドによって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。

入力時に使用した大 / 小文字に関係なく、タグは大文字で格納されます。バックアップ・タグの最大長は 30 バイトです。オペレーティング・システム的环境変数または %T、%D などの特殊な書式は、タグに使用できません。

**参照：** BACKUP ...TAG のデフォルト形式については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

### バックアップ・セットおよびイメージ・コピーのタグの指定

タグで使用する文字は、ターゲットのデータベース・ファイル・システムでファイル名として有効な文字に制限する必要があります。たとえば、**自動ストレージ管理**では、内部的に使用されるファイル名でのダッシュ (-) の使用はサポートされていません。このため、ASM ディスク・グループでのバックアップの場合、ダッシュが含まれているタグ (weekly-incr など) は無効なタグ名になります。

バックアップ・セットにタグを指定すると、そのタグは、指定したバックアップ・セットのコピーに含まれる各バックアップ・ピースの属性になります。**多重バックアップ・セット**を作成すると、バックアップ・セットの各コピーに同じタグが割り当てられます。例 8-4 では、タグ MONDAYBKP が指定されたバックアップ・セットを 1 つ作成します。

#### 例 8-4 バックアップ・セットへのタグの適用

```
BACKUP AS BACKUPSET
  COPIES 1
  DATAFILE 7
  TAG mondaybkp;
```

イメージ・コピーにタグを指定すると、そのタグは各コピーに適用されます。例 8-5 では、表領域 users および tools のデータファイルのコピーに、タグ MONDAYCPY を割り当てます。

#### 例 8-5 イメージ・コピーへのタグの適用

```
BACKUP AS COPY
  TABLESPACE users, tools
  TAG mondaycpy;
```

FROM TAG を使用して特定のタグが含まれているイメージ・コピーをコピーした後、TAG を使用して出力コピーに別のタグを割り当てることができます。例 8-6 では、タグ full\_cold\_copy が含まれている、データベースのすべてのイメージ・コピーのコピーを新しく作成し、その新しいコピーに new\_full\_cold\_copy を割り当てます。

#### 例 8-6 出力コピーへのタグの割り当て

```
BACKUP AS COPY
  COPY OF DATABASE
  FROM TAG full_cold_copy
  TAG new_full_cold_copy;
```

## 圧縮バックアップの作成

バックアップ・セットを作成する BACKUP コマンドを実行する場合は、Recovery Manager でサポートされている、バックアップ・セットの**バイナリ圧縮**を利用できます。BACKUP コマンドに AS COMPRESSED BACKUPSET オプションを指定します。作成されるバックアップ・セットは、Oracle Database ファイルが効率的に圧縮されるように最適化されたアルゴリズムを使用して圧縮されます。リカバリ中に特別な解凍手順を実行する必要はありません。

---

---

**注意：** バックアップ先がテープであり、テープ・デバイスで独自の圧縮が実行される場合、Recovery Manager によるバックアップ・セットの圧縮とメディア・マネージャ・ベンダーによる圧縮の両方は使用しないでください。ほとんどの場合、メディア・マネージャの圧縮を使用した方がよりよい結果を得ることができます。詳細は、第 21 章「Recovery Manager のパフォーマンスのチューニング」の Recovery Manager によるテープ・バックアップのパフォーマンス・チューニングに関する項を参照してください。

---

---

例 8-7 では、データベース全体とアーカイブ・ログを、構成済みのデフォルトのバックアップ先（ディスクまたはテープ）にバックアップし、圧縮バックアップ・セットを作成します。

#### 例 8-7 圧縮バックアップの作成

```
BACKUP
AS COMPRESSED BACKUPSET
DATABASE PLUS ARCHIVELOG;
```

バイナリ圧縮では、バックアップおよびリカバリ操作中に、パフォーマンスにある程度のオーバーヘッドが発生します。バイナリ圧縮は CPU リソースを消費するため、CPU 使用率がすでに高い場合は、圧縮バックアップはスケジュールしないでください。ただし、次の状況では、パフォーマンスが低下する可能性があります。

- フラッシュ・リカバリ領域またはその他のディスクベースのバックアップ先内のディスク領域が制限されている状況で、ディスクベースのバックアップを使用している場合
- CPU 使用率よりネットワーク帯域幅を削減することのほうが重要となる状況で、ネットワークを介して特定のデバイスへのバックアップを実行している場合
- CD、DVD などのアーカイブ・バックアップ・メディアを使用しており、バックアップ・サイズを削減することでメディア・コストおよびアーカイブ・ストレージを節約できる場合

**参照：** バックアップ・セットのバイナリ圧縮を使用した場合のパフォーマンスについては、『Oracle Database バックアップおよびリカバリ・リファレンス』の BACKUP コマンドの AS COMPRESSED BACKUPSET オプションに関する項を参照してください。

## Recovery Manager を使用したデータベース・ファイルのバックアップ

この項の内容は、次のとおりです。

- [Recovery Manager を使用したデータベース全体のバックアップの作成](#)
- [Recovery Manager を使用した表領域およびデータファイルのバックアップ](#)
- [Recovery Manager を使用した制御ファイルのバックアップ](#)
- [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](#)
- [NOARCHIVELOG モードでのデータベースのバックアップ](#)



## Recovery Manager を使用したデータベース全体のバックアップの作成

**データベース全体のバックアップ**は、データベースをマウントまたはオープンして実行できます。データベース全体のバックアップを実行するには、Recovery Manager プロンプトで `BACKUP DATABASE` 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、指定した表領域を除外することができます。6-6 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](#)」の説明に従って、常にスキップする各表領域に対して `CONFIGURE EXCLUDE` コマンドを実行すると、すべての Recovery Manager セッションで永続的に表領域をスキップできます。 `BACKUP ... NOEXCLUDE` を使用すると、構成済の設定を上書きできます。

### データベースをバックアップする手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. 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。
3. Recovery Manager プロンプトで、`BACKUP DATABASE` コマンドを発行します。

次の例に示すように、このコマンドの最も簡単な形式にはオプションまたはパラメータは必要ありません。

```
BACKUP DATABASE;
```

次の例では、データベースをバックアップし、オンライン REDO ログを切り替え、アーカイブ・ログをバックアップに格納します。

```
BACKUP DATABASE PLUS ARCHIVELOG;
```

バックアップの直後にログをアーカイブすることによって、バックアップ時間全体のアーカイブ・ログの完全なセットを取得します。これによって、このバックアップのリストア後にメディア・リカバリを実行できることが保証されます。

#### 参照：

- `BACKUP` コマンドについては『Oracle Database バックアップおよびリカバリ・リファレンス』、`CONNECT` コマンドについては『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- `BACKUP ... SKIP` を使用して、アクセスできないデータファイル、オフライン表領域内のデータファイルまたは読取り専用表領域内のデータファイルをスキップする方法については、9-7 ページの「[オフライン・ファイル、読取り専用ファイルおよびアクセスできないファイルのスキップ](#)」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特定の表領域を除外する方法については、6-6 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。

## Recovery Manager を使用した表領域およびデータファイルのバックアップ

`BACKUP TABLESPACE` コマンドでは 1 つ以上の表領域、`BACKUP DATAFILE` コマンドでは 1 つ以上のデータファイルをバックアップできます。表領域を指定すると、Recovery Manager は表領域の名前を一連のデータファイルに内部的に変換します。データベースは、マウントされている状態でもオープンされている状態でもかまいません。表領域は、読取り / 書込みでも読取り専用でもかまいません。

---

---

**注意：** 以前のリリースと同様に、トランSPORTABLE 表領域は、バックアップに対して読取り / 書込みモードである必要はありません。

---

---



バックアップにデータファイル 1 が含まれている場合、Recovery Manager は、制御ファイルおよびサーバー・パラメータ・ファイル（インスタンスがサーバー・パラメータ・ファイルで起動されている場合）を自動的にバックアップします。**制御ファイルの自動バックアップ**が有効になっている場合、Recovery Manager は、現行の制御ファイルおよびサーバー・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。そうでない場合、Recovery Manager は、データファイル 1 が含まれているバックアップ・セットにこれらのファイルを書き込みます。

### 表領域またはデータファイルをバックアップする手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. データベース・インスタンスが起動されていない場合は、データベースをマウントまたはオープンします。
3. Recovery Manager プロンプトで、BACKUP TABLESPACE コマンドまたは BACKUP DATAFILE コマンドを実行します。

次の例では、users および tools 表領域をテープにバックアップします。

```
BACKUP
  DEVICE TYPE sbt
  TABLESPACE users, tools;
```

次の例では、SBT チャンネルを使用して、データファイル 1 から 4 および /tmp/system01.dbf に格納されているデータファイルのコピーをテープにバックアップします。

```
BACKUP
  DEVICE TYPE sbt
  DATAFILE 1,2,3,4
  DATAFILECOPY '/tmp/system01.dbf';
```

## Recovery Manager を使用した制御ファイルのバックアップ

制御ファイルは、データベースがマウントまたはオープンされているときにバックアップできません。Recovery Manager は、**スナップショット制御ファイル**を使用して、読取り一貫性のバージョンを保証します。CONFIGURE CONTROLFILE AUTOBACKUP コマンドが ON に設定されている場合（デフォルトでは OFF）、Recovery Manager は、すべてのバックアップ後およびデータベースの構造変更後に、制御ファイルおよびサーバー・パラメータ・ファイルを自動的にバックアップします。**制御ファイルの自動バックアップ**には、以前のバックアップに関するメタデータが含まれます。このメタデータは、**障害リカバリ**に重要です。

---

**注意：** 1 つの Data Guard データベースで作成された制御ファイルのバックアップを、環境内の他のすべてのデータベースにリストアすることができます。プライマリ制御ファイルのバックアップとスタンバイ制御ファイルのバックアップには互換性があります。Recovery Manager を使用してスタンバイ・データベースにリストアする方法については、『Oracle Data Guard 概要および管理』を参照してください。

---

自動バックアップ機能が設定されていない場合は、次のいずれかの方法を使用して、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。

- BACKUP CURRENT CONTROLFILE を実行します。
- 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いずれかのバックアップに含める
- データファイル 1 をバックアップします（制御ファイルおよびサーバー・パラメータ・ファイルは、Recovery Manager によって自動的にデータファイル 1 のバックアップに格納されるため）。

---



---

**注意：** 制御ファイルのブロック・サイズがデータファイル1のブロック・サイズと等しくない場合、制御ファイルをデータファイルと同じバックアップ・セットに書き込むことはできません。ブロック・サイズが異なる場合、Recovery Manager は、制御ファイルを単独で別のバックアップ・セットに書き込みます。V\$CONTROLFILE.BLOCK\_SIZE は制御ファイルのブロック・サイズを示し、DB\_BLOCK\_SIZE 初期化パラメータはデータファイル1のブロック・サイズを示します。

---



---

## 制御ファイルの手動バックアップの作成

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は異なります。Recovery Manager は、BACKUP コマンドで指定されたファイルをバックアップした後、制御ファイルの自動バックアップを作成します。このため、制御ファイルの手動バックアップとは異なり、自動バックアップには、完了直後のバックアップに関するメタデータが含まれます。また、Recovery Manager は、リカバリ・カタログを使用しないで自動バックアップを自動的にリストアすることもできます。

手動バックアップを作成するには、他のファイルをバックアップする際に INCLUDE CURRENT CONTROLFILE を指定するか、または BACKUP CONTROLFILE を指定します。また、CONTROLFILECOPY パラメータを指定して、制御ファイルのコピーをディスクにバックアップすることもできます。

### 制御ファイルを手動でバックアップする手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. ターゲット・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。
3. 必要な制御ファイル句を指定して BACKUP コマンドを実行します。

次の例では、表領域 users をテープにバックアップし、そのバックアップに現行の制御ファイルを含めます。

```
BACKUP DEVICE TYPE sbt
    TABLESPACE users
    INCLUDE CURRENT CONTROLFILE;
```

次の例では、現行の制御ファイルをデフォルトのディスク・デバイスにバックアップします。

```
BACKUP AS COPY
    CURRENT CONTROLFILE
    FORMAT '/tmp/control01.ct1';
```

次の例では、前述の例で作成した制御ファイルのコピーをテープにバックアップします。

```
BACKUP AS COPY
    CURRENT CONTROLFILE
    FORMAT '/tmp/control01.ct1';
BACKUP DEVICE TYPE sbt
    CONTROLFILECOPY '/tmp/control01.ct1';
```

制御ファイルの自動バックアップ機能が有効になっている場合、Recovery Manager は、これらの例で制御ファイルのバックアップを2つ作成します。1つは BACKUP コマンドで指定したファイルの明示的なバックアップで、もう1つは制御ファイルおよびサーバー・パラメータ・ファイルの自動バックアップです。

**参照：** CONFIGURE CONTROLFILE AUTOBACKUP コマンドについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## 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

8-9 ページの「[Recovery Manager を使用した制御ファイルのバックアップ](#)」で説明したとおり、Recovery Manager は、特定の状況下で、現行のサーバー・パラメータ・ファイルを自動的にバックアップします。BACKUP SPFILE コマンドは、パラメータ・ファイルを明示的にバックアップします。バックアップされるサーバー・パラメータ・ファイルは、インスタンスで現在使用されているサーバー・パラメータ・ファイルです。

### サーバー・パラメータ・ファイルをバックアップする方法

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. ターゲット・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。  
サーバー・パラメータ・ファイルを使用して、データベースを起動している必要があります。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 BACKUP ... SPFILE を実行すると、Recovery Manager によってエラーが発生されます。
3. BACKUP ... SPFILE コマンドを実行します。

次の例では、サーバー・パラメータ・ファイルがテープにバックアップされます。

```
BACKUP DEVICE TYPE sbt SPFILE;
```

## NOARCHIVELOG モードでのデータベースのバックアップ

NOARCHIVELOG モードのデータベースの有効なバックアップは、クローズ状態の一貫性バックアップのみです。次のスクリプトを実行すると、データベースがデータベース全体の一貫性バックアップを行うための正しいモードになり、バックアップされます。次のスクリプトでは、[制御ファイルの自動バックアップ](#)がデータベースに対して有効になっていると想定されています。

### 例 8-8 NOARCHIVELOG モードでのデータベースのバックアップ

```
SHUTDOWN IMMEDIATE;
# Start up the database in case it suffered instance failure or was
# closed with SHUTDOWN ABORT before starting this script.
STARTUP FORCE DBA;
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
# this example uses automatic channels to make the backup
BACKUP
    INCREMENTAL LEVEL 0
    MAXSETSIZE 10M
    DATABASE
    TAG 'BACKUP_1';
# Now that the backup is complete, open the database.
ALTER DATABASE OPEN;
```

読取り専用表領域などの表領域はスキップできますが、バックアップからデータベースをリストアする必要がある場合、最新のバックアップ以降にオフラインまたは読取り専用になっていなかった表領域はスキップすると消失します。

## 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、メディア・リカバリを正常に実行するために重要です。アーカイブ REDO ログは、定期的にバックアップする必要があります。

### アーカイブ REDO ログのバックアップ

Recovery Manager バックアップのいくつかの機能は、アーカイブ REDO ログ専用です。たとえば、BACKUP ... DELETE を使用すると、アーカイブ REDO ログをバックアップ・セットにバックアップした後、そのアーカイブ REDO ログの 1 つのコピーまたはすべてのコピーをディスクから削除できます。

#### アーカイブ REDO ログ・フェイルオーバー

REDO ログが複数のアーカイブ先にアーカイブされている場合でも、Recovery Manager を使用してアーカイブ REDO ログをバックアップすると、Recovery Manager はアーカイブ REDO ログ・ファイルのコピーを 1 つのみ選択してバックアップに含めます。同じログ順序番号を持つログは同一であるため、複数のログ・コピーを含める必要はありません。

**アーカイブ REDO ログ・フェイルオーバー**機能を使用すると、Recovery Manager は、一部のアーカイブ先でログが欠落している場合またはログに破損ブロックが存在する場合でも、バックアップを完了できます。特定のログ順序およびスレッドに対応する 1 つ以上のログが、フラッシュ・リカバリ領域またはいずれかのアーカイブ先で使用可能な場合、Recovery Manager はそのログのバックアップを試みます。バックアップ中にログ・ファイルで破損ブロックが検出された場合、Recovery Manager は、他の出力先で破損ブロックのないそのログのコピーを検索します。

たとえば、/arch1 および /arch2 の 2 つのアーカイブ先に、ログ 121 から 124 をアーカイブするとします。表 8-1 に、制御ファイル内のアーカイブ REDO ログ・レコードを示します。

表 8-1 アーカイブ REDO ログ・レコードの例

| 順序  | /arch1 でのファイル名          | /arch2 でのファイル名          |
|-----|-------------------------|-------------------------|
| 121 | /arch1/archive1_121.arc | /arch2/archive1_121.arc |
| 122 | /arch1/archive1_122.arc | /arch2/archive1_122.arc |
| 123 | /arch1/archive1_123.arc | /arch2/archive1_123.arc |
| 124 | /arch1/archive1_124.arc | /arch2/archive1_124.arc |

ここで、あるユーザーが、Recovery Manager を使用せずに /arch1 ディレクトリからログ 122 および 124 を削除したとします。その後、次のバックアップを実行したとします。

```
BACKUP ARCHIVELOG
  FROM SEQUENCE 121
  UNTIL SEQUENCE 125;
```

フェイルオーバーによって、Recovery Manager は、/arch2 のログ 122 および 124 を使用してバックアップを完了します。

#### オンライン REDO ログ・スイッチ

Recovery Manager のもう 1 つの重要な機能は、自動オンライン REDO ログ・スイッチです。最新のオンライン REDO ログが含まれている、アーカイブ REDO ログのオープンされているデータベースのバックアップを作成するには、次のいずれかの句を指定して BACKUP コマンドを実行します。

- PLUS ARCHIVELOG
- ARCHIVELOG ALL
- ARCHIVELOG FROM ...

Recovery Manager は、バックアップを開始する前に、現行の REDO ログ・グループからの切替えを行い、コマンドの発行時に最新だった REDO ログ・グループまでのアーカイブされていないすべてのオンライン REDO ログをアーカイブします。この機能によって、コマンド開始前に生成されたすべての REDO がバックアップに含まれるようになります。

アーカイブ REDO ログをバックアップする最も効果的な方法の 1 つとして、BACKUP ... PLUS ARCHIVELOG 句を使用する方法があります。これによって、Recovery Manager で次の操作が実行されます。

1. ALTER SYSTEM ARCHIVE LOG CURRENT 文を実行します。
2. BACKUP ARCHIVELOG ALL を実行します。バックアップの最適化が有効になっている場合、Recovery Manager は、指定したデバイスにすでにバックアップされているログをスキップします。
3. BACKUP コマンドに指定された残りのファイルをバックアップします。
4. ALTER SYSTEM ARCHIVE LOG CURRENT 文を実行します。
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。バックアップの最適化が有効になっていない場合、Recovery Manager は、ステップ 1 で生成されたログおよびバックアップ中に生成されたすべてのログをバックアップします。

前述のステップによって、コマンド実行中に作成されるデータファイルのバックアップを一貫性のある状態にリカバリできます。また、バックアップ終了時にオンライン REDO ログがアーカイブされていない場合、そのバックアップで DUPLICATE を実行することはできません。

## アーカイブ REDO ログ・ファイルのバックアップ

アーカイブ・ログをバックアップするには、BACKUP ARCHIVELOG コマンドを使用します。バックアップの最適化が有効になっている場合、Recovery Manager は、指定したデバイスにすでにバックアップされているアーカイブ・ログのバックアップをスキップします。

### アーカイブ REDO ログ・ファイルをバックアップする手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. ターゲット・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。
3. BACKUP ARCHIVELOG または BACKUP ... PLUS ARCHIVELOG コマンドを実行します。

次の例では、データベースおよび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;
```

次の例では、構成済のディスクまたは SBT のチャネルを使用して、すべてのアーカイブ REDO ログの各ログ順序番号のコピーを 1 つバックアップします。

```
BACKUP ARCHIVELOG ALL;
```

また、アーカイブ REDO ログの範囲を、時間、SCN、またはログ順序番号で指定することもできます。次に例を示します。

```
BACKUP ARCHIVELOG
  FROM TIME 'SYSDATE-30'
  UNTIL TIME 'SYSDATE-7';
```

## バックアップが必要なアーカイブ REDO ログのみのバックアップ

Recovery Manager がアーカイブ REDO ログのバックアップを次の方法で自動的にスキップするように指定できます。

- **バックアップの最適化**の構成

5-23 ページの「[バックアップの最適化の構成](#)」で説明されているように、[バックアップの最適化](#)が有効な場合、指定したデバイス・タイプにすでにファイルがバックアップされていると、BACKUP ARCHIVELOG コマンドは同一アーカイブ・ログのバックアップをスキップします。DBID、スレッド、シーケンス番号、RESETLOGS SCN および時間が同じ場合、アーカイブ・ログは別のログと同一とみなされます。

- **アーカイブ REDO ログの削除方針**の構成

5-27 ページの「[アーカイブ REDO ログの削除方針の構成](#)」で説明されているように、削除方針が BACKED UP *integer* TIMES 句で構成されている場合、指定したデバイス・タイプ上に *integer* 個のバックアップがすでに存在していないかぎり、BACKUP ARCHIVELOG コマンドはログをコピーします。ログの *integer* 個のバックアップが存在している場合、BACKUP ARCHIVELOG コマンドはログをスキップします。

BACKUP ... NOT BACKED UP *integer* TIMES コマンドを指定すると、Recovery Manager は、指定したデバイスに *integer* 回以上バックアップされていないアーカイブ・ログ・ファイルのみをバックアップします。ファイルのバックアップの数を決定する場合、Recovery Manager は、現行のバックアップと同じデバイス・タイプに作成されているバックアップのみを考慮します。

BACKED UP 句は、指定したメディアにアーカイブ・ログをバックアップする場合に有効です。たとえば、Recovery Manager がテープに各アーカイブ REDO ログのコピーを 2 つ保持し、追加のバックアップをスキップするように指定できます。

### バックアップが必要なアーカイブ REDO ログをバックアップする手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. ターゲット・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。
3. バックアップに適切なチャネルが構成されていることを確認します。
4. NOT BACKED UP 句を指定して BACKUP ARCHIVELOG コマンドを実行します。

```
BACKUP ARCHIVELOG ALL NOT BACKED UP 2 TIMES;
```

**参照：** NOT BACKED UP の使用例については、9-4 ページの「[バックアップの最適化を使用したファイルのスキップ](#)」を参照してください。

## バックアップ後のアーカイブ REDO ログの削除

BACKUP ARCHIVELOG ... DELETE INPUT コマンドは、アーカイブ・ログ・ファイルをバックアップした後、それらのファイルを削除します。このコマンドを使用すると、アーカイブ REDO ログを手動で削除する手順を実行する必要がなくなります。

DELETE INPUT を実行すると、Recovery Manager は、バックアップ・セットに選択されたアーカイブ・ログの特定のコピーのみを削除します。DELETE ALL INPUT を実行すると、Recovery Manager は、バックアップ済の各アーカイブ REDO ログ・ファイルを、ログのすべてのアーカイブ先から削除します。

5-27 ページの「[アーカイブ REDO ログの削除方針の構成](#)」で説明されているように、BACKUP ... DELETE INPUT および DELETE ARCHIVELOG コマンドは、すべてのアーカイブ場所にあるログに関して[アーカイブ REDO ログの削除方針](#)に従います。たとえば、2 回以上テープにバックアップされたログのみを削除するように指定した場合、BACKUP ... DELETE はこの方針に従います。

次の手順では、/arc\_dest1、/arc\_dest2 およびフラッシュ・リカバリ領域にアーカイブすることを想定しています。

### バックアップ後にアーカイブ REDO ログを削除する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. ターゲット・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。
3. DELETE INPUT 句を指定して BACKUP コマンドを実行します。

次の BACKUP コマンドを実行するとします。

```
BACKUP DEVICE TYPE sbt
  ARCHIVELOG ALL
  DELETE ALL INPUT;
```

この場合、Recovery Manager は、これらのアーカイブ場所にある各ログ順序番号のコピーを1つのみバックアップします。Recovery Manager は、フラッシュ・リカバリ領域内のログは削除しませんが、他のアーカイブ先内のバックアップ済のログのコピーはすべて削除します。削除対象となっているフラッシュ・リカバリ領域内のログは、領域が必要になった場合に自動的にデータベースによって削除されます。

DELETE ALL INPUT ではなく DELETE INPUT を指定すると、Recovery Manager は、バックアップ済の特定のアーカイブ REDO ログ・ファイルのみを削除します。たとえば、Recovery Manager は、/arc\_dest1 内のログがバックアップのソースとして使用された場合はそれらのファイルを削除しますが、/arc\_dest2 の内容はそのまま残します。

#### 参照：

- スタンバイ・データベースでのアーカイブ REDO ログの管理については、『Oracle Data Guard 概要および管理』を参照してください。
- CONFIGURE ARCHIVELOG DELETION POLICY コマンドについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- DELETE ARCHIVELOG コマンドについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- 「[Recovery Manager バックアップおよびアーカイブ REDO ログの削除](#)」(11-20 ページ)

## 増分バックアップの作成および更新

7-13 ページの「[増分バックアップ](#)」で説明されているように、[増分バックアップ](#)では、指定した以前のバックアップ以降に変更されたデータファイル・ブロックのみがコピーされます。増分バックアップは、[累積増分バックアップ](#)または[差分増分バックアップ](#)のいずれかです。

バックアップの内容は同じですが、BACKUP DATABASE と BACKUP INCREMENTAL LEVEL 0 DATABASE は異なります。[全体バックアップ](#)は増分計画の一部として使用できませんが、[レベル0の増分バックアップ](#)は増分計画の基礎となります。Recovery Manager コマンドでは、全体バックアップをレベル0には変更できません。

Recovery Manager は、全体バックアップの場合と同様に、ARCHIVELOG モードでオープンまたはマウントされているデータベースの増分バックアップを作成できます。データベースが NOARCHIVELOG モードの場合、Recovery Manager は、一貫性のある状態でデータベースを停止した後でのみ、増分バックアップを作成できます。



## 増分バックアップの目的

増分バックアップを計画の一環として作成する主な理由は次のとおりです。

- **ブロック・チェンジ・トラッキング**を有効にすると、日次バックアップが高速になります (8-21 ページの「**ブロック・チェンジ・トラッキングを使用した、増分バックアップのパフォーマンスの向上**」を参照)。
- データファイルのイメージ・コピーをロールフォワードする機能によって、リカバリ時間を短縮し、全体バックアップが繰り返されないようにするため
- ネットワークを介してバックアップする場合に帯域幅の消費を削減するため
- テープの書き込み I/O に使用するテープ帯域幅の集計がディスクの読取り I/O に使用するディスク帯域幅の集計より少ない場合にパフォーマンスを向上させるため
- NOLOGGING オプションで作成したオブジェクトへの変更をリカバリできるようにするため  
たとえば、ダイレクト・ロード・インサートでは REDO ログ・エントリが作成されないため、ダイレクト・ロード・インサートによる変更はメディア・リカバリでは再作成できません。ただし、ダイレクト・ロード・インサートではデータ・ブロックが変更されるため、これらのブロックが増分バックアップによって取得されます。
- フィジカル・スタンバイ・データベースをプライマリ・データベースと同期化します。  
Recovery Manager の BACKUP INCREMENTAL FROM SCN コマンドを使用すると、スタンバイの現行の SCN から始まるバックアップをプライマリ・データベースに作成できます。このバックアップを使用して、後でスタンバイ・データベースをロールフォワードできます。増分バックアップをスタンバイ・データベースに適用する方法については、『Oracle Data Guard 概要および管理』を参照してください。

**参照：** NOLOGGING モードの詳細は、『Oracle Database 管理者ガイド』を参照してください。

## 増分バックアップ計画の設計

許容可能な **MTTR** (平均リカバリ時間) に応じて、バックアップ計画を選択します。たとえば、3つのレベルのバックアップ・スキームを実装して、全体バックアップまたはレベル0のバックアップを月に1回、レベル1の累積バックアップを週に1回、レベル1の差分バックアップを毎日作成するように設定できます。この計画では、完全リカバリのために、1日分を超える REDO を適用する必要はありません。

全体バックアップまたはレベル0のバックアップを実行する頻度の目安として、データの20%以上が変更された時点で、レベル0の新しいバックアップを実行するようにします。データベースへの変更率を予想できる場合は、増分バックアップのサイズを監視して、レベル0の新しいバックアップが必要な時点を判断できます。次の SQL 問合せを実行すると、ブロックの20%以上がバックアップされている各データファイルの、レベル1の増分バックアップに書き込まれたブロックの数が表示されます。

```
SELECT  FILE#, INCREMENTAL_LEVEL, COMPLETION_TIME,
        BLOCKS, DATAFILE_BLOCKS
FROM    V$BACKUP_DATAFILE
WHERE   INCREMENTAL_LEVEL > 0
AND     BLOCKS / DATAFILE_BLOCKS > .2
ORDER BY COMPLETION_TIME;
```

レベル1のバックアップのブロック数を、レベル0のバックアップと比較します。たとえば、レベル1の累積バックアップのみを作成する場合は、レベル1の最新のバックアップのサイズがレベル0のバックアップの約半分であれば、レベル0の新しいバックアップを作成します。



ディスク領域を節約する効果的な方法の1つとして、増分バックアップをディスクに作成してから、BACKUP AS BACKUPSET でバックアップをテープにオフロードする方法があります。通常、増分バックアップは全体バックアップより小さいため、テープに移動されるまでの格納に必要な領域は制限されます。ディスク上の増分バックアップをテープにバックアップすると、増分バックアップのすべてのブロックがテープにコピーされるため、テープがストリーム化する可能性があります。Recovery Manager でデータファイル内の変更されたブロックの特定にかかる時間のため、遅延が発生する可能性はありません。

もう1つの方法としては、8-18 ページの「増分更新バックアップ」で説明されている増分更新バックアップを使用する方法があります。この方法では、各データファイルのイメージ・コピーを作成した後、レベル1の増分バックアップを作成および適用して、定期的にこのコピーをロールフォワードします。これによって、データファイルの完全なイメージ・コピーを繰り返し作成することによって発生するオーバーヘッドが回避され、すべてのメリットを活用できるようになります。

Data Guard 環境では、増分バックアップを**フィジカル・スタンバイ・データベース**にオフロードできます。スタンバイ・データベースの増分バックアップとプライマリ・データベースの増分バックアップには互換性があります。つまり、スタンバイ・データベースの増分バックアップをプライマリ・データベースに適用したり、プライマリ・データベースの増分バックアップをスタンバイ・データベースに適用することができます。また、スタンバイ・データベースでは、Data Guard 環境の他のデータベースでブロック・チェンジ・トラッキングが有効になっているかどうかに関係なく、ブロック・チェンジ・トラッキングを有効にできます。これによって、Recovery Manager は、スタンバイ・データベースの増分バックアップを自動的に最適化します。

**参照：** Recovery Manager でスタンバイ・データベースをバックアップする方法については、『Oracle Data Guard 概要および管理』を参照してください。

## 増分バックアップの作成

Recovery Manager を起動した後、Recovery Manager プロンプトで BACKUP INCREMENTAL コマンドを実行します。デフォルトでは、増分バックアップは差分バックアップです。

### 増分バックアップを作成する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. ターゲット・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。
3. 必要なオプションを指定して BACKUP INCREMENTAL コマンドを実行します。

増分レベルを指定するには、LEVEL パラメータを使用します。次の例では、データベースのレベル0の増分バックアップを作成します。

```
BACKUP
  INCREMENTAL LEVEL 0
  DATABASE;
```

次の例では、SYSTEM および tools 表領域の**差分増分バックアップ**をレベル1で作成します。レベル1またはレベル0の最新のバックアップ以降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, tools;
```

次の例では、users 表領域の**累積増分バックアップ**をレベル 1 で作成し、最新のレベル 0 のバックアップ以降に変更されたすべてのブロックをバックアップします。

```
BACKUP  
  INCREMENTAL LEVEL 1 CUMULATIVE  
  TABLESPACE users;
```

## VSS スナップショットの増分バックアップの作成

**Volume Shadow Copy Service (VSS)** を Oracle VSS ライターとともに使用すると、Oracle Database 内のファイルの**シャドウ・コピー**またはスナップショットを作成できます。Oracle VSS ライターで VSS スナップショットを作成するには、Recovery Manager ではなく、サードパーティのバックアップ・プログラムを使用する必要があります。この場合、**フラッシュ・リカバリ領域**によって、VSS スナップショットにすでにバックアップされているファイルの管理が自動化され、必要に応じてそれらのファイルが削除されます。

Recovery Manager で BACKUP INCREMENTAL LEVEL 1 ... FROM SCN コマンドを使用すると、フラッシュ・リカバリ領域に増分バックアップを作成できます。つまり、このコマンドを使用すると、VSS シャドウ・コピーのレベル 1 の増分バックアップを作成できます。Recovery Manager では、リカバリ中に増分バックアップを透過的に適用できます。

**参照：** Recovery Manager を使用した VSS バックアップの作成方法については、Oracle Database のプラットフォーム・ガイドを参照してください。

## 増分更新バックアップ

バックアップを増分更新することによって、データファイルの完全なイメージ・コピー・バックアップを作成する場合に発生するオーバーヘッドを回避できます。また、データベースのメディア・リカバリにかかる時間を最小限に抑えることもできます。たとえば、日次バックアップ・スクリプトを実行する場合、メディア・リカバリのために、1 日分を超える REDO を適用する必要はありません。

### データファイルのバックアップを増分更新する手順

1. データファイルの完全なイメージ・コピー・バックアップを作成します。
2. 定期的（毎日など）に、データファイルのレベル 1 の増分バックアップを作成して、イメージ・コピー・バックアップに適用します。

この方法では、レベル 1 の増分が作成された時点でバックアップがロールフォワードされます。Recovery Manager は、この**増分更新バックアップ**をリストアし、REDO ログから変更を適用できます。この場合、最後に適用されたレベル 1 の増分バックアップの SCN で作成されたデータファイルのバックアップをリストアする場合と同じ結果になります。

---

**注意：** RECOVER COPY コマンドを毎日実行すると、イメージ・コピーは継続的に更新されるため、1 日を超える**リカバリ期間**を満たすことができません。増分更新バックアップ機能は、高速のメディア・リカバリを実現するための最適な方法です。

---

### 増分更新バックアップ：基本例

増分更新バックアップ計画で使用する増分バックアップを作成するには、BACKUP ... FOR RECOVER OF COPY WITH TAG 形式の BACKUP コマンドを使用します。このコマンドについては、増分更新バックアップ計画を実装するサンプル・スクリプトのコンテキストを参照してください。

増分更新バックアップに基づいた計画を実装するには、[例 8-9](#) のスクリプトを定期的に行うことのみが必要となります。

## 例 8-9 基本的な増分更新スクリプト

```

RUN
{
  RECOVER COPY OF DATABASE
  WITH TAG 'incr_update';
  BACKUP
  INCREMENTAL LEVEL 1
  FOR RECOVER OF COPY WITH TAG 'incr_update'
  DATABASE;
}

```

このスクリプトおよび計画を理解するには、データファイルのコピーまたは増分バックアップが存在しない場合の、使用されている2つのコマンドの影響について理解する必要があります。次の2つの点に注意してください。

- 例 8-9 の BACKUP コマンドによって、常にレベル 1 の増分バックアップが作成されるわけではありません。
- 例 8-9 の RECOVER コマンドによって、Recovery Manager は、指定したタグが含まれているデータファイルのコピーのセットに、使用可能なすべてのレベル 1 の増分バックアップを適用します。

次の表に、スクリプトの影響を示します。列には、スクリプトを最初に実行した場合の影響、スクリプトを2回目に実行した場合の影響、スクリプトを3回目およびそれ以降に実行した場合の影響が示されています。

表 8-2 スクリプトを複数回実行した場合の影響

| コマンド    | 1 回目  | 2 回目  | 3 回目以降   |
|---------|---|---|--|
| RECOVER | 増分バックアップまたはデータファイルのコピーが存在しないため、コマンドによってメッセージが生成されます（エラーは生成されません）。つまり、コマンドによる影響はありません。   | データファイルのコピーは存在しますが、そのコピーのリカバリに使用するレベル 1 の増分バックアップは存在しません。つまり、RECOVER コマンドによる影響はありません。 | 前回の実行によって、データファイルのコピーおよびレベル 1 の増分バックアップが存在しています。レベル 1 の増分バックアップがデータファイルのコピーに適用され、そのコピーがレベル 1 の増分バックアップのチェックポイント SCN まで戻されます。 |
| BACKUP  | レベル 0 のイメージ・コピー・バックアップが存在しないため、指定したタグが含まれている、データファイルのイメージ・コピー・バックアップがコマンドによってディスクに作成されます。つまり、スクリプトによって、増分更新のサイクルを開始するために必要なイメージ・コピーが作成されます。<br><b>注意:</b> スクリプトによって DEVICE TYPE sbt が設定される場合は、最初の実行によって、テープ上ではなくディスク上にコピーが作成されます。後続の実行では、テープ上にレベル 1 の増分バックアップが作成されます。 | コマンドによって、前日のブロック変更が含まれているレベル 1 の増分バックアップが作成されます。                                      | コマンドによって、前日のブロック変更が含まれているレベル 1 の増分バックアップが作成されません。  |

例 8-9 の動作については、次の点にも注意してください。

- データファイルがデータベースに追加されるたびに、新しいデータファイルのイメージ・コピーが、次のスクリプト実行時に作成されます。その次の実行では、追加されたデータファイルの、最初のレベル 1 の増分バックアップが作成されます。後続のすべての実行では、新しいデータファイルは他のデータファイルと同様に処理されます。
- 実装する他のバックアップ計画に影響を与えないように、タグを使用して、この計画で使用するために作成されたデータファイルのコピーのレベル 0 の増分バックアップを識別する必要があります。複数の増分バックアップ計画を使用している場合、レベル 0 のバックアップにタグを指定していないかぎり、Recovery Manager はレベル 1 の増分バックアップを確実に作成することはできません。

これらのイメージ・コピーに適用されるレベル 1 の増分バックアップは、データファイルのイメージ・コピーのチェックポイント SCN および使用可能なレベル 1 の増分バックアップのチェックポイント SCN に基づいて選択されます。リカバリされるイメージ・コピーで使用されているタグは、適用されるレベルの増分バックアップを選択する場合の要因にはなりません。

実際には、例 8-9 のスクリプトは、毎日日が変わった後（可能なかぎり午前 0 時）に実行されるようにスケジュールします。スクリプトの 3 回目の実行後は、次のファイルを Point-in-Time リカバリで使用できるようになります。

- 前回（24 時間前に）スクリプトを実行したときのチェックポイント SCN 時点でのデータベースのイメージ・コピー
- 前回実行したときのチェックポイント SCN 以降に行われた変更の増分バックアップ
- イメージ・コピーのチェックポイント SCN と現在の時点の間で行われたすべての変更を含むアーカイブ REDO ログ

スクリプトを実行した後の 24 時間のいずれかの時点で、データベースのリストアおよびリカバリが必要になった場合は、増分更新されたデータファイルのコピーからデータファイルをリストアできます。これによって、必要な SCN に達するまで、レベル 1 の最新の増分バックアップおよび REDO ログから変更を適用できます。最大 24 時間分の適用する REDO があるため、Point-in-Time リカバリにかかる時間は制限されます。

**参照：** Oracle Enterprise Manager での Oracle 推奨バックアップ計画の使用方法については、『Oracle Database 2 日でデータベース管理者』を参照してください。

## 増分更新バックアップ：高度な例

例 8-9 の基本スクリプトを拡張すると、24 時間を超える期間を高速でリカバリできます。例 8-10 に、RECOVER コマンドでリカバリ可能期間の開始時間を指定して、7 日間保持する方法を示します。

### 例 8-10 高度な増分更新スクリプト

```

RUN
{
  RECOVER COPY OF DATABASE
    WITH TAG 'incr_update'
    UNTIL TIME 'SYSDATE - 7';
  BACKUP
    INCREMENTAL LEVEL 1
    FOR RECOVER OF COPY WITH TAG 'incr_update'
    DATABASE;
}

```

次の表に、スクリプトの影響を示します。列には、スクリプトを最初に実行した場合の影響、スクリプトを2から7回目に実行した場合の影響、スクリプトをそれ以降に実行した場合の影響が示されています。

表 8-3 スクリプトを複数回実行した場合の影響

|                 | 1 回目                                     | 2 から 7 回目   | 8 回目以降  |
|-----------------|--|---|---|
| RECOVER<br>コマンド | リカバリするバックアップが存在しないため、何も実行されません。          | UNTIL TIME SYSDATE-7 は将来のある時点を指定しているため、何も実行されません。 | 7 日前に作成されたレベル 1 の増分バックアップが既存のデータベース・コピーに適用されます。 |
| BACKUP<br>コマンド  | レベル 0 のコピーが存在しないため、レベル 0 の増分コピーが作成されません。 | 前日のブロック変更が含まれているレベル 1 の増分バックアップが作成されます。           | 前日以降の変更が含まれている増分バックアップが作成されます。                  |

例 8-9 の基本スクリプトと同様に、データファイルのコピーの SCN と現在の間の任意の時点に高速でリカバリできます。Recovery Manager は、増分バックアップのブロック変更および REDO ログの個々の変更の両方を使用できます。レベル 1 の日次増分バックアップがあるため、1 日分を超える REDO を適用する必要はありません。

**参照：** RECOVER コマンドについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## ブロック・チェンジ・トラッキングを使用した、増分バックアップのパフォーマンスの向上

増分バックアップの**ブロック・チェンジ・トラッキング**機能を使用すると、データファイルごとに変更されたブロックを記録することによってバックアップのパフォーマンスが向上します。

### ブロック・チェンジ・トラッキング

プライマリ・データベースまたはスタンバイ・データベースでブロック・チェンジ・トラッキングが有効になっている場合、Recovery Manager は、**ブロック・チェンジ・トラッキング・ファイル**を使用して、増分バックアップ用に変更されたブロックを識別します。この小さなビットマップ・ファイルを読み取り、変更されたブロックを確認することによって、バックアップしているデータファイルのすべてのブロックのスキャンを回避します。

ブロック・チェンジ・トラッキングは、デフォルトでは無効になっています。ただし、バックアップ中にデータファイル全体をスキャンする必要がなくなるというメリットは無視できません。バックアップを実行してから次のバックアップを実行するまでの間に変更されたデータ・ブロックが少量の場合は特にです。バックアップ計画に増分バックアップが含まれている場合は、ブロック・チェンジ・トラッキングを有効にすることをお勧めします。ブロック・チェンジ・トラッキングによって、増分バックアップの実行に使用されるコマンドが変更されることはありません。通常、初期構成後にチェンジ・トラッキング・ファイル自体にメンテナンスを行う必要はほとんどありません。

**ブロック・チェンジ・トラッキング・ファイルの領域管理** チェンジ・トラッキング・ファイルには、バックアップ間のデータファイルの変更をマークするビットマップが保持されます。データベースでは、各バックアップの前にビットマップの切替えが実行されます。Oracle Database では、最新の 8 つのバックアップに対応するブロック・チェンジ・データを保持するためにチェンジ・トラッキング・ファイルの領域が自動的に管理されます。最大 8 つのビットマップに達すると、最新のビットマップは現行の変更を追跡するビットマップによって上書きされます。

最初のレベル 0 の増分バックアップでは、データベース全体がスキャンされます。それ以降の増分バックアップでは、ブロック・チェンジ・トラッキング・ファイルを使用して、最後のバックアップ以降に変更されたとマークされているブロックのみがスキャンされます。増分バックアップは、ブロック・チェンジ・トラッキング・ファイル内の最も古いビットマップの開始後に作成された親ビットマップに基づいている場合にのみ最適化できます。

増分バックアップ計画を作成する場合は、ビットマップ数の制限 (8) を考慮してください。たとえば、レベル 0 のデータベース・バックアップを作成した後で 7 つの差分増分バックアップを作成すると、ブロック・チェンジ・トラッキング・ファイルには 8 つのビットマップが含まれます。次にレベル 1 の累積増分バックアップを作成すると、レベル 0 の親バックアップに対応するビットマップが現行の変更を追跡するビットマップで上書きされるため、Recovery Manager はバックアップを最適化できません。

**ブロック・チェンジ・トラッキング・ファイルの場所** データベース全体に対して、1 つのブロック・チェンジ・トラッキング・ファイルが作成されます。デフォルトでは、チェンジ・トラッキング・ファイルは、DB\_CREATE\_FILE\_DEST 初期化パラメータで指定された出力先に **Oracle 管理ファイル** として作成されます。また、ブロック・チェンジ・トラッキング・ファイルは、選択した場所に格納することによってその名前を指定することもできます。ブロック・チェンジ・トラッキング・ファイルに RAW デバイス (ファイル・システムを持たないディスク) を使用できますが、デバイスは十分に大きい必要があります。

---

**注意:** Oracle RAC 環境では、チェンジ・トラッキング・ファイルは、クラスタ内のすべてのノードからアクセスできる共有ストレージに格納する必要があります。

---

Recovery Manager では、チェンジ・トラッキング・ファイルのバックアップおよびリカバリはサポートされていません。データベースは、チェンジ・トラッキング・ファイルが無効であると判断した場合、チェンジ・トラッキング・ファイルを再設定します。データベース全体またはサブセットをリストアおよびリカバリする場合、データベースは、ブロック・チェンジ・トラッキング・ファイルを消去し、変更の追跡を再度開始します。レベル 0 の増分バックアップを作成した後の次の増分バックアップでは、チェンジ・トラッキング・データを使用できます。

**ブロック・チェンジ・トラッキング・ファイルのサイズ** ブロック・チェンジ・トラッキング・ファイルのサイズは、データベースのサイズおよび REDO の有効になっているスレッドの数に比例します。ブロック・チェンジ・トラッキング・ファイルのサイズは、データベースの変更に応じて増減します。このサイズは、データベースの更新頻度とは関係ありません。

通常、ブロック・チェンジ・トラッキングに必要な領域は、追跡するデータ・ブロックのサイズの約 1/30,000 です。この見積りが示すサイズよりファイルが大きくなる場合の要因を次に示します。

- データベースの拡張に応じて領域を割り当てた場合にオーバーヘッドが発生しないようにするために、ブロック・チェンジ・トラッキング・ファイルのサイズは 10MB から始まります。新しい領域は 10MB ずつ割り当てられます。つまり、ファイルのサイズは、約 300GB までのデータベースに対しては 10MB 以上、約 600GB までのデータベースに対しては 20MB 以上になります。
- データファイルのサイズとは関係なく、データファイルごとに 320KB 以上の領域がブロック・チェンジ・トラッキング・ファイルに割り当てられます。このため、比較的小さなデータファイルが多数ある場合は、同じデータを含む大きなデータファイルが少数ある場合よりチェンジ・トラッキング・ファイルのサイズが大きくなります。

## ブロック・チェンジ・トラッキングの有効化および無効化

ブロック・チェンジ・トラッキングは、データベースがオープンまたはマウントされている場合に有効にできます。この項では、ブロック・チェンジ・トラッキングを Oracle 管理ファイルとして **データベース領域** に作成することを想定しています。データベース領域とは、データファイル、制御ファイル、オンライン REDO ログ・ファイルなどのアクティブなデータベース・ファイルをデータベースが保持する場所です。データベースおよび **フラッシュ・リカバリ領域** については、5-15 ページの「**フラッシュ・リカバリ領域の概要**」を参照してください。



**ブロック・チェンジ・トラッキングを有効にする手順**

1. SQL\*Plus を起動し、管理者権限でターゲット・データベースに接続します。
2. DB\_CREATE\_FILE\_DEST 初期化パラメータが設定されていることを確認します。

```
SHOW PARAMETER DB_CREATE_FILE_DEST
```

このパラメータが設定されていない状態でデータベースがオープンされている場合は、次の形式の ALTER SYSTEM 文を使用してこのパラメータを設定できます。

```
ALTER SYSTEM SET
  DB_CREATE_FILE_DEST = '/disk1/bct/'
  SCOPE=BOTH SID='*';
```

3. ブロック・チェンジ・トラッキングを有効にします。

次の ALTER DATABASE 文を実行します。

```
ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;
```

また、次の形式の SQL 文を使用して、自分で選択した場所にチェンジ・トラッキング・ファイルを作成することもできます。

```
ALTER DATABASE ENABLE BLOCK CHANGE TRACKING
  USING FILE '/mydir/rman_change_track.f' REUSE;
```

REUSE オプションは、Oracle Database に、指定した名前を持つ既存のブロック・チェンジ・トラッキング・ファイルを上書きするように指示します。

**ブロック・チェンジ・トラッキングの無効化**

この項では、ブロック・チェンジ・トラッキング機能が現在有効になっていると想定しています。ブロック・チェンジ・トラッキングを無効にすると、データベースはオペレーティング・システムからブロック・チェンジ・トラッキング・ファイルを削除します。

**ブロック・チェンジ・トラッキングを無効にする手順**

1. SQL\*Plus を起動し、管理者権限でターゲット・データベースに接続します。
2. ターゲット・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。
3. ブロック・チェンジ・トラッキングを無効にします。

次の ALTER DATABASE 文を実行します。

```
ALTER DATABASE DISABLE BLOCK CHANGE TRACKING;
```

**チェンジ・トラッキングが有効かどうかの確認**

V\$BLOCK\_CHANGE\_TRACKING ビューを問い合わせ、チェンジ・トラッキングが有効になっているかどうかを確認できます。有効になっている場合は、ブロック・チェンジ・トラッキング・ファイルのファイル名も確認できます。

**チェンジ・トラッキングが有効かどうかを確認する方法**

- SQL\*Plus で次の問合せを入力します（出力例も示します）。

```
COL STATUS      FORMAT A8
COL FILENAME    FORMAT A60
```

```
SELECT STATUS, FILENAME
FROM   V$BLOCK_CHANGE_TRACKING;
```

```
STATUS      FILENAME
```

```
-----
ENABLED    /disk1/bct/RDBMS/changetracking/o1_mf_2f71np5j_.chg
```

## ブロック・チェンジ・トラッキング・ファイルの場所の変更

チェンジ・トラッキング・ファイルを移動するには、ALTER DATABASE RENAME FILE 文を使用します。データベースはマウントされている必要があります。この文は、新しい場所を参照するように制御ファイルを更新し、チェンジ・トラッキング・ファイルの内容を保持します。データベースを停止できない場合は、ブロック・チェンジ・トラッキングを無効にしてから有効にすることができます。この場合、既存のブロック・チェンジ・トラッキング・ファイルの内容は消失します。

### チェンジ・トラッキング・ファイルの場所を変更する手順

1. SQL\*Plus を起動し、ターゲット・データベースに接続します。
2. 必要に応じて、チェンジ・トラッキング・ファイルの現在の名前を確認します。

```
SQL> SELECT FILENAME FROM V$BLOCK_CHANGE_TRACKING;
```

3. 可能な場合は、データベースを停止します。たとえば、次のように入力します。

```
SQL> SHUTDOWN IMMEDIATE
```

データベースを停止する場合は、次の手順にスキップします。データベースを停止しない場合は、次の SQL 文を実行し、残りの手順をすべてスキップします。

```
SQL> ALTER DATABASE DISABLE BLOCK CHANGE TRACKING;
SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING USING FILE 'new_location';
```

この場合、ブロック・チェンジ・トラッキング・ファイルの内容は消失します。次回レベル 0 の増分バックアップを完了するまで、Recovery Manager はファイル全体をスキャンする必要があります。

4. ホスト・オペレーティング・システムのコマンドを使用して、チェンジ・トラッキング・ファイルを新しい場所に移動します。
5. データベースをマウントして、より多くの領域がある場所にチェンジ・トラッキング・ファイルを移動します。たとえば、次のように入力します。

```
ALTER DATABASE RENAME FILE
  '/disk1/bct/RDEMS/changetracking/o1_mf_2f71np5j_.chg' TO
  '/disk2/bct/RDEMS/changetracking/o1_mf_2f71np5j_.chg';
```

この文は、内容を保持した状態でチェンジ・トラッキング・ファイルの場所を変更します。

6. データベースをオープンします。

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

**参照：** ALTER DATABASE 文および ALTER SYSTEM 文については、『Oracle Database SQL 言語リファレンス』を参照してください。

## 長期格納用のデータベース・バックアップの作成

この項では、長期格納用のバックアップを作成する場合の基本的な概念およびタスクについて説明します。

### アーカイブ・バックアップの目的

BACKUP ... KEEP を使用すると、[バックアップの保存方針](#)から除外される包括的なバックアップを作成できます。データベースのリストアおよびリカバリに必要なすべてのファイルが単一のディスクまたはテープにバックアップされるため、このバックアップは包括的になります。また、KEEP オプションは、保存方針からのバックアップの除外を永続的または指定した期間行うように指定します。BACKUP ... KEEP で作成されるバックアップの一般名は [アーカイブ・バックアップ](#) です。



1-3 ページの「データの保持」で説明されているように、バックアップおよびリカバリ計画の目的の1つは、データを保存することです。BACKUP ... KEEP を使用すると、保存方針に指定された期間より長くデータベースのバックアップを保持できます。たとえば、規定の要件を満たすために毎年元旦にデータベースをバックアップして、メディアをオフサイトに格納することができます。アーカイブ・バックアップを作成して数年後に、このバックアップをリストアおよびリカバリしてバックアップ時のデータの状態を問い合わせることができます。

アーカイブ・バックアップのもう1つの目的は、テストのためにリストアするバックアップを作成し、後で削除することです。たとえば、データベースをバックアップしてテスト環境でリストアした後、テスト・データベースが操作可能になったらアーカイブ・バックアップを破棄することができます。また、これと関連して、別のユーザーまたはホストへの転送完了後に削除可能な自己完結型のバックアップを作成するという目的もあります。たとえば、別のユーザーが、レポートまたはテスト用にデータベースのコピーを必要とする場合があります。

## アーカイブ・バックアップの基本的な概念

BACKUP コマンドで KEEP オプションを使用すると、バックアップを保存方針から除外できます。また、CHANGE コマンドで KEEP および NOKEEP オプションを使用すると、既存のバックアップのステータスを変更できます。KEEP 属性を使用したバックアップは、他のすべてのバックアップと同様にリカバリできる有効なバックアップです。

KEEP UNTIL TIME 句を使用してアーカイブ・バックアップの終了日を指定したり、FOREVER でバックアップを永続的に保持するように指定できます。UNTIL を指定した場合、構成されている保存方針に関係なく、UNTIL で指定した時間が経過すると、バックアップは Recovery Manager によって不要とマークされます。たとえば、KEEP UNTIL TIME '01-JAN-08' と指定すると、1月1日の深夜0時を1秒経過した後にバックアップが不要とマークされます。UNTIL TIME を午後9時に指定すると、午後9時1分にバックアップが不要とマークされません。

BACKUP コマンドに KEEP を指定すると、Recovery Manager は複数のバックアップ・セットを生成します。BACKUP ... KEEP コマンドには次の特性があります。

- データファイル、制御ファイル（**制御ファイルの自動バックアップ**が無効になっている場合でも）およびサーバー・パラメータ・ファイルが自動的にバックアップされます。
- データベースのバックアップを一貫性のある状態にリカバリできるように、アーカイブ REDO ログのバックアップが自動的に生成されます。
- FORMAT、POOL または TAG パラメータを指定した場合、これらのパラメータはすべてのバックアップに使用されます。このため、FORMAT 文字列で、複数のバックアップ・ピースを作成できるようにする必要があります。%U 置換変数を指定すると、最も簡単にこの要件を満たすことができます。
- **通常のリストア・ポイント**を作成するオプションの RESTORE POINT 句がサポートされています。通常のリストア・ポイントとは、バックアップを一貫性のある状態にするためにリカバリする必要がある時点の SCN のラベルのことです。この SCN は、データファイルのバックアップの完了直後に取得されます。Recovery Manager では、**リカバリ・カタログ**を使用してリストア・ポイントが再同期化され、バックアップが存在するかぎり、そのリストア・ポイントが保持されます。リストア・ポイントの表示方法については、10-10 ページの「**リストア・ポイントの表示**」を参照してください。

**参照：** CHANGE 構文については『Oracle Database バックアップおよびリカバリ・リファレンス』、BACKUP ... KEEP 構文については『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## 長期格納用のアーカイブ・バックアップの作成

この項では、アーカイブを目的としたデータベース・バックアップの作成方法について説明します。通常、アーカイブ・バックアップはテープに作成します。データ保護を目的としたバックアップは、アクセス可能な状態のまま再利用されるテープのセットに格納されることが多いため、アーカイブ・バックアップ用にテープのセットを取っておくことをお勧めします。この特別なテープのセットにアーカイブ・バックアップを書き込んだ後、オフサイトの保管場所に格納することができます。

この項では、アーカイブ・バックアップを作成する場合の標準的な方法について説明します。手順を変更すると、動的に更新できる**ストアド・スクリプト**またはシェル・スクリプトを作成できます。このスクリプトの実行時に、リストア・ポイントの名前、バックアップ形式などを動的に設定できます。

### 参照：

- アーカイブ・バックアップをリストアする場合の推奨方法については、23-21 ページの「**DUPLICATE を使用したアーカイブ・バックアップのリストア**」を参照してください。
- Recovery Manager コマンド・ファイルを使用してアーカイブ・バックアップを作成する方法については、4-5 ページの「**コマンド・ファイルでの置換変数の使用**」を参照してください。
- Recovery Manager コマンド・ファイルを使用してアーカイブ・バックアップを作成する方法については、12-19 ページの「**動的ストアド・スクリプトの作成および実行**」を参照してください。

## アーカイブ・バックアップの作成

次の例では、QUARTERLY というバックアップの**タグ**を使用して長期用のアーカイブ・バックアップを作成し、長期格納用に確保されている Oracle Secure Backup の専用のテープ・ファミリーに割り当てます。この例では、次の点に注意してください。

- FOREVER キーワードは、このバックアップがバックアップ**保存方針**による削除の対象にはならないことを示します。
- BACKUP コマンドは、このバックアップが一貫性のある状態となる SCN と一致する FY06Q4 という名前のリストア・ポイントを作成します。

### 長期用のアーカイブ・バックアップを作成する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログに接続します。

ターゲット・データベースは、オープンされている状態でもマウントされている状態でもかまいません。リカバリ・カタログは、KEEP FOREVER には必要ですが、その他の KEEP オプションには必要ありません。

2. BACKUP ... KEEP を実行して、バックアップを作成します。

次の例では、データファイルおよびアーカイブ・ログのバックアップを生成し、**通常のリストア・ポイント**を作成します。指定したリストア・ポイントは、存在していない必要があることに注意してください。

ログのバックアップには、このバックアップを一貫性のある状態にリストアするためのアーカイブ・ログのみが含まれています。この新しいバックアップを一貫性のある状態にするために必要な、現行のオンライン・ログにある REDO をアーカイブするために、データベースでオンライン REDO ログの切替えが実行されます。制御ファイルの自動バックアップにはリストア・ポイントのコピーが含まれているため、制御ファイルをリストアするとすぐにこのコピーを参照できます。

```

RUN
{
  ALLOCATE CHANNEL c1
    DEVICE TYPE sbt
    PARMS 'ENV=(OB_MEDIA_FAMILY=archival_backup)';
  BACKUP DATABASE
    TAG quarterly
    KEEP FOREVER
    RESTORE POINT FY06Q4;
}

```

次の例では、バックアップを永続的ではなく、365 日間保存します。1 年が経過すると、バックアップ保存方針に関係なく、バックアップは不要とマークされます。

```

RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt
    PARMS 'ENV=(OB_MEDIA_FAMILY=archival_backup)';
  BACKUP DATABASE
    TAG quarterly
    KEEP UNTIL TIME 'SYSDATE+365'
    RESTORE POINT FY06Q4;
}

```

**参照：** リストア・ポイントについては、5-29 ページの「[リストア・ポイントおよびフラッシュバック・データベース](#)」を参照してください。

## 一時的なアーカイブ・バックアップの作成

アーカイブ・バックアップの目的の 1 つは、テスト・データベースを作成することです。テスト・データベースの作成方法は、8-26 ページの「[長期格納用のアーカイブ・バックアップの作成](#)」で説明されている方法と基本的に同じです。異なる点は、バックアップを、作成後すぐに削除することです。

BACKUP ... KEEP UNTIL パラメータを使用すると、バックアップの一時的なステータスを指定できます。バックアップを作成後、そのバックアップを同じ日に新しいホストにリストアするとします。この場合、KEEP UNTIL TIME SYSDATE+1 を指定して、1 日間のみこのバックアップの保存方針を上書きするように Recovery Manager に指示できます。1 日が経過すると、構成されているバックアップ保存方針に関係なく、バックアップは不要とマークされます。

**例 8-11** のコマンドは、タグ TESTDB が含まれているアーカイブ・バックアップを一時ディスクに作成します。この例では、バックアップをリカバリする時点のラベルとなる通常のリストア・ポイントを作成します。バックアップ中にデータベースがオープンしている場合、Recovery Manager はアーカイブ REDO ログのみをバックアップします。アーカイブ・ログは、オフライン・バックアップでは必要ないためバックアップされません。

### 例 8-11 一時的なアーカイブ・バックアップの作成

```

BACKUP DATABASE
  FORMAT '/disk1/oraclebck/%U'
  TAG TESTDB
  KEEP UNTIL TIME 'SYSDATE+1'
  RESTORE POINT TESTDB06;

```

アーカイブ・バックアップをリストアする場合は、DUPLICATE コマンドを使用する方法をお勧めします。詳細は、23-21 ページの「[DUPLICATE を使用したアーカイブ・バックアップのリストア](#)」を参照してください。

## Recovery Manager バックアップのバックアップ

この項では、バックアップ・セットおよびイメージ・コピーのバックアップ方法について説明します。

### バックアップのバックアップ

BACKUP BACKUPSET コマンドを使用すると、他のバックアップ・ジョブによって作成されたバックアップ・セットをバックアップできます。また、BACKUP RECOVERY AREA を使用すると、現行および以前のすべての**フラッシュ・リカバリ領域**に作成されたリカバリ・ファイルをバックアップすることもできます。リカバリ・ファイルとは、全体および増分バックアップ・セット、制御ファイルの自動バックアップ、データファイルのコピーおよびアーカイブ REDO ログのことです。BACKUP RECOVERY AREA では、SBT バックアップのみがサポートされています。

これらのコマンドは、特に次の場合に有効です。

- すべてのバックアップをディスクおよびテープの両方に存在させる場合。
- バックアップをディスクからテープに移動して、ディスク上の領域を解放する場合。このタスクは、必要に応じて領域を再利用できるようにデータベースでフラッシュ・リカバリを使用している場合、特に重要です。

---

**注意：** BACKUP BACKUPSET を実行する場合は、バックアップを多重化できません。BACKUP BACKUPSET を実行すると、指定したメディア上にコピーが常に1つのみ作成されます。

---

また、BACKUP COPY OF コマンドを使用すると、データファイル、制御ファイルおよびアーカイブ REDO ログのイメージ・コピーをバックアップすることもできます。このコマンドでは、バックアップ・セットまたはイメージ・コピーのいずれかを出力できるため、イメージ・コピーからバックアップ・セットを生成できます。ディスク上にイメージ・コピーとして作成されたデータベースのバックアップをテープにバックアップするには、この形式のバックアップを使用します。

### バックアップ・セットの複数のコピー

BACKUP BACKUPSET を実行すると、バックアップ・セットにバックアップ・ピースの追加コピーが作成されますが、新しいバックアップ・セットは作成されません。つまり、BACKUP BACKUPSET は、BACKUP の DUPLEX または MAXCOPIES オプションを使用する場合と類似しています (9-7 ページの「[バックアップ・セットの多重化](#)」を参照)。他の形式の BACKUP コマンドによって生成されたバックアップ・セットのコピーが個別のバックアップ・セットにならないのと同様に、BACKUP BACKUPSET によって作成されたバックアップ・セットの追加コピーも新しいバックアップ・セットにはなりません。

### バックアップをバックアップする場合のバックアップの保存方針の影響

[冗長性](#)に基づいたバックアップ**保存方針**では、バックアップ・セットはバックアップの1つのインスタンスとみなされます。これは、バックアップ・セットを構成するバックアップ・ピースの複数のコピーが存在する場合（バックアップ・セットがディスクからテープにバックアップされている場合など）でも該当します。

リカバリ期間に基づく保存方針では、バックアップ・セットのすべてのコピーが不要とみなされるか、またはすべてが不要でないかとみなされます。これは、LIST および REPORT コマンドの出力を参照すると最も簡単に理解できます。

**バックアップをバックアップする場合のバックアップの保存方針の影響を表示する手順**

1. データファイルをバックアップします。

この例では、データファイル 5 をバックアップします。

```
BACKUP AS BACKUPSET DATAFILE 5;
```

2. 前の手順で作成したデータファイルのバックアップに対して LIST 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（出力例も示します）。

```
LIST BACKUP OF DATAFILE 5 SUMMARY;
```

```
List of Backups
=====
Key      TY LV S Device Type Completion Time #Pieces #Copies Compressed Tag
-----
18       B F A DISK          04-AUG-07      1      1      NO
TAG20070804T160 134
```

3. 前の手順のバックアップ・セット・キーを使用して、バックアップ・セットをバックアップします。

たとえば、次のコマンドを入力します。

```
BACKUP BACKUPSET 18;
```

4. 手順 2 で実行した LIST コマンドを再度実行します。

たとえば、次のコマンドを実行します（出力例も示します）。

```
LIST BACKUP OF DATAFILE 5 SUMMARY;
```

```
List of Backups
=====
Key      TY LV S Device Type Completion Time #Pieces #Copies Compressed Tag
-----
18       B F A DISK          04-AUG-07      1      2      NO
TAG20070804T160 134
```

この出力に表示されているバックアップ・セットは 1 つのみですが、これでバックアップ・セットのコピーは 2 つ存在しています。

5. レポートを生成して、冗長性に基づくバックアップの保存方針でのこれらのコピーの影響を確認します。

たとえば、次のコマンドを発行します。

```
REPORT OBSOLETE REDUNDANCY 1;
```

バックアップ・セットの両方のコピーの `set_stamp` と `set_count` の値が同じであるため、いずれのコピーも不要とはみなされません。

6. レポートを生成して、リカバリ期間に基づくバックアップの保存方針でのこれらのコピーの影響を確認します。

たとえば、次のコマンドを発行します。

```
REPORT OBSOLETE RECOVERY WINDOW 1 DAY;
```

現在の時点と他のバックアップの可用性に関して、バックアップ・セットのいずれのコピーも不要であるとはみなされず、またこのバックアップ・セットの `CHECKPOINT_CHANGE#` に基づくとはみなされません。

**参照：**

- 「冗長性に基づく保存方針の構成」(5-22 ページ)
- LIST および REPORT コマンドの使用方法については、[第 10 章「Recovery Manager 操作に関するレポート」](#)を参照してください。

**Recovery Manager を使用したバックアップ・セットのバックアップ**

この項では、BACKUP BACKUPSET コマンドを使用して、バックアップ・セットをディスクからテープにコピーする方法について説明します。この手順では、デフォルト・デバイスとして **SBT** デバイスがすでに構成されていることを想定しています。

**ディスクからテープにバックアップ・セットをバックアップする手順**

1. 使用可能なバックアップ・セットのサブセットをバックアップする場合は、LIST BACKUPSET コマンドを実行してそれらの主キーを取得します。

次の例では、バックアップ・セットをサマリー形式で表示します。

```
RMAN> LIST BACKUPSET SUMMARY;
```

```
List of Backups
```

```
=====
```

| Key | TY | LV | S | Device | Type | Completion Time | #Pieces | #Copies | Comp | Tag                |
|-----|----|----|---|--------|------|-----------------|---------|---------|------|--------------------|
| 1   | B  | F  | A | DISK   |      | 28-MAY-07       | 1       | 1       | NO   | TAG20070528T132432 |
| 2   | B  | F  | A | DISK   |      | 29-MAY-07       | 1       | 1       | NO   | TAG20070529T132433 |
| 3   | B  | F  | A | DISK   |      | 30-MAY-07       | 1       | 1       | NO   | TAG20070530T132434 |

次の例では、バックアップ・セット 3 を詳細形式で表示します。

```
RMAN> LIST BACKUPSET 3;
```

```
List of Backup Sets
```

```
=====
```

| BS Key | Type | LV | Size  | Device | Type | Elapsed Time | Completion Time |
|--------|------|----|-------|--------|------|--------------|-----------------|
| 3      | Full |    | 8.33M | DISK   |      | 00:00:01     | 30-MAY-07       |

BP Key: 3 Status: AVAILABLE Compressed: NO Tag: TAG20070530T132434  
 Piece Name: /disk1/oracle/dbs/c-35764265-20070530-02  
 Control File Included: Ckp SCN: 397221 Ckp time: 30-MAY-07  
 SPFILE Included: Modification time: 30-MAY-07  
 SPFILE db\_unique\_name: PROD

2. BACKUP BACKUPSET コマンドを実行します。

次の例では、ディスクの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、入力ディスクのバックアップを削除します。

```
BACKUP BACKUPSET ALL  
DELETE INPUT;
```

次の例では、主キー 1 および 2 を含むバックアップ・セットのみをテープにバックアップし、入力ディスクのバックアップを削除します。

```
BACKUP BACKUPSET 1,2  
DELETE INPUT;
```

3. 必要に応じて、LIST コマンドを実行して、バックアップ・セットおよびバックアップ・ピースのリストを表示します。

BACKUP BACKUPSET によって作成されたバックアップ・ピースのコピーを含むすべてのコピーが出力に含まれます。

## Recovery Manager を使用したイメージ・コピーのバックアップ

この項では、BACKUP コマンドを使用してイメージ・コピーをテープにバックアップする方法について説明します。デフォルト・デバイスとして **SBT** デバイスが構成されていることを想定しています。

データファイルの複数のコピーを含むイメージ・コピーをバックアップする場合は、バックアップにタグを指定すると、入力イメージ・コピーの識別が簡単になります。データファイルのすべてのイメージ・コピーにタグが指定されます。イメージ・コピーが新しいイメージ・コピーとしてバックアップされると、デフォルトで、そのイメージ・コピーのタグが継承されます。

### ディスクからテープにイメージ・コピーをバックアップする手順

1. BACKUP ... COPY OF または BACKUP DATAFILECOPY コマンドを発行します。

次の例では、タグ DBCopy が含まれているデータファイルのコピーをバックアップします。

```
BACKUP DATAFILE COPY FROM TAG monDBCOPY;
```

次の例では、データベースの最新イメージ・コピーをテープにバックアップし、QUARTERLY\_BACKUP というタグを割り当て、入力のディスク・バックアップを削除します。

```
BACKUP DEVICE TYPE sbt
  TAG "quarterly_backup"
  COPY OF DATABASE
  DELETE INPUT;
```

2. 必要に応じて、LIST コマンドを実行して、バックアップ・セットのリストを表示します。BACKUP BACKUPSET によって作成されたバックアップ・ピースのコピーを含むすべてのコピーが出力に含まれます。





---

## データベースのバックアップ：高度なトピック

この章では、Recovery Manager の高度なバックアップ手順について説明します。この章の内容は、次のとおりです。

- Recovery Manager バックアップ・セットのサイズの制限
- バックアップの最適化を使用したファイルのスキップ
- オフライン・ファイル、読取り専用ファイルおよびアクセスできないファイルのスキップ
- バックアップ・セットの多重化
- Recovery Manager を使用したミラーの分割によるバックアップの実行
- Recovery Manager バックアップの暗号化
- Recovery Manager バックアップの再開
- バックアップ期間の管理

**参照：** 基本的なバックアップ手順については、第8章「データベースのバックアップ」を参照してください。

## Recovery Manager バックアップ・セットのサイズの制限

6-4 ページの「バックアップ・セットの最大サイズの構成」の説明に従って CONFIGURE コマンドを使用すると、バックアップ・セットのサイズを制御する永続的な設定を作成できます。この制御は、大規模なファイルをバックアップする場合に有効です。バックアップ・セットのサイズを永続的に構成しない場合は、BACKUP ... MAXSETSIZE コマンドを使用して、バックアップ・セットのサイズを制限することができます。

BACKUP コマンドではなく、CONFIGURE コマンドを使用すると、個々のバックアップ・ピースのサイズに制限を設定できます。この制御は、ファイル・サイズに制限があるメディア・マネージャを使用する場合、または大規模なファイルをバックアップする必要がある場合に特に有効です。詳細は、6-5 ページの「バックアップ・ピースの最大サイズの構成」を参照してください。

### バックアップ・セットのサイズ

BACKUP コマンドの MAXSETSIZE パラメータには、バイト（デフォルト）、KB、MB または GB 単位でバックアップ・セットの最大サイズを指定します。したがって、バックアップ・セットのサイズを 305MB に制限するには、MAXSETSIZE 305M と指定します。Recovery Manager は、すべてのバックアップ・セットをこのサイズに制限します。

BACKUP ... MAXSETSIZE を使用して、データベースが複数のバックアップ・セットに分割されるように、バックアップ・セットのサイズを制限できます。バックアップの途中で障害が発生した場合は、再開可能バックアップ機能を使用して、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できます。Recovery Manager バックアップを再起動する方法については、9-13 ページの「Recovery Manager バックアップの再開」を参照してください。

場合によっては、MAXSETSIZE の値が小さすぎて、バックアップ中の最大のファイルを格納できない場合があります。MAXSETSIZE が小さすぎるかどうかを判断する場合、Recovery Manager は、圧縮後のファイル・サイズではなく元のデータファイルのサイズを使用します。Recovery Manager は、次のようなエラー・スタックを表示します。

```

RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of backup command at 11/03/06 14:40:33
RMAN-06182: archive log larger than MAXSETSIZE: thread 1 seq 1
              /oracle/oradata/trgt/arch/archive1_1.dbf

```

**参照：** CONFIGURE MAXSETSIZE コマンドについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

### BACKUP ... MAXSETSIZE を使用したバックアップ・セットのサイズの制限

バックアップ・ピースのサイズがファイル・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・サイズより大きい場合、問題が発生します。バックアップ・ピースのサイズを制限するには、CONFIGURE CHANNEL コマンドまたは ALLOCATE CHANNEL コマンドの MAXSETSIZE パラメータを使用します。

#### バックアップ・セットのサイズを制限する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. MAXSETSIZE パラメータを指定して BACKUP コマンドを実行します。

次の例では、各バックアップ・セットのサイズを 100MB に制限して、アーカイブ・ログをテープにバックアップします。

```

BACKUP DEVICE TYPE sbt
      MAXSETSIZE 100M
      ARCHIVELOG ALL;

```

## セクションへの大規模なデータファイルのバックアップの分割

BACKUP コマンドに SECTION SIZE パラメータを指定すると、Recovery Manager によって、各バックアップ・ピースに 1 つのファイル・セクションのブロックが含まれているバックアップ・セットが作成されます。**ファイル・セクション**とは、ファイル内の連続するブロックの範囲のことです。このタイプのバックアップは**マルチセクション・バックアップ**と呼ばれます。

---

**注意：** SECTION SIZE は、MAXPIECESIZE とともに指定することはできません。

---

マルチセクション・バックアップの目的は、Recovery Manager チャンネルが単一の大規模なファイルを実行してバックアップできるようにすることです。Recovery Manager は、複数のチャンネルに作業を分割し、各チャンネルでファイル内の 1 つのファイル・セクションをバックアップします。個別のセクションでファイルをバックアップすることによって、大規模データファイルのバックアップのパフォーマンスを向上させることができます。

マルチセクション・バックアップが正常に完了した場合は、バックアップ時に生成されたバックアップ・セットに、部分的なデータファイルは含まれません。マルチセクション・バックアップに失敗すると、Recovery Manager メタデータに部分的なバックアップ・セットのレコードが含まれる可能性があります。Recovery Manager は、部分的なバックアップをリストアおよびリカバリ対象とみなしません。DELETE コマンドを使用して、部分的なバックアップ・セットを削除する必要があります。

ファイルのサイズより大きいセクション・サイズを指定した場合、Recovery Manager はファイルのマルチセクション・バックアップを使用しません。小さなセクション・サイズを指定した結果、セクションの数が 256 を超えると、Recovery Manager は、正確に 256 になる値までセクション・サイズを増やします。

### マルチセクション・バックアップを作成する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. 必要に応じて、Recovery Manager がバックアップを平行化できるように、チャンネルの平行化を構成します。
3. SECTION SIZE パラメータを指定して BACKUP を実行します。

たとえば、users 表領域に 900MB の単一のデータファイルが含まれているとします。また、SBT デバイス・セットの平行化設定が 3 に設定され、3 つの SBT チャンネルが構成されているとします。この表領域内のデータファイルは、次の例に示すようにファイル・セクションに分割できます。

```
BACKUP
SECTION SIZE 300M
TABLESPACE users;
```

この例では、3 つの SBT チャンネルのそれぞれによって、users データファイルの 300MB のファイル・セクションがバックアップされます。

**参照：** 大規模なデータファイルのセクションを検証する方法については、15-6 ページの「[データファイルの検証の平行化](#)」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スキップ

5-23 ページの「バックアップの最適化の構成」で説明されているように、CONFIGURE BACKUP OPTIMIZATION コマンドを実行すると、バックアップの最適化が有効になります。特定の条件が満たされた場合、Recovery Manager は、すでにバックアップされているファイルと同じファイルのバックアップをスキップします。

後続の例では、バックアップの最適化および保存方針を次の例のように構成していると想定しています。

### 例 9-1 バックアップの最適化の構成

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE BACKUP OPTIMIZATION ON;
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 4 DAYS;
```

Recovery Manager を例 9-1 のように構成して、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、リカバリ期間内に最新のバックアップが実行されている場合のみ、Recovery Manager は、オフラインおよび読取り専用のデータファイルのバックアップをスキップします。最新のバックアップがリカバリ期間より前に実行されている場合、Recovery Manager はバックアップをスキップしません。たとえば、最適化を行うと、読取り専用データファイルを含むバックアップ・セットがリカバリ期間内に 1 つ存在しているかぎり、このデータファイルの新しいバックアップを毎晩実行する必要がなくなります。

#### 参照：

- バックアップの最適化およびリカバリ期間に関連する例は、5-25 ページの「リカバリ期間に基づく保存方針による SBT バックアップのバックアップの最適化」を参照してください。
- ファイルが同一で、スキップできる可能性があるかどうかを判断するために CONFIGURE BACKUP OPTIMIZATION で使用される条件の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## 1 つのテープへの毎日のアーカイブ・ログ・バックアップの最適化の例

毎晩すべてのアーカイブ・ログをバックアップすると想定しています。ただし、各ログ順序番号の複数のコピーが作成されないようにします。Recovery Manager を例 9-1 のように構成して、次のコマンドをスクリプトで毎晩午前 1 時に実行します。

```
BACKUP DEVICE TYPE sbt
ARCHIVELOG ALL;
```

Recovery Manager は、24 時間以内に生成されたログ以外のすべてのログをスキップします。この方法で、各アーカイブ・ログの 1 つのみのコピーをテープ上に保持できます。

## 複数のメディア・ファミリへの毎日のアーカイブ・ログ・バックアップの最適化の例

Oracle Secure Backup では、メディア・ファミリは、共有のユーザー定義属性のセットが含まれている名前付きのボリューム・グループです。この例では、テープ上に存在しないログを 1 つのメディア・ファミリにバックアップし、同じログを別のメディア・ファミリにバックアップします。最後に、古いログを削除します。

Recovery Manager を例 9-2 に示すように構成して、毎晩同じ時刻に次のスクリプトを実行し、前日に生成されたログを 2 つの個別のメディア・ファミリにバックアップします。

**例 9-2 複数のメディア・ファミリへのアーカイブ REDO ログのバックアップ**

```
# The following command backs up just the logs that are not on tape. The
# first copies are saved to the tapes from the media family "log_family1".
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt
  PARS 'ENV=(OB_MEDIA_FAMILY=log_family1)';
  BACKUP ARCHIVELOG ALL;
}
# Make one more copy of the archived logs and save them to tapes from a
# different media family
RUN
{
  ALLOCATE CHANNEL c2 DEVICE TYPE sbt
  PARS 'ENV=(OB_MEDIA_FAMILY=log_family2)';
  BACKUP ARCHIVELOG
  NOT BACKED UP 2 TIMES;
}
```

SBT に 2 回バックアップしたログをディスクから削除することが目標である場合は、**アーカイブ REDO ログの削除方針**を使用する方法が目標を達成する最も簡単な方法です。次のワнтаイム構成では、テープに 2 つのアーカイブ・ログが存在する場合にアーカイブ REDO ログがディスクからの削除対象となるように指定されます。

```
CONFIGURE ARCHIVELOG DELETION POLICY
  TO BACKED UP 2 TIMES TO DEVICE TYPE sbt;
```

例 9-2 のスクリプトを実行した後に、DELETE ARCHIVELOG ALL を実行すると、不要なログを削除できます。

**アーカイブ・ログの週ごとのセカンダリ・バックアップの作成の例**

アーカイブ・ログを毎日テープにバックアップするための、より高度な例を想定します。ただし、テープ破損を考慮して、週ごとにディスクからログを削除する前に、別々のテープに各ログ順序番号の複数のコピーが作成されるようにします。この例では、データベースでフラッシュ・リカバリ領域が使用されていないことを想定しています。

まず、ワнтаイム構成を次のように実行します。

```
CONFIGURE BACKUP OPTIMIZATION ON;
CONFIGURE DEVICE TYPE sbt PARALLELISM 1;
CONFIGURE default DEVICE TYPE TO sbt;
CONFIGURE CHANNEL DEVICE TYPE sbt PARS 'ENV=(OB_MEDIA_FAMILY=first_copy);
```

最適化を有効にしているため、次のコマンドを毎晩実行して、まだバックアップされていないすべてのアーカイブ・ログを first\_copy メディア・ファミリにバックアップできます。

```
BACKUP ARCHIVELOG ALL TAG first_copy;
```

毎週金曜日の夜に、すべてのアーカイブ・ログの追加のバックアップを別のメディア・ファミリに作成します。バックアップの終了時に、すでにテープ上に 2 つ以上のコピーが作成されているすべてのアーカイブ・ログを削除します。これを行うには、次のスクリプトを実行します。

```
RUN
{
  # manually allocate a channel, in order to specify that the backup run by this
  # channel should go to both media families "first_copy" and "second_copy"
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt
  PARS 'ENV=(OB_MEDIA_FAMILY=second_copy)';
  ALLOCATE CHANNEL c2 DEVICE TYPE sbt
  PARS 'ENV=(OB_MEDIA_FAMILY=first_copy)';
```

```

BACKUP
  CHANNEL c1
  ARCHIVELOG
  UNTIL TIME 'SYSDATE'
  NOT BACKED UP 2 TIMES # back up only logs without 2 backups on tape
  TAG SECOND_COPY;
BACKUP
  CHANNEL c2
  ARCHIVELOG
  UNTIL TIME 'SYSDATE'
  NOT BACKED UP 2 TIMES # back up only logs without 2 backups on tape
  TAG FIRST_COPY;
}

# 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;

```

次の表に、日次バックアップ・スクリプトおよび週次バックアップ・スクリプトの効果を示します。

表 9-1 日次スクリプトおよび週次スクリプトの効果

| スクリプト | スクリプト実行後のテープの内容  | スクリプト実行後のディスクの内容                                       |
|-------|--|--|
| 日次    | バックアップされていないアーカイブ・ログは、この時点でメディア・ファミリー first_copy に存在します。                           | 最後に DELETE コマンドを実行した後に作成されたすべてのアーカイブ・ログは、ディスク上に存在しません。 |
| 週次    | テープに 2 つより少ないバックアップがあるアーカイブ・ログは、この時点でメディア・ファミリー first_copy および second_copy に存在します。 | 2 回以上テープにバックアップされたすべてのアーカイブ・ログが削除されます。                 |

週次バックアップの後に、second\_copy メディア・ファミリーのテープをオフサイトのストレージに送信できます。このテープ・バックアップは、プール first\_copy のプライマリ・テープが損傷した場合にのみ使用します。セカンダリ・テープはオフサイトに存在するため、Recovery Manager でのリカバリには使用しません。そのため、このバックアップには使用不可能のマークを付けることができます。

```
CHANGE BACKUP OF ARCHIVELOG TAG SECOND_COPY UNAVAILABLE;
```

**参照：**

- バックアップのステータスを変更する方法およびバックアップを削除する方法については、第 11 章「Recovery Manager バックアップおよびリポジトリ・レコードのメンテナンス」を参照してください。
- CHANGE および DELETE コマンドについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## オフライン・ファイル、読取り専用ファイルおよびアクセスできないファイルのスキップ

デフォルトでは、BACKUP コマンドでデータファイルにアクセスできない場合、このコマンドは終了します。表 9-2 に示すようにパラメータを指定すると、コマンドの終了を回避できます。

表 9-2 BACKUP ...SKIP のオプション

| 指定するパラメータ         | Recovery Manager がスキップするデータファイル  |
|-------------------|--|
| SKIP INACCESSIBLE | Recovery Manager で読み取ることができないデータファイル。  |
| SKIP OFFLINE      | オフライン・データファイル。一部のオフライン・データファイルは、ディスク上に存在するため、読取り可能になる可能性があります。ディスクから削除または移動されたデータファイルは読み取れないため、アクセス不可能になります。 |
| SKIP READONLY     | 読取り専用表領域のデータファイル。  |

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、バックアップ・ジョブを終了させる可能性のあるデータファイルをすべてスキップします。

### 例 9-3 Recovery Manager バックアップ時のファイルのスキップ

```
BACKUP DATABASE
  SKIP INACCESSIBLE
  SKIP READONLY
  SKIP OFFLINE;
```

## バックアップ・セットの多重化

Recovery Manager は、バックアップ・セットの最大 4 つのコピーを同時に作成できます。これらのコピーは、それぞれが完全な複製です。**多重バックアップ・セット**のコピーは、バックアップ・セットに含まれる各バックアップ・ピースのコピーで、各コピーに一意的なコピー番号が付けられます (0tcm8u2s\_1\_1、0tcm8u2s\_1\_2 など)。バックアップ・セットを**フラッシュ・リカバリ領域**に多重化することはできません。

BACKUP ... COPIES または CONFIGURE ... BACKUP COPIES を使用すると、バックアップ・セットを多重化できます。Recovery Manager では、ディスクまたはテープにバックアップを多重化できますが、テープとディスクにバックアップを同時に多重化することはできません。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、異なる物理ディスクに複数のコピーを作成します。SBT チャネルの場合は、バージョン 2 の SBT API をサポートするメディア・マネージャを使用すると、そのメディア・マネージャによって各コピーが自動的に個別のメディア (個別のテープなど) に書き込まれます。テープへのバックアップ時に、コピーの数が、使用可能なテープ・デバイスの数を超えないようにしてください。

多重化はバックアップ・セットにのみ適用され、イメージ・コピーには適用されません。イメージ・コピー・バックアップの作成時に BACKUP ... COPIES を指定すると、エラーが発生します。また、イメージ・コピー・バックアップに CONFIGURE ... BACKUP COPIES を設定しても、この設定は無視されます。

**参照:** Recovery Manager のバックアップ・コピーの概要は、7-10 ページの「Recovery Manager を使用したバックアップの複数のコピー」を参照してください。



## CONFIGURE BACKUP COPIES を使用したバックアップ・セットの多重化

6-5 ページの「バックアップの多重化の構成」の説明に従って CONFIGURE ... BACKUP COPIES コマンドを使用すると、指定したデバイス・タイプに作成する同じバックアップ・セットの数を指定できます。この設定は、制御ファイルの自動バックアップ（制御ファイルの自動バックアップでは常に1つのコピーが生成されるため）および BACKUP BACKUPSET コマンドを使用してバックアップしたバックアップ・セットを除く、すべてのバックアップ・セットに適用されます。

### CONFIGURE ... BACKUP COPIES を使用してバックアップを多重化する手順

1. 任意のデバイス・タイプのデータファイルおよびアーカイブ REDO ログに対して、任意のデバイス・タイプで作成するコピーの数を構成します。

デフォルトでは、各デバイス・タイプの CONFIGURE ... BACKUP COPIES が 1 に設定されています。次の例では、データファイルおよびアーカイブ・ログのテープへの多重化、およびデータファイル（アーカイブ REDO ログは含まない）のディスクへの多重化を構成しています。

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 1;
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '/disk1/%U', '/disk2/%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. BACKUP コマンドを実行します。

次のコマンドを実行すると、データベースおよびアーカイブ・ログがテープにバックアップされ、各データファイルおよびアーカイブ・ログのコピーが2つずつ作成されます。

```
BACKUP DATABASE PLUS ARCHIVELOG; # uses default sbt channel
```

ディスク・チャンネル用のフォーマットが構成されているため、次のコマンドを実行すると、データベースがディスクにバックアップされ、生成されたバックアップ・セットの1つのコピーが /disk1 ディレクトリに、もう1つのコピーが /disk2 ディレクトリに格納されます。

```
BACKUP DEVICE TYPE DISK AS COPY DATABASE;
```

CONFIGURE CHANNEL に FORMAT 句が構成されていない場合は、BACKUP コマンド自体に FORMAT を指定できます。たとえば、次のコマンドを発行できます。

```
BACKUP DATABASE
  FORMAT '/disk1/%U',
         '/disk2/%U';
```

3. LIST BACKUP コマンドを発行して、バックアップ・セットおよびバックアップ・ピースのリストを表示します。

たとえば、次のコマンドを入力します。

```
LIST BACKUP SUMMARY;
```

#Copies 列には、多重化または複数の BACKUP コマンドによって生成されたバックアップ・セットの数が表示されます。

**参照：** CONFIGURE BACKUP COPIES コマンドについては 6-5 ページの「バックアップの多重化の構成」、基本的なバックアップ構成オプションについては 5-2 ページの「Recovery Manager バックアップの環境の構成」を参照してください。



## BACKUP ... COPIES を使用したバックアップ・セットの多重化

BACKUP コマンドの COPIES オプションは、バックアップ・セットの多重化を制御する他のすべての COPIES または DUPLEX の設定より優先されます。

### BACKUP ... COPIES を使用してバックアップを多重化する手順

1. BACKUP コマンドの COPIES オプションを指定して、コピーの数を指定します。たとえば、次のコマンドを実行して、デフォルトの場所である DISK に各バックアップ・セットのコピーを 3 つずつ作成します。

```
BACKUP AS BACKUPSET DEVICE TYPE DISK
COPIES 3
INCREMENTAL LEVEL 0
DATABASE;
```

BACKUP コマンドで COPIES を指定しているため、CONFIGURE DATAFILE COPIES の設定に関係なく、各データファイルのバックアップ・セットが 3 つずつ作成されます。

2. LIST BACKUP コマンドを発行して、バックアップ・セットおよびバックアップ・ピースのリストを表示します (#Copies 列には、多重化によって、または BACKUP コマンドを複数回実行することによって生成されたコピーの数が表示されます)。たとえば、次のように入力します。

```
LIST BACKUP SUMMARY;
```

## Recovery Manager を使用したミラーの分割によるバックアップの実行

多くのサイトでは、プライマリ・データベース上で**メディア障害**が発生したり、ユーザーが不適切な操作を行って **Point-in-Time リカバリ** が必要になった場合のために、データベースのバックアップがディスク上に格納されています。ディスク上のデータファイルのバックアップを使用すると、リカバリのリストア手順が簡単になり、リカバリの速度および信頼性が向上します。

---

**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いずれの方法を使用しても）実行しないでください。オンライン REDO ログのバックアップをリストアすると、順序番号が同じで内容が異なる 2 つのアーカイブ・ログが作成される場合があります。また、制御ファイルのバックアップを実行するには、分割されたミラーではなく BACKUP CONTROLFILE コマンドを使用することをお勧めします。

---

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、データベースの各ファイルの 3 つのコピーを保持できます。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て、1 つのバックアップとして使用できます。

Recovery Manager では、自動的にミラーの分割は行われませんが、バックアップおよびリカバリで分割されたミラーを使用することができます。たとえば、Recovery Manager では、データファイルの分割されたミラーをデータファイルのコピーとして処理できます。また、このコピーをディスクまたはテープにバックアップできます。この項の手順では、ALTER SYSTEM SUSPEND/RESUME 機能を使用して**ミラーの分割によるバックアップ**を作成する方法について説明します。

一部のミラー化技術では、ミラーを分割してバックアップとして使用する前に、Oracle Database ですべての I/O を一時停止する必要があります。データベース・インスタンスで I/O の一時停止が必要かどうかについては、ストレージ・マネージャ、ボリューム・マネージャまたはファイル・システムのドキュメントを参照してください。

**SUSPEND/RESUME を使用して表領域のミラーの分割によるバックアップを実行する手順**

1. Recovery Manager を起動し、ALTER TABLESPACE ... BEGIN BACKUP 文を使用して、バックアップする表領域をバックアップ・モードにします。(すべての表領域をバックアップ・モードにするには、ALTER DATABASE BEGIN BACKUP を使用します。)

たとえば、表領域 users をバックアップ・モードにするには、Recovery Manager をターゲット・データベースに接続し、次の SQL コマンドを実行します。

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

2. ミラー化するソフトウェアまたはハードウェアが必要な場合、I/O を一時停止します。たとえば、Recovery Manager で次のコマンドを入力します。

```
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のミラーを分割します。
4. データベースの一時停止を終了します。たとえば、Recovery Manager で次のコマンドを入力します。

```
SQL 'ALTER SYSTEM RESUME';
```

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

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

ALTER DATABASE END BACKUP を使用して、すべての表領域のバックアップ・モードを終了することもできます。

6. CATALOG コマンドを使用して、ユーザー管理のミラー・コピーを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。

```
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、プロンプトで BACKUP DATAFILECOPY コマンドを実行します。

```
BACKUP DATAFILECOPY '/dk2/oradata/trgt/users01.dbf';
```

8. 分割されたミラーを復元する場合、まず 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。

```
CHANGE DATAFILECOPY '/dk2/oradata/trgt/users01.dbf' UNCATALOG;
```

9. 影響を受けたデータファイルの分割されたミラーを復元します。

**参照：**

- [「SUSPEND モードでのユーザー管理バックアップの作成」](#)  
(27-12 ページ)
- SUSPEND/RESUME 機能の詳細は、『Oracle Database 管理者ガイド』を参照してください。
- ALTER SYSTEM SUSPEND 構文については、『Oracle Database SQL 言語リファレンス』を参照してください。

## Recovery Manager バックアップの暗号化

6-7 ページの「バックアップの暗号化の構成」で説明されているように、バックアップの暗号化で Recovery Manager バックアップ・セットを保護できます。暗号化バックアップは、不正なユーザーが取得しても読み取ることができません。Recovery Manager バックアップの暗号化機能を使用するには、Enterprise Edition のデータベースが必要です。

## Recovery Manager バックアップの暗号化設定

バックアップの暗号化は、次のコマンドで指定される暗号化設定に基づいて実行されます。

- CONFIGURE ENCRYPTION

このコマンドを使用すると、透過的暗号化を永続的に構成できます。デュアル・モードまたはパスワード・モードの暗号化は永続的には構成できません。

- SET ENCRYPTION

このコマンドを使用すると、デュアル・モードまたはパスワード・モードの暗号化を Recovery Manager セッション・レベルで構成できます。

---

**注意：** ウォレットベースの暗号化は、パスワードが必要ないため、パスワードベースの暗号化より安全です。パスワードベースの暗号化は、バックアップをトランスポートする必要があるため絶対に必要な場合にのみ使用してください。

---

データベースでは、各暗号化バックアップに対して新しい暗号化キーが使用されます。バックアップ暗号化キーは、選択した暗号化モードに応じて、パスワードまたはデータベース・マスター・キー（あるいはその両方）を使用して暗号化されます。個々のバックアップ暗号化キーまたはパスワードは、クリアテキストでは格納されません。

1 回のリストア操作で、異なるモードで暗号化されたバックアップを処理できます。Recovery Manager は、リストアするバックアップ・ピースごとに、それらが暗号化されているかどうかを確認します。透過的に暗号化されているバックアップは、Oracle ウォレットがオープンして使用可能な場合、ユーザーの介入を必要としません。

パスワードの暗号化が検出された場合、Recovery Manager は、SET DECRYPTION コマンドで入力したパスワードのリスト内の一致するキーを検索します。使用可能なキーが検出された場合、Recovery Manager はリストア操作を続行します。検出されなかった場合、Recovery Manager は Oracle ウォレット内のキーを検索します。Recovery Manager は、使用可能なキーが検出された場合はリストア操作を続行します。検出されなかった場合はバックアップ・ピースを復号化できないというエラーを通知します。

---

**注意：** 異なるパスワードを使用して作成されたバックアップのセットを Recovery Manager でリストアする場合は、必要なすべてのパスワードを SET DECRYPTION に指定する必要があります。

---

暗号化は、バックアップのパフォーマンスに悪影響を及ぼす場合があります。暗号化バックアップでは、暗号化されていないバックアップの場合より多くの CPU リソースが消費されるため、より多くの Recovery Manager チャンネルを使用すると、ディスクへの暗号化バックアップのパフォーマンスを向上させることができます。

**参照：**

- パスワード暗号化バックアップをリストアする方法については、17-10 ページの「データベースの完全リカバリの実行」を参照してください。
- 「バックアップ・ピースの暗号化ステータスの確認」(10-18 ページ)
- SET コマンドの ENCRYPTION および DECRYPTION オプションについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## 透過モード暗号化バックアップの作成

6-9 ページの「Recovery Manager バックアップの暗号化モードの構成」の説明に従って、CONFIGURE コマンドで透過的暗号化を構成してある場合、暗号化バックアップの作成に必要な追加のコマンドはありません。通常どおりに、Recovery Manager バックアップを作成します。

## パスワード・モード暗号化バックアップの作成

SET ENCRYPTION BY PASSWORD コマンドを実行すると、Recovery Manager セッションで暗号化パスワードを設定できます。透過的暗号化が構成されている場合は、構成されている透過的暗号化ではなく、パスワードでバックアップを保護することを示すために ONLY キーワードを指定します。

---

---

**注意：** 安全なパスワードを作成してください。詳細は、『Oracle Database セキュリティ・ガイド』を参照してください。

---

---

### パスワード暗号化バックアップを作成する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. SET ENCRYPTION ON IDENTIFIED BY *password* ONLY コマンドを実行します。

次の例では、バックアップ内のすべての表領域の暗号化パスワード (*password* は実際に入力するパスワードのプレースホルダ) を設定し、暗号化がパスワードのみであることを示すために ONLY を指定しています。

```
SET ENCRYPTION IDENTIFIED BY password ONLY ON FOR ALL TABLESPACES;
```

3. データベースをバックアップします。  
たとえば、次のコマンドを入力します。

```
BACKUP DATABASE PLUS ARCHIVELOG;
```

## デュアル・モード暗号化バックアップの作成

Recovery Manager プロンプトで SET ENCRYPTION BY PASSWORD コマンドを使用して、パスワード保護されたバックアップを作成します。透過的暗号化が構成されている場合は、パスワードおよび構成されている透過的暗号化でバックアップを保護することを示すために ONLY キーワードを省略します。

---

---

**注意：** 安全なパスワードを作成してください。詳細は、『Oracle Database セキュリティ・ガイド』を参照してください。

---

---

### デュアル・モード暗号化バックアップを作成する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. ONLY キーワードを省略して、SET ENCRYPTION BY PASSWORD コマンドを実行します。

次の例では、バックアップ内のすべての表領域の暗号化パスワード（*password* は実際に入力するパスワードのプレースホルダ）を設定し、デュアル・モードの暗号化を示すために ONLY を省略しています。

```
SET ENCRYPTION IDENTIFIED BY password ON FOR ALL TABLESPACES;
```

3. データベースをバックアップします。  
たとえば、次のコマンドを入力します。

```
BACKUP DATABASE PLUS ARCHIVELOG;
```

## Recovery Manager バックアップの再開

再開可能バックアップ機能を使用すると、指定し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できます。

### 再開可能バックアップ

再開可能最小単位はデータファイルです。ただし、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、このピースに複数のデータファイルからのブロックが含まれている場合、再開可能単位はバックアップ・ピースになります。イメージ・コピーの再開可能単位はデータファイルです。

再開可能バックアップのメリットは、バックアップによって複数のバックアップ・セットが生成される場合、正常に完了したバックアップ・セットを再実行する必要がないことです。ただし、データベース全体が1つのバックアップ・セットに書き込まれる場合、途中でバックアップが失敗すると、バックアップ全体を再開する必要があります。

ファイルの読取り時またはバックアップ・ピースやイメージ・コピーへの書込み時に I/O エラーが検出された場合、Recovery Manager は実行中のバックアップ・ジョブを終了します。たとえば、バックアップを試みたデータファイルがディスク上に存在しなかった場合、バックアップは終了します。ただし、複数のチャネルが使用されていたり、バックアップの冗長コピーが作成されている場合、Recovery Manager はユーザーの介入なしにバックアップを続行できる可能性があります。

Recovery Manager は、指定し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できます。バックアップが失敗した場合にこの機能を使用すると、失敗したバックアップで処理されなかったデータベースの部分をバックアップできます。

BACKUP コマンドに SINCE TIME 句を指定すると、バックアップを再開できます。SINCE TIME に完了時刻より後の値が設定されている場合、Recovery Manager はそのファイルをバックアップします。SINCE TIME パラメータを指定せずに BACKUP DATABASE NOT BACKED UP を使用した場合、Recovery Manager は、一度も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。

**参照：** BACKUP ... NOT BACKED UP 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』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開

BACKUP コマンドの SINCE TIME パラメータを使用すると、その日付を過ぎると新しいバックアップが必要になる日付を指定できます。SINCE TIME に完了時刻より後の値が設定されている場合、Recovery Manager はそのファイルをバックアップします。SINCE TIME パラメータを指定せずに BACKUP DATABASE NOT BACKED UP を使用した場合、Recovery Manager は、一度も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。

### 指定し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手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. BACKUP ... NOT BACKED UP SINCE TIME コマンドを実行します。

SINCE TIME パラメータに有効な日付を指定します。次の例では、デフォルトの構成済チャンネルを使用して、過去2週間にバックアップされていないすべてのデータベース・ファイルおよびアーカイブ REDO ログをバックアップします。

```
BACKUP
NOT BACKED UP SINCE TIME 'SYSDATE-14'
DATABASE PLUS ARCHIVELOG;
```

**参照：** BACKUP コマンドを使用して完了していないバックアップを再開する方法の例については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## バックアップ期間の管理

この項では、バックアップ期間を使用して、バックアップ・ジョブを完了できる期間の制限を設定する方法について説明します。

### バックアップ期間

**バックアップ期間**とは、バックアップを完了する必要がある時間の長さのことです。たとえば、データベースのバックアップを、システム上のユーザー・アクティビティが低下する時間帯（午前2時から6時など）に制限する場合があります。

Recovery Manager は、バックアップの日付が最も古いファイルを最初にバックアップします。デフォルトでは、Recovery Manager はファイルを可能なかぎり速くバックアップします。期間を指定しても、期間終了前にバックアップが完了するように通常より速くデータがバックアップされるわけではありません。

デフォルトでは、バックアップが DURATION の時間内に完了しなかった場合、Recovery Manager はバックアップを中断し、エラーを通知します。BACKUP コマンドが RUN コマンド内に指定されている場合、RUN コマンドは終了します。バックアップ全体が完了しなかった場合でも、バックアップが完了したバックアップ・セットは保持され、リストア操作で使用できます。このため、実行可能な時間が終了して中断されたジョブを再試行すると、試行を繰り返すたびに、バックアップする必要があるファイルの処理が進行していきます。不完全なバックアップ・セットは廃棄されます。

### バックアップ期間の指定

BACKUP コマンドの DURATION パラメータを使用すると、指定したバックアップ・ジョブを実行できる期間を指定できます。

#### バックアップ期間を指定する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. BACKUP DURATION コマンドを実行します。

たとえば、次のコマンドを午前2時に実行して、午前6時までバックアップを実行するように指定できます。

```
BACKUP
DURATION 4:00
TABLESPACE users;
```

**参照：** BACKUP コマンドの構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。



## バックアップ期間での部分バックアップの許可

PARTIAL を指定すると、Recovery Manager は、バックアップ期間が終了してバックアップが中断された場合でも、エラーを通知しません。かわりに、バックアップできなかったファイルを示すメッセージを表示します。BACKUP コマンドを RUN ブロックで実行している場合、RUN ブロック内の残りのコマンドは続行されます。

FILESERSET 1 を指定すると、Recovery Manager は各ファイルを独自のバックアップ・セットに格納します。バックアップ期間が終了してバックアップが中断されると、そのときバックアップ中であったファイルのバックアップのみが消失します。時間内に完了したすべてのバックアップ・セットは保存されるため、バックアップ期間の終了によって無効となる処理は最小限で済みます。

### バックアップが部分的に完了した場合に Recovery Manager がエラーを通知しないようにする手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. PARTIAL オプションを指定して BACKUP DURATION コマンドを実行します。

たとえば、次のコマンドを午前 2 時に実行して、午前 6 時までバックアップを実行し、各データファイルが個別のバックアップ・セットに格納されるように指定できます。

```
BACKUP
  DURATION 4:00 PARTIAL
  TABLESPACE users
  FILESPERSET 1;
```

## バックアップの負荷および時間の最小化

DURATION を使用すると、最高のパフォーマンスでバックアップを実行したり、割り当てられた時間内で可能な限り低速のバックアップを実行して、パフォーマンスへのバックアップ作業の影響を最小限に抑えることができます。最高のパフォーマンスでバックアップを実行するには、例 9-4 のように、DURATION とともに MINIMIZE TIME オプションを使用します。

### 例 9-4 BACKUP DURATION での MINIMIZE TIME の使用

```
BACKUP
  DURATION 4:00 PARTIAL
  MINIMIZE TIME
  DATABASE
  FILESPERSET 1;
```

使用可能なすべての時間を使用してバックアップを実行するには、例 9-5 のように、MINIMIZE LOAD オプションを使用します。

### 例 9-5 BACKUP DURATION での MINIMIZE LOAD の使用

```
BACKUP
  DURATION 4:00 PARTIAL
  MINIMIZE LOAD
  DATABASE
  FILESPERSET 1;
```

例 9-5 では、Recovery Manager はバックアップの進捗状況を監視し、現在のレートでバックアップの完了に必要な時間を定期的に見積もります。バックアップ期間の終了前にバックアップが終了すると判断した場合、Recovery Manager は、使用可能なすべての時間が使用されるように、バックアップのレートを低下させます。これによって、データベースでバックアップに関連するオーバーヘッドが減少します。

DURATION および MINIMIZE LOAD をテープ・バックアップで使用する場合は、次の事項に注意してください。

- テープへの効率的なバックアップには、テープ・ストリームが必要です。MINIMIZE LOAD を使用すると、**Recovery Manager** がバックアップのレートを低下させ、テープ・ストリームが最適ではなくなる場合があります。
- **Recovery Manager** は、バックアップ期間中、常にテープ・リソースを保持します。このため、バックアップ期間中は、テープ・リソースを他の目的に使用できません。

これらの点から、テープへのバックアップ時には MINIMIZE LOAD を使用しないことをお勧めします。

**参照：** 効率的なテープ処理の詳細は、21-10 ページの「[SBT への書込みフェーズにおけるメディア・マネージャの構成要素](#)」を参照してください。



# 第 IV 部

---

## Recovery Manager のバックアップの管理

第 IV 部では、Recovery Manager バックアップを管理する方法について説明します。第 IV 部に含まれる章は次のとおりです。

- 第 10 章「Recovery Manager 操作に関するレポート」
- 第 11 章「Recovery Manager バックアップおよびリポジトリ・レコードのメンテナンス」
- 第 12 章「リカバリ・カタログの管理」



---

## Recovery Manager 操作に関するレポート

この章では、Recovery Manager 操作に関してレポートする方法について説明します。この章の内容は、次のとおりです。

- Recovery Manager レポートの概要
- バックアップおよびリカバリ関連オブジェクトの表示
- バックアップおよびデータベース・スキーマに関するレポート
- V\$ ビューを使用したバックアップ・メタデータの問合せ
- リカバリ・カタログ・ビューの問合せ

## Recovery Manager レポートの概要

この項では、Recovery Manager レポートの目的および基本的な概念について説明します。

### Recovery Manager レポートの目的

バックアップおよびリカバリ計画の一部として、バックアップした内容を示すレポートを定期的に行う必要があります。また、バックアップが必要なデータファイルまたは最近バックアップされていないファイルを確認しておく必要もあります。また、問題発生時に Recovery Manager でリストアする必要があるバックアップをプレビューすることもできます。

バックアップおよびリカバリのもう 1 つ重要な側面は、領域の使用状況の監視です。ディスクにバックアップする場合は、そのディスクが一杯になる可能性があるため、パフォーマンス上の問題が発生したり、データベースが停止することもあります。Recovery Manager を使用すると、バックアップが**不要なバックアップ**で、削除可能かどうかを確認できます。

また、Recovery Manager ジョブについての履歴情報を取得する必要がある場合もあります。たとえば、発行されたバックアップ・ジョブの数、各バックアップ・ジョブのステータス（失敗したか完了したかなど）、ジョブの開始日時と終了日時、および実行されたバックアップのタイプについて確認する必要がある場合があります。

### Recovery Manager レポートの基本的な概念

Recovery Manager は、操作の実行対象となる各ターゲット・データベースの制御ファイルに、そのメタデータの **Recovery Manager リポジトリ** を常に格納します。たとえば、Recovery Manager を使用して、prod1 および prod2 データベースをバックアップするとします。Recovery Manager は、prod1 のバックアップ用メタデータを prod1 の制御ファイルに、prod2 のバックアップ用メタデータを prod2 の制御ファイルに格納します。

必要に応じて、**リカバリ・カタログ**とともに Recovery Manager を使用できます。この場合、Recovery Manager は、別の**リカバリ・カタログ・データベース**内の一連の表に、追加のメタデータ・リポジトリを保持します。たとえば、prod3 にリカバリ・カタログを作成するとします。このリカバリ・カタログには、複数のターゲット・データベースを登録できます。たとえば、prod1 および prod2 を prod3 に格納されているリカバリ・カタログに登録すると、Recovery Manager は、prod1 および prod2 のバックアップに関するメタデータを**リカバリ・カタログ・スキーマ**に格納します。

次に示す様々な方法で、Recovery Manager リポジトリからメタデータにアクセスできます。

- Recovery Manager の LIST および REPORT コマンドでは、使用可能なバックアップと、データベースをリストアおよびリカバリするためにそれをどのように使用できるかという詳細な情報が提供されます。

LIST コマンドについては 10-4 ページの「バックアップおよびリカバリ関連オブジェクトの表示」、REPORT については 10-11 ページの「バックアップ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。

- データベースがオープンしている場合は、多くの V\$ ビューによって、各ターゲット・データベースの制御ファイル内にある Recovery Manager リポジトリ・レコードに直接アクセスできます。

V\$DATAFILE\_HEADER、V\$PROCESS、V\$SESSION などの一部の V\$ ビューには、リカバリ・カタログ・ビューにはない情報が含まれています。V\$ ビューの詳細は、『Oracle Database リファレンス』を参照してください。

- 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は、RC\_ ビューによって、リカバリ・カタログに格納されている Recovery Manager リポジトリ・データに直接アクセスできます。

RC\_ ビューは、V\$ ビューとほぼ対応しています。RC\_ ビューについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

- RESTORE ... PREVIEW および RESTORE ... VALIDATE HEADER コマンドでは、指定した時間まで Recovery Manager でリストア可能なバックアップが表示されます。

RESTORE ... PREVIEW は、メタデータは問い合せますが、実際にバックアップ・ファイルは読み取りません。RESTORE ... VALIDATE HEADER コマンドでも同じ操作が実行されますが、リストアおよびリカバリに必要なファイルの表示の他に、ディスク上またはメディア管理カタログ内のファイルが Recovery Manager リポジトリのメタデータに対応しているかどうかを確認するためにバックアップ・ファイル・ヘッダーが検証されます。これらのコマンドについては、17-6 ページの「リストア操作で使用されるバックアップのプレビュー」を参照してください。

第 11 章「Recovery Manager バックアップおよびリポジトリ・レコードのメンテナンス」で説明されているように、Recovery Manager リポジトリがディスクおよびテープ上の実情を反映していない場合があります。たとえば、ユーザーがオペレーティング・システムのユーティリティを使用してバックアップを削除した場合、そのバックアップは、Recovery Manager リポジトリでは誤って使用可能と表示されます。

CHANGE、CROSSCHECK、DELETE などのコマンドを使用すると、使用可能なバックアップの実際の状態を反映して Recovery Manager リポジトリを更新できます。そうしない場合、コマンドおよびビューの出力が誤って表示され、データベースをリストアおよびリカバリするためのバックアップを Recovery Manager で検出できない可能性があります。

#### 参照：

- Recovery Manager リポジトリを最新の状態に保つ方法については、11-12 ページの「Recovery Manager リポジトリのクロスチェック」を参照してください。
- LIST 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- REPORT 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- RESTORE PREVIEW 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## Data Guard 環境でのレポート

3-8 ページの「Data Guard 環境での Recovery Manager によるファイル管理」で説明されているように、すべてのバックアップがそのバックアップを作成したプライマリ・データベースまたはスタンバイ・データベースに関連付けられています。たとえば、standby1 の DB\_UNIQUE\_NAME を使用してデータベースをバックアップすると、standby1 データベースがこのバックアップに関連付けられます。

Data Guard 環境では、Data Guard を使用しない場合と同様に、LIST、REPORT および SHOW コマンドを使用できます。FOR DB\_UNIQUE\_NAME 句を指定してこれらのコマンドを実行すると、指定したデータベースに関連付けられているバックアップを表示できます。たとえば、次のコマンドでは、sfstandby にのみ関連付けられているアーカイブ REDO ログが表示されません。

```
LIST ARCHIVELOG ALL FOR DB_UNIQUE_NAME sfstandby;
```

Data Guard 環境で FOR DB\_UNIQUE\_NAME 句を指定せずに LIST、REPORT および SHOW コマンドを使用すると、Recovery Manager によって、ターゲット・データベースでアクセス可能なファイルが表示されます。Recovery Manager でバックアップにアクセス可能であるとみなされる場合については、3-9 ページの「Data Guard 環境でのバックアップの関連付け」を参照してください。

Data Guard 環境では、Recovery Manager をリカバリ・カタログとともに使用する必要があります。Recovery Manager では、すべてのバックアップおよびリカバリ・ファイルのメタデータがリカバリ・カタログの Data Guard 環境に格納されます。Recovery Manager のレポート・コマンドの実行時に、マウントまたはオープンされているデータベースに Recovery Manager を TARGET として接続するか、または SET DBID コマンドを使用してデータベースを識別することができます。

**参照：** Data Guard 環境で Recovery Manager 操作に関してレポートする方法については、『Oracle Data Guard 概要および管理』を参照してください。

## バックアップおよびリカバリ関連オブジェクトの表示

LIST コマンドは、Recovery Manager リポジトリの情報を使用して、バックアップやバックアップとリカバリに関連するその他のオブジェクトのリストを表示します。この項の内容は、次のとおりです。

- LIST コマンド
- バックアップおよびコピーの表示
- データベース・インカーネーションの表示
- リストア・ポイントの表示

## LIST コマンド

LIST コマンドの主な目的は、バックアップおよびコピーを表示することです。たとえば、次の内容を表示できます。

- データベース、表領域、データファイル、アーカイブ REDO ログまたは制御ファイルのバックアップおよびプロキシ・コピー
- 期限切れのバックアップ
- 時間、パス名、デバイス・タイプまたはリカバリ可能性で制限されているバックアップ
- アーカイブ REDO ログ・ファイルおよびディスク・コピー

LIST コマンドの主な目的は、バックアップおよびコピーを表示することです。Recovery Manager では、バックアップおよびコピーのみでなく、他のタイプのデータも表示できます。次の表に、表示できる有効なオブジェクトをいくつか示します。

表 10-1 LIST オブジェクト

| リストの内容                | コマンド                | 説明  |
|-----------------------|---------------------|---|
| バックアップ・セットおよびプロキシ・コピー | LIST BACKUP         | データベース、表領域、データファイル、アーカイブ REDO ログ、制御ファイルまたはサーバー・パラメータ・ファイルのすべてのバックアップ・セット、コピーおよびプロキシ・コピーを表示できます。   |
| イメージ・コピー              | LIST COPY           | データファイル・コピーおよびアーカイブ REDO ログ・ファイルを表示できます。デフォルトでは、LIST COPY によってすべてのデータベース・ファイルおよびアーカイブ REDO ログのコピーが表示されます。出力には、リストアできないイメージ・コピー、期限切れのイメージ・コピーまたは使用不可能なイメージ・コピーも含め、使用可能なイメージ・コピーおよび使用不可能なイメージ・コピーの両方が含まれます。 |
| アーカイブ REDO ログ・ファイル    | LIST ARCHIVELOG     | アーカイブ REDO ログ・ファイルを表示できます。  |
| データベース・インカーネーション      | LIST INCARNATION    | データベースのすべてのインカーネーションを表示できます。RESETLOGS オプションを使用してオープンすると、新しいデータベース・ <b>インカーネーション</b> が作成されます。  |
| Data Guard 環境のデータベース  | LIST DB_UNIQUE_NAME | Data Guard 環境のデータベースは、その DB_UNIQUE_NAME 初期化パラメータ設定によって識別されます。同じ DBID を持つすべてのデータベースを表示できます。  |

表 10-1 LIST オブジェクト (続き)

| リストの内容   | コマンド                        | 説明  |
|--|-----------------------------|---|
| Data Guard 環境のプライマリ・データベースまたはスタンバイ・データベースのバックアップおよびコピー | LIST ... FOR DB_UNIQUE_NAME | Data Guard 環境内の指定したデータベースまたはすべてのデータベースのすべてのバックアップおよびコピーを表示できます。<br><br>Recovery Manager では、指定した DB_UNIQUE_NAME を持つデータベースに排他的に関連付けられているファイルまたはオブジェクトに出力が制限されます。たとえば、FOR DB_UNIQUE_NAME を指定して LIST を使用すると、特定のスタンバイ・データベースまたはプライマリ・データベースに関連付けられているアーカイブ REDO ログのリストを表示できます。データベースによって所有されていないオブジェクト (リカバリ・カタログ・ビューの SITE_KEY 列が NULL) は表示されないことに注意してください。 |
| リストア・ポイント  | LIST RESTORE POINT          | Recovery Manager リポジトリで認識されるリストア・ポイントを表示できます。   |
| ストアド・スクリプトの名前  | LIST SCRIPT NAMES           | CREATE SCRIPT または REPLACE SCRIPT コマンドを使用して作成されたリカバリ・カタログ・スクリプトの名前を表示できます。リカバリ・カタログが必要です。  |
| データ・リカバリ・アドバイザとともに使用する場合の障害                            | LIST FAILURE                | <b>障害</b> とは、 <b>修復オプション</b> にマッピングされている永続的なデータの破損のことです。LIST FAILURE を ADVISE および REPAIR コマンドとともに使用する方法については、第 14 章「データ・リカバリ・アドバイザを使用した障害の診断および修復」を参照してください。  |

LIST コマンドでは、出力の表示方法を制御できる多くのオプションがサポートされています。次の表に、最も一般的な LIST のオプションを示します。

表 10-2 最も一般的な LIST のオプション

| LIST のオプション      | 説明  |
|------------------|---|
| LIST EXPIRED     | Recovery Manager リポジトリに記録されているバックアップまたはコピーで、最後に <b>クロスチェック</b> を実行したときに、ディスクまたはテープ上の予測した位置に存在しなかったものを表示します。このようなバックアップは、Recovery Manager の外部で削除された可能性があります。 |
| LIST ... BY FILE | 各データファイル、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・ファイルのバックアップを表示します。各行にファイルのバックアップの説明が示されます。  |
| LIST ... SUMMARY | 各バックアップの 1 行のサマリーを表示します。  |

前述の表に、すべての LIST オブジェクトおよびオプションが示されているわけではありません。たとえば、時間、パス名、デバイス・タイプ、タグまたはリカバリ可能性で制限されたバックアップを表示できます。

**参照：** LIST コマンドの詳細は、『Oracle Database バックアップおよびリカバリ・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表示

`listObjList` または `recordSpec` 句 (『Oracle Database バックアップおよびリカバリ・リファレンス』を参照) 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と、Recovery Manager は、すべてのデータベース・ファイルとアーカイブ REDO ログ・ファイルのコピーを表示します。

デフォルトでは、Recovery Manager は、各バックアップまたはプロキシ・コピーを連続して表示した後、そのバックアップに含まれているファイルを識別します。また、ファイルごとにバックアップを表示することもできます。

デフォルトでは、Recovery Manager は冗長モードで表示を行います。つまり、様々な情報を複数行にわたって表示します。冗長モードによる出力が多すぎる場合は、サマリー・モードでバックアップを表示することもできます。

### バックアップおよびコピーを表示する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. すべてのバックアップおよびコピーのサマリー・レポートを表示するには、SUMMARY オプションを指定して LIST コマンドを実行します。

次のコマンドを実行すると、すべての Recovery Manager バックアップのサマリーが出力されます。

```
LIST BACKUP SUMMARY;
```

出力例を次に示します。

```
List of Backups
=====
Key      TY LV S Device Type Completion Time #Pieces #Copies Compressed Tag
-----
1        B  A  A SBT_TAPE  21-OCT-07      1      1      NO
TAG20071021T094505
2        B  F  A SBT_TAPE  21-OCT-07      1      1      NO
TAG20071021T094513
3        B  A  A SBT_TAPE  21-OCT-07      1      1      NO
TAG20071021T094624
4        B  F  A SBT_TAPE  21-OCT-07      1      1      NO
TAG20071021T094639
5        B  F  A DISK    04-NOV-07      1      1      YES
TAG20071104T195949
```

3. バックアップおよびコピーの詳細な出力を表示するには、SUMMARY オプションを指定せずに LIST コマンドを実行します。

次のコマンドを実行すると、デフォルトの詳細な出力で Recovery Manager バックアップおよびコピーが表示されます。

```
LIST BACKUP;
LIST COPY;
```

LIST BACKUP および LIST COPY の出力例を次に示します。

```
List of Backup Sets
=====
BS Key Size      Device Type Elapsed Time Completion Time
-----
7      136M DISK      00:00:20 04-NOV-06
BP Key: 7 Status: AVAILABLE Compressed: NO Tag: TAG20071104T200759
Piece Name: /d2/RDBMS/backupset/2007_11_04/o1_mf_anmn TAG20071104T200759_
ztjxx3k8_.bkp
```

```
List of Archived Logs in backup set 7
Thrd Seq Low SCN Low Time Next SCN Next Time
-----
1 1 173832 21-OCT-06 174750 21-OCT-06
1 2 174750 21-OCT-06 174755 21-OCT-06
1 3 174755 21-OCT-06 174758 21-OCT-06
```



```

BS Key  Type LV Size      Device Type Elapsed Time Completion Time
-----
8      Full  2M          DISK        00:00:01    04-NOV-06
      BP Key: 8  Status: AVAILABLE Compressed: NO Tag: TAG20071104T200829
      Piece Name: /disk1/oracle/dbs/c-774627068-20071104-01
      Controlfile Included: Ckp SCN: 631510      Ckp time: 04-NOV-06
      SPFILE Included: Modification time: 21-OCT-06
    
```

List of Datafile Copies  
=====

```

Key      File S Completion Time Ckp SCN    Ckp Time
-----
1       7   A 11-OCT-06      360072    11-OCT-06
      Name: /work/orcva/RDBMS/datafile/o1_mf_tbs_2_21v7bf82_.dbf
      Tag: DF7COPY

2       8   A 11-OCT-06      360244    11-OCT-06
      Name: /work/orcva/RDBMS/datafile/o1_mf_tbs_2_21v7qmcj_.dbf
      Tag: TAG20071011T184835
    
```

List of Control File Copies  
=====

```

Key      S Completion Time Ckp SCN    Ckp Time
-----
3       A 11-OCT-06      360380    11-OCT-06
      Name: /d2/RDBMS/controlfile/o1_mf_TAG20071011T185335_21v80zqd_.ctl
      Tag: TAG20071011T185335
    
```

List of Archived Log Copies for database with db\_unique\_name RDBMS  
=====

```

Key      Thrd Seq    S Low Time
-----
1       1    1      A 11-OCT-06
      Name: /work/arc_dest/arcr_1_1_603561743.arc

2       1    2      A 11-OCT-06
      Name: /work/arc_dest/arcr_1_2_603561743.arc

3       1    3      A 11-OCT-06
      Name: /work/arc_dest/arcr_1_3_603561743.arc
    
```

4. ファイルごとにバックアップを表示するには、表示するオブジェクトおよびオプションを BY FILE オプションとともに指定して LIST を実行します。たとえば、次のように入力します。

```
LIST BACKUP BY FILE;
```

出力例を次に示します。

List of Datafile Backups  
=====

```

File Key  TY LV S Ckp SCN    Ckp Time #Pieces #Copies Compressed Tag
-----
1       5   B F A 631092    04-NOV-06 1      1      YES
TAG20071104T195949
      2   B F A 175337    21-OCT-06 1      1      NO
TAG20071021T094513
    
```

```

2      5      B F A 631092    04-NOV-06 1      1      YES
TAG20071104T195949
      2      B F A 175337    21-OCT-06 1      1      NO
TAG20071021T094513

```

... some rows omitted

List of Archived Log Backups

=====

| Thrd | Seq | Low SCN | Low Time  | BS Key | S | #Pieces | #Copies | Compressed | Tag                |
|------|-----|---------|-----------|--------|---|---------|---------|------------|--------------------|
| 1    | 1   | 173832  | 21-OCT-06 | 7      | A | 1       | 1       | NO         | TAG20071104T200759 |
|      |     |         |           | 1      | A | 1       | 1       | NO         | TAG20071021T094505 |
| 1    | 2   | 174750  | 21-OCT-06 | 7      | A | 1       | 1       | NO         | TAG20071104T200759 |
|      |     |         |           | 1      | A | 1       | 1       | NO         | TAG20071021T094505 |
| 1    | 38  | 575472  | 03-NOV-06 | 7      | A | 1       | 1       | NO         | TAG20071104T200759 |
| 1    | 39  | 617944  | 04-NOV-06 | 7      | A | 1       | 1       | NO         | TAG20071104T200759 |

List of Controlfile Backups

=====

| CF     | Ckp       | SCN | Ckp Time | BS Key | S | #Pieces | #Copies            | Compressed | Tag |
|--------|-----------|-----|----------|--------|---|---------|--------------------|------------|-----|
| 631510 | 04-NOV-06 | 8   | A        | 1      | 1 | NO      | TAG20071104T200829 |            |     |
| 631205 | 04-NOV-06 | 6   | A        | 1      | 1 | NO      | TAG20071104T200432 |            |     |

List of SPFILE Backups

=====

| Modification Time | BS Key | S | #Pieces | #Copies | Compressed | Tag                |
|-------------------|--------|---|---------|---------|------------|--------------------|
| 21-OCT-06         | 8      | A | 1       | 1       | NO         | TAG20071104T200829 |
| 21-OCT-06         | 6      | A | 1       | 1       | NO         | TAG20071104T200432 |

**参照：** LIST 出力の様々な列ヘッダーについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## 選択したバックアップおよびコピーの表示

様々な条件を指定して、LIST 出力を制限できます。

### 選択したバックアップおよびコピーを表示する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. `listObjList` または `recordSpec` 句を指定して、LIST COPY または LIST BACKUP を実行します。たとえば、次のいずれかのコマンドを実行します。

```

# lists backups of all files in database
LIST BACKUP OF DATABASE;
# lists copy of specified datafile
LIST COPY OF DATAFILE 'ora_home/oradata/trgt/system01.dbf';
# lists specified backup set

```

```
LIST BACKUPSET 213;
# lists datafile copy
LIST DATAFILECOPY '/tmp/tools01.dbf';
```

*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、次のいずれかのコマンドを実行します。

```
# specify a backup set by tag
LIST BACKUPSET TAG 'weekly_full_db_backup';
# specify a backup or copy by device type
LIST COPY OF DATAFILE 'ora_home/oradata/trgt/system01.dbf' DEVICE TYPE sbt;
# specify a backup 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-2002' AND '17-DEC-2002';
# specify logs backed up at least twice to tape
LIST ARCHIVELOG ALL BACKED UP 2 TIMES TO DEVICE TYPE sbt;
```

### 3. 出力を確認します。

出力は、LIST コマンドに指定したオプションによって異なります。たとえば、次のコマンドでは、データファイル 1 のコピーが表示されます。

```
RMAN> LIST BACKUP OF DATAFILE 1;
```

```
List of Backup Sets
```

```
=====
```

```
BS Key Type LV Size Device Type Elapsed Time Completion Time
```

```
-----
```

```
2 Full 230M SBT_TAPE 00:00:49 21-OCT-06
BP Key: 2 Status: AVAILABLE Compressed: NO Tag: TAG20071021T094513
Handle: 02f4eatc_1_1 Media: /smrdir
```

```
List of Datafiles in backup set 2
```

```
File LV Type Ckp SCN Ckp Time Name
```

```
-----
```

```
1 Full 175337 21-OCT-06 /oracle/dbs/tbs_01.f
```

```
BS Key Type LV Size Device Type Elapsed Time Completion Time
```

```
-----
```

```
5 Full 233M DISK 00:04:30 04-NOV-06
BP Key: 5 Status: AVAILABLE Compressed: NO Tag: TAG20071104T195949
Piece Name: /disk1/2007_11_04/o1_mf_nnndf_TAG20071104T195949_ztjxfvgz_.bkp
```

```
List of Datafiles in backup set 5
```

```
File LV Type Ckp SCN Ckp Time Name
```

```
-----
```

```
1 Full 631092 04-NOV-06 /oracle/dbs/tbs_01.f
```

#### 参照:

- *listObjList* および *recordSpec* 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- LIST 出力の列については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## データベース・インカーネーションの表示

データベースに対して OPEN RESETLOGS 操作を実行するたびに、新しいデータベース・インカーネーションが作成されます。データベース・インカーネーションと、それによってデータベース・リカバリが受ける影響については、13-6 ページの「データベース・インカーネーション」を参照してください。

増分バックアップの実行時、Recovery Manager は、前回のインカーネーションまたは現行のインカーネーションからのバックアップを、後続の増分バックアップの基礎として使用できます。リストアおよびリカバリの実行時、すべてのアーカイブ・ログが使用可能なかぎり、Recovery Manager は、現行のインカーネーションからのバックアップを使用するのと同様に、リストアまたはリカバリ操作で前回のインカーネーションからのバックアップを使用できます。

### データベース・インカーネーションを表示する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. 次の例に示すように、LIST INCARNATION コマンドを実行します。

```
LIST INCARNATION;
```

リカバリ・カタログを使用している場合、および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、それらを区別することができます。

```
LIST INCARNATION OF DATABASE prod3;
```

LIST 出力の様々な列ヘッダーについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。出力例は次のとおりです。

```
RMAN> LIST INCARNATION OF DATABASE;
```

```
List of Database Incarnations
DB Key  Inc Key DB Name  DB ID          STATUS  Reset SCN  Reset Time
-----
1       1       RDBMS    774627068     PARENT   1          21-OCT-06
2       2       RDBMS    774627068     CURRENT 173832     21-OCT-06
```

この出力例では、RESETLOGS が、SCN 164378 でデータベース trgt に対して実行され、新しいインカーネーションが作成されたことが示されています。インカーネーションは、インカーネーション・キー（Inc Key 列）で区別されます。

## リストア・ポイントの表示

LIST コマンドを使用して、特定のリストア・ポイント、または Recovery Manager リポジトリで認識されるすべてのリストア・ポイントを表示できます。このコマンドの例を次に示します。

```
LIST RESTORE POINT restore_point_name;
LIST RESTORE POINT ALL;
```

Recovery Manager は、リストア・ポイントの SCN および時刻、リストア・ポイントのタイプおよびリストア・ポイントの名前を表示します。出力例を次に示します。

```
RMAN> LIST RESTORE POINT ALL;
```

```
using target database control file instead of recovery catalog
SCN          RSP Time  Type          Time          Name
-----
341859       28-JUL-06          28-JUL-06  NORMAL_RS
343690       28-JUL-06  GUARANTEED  28-JUL-06  GUARANTEED_RS
```

また、次のように V\$RESTORE\_POINT ビューを問い合わせ、現在定義されているリストア・ポイントのリストを表示することもできます。

```
SELECT NAME, SCN, TIME, DATABASE_INCARNATION#,
       GUARANTEE_FLASHBACK_DATABASE, STORAGE_SIZE
FROM   V$RESTORE_POINT;
```

各リストア・ポイントの名前、リストア・ポイントが作成された SCN、実時間およびデータベース・**インカネーション**番号、各リストア・ポイントが保証付きリストア・ポイントであるかどうか、およびそのリストア・ポイントに対するフラッシュバック・データベース操作に必要なデータに使用されているリカバリ領域の量を表示できます。

また、次の問合せを使用して、保証付きリストア・ポイントのみを表示することもできます。

```
SELECT NAME, SCN, TIME, DATABASE_INCARNATION#,
       GUARANTEE_FLASHBACK_DATABASE, STORAGE_SIZE
FROM   V$RESTORE_POINT
WHERE  GUARANTEE_FLASHBACK_DATABASE='YES';
```

通常のリストア・ポイントの場合、STORAGE\_SIZE が 0 (ゼロ) になります。保証付きリストア・ポイントの場合、STORAGE\_SIZE は、フラッシュ・リカバリ領域のディスク領域の容量を示します。この領域は、そのリストア・ポイントに対する FLASHBACK DATABASE を保証するために必要なログの保持に使用されます。

#### 参照:

- V\$RESTORE\_POINT については、『Oracle Database リファレンス』を参照してください。
- 「フラッシュバック・データベースを使用したデータベースの巻戻し」(16-11 ページ)

## バックアップおよびデータベース・スキーマに関するレポート

Recovery Manager の REPORT コマンドを使用すると、使用可能なバックアップおよびデータベースが分析されます。この項の内容は、次のとおりです。

- [Recovery Manager バックアップに関するレポート](#)
- [保存方針に基づくバックアップが必要なファイルに関するレポート](#)
- [リカバリ不能な操作によって影響を受けるデータファイルに関するレポート](#)
- [不要なバックアップに関するレポート](#)
- [データベース・スキーマに関するレポート](#)

### Recovery Manager バックアップに関するレポート

REPORT コマンドを使用すると、次の重要な質問に回答することができます。

- バックアップが必要なファイルはどれですか。
- リカバリ不能な操作を実行したファイルはどれですか。
- 不要なため削除可能なバックアップはどれですか。
- 過去のある時点でのターゲット・データベースまたは Data Guard 環境のデータベースの物理スキーマは何ですか。
- 最近バックアップされていないファイルはどれですか。

レポートを使用すると、バックアップおよびリカバリ計画が実際にデータベースのリカバリ可能性の要件を満たしていることを確認できます。データベースがリカバリ可能であるかどうかを判断するために使用する REPORT には、主に次の 2 つの形式があります。

■ REPORT NEED BACKUP

構成済の保存方針または指定した保存方針を満たすためにバックアップする必要があるデータベース・ファイルがレポートされます。

■ REPORT UNRECOVERABLE

ダイレクト・パス・インサートなどの NOLOGGING 操作の影響を受けているためバックアップを必要とするデータベース・ファイルがレポートされます。

Recovery Manager リポジトリには、REPORT コマンドを使用してアクセスできる他の情報が含まれています。REPORT オプションの概要は、表 10-3 を参照してください。

表 10-3 REPORT のオプション

| レポートの内容     | コマンド            | 説明  |
|-------------|-----------------|---|
| 不要なバックアップ   | REPORT OBSOLETE | Recovery Manager リポジトリに記録され、不要になったため削除できる全体バックアップ、データファイルのコピーおよびアーカイブ REDO ログ。  |
| データベース・スキーマ | REPORT SCHEMA   | 指定した時点のターゲット・データベースに対するすべてのデータファイル（永続的および一時的）および表領域の名前。Data Guard 環境で Recovery Manager を使用する場合は、指定した DB_UNIQUE_NAME のスキーマをレポートできます。 |

**参照：** REPORT コマンドについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## 保存方針に基づくバックアップが必要なファイルに関するレポート

REPORT NEED BACKUP コマンドを使用して、特定の保存方針に基づくバックアップが必要なデータベース・ファイルを判断します。

引数を指定せずに REPORT NEED BACKUP を実行すると、現行の保存方針でバックアップが必要なオブジェクトがレポートされます。REDUNDANCY が 1 に設定されている構成済の保存方針の出力は、次の例のようになります。

```

RMAN> 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
    
```

**注意：** CONFIGURE RETENTION POLICY TO NONE を使用して保存方針を無効にしている場合、REPORT NEED BACKUP はエラー・メッセージを戻します。これは、保存方針がないと、Recovery Manager はバックアップする必要のあるファイルを決定できないためです。

## 様々な保存方針での Recovery Manager の REPORT NEED BACKUP の使用

次のいずれかの形式のコマンドを使用して、REPORT NEED BACKUP に様々な条件を指定できます。

- `REPORT NEED BACKUP RECOVERY WINDOW OF n DAYS`  
バックアップがリカバリ期間ベースの保存方針を満たす必要があるオブジェクトが表示されます。
- `REPORT NEED BACKUP REDUNDANCY n`  
バックアップが冗長性ベースの保存方針を満たす必要があるオブジェクトが表示されます。
- `REPORT NEED BACKUP DAYS n`  
リカバリ用に  $n$  日分より多いアーカイブ REDO ログを必要とするファイルが表示されません。
- `REPORT NEED BACKUP INCREMENTAL n`  
リカバリ用に  $n$  個より多い増分バックアップの適用を必要とするファイルが表示されます。

## 表領域およびデータファイルでの Recovery Manager の REPORT NEED BACKUP の使用

REPORT NEED BACKUP を使用して、データベース全体の確認、指定された表領域のスキップ、または様々な保存方針に対する特定の表領域またはデータファイルのみの確認を行うことができます。次に例を示します。

```
REPORT NEED BACKUP RECOVERY WINDOW OF 2 DAYS DATABASE SKIP TABLESPACE TBS_2;
REPORT NEED BACKUP REDUNDANCY 2 DATAFILE 1;
REPORT NEED BACKUP TABLESPACE TBS_3; # uses configured retention policy
REPORT NEED BACKUP INCREMENTAL 2; # checks entire database
```

**参照：** REPORT NEED BACKUP の使用可能なすべてのオプションおよび出力の様々な列ヘッダーについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## テープまたはディスク上のバックアップのみでの REPORT NEED BACKUP の使用

REPORT NEED BACKUP でテストするバックアップをディスクベースまたはテープベースのバックアップのみに制限できます。次に例を示します。

```
REPORT NEED BACKUP RECOVERY WINDOW OF 2 DAYS DATABASE DEVICE TYPE sbt;
REPORT NEED BACKUP DEVICE TYPE DISK;
REPORT NEED BACKUP TABLESPACE TBS_3 DEVICE TYPE sbt;
```

## リカバリ不能な操作によって影響を受けるデータファイルに関するレポート

ダイレクト・ロード・インサートなどのリカバリ不能な操作によってデータファイルが変更されている場合、リカバリ不能な操作では REDO が生成されないため、通常のメディア・リカバリを使用してファイルをリカバリすることはできません。このような操作の後に影響を受けるデータファイルの全体バックアップまたは増分バックアップのいずれかを実行して、リカバリ不能な操作の影響を受けるデータ・ブロックを Recovery Manager を使用してリカバリできるようにする必要があります。

### リカバリ不能な操作の影響を受けるデータファイルを識別する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。

- REPORT UNRECOVERABLE コマンドを実行します。

出力例を次に示します。

```

RMAN> REPORT UNRECOVERABLE;

Report of files that need backup due to unrecoverable operations
File Type of Backup Required Name
-----
1    full                               /oracle/oradata/trgt/system01.dbf

```

## 不要なバックアップに関するレポート

OBSOLETE キーワードを指定すると、不要な（指定した保存方針を満たす必要がない）バックアップ・セット、バックアップ・ピースおよびデータファイルのコピーをレポートできます。

### 不要なバックアップをレポートする手順

- Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
- CROSSCHECK コマンドを実行して、ディスク上のバックアップのステータスと比較してリポジトリのバックアップのステータスを更新します。

最も簡単な方法としては、次のいずれかのコマンドを使用して、ディスクまたはテープ（あるいはその両方）上のすべてのバックアップをクロスチェックすることができます。

```

CROSSCHECK BACKUP DEVICE TYPE DISK;
CROSSCHECK BACKUP DEVICE TYPE sbt;
CROSSCHECK BACKUP; # crosschecks all backups on all devices

```

実際に使用可能なバックアップ・セットが含まれるように Recovery Manager リポジトリを更新する方法の詳細は、第 11 章「[Recovery Manager バックアップおよびリポジトリ・レコードのメンテナンス](#)」を参照してください。

- REPORT OBSOLETE を実行して、リカバリの必要がなくなったために不要となったバックアップを識別します。

他のオプションを指定せずに REPORT OBSOLETE を実行すると、現行の保存方針で不要とみなされるバックアップが表示されます。次に例を示します。

```

RMAN> REPORT OBSOLETE;

Datafile Copy      44      08-FEB-06      /backup/ora_df549738566_s70_s1
Datafile Copy      45      08-FEB-06      /backup/ora_df549738567_s71_s1
Datafile Copy      46      08-FEB-06      /backup/ora_df549738568_s72_s1
Backup Set         26      08-FEB-06
  Backup Piece      26      08-FEB-06      /backup/ora_df549738682_s76_s1
.
.
.

```

RECOVERY WINDOW および REDUNDANCY オプションを指定して REPORT OBSOLETE を使用することによって、様々なリカバリ期間ベースまたは冗長性ベースの保存方針に基づいて不要とみなされるバックアップを確認できます。次に例を示します。

```

REPORT OBSOLETE RECOVERY WINDOW OF 3 DAYS;
REPORT OBSOLETE REDUNDANCY 1;

```



**参照：**

- Recovery Manager のバックアップ保存方針の概要については、5-22 ページの「バックアップの保存方針の構成」を参照してください。
- Recovery Manager バックアップを削除する方法および Recovery Manager リポジトリから Recovery Manager バックアップのレコードを削除する方法の詳細は、11-24 ページの「期限切れの Recovery Manager バックアップおよびコピーの削除」を参照してください。

## データベース・スキーマに関するレポート

REPORT SCHEMA コマンドを実行すると、データベース・ファイル、表領域などに関する情報が表示されます。REPORT SCHEMA の出力については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

FOR DB\_UNIQUE\_NAME を REPORT SCHEMA とともに指定しない場合、リカバリ・カタログ接続は任意ですが、ターゲット・データベース接続は必須です。Data Guard 環境では、REPORT SCHEMA FOR DB\_UNIQUE\_NAME を指定して環境内のデータベースのスキーマをレポートできます。この場合、Recovery Manager をターゲット・データベースに接続する必要はありません。かわりに、Recovery Manager をリカバリ・カタログに接続し、DBID を設定できます。

### データベース・スキーマに関してレポートする手順

1. Recovery Manager を起動し、対象となるデータベースに接続します。
2. 前の手順で Recovery Manager をターゲット・データベースに接続しておらず、REPORT SCHEMA に FOR DB\_UNIQUE\_NAME 句を指定する場合は、データベース DBID を設定します。たとえば、次のコマンドを入力します。

```
RMAN> SET DBID 28014364;
```

3. 次の例に示すように、REPORT SCHEMA コマンドを実行します。

```
RMAN> REPORT SCHEMA;
```

```
Report of database schema for database with db_unique_name DGRDBMS
```

```
List of Permanent Datafiles
```

```
=====
File Size(MB) Tablespace          RB segs Datafile Name
-----
1      450      SYSTEM                YES     /disk1/oracle/dbs/t_db1.f
2      141      SYSAUX                  NO      /disk1/oracle/dbs/t_ax1.f
3       50      UD1                     YES     /disk1/oracle/dbs/t_undo1.f
4       50      TBS_11                  NO      /disk1/oracle/dbs/tbs_111.f
5       50      TBS_11                  NO      /disk1/oracle/dbs/tbs_112.f
```

```
List of Temporary Files
```

```
=====
File Size(MB) Tablespace          Maxsize(MB) Tempfile Name
-----
1       40      TEMP                    32767   /disk1/oracle/dbs/t_tmp1.f
```

リカバリ・カタログを使用すると、atClause を使用して、過去の時刻、SCN またはログ順序番号を指定できます。次に例を示します。

```
RMAN> REPORT SCHEMA AT TIME 'SYSDATE-14';      # schema 14 days ago
RMAN> REPORT SCHEMA AT SCN 1000;              # schema at scn 1000
RMAN> REPORT SCHEMA AT SEQUENCE 100 THREAD 1; # schema at sequence 100
RMAN> REPORT SCHEMA FOR DB_UNIQUE_NAME standby1; # schema for database standby1
```

## V\$ ビューを使用したバックアップ・メタデータの間合せ

LIST および REPORT コマンドでは表示できない情報が V\$ ビューに表示される場合があります。この項では、V\$ ビューが特に役立つ場合について説明します。

### 過去および現行の Recovery Manager ジョブの詳細の間合せ

**Recovery Manager ジョブ**は、**Recovery Manager セッション**内で実行されるコマンドのセットです。したがって、1つの Recovery Manager ジョブに複数のコマンドを含めることができます。たとえば、2つの別々のコマンド BACKUP および RECOVER COPY を、1つのセッションで実行できます。**Recovery Manager バックアップ・ジョブ**は、1つの Recovery Manager ジョブで実行される BACKUP コマンドのセットです。たとえば、同じ Recovery Manager ジョブで実行される BACKUP DATABASE および BACKUP ARCHIVELOG ALL コマンドで、1つの Recovery Manager バックアップ・ジョブが構成されます。

V\$RMAN\_BACKUP\_JOB\_DETAILS と V\$RMAN\_BACKUP\_SUBJOB\_DETAILS の各ビューおよびこれらに対応するリカバリ・カタログのバージョンによって、Recovery Manager バックアップ・ジョブの詳細が提供されます。たとえば、これらのビューには、バックアップにかかった時間、発行されたバックアップ・ジョブの数、各バックアップ・ジョブのステータス（失敗したか完了したかなど）、ジョブの開始日時と終了日時、および実行されたバックアップのタイプが表示されます。SESSION\_KEY 列は、バックアップ・ジョブが発生した Recovery Manager セッションの一意のキーです。

多くの場合、Recovery Manager によるバックアップでは、書込みを読み取りほど行いません。Recovery Manager 圧縮のため、OUTPUT\_BYTES\_PER\_SEC 列をバックアップ速度の測定には使用できません。バックアップ速度の測定に適した列は INPUT\_BYTES\_PER\_SEC です。読み取りデータと書込みデータの比率は COMPRESSION\_RATIO 列に示されます。

#### 過去および現行の Recovery Manager ジョブの詳細を問い合わせる手順

1. 問い合わせるバックアップ履歴を持つデータベースに SQL\*Plus を接続します。
2. バックアップのタイプ、ステータスおよび開始日時と終了日時についての情報を表示するために V\$RMAN\_BACKUP\_JOB\_DETAILS ビューを問い合わせます。

次の問合せによって、バックアップ・ジョブの履歴が、Recovery Manager セッションの主キーであるセッション・キーの順序で表示されます。

```
COL STATUS FORMAT a9
COL hrs      FORMAT 999.99
SELECT SESSION_KEY, INPUT_TYPE, STATUS,
       TO_CHAR(START_TIME, 'mm/dd/yy hh24:mi') start_time,
       TO_CHAR(END_TIME, 'mm/dd/yy hh24:mi')   end_time,
       ELAPSED_SECONDS/3600                    hrs
FROM V$RMAN_BACKUP_JOB_DETAILS
ORDER BY SESSION_KEY;
```

次に、バックアップ・ジョブの履歴の出力例を示します。

| SESSION_KEY | INPUT_TYPE    | STATUS    | START_TIME     | END_TIME       | HRS |
|-------------|---------------|-----------|----------------|----------------|-----|
| 9           | DATAFILE FULL | COMPLETED | 04/18/07 18:14 | 04/18/07 18:15 | .02 |
| 16          | DB FULL       | COMPLETED | 04/18/07 18:20 | 04/18/07 18:22 | .03 |
| 113         | ARCHIVELOG    | COMPLETED | 04/23/07 16:04 | 04/23/07 16:05 | .01 |

3. Recovery Manager セッションのバックアップ・ジョブの割合を表示するために V\$RMAN\_BACKUP\_JOB\_DETAILS ビューを問い合わせます。

次の間合せによって、バックアップ・ジョブの速度が、Recovery Manager セッションの主キーであるセッション・キーの順序で表示されます。列 in\_sec および out\_sec に、1 秒当たりのデータの入力と出力が表示されます。

```
COL in_sec FORMAT a10
COL out_sec FORMAT a10
COL TIME TAKEN DISPLAY FORMAT a10
SELECT SESSION_KEY,
       OPTIMIZED,
       COMPRESSION_RATIO,
       INPUT_BYTES_PER_SEC_DISPLAY in_sec,
       OUTPUT_BYTES_PER_SEC_DISPLAY out_sec,
       TIME_TAKEN_DISPLAY
FROM   V$RMAN_BACKUP_JOB_DETAILS
ORDER BY SESSION_KEY;
```

次に、バックアップ・ジョブの速度の出力例を示します。

| SESSION_KEY | OPT | COMPRESSION_RATIO | IN_SEC | OUT_SEC | TIME_TAKEN     |
|-------------|-----|-------------------|--------|---------|----------------|
| 9           | NO  |                   | 1      | 8.24M   | 8.24M 00:01:14 |
| 16          | NO  | 1.32732239        |        | 6.77M   | 5.10M 00:01:45 |
| 113         | NO  |                   | 1      | 2.99M   | 2.99M 00:00:44 |

4. Recovery Manager セッションのバックアップのサイズを表示するために V\$RMAN\_BACKUP\_JOB\_DETAILS ビューを問い合わせます。

BACKUP DATABASE を実行すると、V\$RMAN\_BACKUP\_JOB\_DETAILS.OUTPUT\_BYTES に よって、バックアップ中のデータベースのバックアップ・ジョブによって書き込まれるバックアップ・セットの合計サイズが表示されます。登録されているすべてのデータベースのバックアップ・セットのサイズを表示するには、RC\_RMAN\_BACKUP\_JOB\_DETAILS を問い合わせます。

次の間合せによって、バックアップ・ジョブのサイズが、Recovery Manager セッションの主キーであるセッション・キーの順序で表示されます。列 in\_size および out\_size に、1 秒当たりのデータの入力と出力が表示されます。

```
COL in_size FORMAT a10
COL out_size FORMAT a10
SELECT SESSION_KEY,
       INPUT_TYPE,
       COMPRESSION_RATIO,
       INPUT_BYTES_DISPLAY in_size,
       OUTPUT_BYTES_DISPLAY out_size
FROM   V$RMAN_BACKUP_JOB_DETAILS
ORDER BY SESSION_KEY;
```

次に、バックアップ・ジョブのサイズの出力例を示します。

| SESSION_KEY | INPUT_TYPE    | COMPRESSION_RATIO | IN_SIZE | OUT_SIZE |
|-------------|---------------|-------------------|---------|----------|
| 10          | DATAFILE FULL |                   | 1       | 602.50M  |
| 17          | DB FULL       | 1.13736669        | 634.80M | 558.13M  |

**参照:** V\$RMAN\_BACKUP\_JOB\_DETAILS ビューについては、『Oracle Database リファレンス』を参照してください。

## バックアップ・ピースの暗号化ステータスの確認

V\$BACKUP\_PIECE および RC\_BACKUP\_PIECE の ENCRYPTED 列は、バックアップ・ピースが暗号化されているか (YES)、暗号化されていないか (NO) を示します。たとえば、SQL\*Plus で次の問合せを実行すると、暗号化されているバックアップ・ピースを確認できます。

```
COL BS_REC      FORMAT 99999
COL BP_REC      FORMAT 99999
COL MB          FORMAT 9999999
COL ENCRYPTED   FORMAT A7
COL TAG         FORMAT A25

SELECT S.RECID AS "BS_REC", P.RECID AS "BP_REC", P.ENCRYPTED,
       P.TAG, P.HANDLE AS "MEDIA_HANDLE"
FROM   V$BACKUP_PIECE P, V$BACKUP_SET S
WHERE  P.SET_STAMP = S.SET_STAMP
AND    P.SET_COUNT = S.SET_COUNT;
```

次に、バックアップが暗号化されていることを示す出力例を示します。

```
BS_REC BP_REC ENCRYPT TAG
-----
MEDIA_HANDLE
-----
      1      1 YES    TAG20070711T140124
/disk1/c-39525561-20070711-00

      2      2 YES    TAG20070711T140128
/disk1/c-39525561-20070711-01

      3      3 YES    TAG20070711T140130
/disk1/c-39525561-20070711-02
```

**参照:** V\$BACKUP\_PIECE ビューについては、『Oracle Database リファレンス』を参照してください。

## リカバリ・カタログ・ビューの問合せ

LIST、REPORT および SHOW コマンドを使用すると、制御ファイルおよびリカバリ・カタログ内のデータに簡単にアクセスできます。また、リカバリ・カタログ・ビューから有効な情報を取得できる場合もあります。リカバリ・カタログ・ビューとは、リカバリ・カタログ・スキーマ内に存在する RC\_ 接頭辞が付いたビューのことです。

## リカバリ・カタログ・ビュー

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のメタデータを取得し、リカバリ・カタログの表に格納します。リカバリ・カタログ・ビューは、これらの表から導出されます。リカバリ・カタログ・ビューは、ユーザーによる問合せに対して正規化または最適化されていません。

通常、リカバリ・カタログ・ビューは、Recovery Manager のレポート・コマンドほど簡単に使用できません。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場合、LIST、REPORT および SHOW コマンドを発行するのみでこのターゲット・データベースの情報を取得できます。同じリカバリ・カタログに 10 個の異なるターゲット・データベースを登録している場合、カタログ・ビューの問合せによって、10 個すべてのデータベースのすべてのインカネーションのメタデータが表示されます。多くの場合、ビュー間で複雑な選択および結合を実行して、データベース・**インカネーション**に関する使用可能な情報を抽出する必要があります。

ほとんどのカタログ・ビューには、対応する V\$ がデータベース内に存在します。たとえば、RC\_BACKUP\_PIECE は V\$BACKUP\_PIECE に対応しています。リカバリ・カタログ・ビューと対応する V\$ ビューとの主な違いとしては、各リカバリ・カタログ・ビューにはリカバリ・カタログに登録されているすべてのターゲット・データベースに関するメタデータが含まれていることがあげられます。V\$ には、ビュー自体の情報のみが含まれています。

**参照：** リカバリ・カタログ・ビューの詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## 登録されているデータベースの一意的識別子

ほとんどのリカバリ・カタログ・ビューには、列 DB\_KEY および DBINC\_KEY が含まれています。リカバリ・カタログに登録されている各データベースは、主キー (DB\_KEY 列値) または DBID (32 ビットの一意的データベース識別子) のいずれかによって一意に識別できます。データベースの各インカネーションは、DBINC\_KEY 列によって一意に識別されます。

DB\_KEY および DBINC\_KEY を使用すると、ターゲット・データベースの特定のインカネーションのレコードを取得できます。その後、他のほとんどのカタログ・ビューとの結合を実行して、このインカネーションに属するレコードを分離できます。

カタログ・ビューと V\$ ビューとの主な違いとしては、バックアップ・ファイルおよびリカバリ・ファイルに使用される一意の識別子の形式が異なることがあげられます。たとえば、V\$ARCHIVED\_LOG などの多くの V\$ ビューでは、RECID 列および STAMP 列によって連結主キーが構成されます。対応するリカバリ・カタログ・ビューでは、導出された値が主キーとして使用され、この値は単一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 の列値は、LIST コマンドの出力に表示される主キーです。

## Data Guard 環境の一意的識別子

Data Guard 環境でリカバリ・カタログを問い合わせる場合は、特別な考慮事項が適用されます。Data Guard 環境では、複数のデータベースで同じ DBID が共有されます。複数のビューに、(レコードが含まれているデータベース・インカネーションの DB\_UNIQUE\_NAME を示す) DB\_UNIQUE\_NAME 列が含まれています。Data Guard 環境のすべてのデータベースで同じ DBID が共有されますが、DB\_UNIQUE\_NAME の値は異なります。

データベース名がカタログで認識されない場合は、リカバリ・カタログに登録されている Oracle9i データベースの場合と同様に、DB\_UNIQUE\_NAME の値は null になります。また、データベースを Oracle Database 11g にアップグレードしたにもかかわらず、リカバリ・カタログのスキーマとすべてのファイルのデータベース名が一致していない場合も、列値は null になります。

リカバリ・カタログ・ビューでは、プライマリ・データベースとスタンバイ・データベースは同じ DB\_KEY を共有します。ただし、Data Guard 環境のすべてのデータベースに、一意の RC\_SITE.SITE\_KEY 値があります。たとえば、プライマリ・データベース prod およびスタンバイ・データベース standby1 の両方に値が 1 の DB\_KEY がありますが、prod の SITE\_KEY は 3 で、standby1 の SITE\_KEY は 30 です。

一部のリカバリ・カタログ・ビューには、DB\_UNIQUE\_NAME 列はありませんが、SITE\_KEY 列があります。SITE\_KEY 列を使用して RC\_SITE.SITE\_KEY と結合し、ファイルに関連付けられているデータベースの DB\_UNIQUE\_NAME を決定できます。3-8 ページの「[Data Guard 環境での Recovery Manager によるファイル管理](#)」で説明されているように、Data Guard 環境では、すべてのファイルがそのファイルを作成したプライマリ・データベースまたはスタンバイ・データベースに関連付けられています。

**参照：** Data Guard 環境でのファイルのレポート方法および管理方法については、『Oracle Data Guard 概要および管理』を参照してください。

## カタログ・ビューへのターゲット DB\_KEY 値または DBID 値の問合せ

DB\_KEY 値は、登録されているデータベースの主キーであり、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、ターゲット・データベースの DBID を使用する方法です。この DBID は、Recovery Manager を TARGET としてデータベースに接続するたびに表示されます。DBID によって、Recovery Manager のリカバリ・カタログ内に登録されている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いずれかのデータベースに関する情報を取得するとします。

### データベースの現行のインカーネーションに関する情報をカタログに問い合わせる手順

1. レコードを表示するデータベースの DBID を確認します。

DBID は、Recovery Manager を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、V\$RMAN\_OUTPUT を問い合わせるか、または V\$DATABASE ビューを問い合わせることによって取得できます。次の例では、SQL\*Plus を目的のデータベースに接続して DBID を問い合わせます。

```
SQL> CONNECT / AS SYSDBA
SQL> SELECT DBID
       2 FROM   V$DATABASE;
```

```
DBID
-----
598368217
```

2. SQL\*Plus を起動し、リカバリ・カタログの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。
3. 手順 1 で取得した DBID を持つデータベースの、データベース・キーを取得します。

その後、次の問合せを実行して、データベースの DB\_KEY を取得することができます。ここで、*dbid\_of\_target* は、手順 1 で取得した DBID です。

```
SELECT DB_KEY
FROM   RC_DATABASE
WHERE  DBID = dbid_of_target;
```

4. 手順 1 で DBID を取得したデータベースの現行のインカーネーションのレコードを問い合わせます。

ターゲット・データベースの現行のインカー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。WHERE 条件を使用して、CURRENT\_INCARNATION 列値が YES である条件を指定します。たとえば、DB\_KEY 値を 1 に指定して、ターゲット・データベースの現行のインカーネーションに設定されたバックアップ・セットに関する情報を取得するには、次の問合せを実行します。

```
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';
```

#### 参照：

- RC\_DATABASE\_INCARNATION ビューの詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- 「データベース・インカーネーション」(13-6 ページ)

## RC\_BACKUP\_FILES の問合せ

ビュー RC\_BACKUP\_FILES に対しては、リカバリ・カタログに登録されているデータベースのすべてのバックアップに関する情報を問い合わせることができます。ただし、RC\_BACKUP\_FILES を問い合わせる前に、DBMS\_RCVMAN.SETDATABASE をコールする必要があります。次の例に示すように、リカバリ・カタログに登録されているいずれかのデータベースの DBID を指定します。

```
SQL> CALL DBMS_RCVMAN.SETDATABASE(null,null,null,2283997583);
```

4 番目のパラメータは、リカバリ・カタログに登録されているデータベースの DBID にする必要があります。その他のパラメータは、すべて NULL にする必要があります。

### 参照：

- RC\_BACKUP\_FILES ビューの詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- データベースの DBID を確認する方法については、17-6 ページの「[データベースの DBID の確認](#)」を参照してください。





---

## Recovery Manager バックアップおよびリポジトリ・レコードのメンテナンス

この章では、Recovery Manager のリポジトリ・レコードおよび Recovery Manager のバックアップとコピーの管理方法について説明します。また、フラッシュ・リカバリ領域に関連するメンテナンス・タスクについても説明します。この章の内容は、次のとおりです。

- [Recovery Manager バックアップおよびリポジトリのメンテナンスの概要](#)
- [制御ファイルのリポジトリのメンテナンス](#)
- [フラッシュ・リカバリ領域のメンテナンス](#)
- [Recovery Manager リポジトリの更新](#)
- [Recovery Manager バックアップおよびアーカイブ REDO ログの削除](#)
- [データベースの削除](#)

**参照：** [リカバリ・カタログ](#)に固有の Recovery Manager のメンテナンスの問題は、[第 12 章「リカバリ・カタログの管理」](#)を参照してください。

## Recovery Manager バックアップおよびリポジトリのメンテナンスの概要

この項では、Recovery Manager リポジトリのメンテナンスの目的および基本的な概念について説明します。

### バックアップおよびリポジトリのメンテナンスの目的

メンテナンス計画では、[フラッシュ・リカバリ領域](#)、[バックアップの保存方針](#)および[アーカイブ REDO ログの削除方針](#)を構成することをお勧めします。この場合、バックアップおよびアーカイブ REDO ログは、必要に応じてデータベースによって自動的にメンテナンスおよび削除されます。ただし、データベース・バックアップおよびアーカイブ REDO ログを手動でメンテナンスする必要がある場合もあります。

Recovery Manager バックアップを管理する場合は、次の関連タスクを実行します。

- ディスクまたはテープに格納されているデータベース・バックアップの管理
- Recovery Manager リポジトリ内のバックアップのレコードの管理

Recovery Manager のメンテナンスでの重要なタスクに、不要となったバックアップの削除があります。[フラッシュ・リカバリ領域](#)を構成すると、この領域内の不要なファイルはデータベースによって自動的に削除されます。ただし、この場合も、バックアップおよびコピーをテープから削除する必要がある場合があります。データベース全体を削除する必要がある場合もあります。Recovery Manager コマンドを使用すると、これらのタスクを実行できます。

フラッシュ・リカバリ領域には、不定期のメンテナンスが必要な場合があります。たとえば、フラッシュ・リカバリ領域が一杯になり、フラッシュ・リカバリ領域に領域を追加する必要がある場合があります。また、リカバリ領域の場所を変更する必要がある場合もあります。

ディスクおよびテープ上のファイルの実際の状態が [Recovery Manager リポジトリ](#) に反映されない場合があります。たとえば、ユーザーがオペレーティング・システムのユーティリティを使用して、バックアップをディスクから削除する場合があります。この場合、Recovery Manager リポジトリには、実際には存在しないファイルが存在しているように表示されます。Recovery Manager バックアップが格納されているテープが破損した場合も、同様の状況が発生します。[Recovery Manager のメンテナンス・コマンド](#)を使用すると、リポジトリを正しい情報で更新できます。

### バックアップおよびリポジトリのメンテナンスの基本的な概念

Recovery Manager のメンテナンス・コマンドの概要を次に示します。

- CATALOG コマンドを使用すると、Recovery Manager リポジトリに現在記録されていない Recovery Manager バックアップおよびユーザー管理のバックアップに関するレコードを追加したり、記録されているバックアップのレコードを削除することができます。
- CHANGE コマンドを使用すると、Recovery Manager リポジトリのレコードのステータスを更新できます。
- CROSSCHECK コマンドを使用すると、論理バックアップ・レコードをバックアップ・ストレージ内のファイルの物理的な状態と同期化できます。
- DELETE コマンドを使用すると、オペレーティング・システムからバックアップを削除できます。

#### メンテナンス・コマンドおよび Recovery Manager リポジトリ・メタデータ

Recovery Manager は、操作の実行対象となる各ターゲット・データベースの制御ファイルに、そのメタデータを常に格納します。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と、Recovery Manager は、このターゲット・データベースのメタデータをリカバリ・カタログに格納します。Recovery Manager のすべてのメンテナンス・コマンドは、リカバリ・カタログを使用しているかどうかに関係なく動作します。

**参照:** [「リカバリ・カタログのメンテナンス」](#) (12-22 ページ)

## Data Guard 環境でのメンテナンス・コマンド

3-8 ページの「Data Guard 環境での Recovery Manager によるファイル管理」で説明されているように、Data Guard 環境では、バックアップまたはコピーを作成するデータベースはファイルに関連付けられています。たとえば、Recovery Manager がターゲット・データベース standby1 に接続し、このデータベースをバックアップする場合、このバックアップは standby1 に関連付けられます。

3-8 ページの「Data Guard 環境での Recovery Manager によるファイル管理」で指定されている基準に従って Recovery Manager からバックアップにアクセスできるかぎり、プライマリ・データベースまたはスタンバイ・データベースへの接続時に、バックアップに対して CHANGE、DELETE および CROSSCHECK などの Recovery Manager メンテナンス・コマンドを使用できます。

**Data Guard 環境でのクロスチェック** クロスチェックする場合、Recovery Manager は、ファイルに関連付けられているデータベースへの接続時にのみ、ファイルのステータスを AVAILABLE から EXPIRED に更新できます。このため、Recovery Manager でクロスチェックしてもファイルが検出されず、Recovery Manager が TARGET として接続されていないデータベースにファイルに関連付けられている場合は、そのファイルに関連付けられているターゲット・データベースへの接続時に、Recovery Manager によってクロスチェックの実行が求められます。Recovery Manager は、CROSSCHECK または CHANGE ... AVAILABLE コマンドの実行時にクロスチェックを行うことに注意してください。

**Data Guard 環境での削除** Recovery Manager は、データベースへの接続時にファイルを削除できます。ファイルに関連付けられているデータベースに Recovery Manager が TARGET として接続されておらず、Recovery Manager でファイルを正常に削除できない場合は、Recovery Manager によって、ファイルに関連付けられているデータベースに TARGET として接続するように求められます。このため、ファイルのメタデータを削除するには、DELETE ... FORCE を使用する必要があります。

**Data Guard 環境での Recovery Manager メタデータへの更新** メンテナンス・コマンドで Recovery Manager のメタデータのみを変更した場合は、Recovery Manager を TARGET として Data Guard 環境のデータベースに接続できます。メタデータのみを変更するコマンドは、次のとおりです。

- CHANGE ... UNAVAILABLE または CHANGE ... UNCATALOG
- CHANGE ... KEEP または CHANGE ... NOKEEP
- CHANGE ... RESET DB\_UNIQUE\_NAME

デフォルトでは、CHANGE コマンドを使用すると、3-9 ページの「Data Guard 環境でのバックアップのアクセシビリティについて」で指定されている規則に従ってアクセス可能なファイルに対してのみ操作が行われます。ただし、FOR DB\_UNIQUE\_NAME オプションを使用すると、ターゲット・データベース以外のデータベースに関連付けられているファイルのステータスを変更できます。

**データベースに関連付けられていないファイル** 特定のファイルで DB\_UNIQUE\_NAME が認識されない場合があります。たとえば、データベース名がカタログで認識されない場合は、リカバリ・カタログに登録されている Oracle9i データベースの場合と同様に、DB\_UNIQUE\_NAME の値は null になります。また、データベースを現行のバージョンにアップグレードしたにもかかわらず、リカバリ・カタログ・スキーマとすべてのファイルの DB\_UNIQUE\_NAME 値が一致していないため、行の DB\_UNIQUE\_NAME が null になる場合もあります。デフォルトでは、Recovery Manager は、TARGET として接続されているデータベースに SITE\_KEY が null のファイルに関連付けます。CHANGE ... RESET DB\_UNIQUE\_NAME を明示的に使用して別のデータベースに関連付けないかぎり、バックアップはそのデータベースに関連付けられたままです。

**参照：**

- Data Guard 環境で Recovery Manager を使用してファイルのバックアップおよびリストアを実行する方法については、『Oracle Data Guard 概要および管理』を参照してください。
- Recovery Manager メンテナンス・コマンドについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## 制御ファイルのリポジトリのメンテナンス

Recovery Manager は、リカバリ・カタログを使用しなくても動作するように設計されています。ただし、リカバリ・カタログを使用しないことを選択すると、各ターゲット・データベースの制御ファイルは、Recovery Manager メタデータ用の排他的なリポジトリになります。制御ファイルへの情報の格納方法を理解し、バックアップおよびリカバリ計画によって制御ファイルが保護されるようにする必要があります。

**参照：** 制御ファイルの概要および管理方法の詳細は、『Oracle Database 管理者ガイド』を参照してください。

## 制御ファイル・レコード

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つの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には、必要に応じて上書き可能な、重要度が低い情報が含まれます。これらのレコードには、データベースによって持続的に生成される情報が含まれます。使用可能なレコード・スロットがすべて使用されると、データベースは、制御ファイルを拡張して新規レコード用の領域を作成するか、または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は、レコードを再利用するまでの最短保管日数を指定します。

非循環再利用レコードには、頻繁には変更されず、上書きできない重要な情報が含まれます。非循環再利用レコードに含まれる情報は、データファイル、オンライン REDO ログ・ファイル、REDO スレッドなどです。

ターゲット・データベースのバックアップを作成すると、データベースによってそれらのバックアップが制御ファイルに記録されます。新しいレコードを追加したことによって制御ファイルが過度に大きくなるように、指定したしきい値より古いレコードは再利用可能になります。CONTROL\_FILE\_RECORD\_KEEP\_TIME 初期化パラメータは、レコードが上書き可能になるまでの最小日数を決定します。

```
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となります。上書きされたレコードの情報は失われます。再利用可能な最も古いレコードが最初に使用されます。

データベースでは、新しい Recovery Manager リポジトリ・レコードを制御ファイルに追加する必要があるにもかかわらず、しきい値より古いレコードが存在しない場合、制御ファイル・サイズの拡大が試行されます。基礎となるオペレーティング・システムで（ディスクが一杯などの状況によって）制御ファイルを拡大できない場合は、データベースによって制御ファイル内の最も古いレコードが上書きされます。

上書きは、**自動診断リポジトリ**内のアラート・ログに記録されます。上書きされる各レコードに対するエントリが、アラート・ログに次のように記録されます。

```
kccwnc: following control file record written over:
RECID #72 Recno 72 Record timestamp
07/28/06 22:15:21
Thread=1 Seq#=3460
Backup set key: stamp=372031415, count=17
Low scn: 0x0000.3af33f36
07/27/06 21:00:08
Next scn: 0x0000.3af3871b
07/27/06 23:23:54
Resetlogs scn and time
scn: 0x0000.00000001
```

## フラッシュ・リカバリ領域および制御ファイル・レコード

フラッシュ・リカバリ領域に作成されたファイルに関する情報を含む制御ファイルのレコードが再利用される際、そのファイルが削除対象である場合は、データベースによってフラッシュ・リカバリ領域からのそのファイルの削除が試行されます。削除対象でない場合は、このファイルのレコードを含む制御ファイルのセクションのサイズが拡大されます。この拡大に関して、アラート・ログに次の例のようなメッセージが記録されます。ここで、*nnnn*は、制御ファイル・レコードのタイプの番号です。

```
kccwnc: trying to expand control file section nnnn for Oracle Managed Files
```

制御ファイルがホスト・オペレーティング・システムでサポートされている最大サイズに達している場合、制御ファイルを拡大することはできません。このような場合は、次の警告がアラート・ログに表示されます。

```
WARNING: Oracle Managed File filename is unknown to control file. This is the result of limitation in control file size that could not keep all recovery area files.
```

前述のメッセージは、構成されている保存方針を満たすために必要なすべてのフラッシュ・リカバリ領域ファイルのレコードを制御ファイルで保持できないことを示しています。次の項では、この状況に対応する方法について説明します。

**参照：** CONTROL\_FILE\_RECORD\_KEEP\_TIME 初期化パラメータについては、『Oracle Database リファレンス』を参照してください。

## 制御ファイル・レコードの消失の防止

制御ファイル・レコードの上書きによって Recovery Manager メタデータが消失することを防止するには、リカバリ・カタログを使用する方法が最適です。リカバリ・カタログを使用できない場合は、次の方法を使用できます。

- CONTROL\_FILE\_RECORD\_KEEP\_TIME の値を、保持する必要がある最も古いファイルの期間より少し長く設定します。たとえば、データベース全体を 1 週間に 1 回バックアップする場合は、すべてのバックアップを 7 日間以上保持する必要があります。CONTROL\_FILE\_RECORD\_KEEP\_TIME の値を 10 や 14 などに設定します。CONTROL\_FILE\_RECORD\_KEEP\_TIME のデフォルト値は 7 日間です。

---

**注意：** リカバリ・カタログの使用にかかわらず、CONTROL\_FILE\_RECORD\_KEEP\_TIME が 0 に設定されている場合は Recovery Manager を使用しないでください。Recovery Manager を使用すると、バックアップ・レコードが失われる場合があります。

---

- 制御ファイルは、拡大できるように **RAW デバイス**ではなくファイル・システムに格納します。
- アラート・ログを監視して、Oracle によって制御ファイル・レコードが上書きされないようにします。アラート・ログは、**自動診断リポジトリ**内にあります。

フラッシュ・リカバリ領域を使用する場合は、次のガイドラインに従って、バックアップ保存方針を満たすために必要なすべてのフラッシュ・リカバリ領域ファイルのレコードを制御ファイルで保持できなくなるような状況を回避します。

- 制御ファイルのブロック・サイズがすでに最大に達している場合は、より大きなブロック・サイズ（可能な場合 32KB）を使用します。  
これを実現するには、SYSTEM 表領域のブロック・サイズを制御ファイルのブロック・サイズ以上に設定し、DB\_BLOCK\_SIZE の変更後に制御ファイルを再作成する必要があります。フラッシュ・リカバリ領域内のファイルはカタログに再度追加されますが、テープに格納されているファイルのレコードは失われることに注意してください。
- フラッシュ・リカバリ領域内のファイルをテープなどの 3 次ストレージ・デバイスにバックアップして、削除対象にします。  
たとえば、BACKUP RECOVERY AREA を使用すると、フラッシュ・リカバリ領域内のファイルを明示的にメディア・マネージャにバックアップできます。
- バックアップの保存方針によってバックアップおよびアーカイブ・ログがビジネス要件より長く保持されている場合は、その保存方針をより短いリカバリ期間またはより低い冗長性のレベルに変更して、フラッシュ・リカバリ領域内のより多くのファイルを削除対象にすることができます。

## 制御ファイルの保護

リカバリ・カタログを使用して Recovery Manager メタデータを格納していない場合は、各ターゲット・データベースの制御ファイルを保護することがさらに重要となります。次の計画を使用すると、制御ファイルを保護することができます。

### 制御ファイルを保護する手順

1. **多重化**またはオペレーティング・システムの**ミラー化**によって、制御ファイルの冗長コピーを作成します。

これによって、制御ファイルのサブセットが消失した場合に、ユーザーが制御ファイルをバックアップからリストアする必要がなくなります。2 つ以上の多重制御ファイルまたはミラー化制御ファイルを別々のディスクで使用することをお勧めします。

2. **制御ファイルの自動バックアップ**機能を構成します。

この場合、特定の Recovery Manager コマンドを実行すると、制御ファイルが自動的にバックアップされます。制御ファイルの自動バックアップを使用できるかぎり、Recovery Manager は、サーバー・パラメータ・ファイルおよびバックアップ制御ファイルをリストアしてデータベースをマウントできます。制御ファイルをマウントした後、データベースの残りの部分をリストアできます。

3. データベースの DBID のレコードを保持します。

制御ファイルが失われると、データベースのリカバリに DBID が必要になる場合があります。

#### 参照：

- 制御ファイルの手動バックアップおよび自動バックアップについては、8-9 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。
- 「**制御ファイルおよびサーバー・パラメータ・ファイルの自動バックアップ**」(7-12 ページ)



## フラッシュ・リカバリ領域のメンテナンス

フラッシュ・リカバリ領域は多くの場合自動管理されますが、DBA の介入が必要になることもあります。

### フラッシュ・リカバリ領域の規則の削除

フラッシュ・リカバリ領域の内容および永続的なファイルと一時的なファイルの違いについては、5-15 ページの「[フラッシュ・リカバリ領域の概要](#)」を参照してください。この章を読む前に、これらの項を参照してください。次の規則によって、ファイルがリカバリ領域からの削除対象となるタイミングが制御されます。

- 永続的なファイルは削除の対象となりません。
- 保存方針に従って不要になったファイルは削除対象となります。  
保存方針の構成方法については、5-22 ページの「[バックアップの保存方針の構成](#)」を参照してください。
- テープにコピーされた一時ファイルは削除対象となります。
- アーカイブ REDO ログは、ログのすべてのコンシューマが要件を満たすまで削除対象となりません。  
ログが削除対象となるタイミングを決定する[アーカイブ REDO ログの削除方針](#)の構成方法については、5-27 ページの「[アーカイブ REDO ログの削除方針の構成](#)」を参照してください。ログのコンシューマには、Recovery Manager、スタンバイ・データベース、Oracle Streams データベースおよびフラッシュバック・データベース機能などがあります。Data Guard 環境でのアーカイブ REDO ログ管理については、『Oracle Data Guard 概要および管理』を参照してください。
- ロジカル・スタンバイ・データベースで LogMiner セッションによってマイニングされた外部のアーカイブ・ログは削除対象になります。[外部のアーカイブ REDO ログ](#)の DBID は、通常のアーカイブ REDO ログの DBID とは異なっています。

フラッシュ・リカバリ領域からのファイルの削除を制御する安全で確実な方法は、保存方針 (5-22 ページの「[バックアップの保存方針の構成](#)」を参照) およびアーカイブ・ログの削除方針 (5-27 ページの「[アーカイブ REDO ログの削除方針の構成](#)」を参照) を構成する方法です。テープに移動したファイルがディスクに保存される可能性を高くするには、フラッシュ・リカバリ領域の割当て制限を大きくします。

### フラッシュ・リカバリ領域の領域使用状況の監視

V\$RECOVERY\_FILE\_DEST ビューおよび V\$FLASH\_RECOVERY\_AREA\_USAGE ビューを使用すると、フラッシュ・リカバリ領域に十分な領域が割り当てられているかどうかを確認できます。V\$RECOVERY\_FILE\_DEST ビューを問い合わせ、フラッシュ・リカバリ領域の現在の位置、ディスク割当て制限、使用領域、ファイル削除による再利用可能な領域、およびファイルの合計数を確認します。たとえば、[例 11-1](#) に示す問合せを入力します (出力例も示します)。SPACE 列はバイト単位で指定することに注意してください。

#### 例 11-1 フラッシュ・リカバリ領域の領域消費量

```
SELECT * FROM V$RECOVERY_FILE_DEST;
```

| NAME         | SPACE_LIMIT | SPACE_USED | SPACE_RECLAIMABLE | NUMBER_OF_FILES |
|--------------|-------------|------------|-------------------|-----------------|
| /mydisk/rcva | 5368709120  | 109240320  | 256000            | 28              |

異なるタイプのファイルで使用されている合計ディスク割当て制限の割合を確認するには、V\$FLASH\_RECOVERY\_AREA\_USAGE ビューを問い合わせます。また、不要なファイル、冗長なファイルまたはすでにテープにバックアップされているファイルを削除することによって、各タイプのファイル用に再利用できる領域の量を判断することもできます。たとえば、次の問合せを入力します (出力例も示します)。

```
SELECT * FROM V$FLASH_RECOVERY_AREA_USAGE;
```

| FILE_TYPE    | PERCENT_SPACE_USED | PERCENT_SPACE_RECLAIMABLE | NUMBER_OF_FILES |
|--------------|--------------------|---------------------------|-----------------|
| CONTROLFILE  | 0                  | 0                         | 0               |
| ONLINELOG    | 2                  | 0                         | 22              |
| ARCHIVELOG   | 4.05               | 2.01                      | 31              |
| BACKUPPIECE  | 3.94               | 3.86                      | 8               |
| IMAGECOPY    | 15.64              | 10.43                     | 66              |
| FLASHBACKLOG | .08                | 0                         | 1               |

保証付きリストア・ポイントがデータベースに定義されている場合は、保証を満たすために必要なファイル用にフラッシュ・リカバリ領域で使用される領域の量を監視する必要があります。保証付きリストア・ポイントを表示する問合せ（10-10 ページの「リストア・ポイントの表示」を参照）を使用し、STORAGE\_SIZE 列を参照して、各保証付きリストア・ポイントに関連するファイルに必要な領域を決定します。

**参照：** V\$RECOVERY\_FILE\_DEST および V\$FLASH\_RECOVERY\_AREA ビューの詳細は、『Oracle Database リファレンス』を参照してください。

## フラッシュ・リカバリ領域でのフラッシュバック・ログの領域の管理

フラッシュバック・ログの削除規則については、5-28 ページの「Oracle Flashback Database およびリストア・ポイントの構成」を参照してください。フラッシュ・リカバリ領域のフラッシュバック・ログは、フラッシュバックの保存ターゲットを設定するか、または保証付きリストア・ポイントを使用する以外に直接管理することはできません。ただし、フラッシュバック・ログの保存に使用できる領域を最大化するために、フラッシュ・リカバリ領域を全体として管理することはできます。これによって、フラッシュバック目標を達成する可能性が高くなります。

フラッシュバック・ログ用の領域を確保するには、BACKUP RECOVERY AREA、BACKUP BACKUPSET などのコマンドを使用して、フラッシュバック・リカバリ領域の他の内容をテープにバックアップします。Oracle Database では、不要になったファイルがフラッシュ・リカバリ領域から自動的に削除されます。バックアップをテープにオフロードしても、バックアップ保存方針およびフラッシュバック保存目標を満たす十分な領域が作成されない場合は、フラッシュ・リカバリ領域により多くの領域を割り当てます。

---

**注意：** フラッシュバック・ログはバックアップできません。このため、フラッシュ・リカバリ領域の内容をテープにバックアップする場合、BACKUP RECOVERY AREA を使用してもフラッシュバック・ログは含まれません。

---

## フラッシュ・リカバリ領域が一杯になった場合の対応

Recovery Manager の保存方針によってフラッシュバック・リカバリ領域のディスク割当て制限より大きいバックアップのセットを保存する必要がある場合、またはこの保存方針が NONE に設定されている場合、フラッシュ・リカバリ領域は再利用可能な領域がなくなるまで完全に一杯になることがあります。

再利用可能な領域が 15% 未満になると警告アラートが発行され、再利用可能な領域が 3% 未満になるとクリティカル・アラートが発行されます。この状況を DBA に警告するために、アラート・ログおよび DBA\_OUTSTANDING\_ALERTS 表（Enterprise Manager で使用）にエントリが追加されます。ただし、フラッシュ・リカバリ領域内の領域は、再利用可能な領域がなくなるまでデータベースによって継続して消費されます。

リカバリ領域が完全に一杯になると、次のエラーを受信します。ここで、*nnnnn* は必要なバイト数、*nnnnnn* はディスク割当て制限です。

```
ORA-19809: limit exceeded for recovery files
ORA-19804: cannot reclaim nnnnn bytes disk space from nnnnnn limit
```



削除対象のファイルがない場合にフラッシュ・リカバリ領域が一杯になった状態を解消する方法として、次のいくつかの方法があります。

- 使用可能なディスク領域を追加し、DB\_RECOVERY\_FILE\_DEST\_SIZE を増加して、追加した領域を反映させます。
- フラッシュ・リカバリ領域からテープなどの 3 次ストレージ・デバイスにバックアップを移動します。

リカバリ領域のすべてのファイルを一度にテープにバックアップする簡単な方法の 1 つに、BACKUP RECOVERY AREA コマンドがあります。バックアップをリカバリ領域からテープに転送した後、フラッシュ・リカバリ領域からファイルを削除できます (11-20 ページの「[Recovery Manager バックアップおよびアーカイブ REDO ログの削除](#)」を参照)。フラッシュバック・ログは、リカバリ領域外にはバックアップできないため、BACKUP RECOVERY AREA ではバックアップされないことに注意してください。

- オペレーティング・システム・ユーティリティを使用して削除されたファイルに対しては、DELETE を実行します。

ホスト・オペレーティング・システムのコマンドを使用してファイルを削除した場合、その結果できる空き領域はデータベースでは認識されません。Recovery Manager の CROSSCHECK コマンドを実行して、フラッシュ・リカバリ領域の内容を再確認し、期限切れのファイルを特定した後、DELETE EXPIRED コマンドを使用して、Recovery Manager リポジトリからすべての[期限切れのバックアップ](#)を削除することができます。

- 保証付きリストア・ポイントが必要であることを確認します。必要ない場合は、11-9 ページの「[リストア・ポイントの削除](#)」の説明に従って削除します。

保証付きリストア・ポイントで必要ないフラッシュバック・ログは、フラッシュ・リカバリ領域内の他のファイルの領域を確保するために自動的に削除されます。保証付きリストア・ポイントを使用すると、リストア・ポイント SCN までフラッシュバック・データベースを実行するために必要なフラッシュバック・ログを強制的に保存できます。

- バックアップ保存方針を確認します。Data Guard を使用している場合は、アーカイブ REDO ログの削除方針も確認します。

**参照：** 保存方針の決定については第 8 章「[データベースのバックアップ](#)」、Data Guard 使用時のアーカイブ・ログの削除方針については『[Oracle Data Guard 概要および管理](#)』を参照してください。

## リストア・ポイントの削除

既存のリストア・ポイントが不要であること確認した場合、または既存のリストア・ポイントの名前を使用して新しいリストア・ポイントを作成する必要がある場合は、SQL 文 DROP RESTORE POINT を使用して、リストア・ポイントを削除できます。たとえば、次のように入力します。

```
DROP RESTORE POINT before_app_upgrade;
```

通常のリストア・ポイントおよび保証付きリストア・ポイントの両方の削除に、同じ文を使用します。

通常のリストア・ポイントは、明示的に削除しなくても、最終的には制御ファイルからエージ・アウトされます。制御ファイル内のリストア・ポイントは、次の規則に基づいて保存されます。

- 最新の 2048 個のリストア・ポイントは、その保存期間に関係なく、常に制御ファイルに保存されます。
- CONTROL\_FILE\_RECORD\_KEEP\_TIME の値よりも新しいリストア・ポイントは、定義されているリストア・ポイント数に関係なく、いずれも保持されます。

前述の条件のいずれかを満たさない通常のリストア・ポイントは、制御ファイルからエージ・アウトされる場合があります。保証付きリストア・ポイントが、制御ファイルからエージ・アウトされることはありません。これらは、明示的に削除されるまで保存されます。

**参照：** SQL の DROP RESTORE POINT 文の詳細は、『Oracle Database SQL 言語リファレンス』を参照してください。

## 新しい場所へのフラッシュ・リカバリ領域の変更

データベースのフラッシュ・リカバリ領域を新しい場所に移動する必要がある場合は、次の手順を実行します。

1. ターゲット・データベースで SQL\*Plus を起動し、DB\_RECOVERY\_FILE\_DEST 初期化パラメータを変更します。たとえば、次のコマンドを入力し、移動先を ASM ディスク・グループ disk1 に設定します。

```
ALTER SYSTEM SET DB_RECOVERY_FILE_DEST='+disk1' SCOPE=BOTH SID='*';
```

このパラメータを変更すると、すべての新しいフラッシュ・リカバリ領域ファイルが新しい場所に作成されます。

2. 古いフラッシュ・リカバリの場所にある永続的なファイル、フラッシュバック・ログおよび一時的なファイルをそのままにするか、または移動します。

既存のファイルをフラッシュ・リカバリに残した場合、古いフラッシュ・リカバリ領域の一時的なファイルは、削除対象となったときに削除されます。

古いファイルを新しいフラッシュ・リカバリ領域に移動する必要がある場合は、[第 26 章「ASM でのデータの移行の実行」](#)を参照してください。ファイルをフラッシュ・リカバリ領域に移動したり、フラッシュ・リカバリ領域から移動する場合は、Recovery Manager を使用してデータベース・ファイルを ASM ディスク・グループに移動したり、ASM ディスク・グループから移動するときと同じ手順を使用できます。

## フラッシュ・リカバリ領域の無効化

フラッシュ・リカバリ領域を無効にする前に、すべての保証付きリストア・ポイントを削除してから、フラッシュバック・データベースを無効にする必要があります。これらの前提条件が満たされたら、DB\_RECOVERY\_FILE\_DEST 初期化パラメータを NULL 文字列に設定することによって、フラッシュ・リカバリ領域を無効にできます。たとえば、次の SQL 文を使用して、実行中のデータベースのパラメータを変更します。

```
ALTER SYSTEM SET DB_RECOVERY_FILE_DEST='' SCOPE=BOTH SID='*';
```

これで、データベースは以前の DB\_RECOVERY\_FILE\_DEST の位置に格納されているファイルにフラッシュ・リカバリ領域の領域管理機能を提供しなくなります。ただし、これらのファイルは、Recovery Manager リポジトリで認識され、バックアップおよびリストア・アクティビティで使用できます。

## ファイル作成時のインスタンスのクラッシュへの対応

通常、フラッシュ・リカバリ領域は自動管理されます。ただし、フラッシュ・リカバリ領域内のファイルの作成中にインスタンスがクラッシュした場合、そのファイルがフラッシュ・リカバリ領域内に残存することがあります。この状況が発生した場合は、アラート・ログに次のエラーが表示されます。ここで、location は、フラッシュ・リカバリ領域の場所です。

```
ORA-19816: WARNING: Files may exist in location that are not known to database.
```

このような場合は、Recovery Manager コマンド CATALOG RECOVERY AREA を使用して、このようなファイルをカタログに再度追加します。問題のファイルのヘッダーが破損している場合は、オペレーティング・システム・ユーティリティを使用してファイルを手動で削除します。

## フラッシュバック・データベースのパフォーマンスの影響の監視

自動ワークロード・リポジトリ (AWR) では、データベースの問題検出および自己チューニングのためにパフォーマンス統計の収集、処理および保持を行うことによって、データベースの統計収集が自動化されています。システムにおけるフラッシュバック・データベースのワークロードの監視を行う場合、そのデータ分析方法は複数あります。たとえば、フラッシュバック・データベースを有効にする前と後で、AWR レポートを比較できます。また、AWR スナップショットを参照して、フラッシュバック・ロギングによって発生するシステムの使用状況を明確にすることもできます。たとえば、flashback buf free by RVWR が最上位の待機イベントとして表示されている場合は、Oracle Database ではフラッシュバック・ログへの迅速な書き込みを行うことができないことを示しています。このような場合は、5-35 ページの「最適なフラッシュバック・データベースのパフォーマンスのための環境の構成」で説明したいずれかの方法で、フラッシュ・リカバリ領域に使用されているファイル・システムおよび記憶域をチューニングすることをお勧めします。

V\$FLASHBACK\_DATABASE\_STAT ビューには、データベースによって記録されたフラッシュバック・データのバイト数が表示されます。ビューの各行には、(通常は 1 時間にわたって) 累積された統計が表示されます。FLASHBACK\_DATA および REDO\_DATA 列には、一定期間中に書き込まれたフラッシュバック・データおよび REDO データのバイト数がそれぞれ示されます。DB\_DATA 列には、読取りおよび書き込みが行われたデータ・ブロックのバイト数が示されます。FLASHBACK\_DATA および REDO\_DATA は順次書き込みに対応し、DB\_DATA はランダム読取りおよび書き込みに対応することに注意してください。

順次 I/O とランダム I/O の違いのため、I/O オーバーヘッドは、フラッシュバック・ログに対して発行された I/O 操作の数によってより明確に示されます。表 11-1 に示す V\$SYSSTAT 統計は、様々な目的のためにインスタンスによって発行された I/O 操作の数を示しています。

表 11-1 V\$SYSSTAT 統計

| 列名                         | 列の意味                               |
|----------------------------|------------------------------------|
| physical write I/O request | データ・ブロックの書き込みのために発行された書き込み操作の数     |
| physical read I/O request  | データ・ブロックの読取りのために発行された読取り操作の数       |
| redo writes                | REDO ログへの書き込みのために発行された書き込み操作の数     |
| flashback log writes       | フラッシュバック・ログへの書き込みのために発行された書き込み操作の数 |

### 参照:

- V\$SYSSTAT ビューの列の詳細は、『Oracle Database リファレンス』を参照してください。
- 自動ワークロード・リポジトリの詳細は、『Oracle Database パフォーマンス・チューニング・ガイド』を参照してください。
- AWR レポートの詳細は、『Oracle Database 2 日でパフォーマンス・チューニング・ガイド』を参照してください。

## I/O エラーが発生した場合のフラッシュバック・ライター (RVWR) の動作

フラッシュバックが有効になっている場合、または保証付きリストア・ポイントが存在する場合は、バックグラウンド・プロセス RVWR によって、フラッシュバック・リカバリ領域内のフラッシュバック・データベース・ログにフラッシュバック・データが書き込まれます。RVWR で I/O エラーが発生した場合は、次の動作が予測されます。

- 保証付きリストア・ポイントが定義されている場合は、RVWR で I/O エラーが発生すると、インスタンスで障害が発生します。

- 保証付きリストア・ポイントが定義されていない場合は、RVWRでI/Oエラーが発生しても、インスタンスで障害は発生しません。次の場合に注意してください。
  - プライマリ・データベースでは、データベースをオープンしている間はフラッシュバック・データベースが自動的に無効になります。既存のすべてのトランザクションおよび問合せは、影響を受けずに続行されます。これは、シングル・インスタンス・データベースおよび Oracle RAC データベースの両方で予測される動作です。
  - フィジカル・スタンバイ・データベースまたはロジカル・スタンバイ・データベースでは、RVWRがハングアップしてI/Oを定期的に再試行しているように見えます。最終的に、これによってロジカル・スタンバイ・データベースがハングアップするか、またはフィジカル・スタンバイ・データベースの管理リカバリがハングアップする場合があります（Oracle Databaseでは、スタンバイ・インスタンスの障害は発生しません。最大保護モードでプライマリ・データベースの障害が発生しないようにするためです。）。このハングアップを解決するには、SHUTDOWN ABORT または ALTER DATABASE FLASHBACK OFF を発行します。

## Recovery Manager リポジトリの更新

この項では、Recovery Manager リポジトリが、ディスクおよびテープに格納されている Recovery Manager 関連ファイルの実情を正確に反映していることを確認する方法について説明します。リポジトリとリポジトリによって記録されるファイル間の相違は、テープまたはディスクの障害、ユーザー管理のコピー、Recovery Manager 関連ファイルの削除などのいくつかの原因によって発生する可能性があります。

この項の内容は、次のとおりです。

- [Recovery Manager リポジトリのクロスチェック](#)
- [バックアップおよびコピーのリポジトリ・ステータスの変更](#)
- [Recovery Manager リポジトリへのバックアップ・レコードの追加](#)
- [Recovery Manager リポジトリからのレコードの削除](#)

## Recovery Manager リポジトリのクロスチェック

リカバリ・カタログまたは制御ファイルのバックアップに関するデータがディスクまたはメディア管理カタログの該当するデータと同期されていることを確認するには、[クロスチェック](#)を実行します。CROSSCHECK コマンドでは、Recovery Manager リポジトリに現在記録されているファイルのみが処理されます。

フラッシュ・リカバリ領域、バックアップの保存方針およびアーカイブ REDO ログの削除方針を使用する場合は、クロスチェックを頻繁に実行する必要はありません。Recovery Manager 以外の方法でファイルを削除する場合は、クロスチェックを定期的に行って、リポジトリ・データが最新の状態で保持されるようにする必要があります。

### Recovery Manager のクロスチェック

クロスチェックは、リポジトリ・レコードが物理的な状態と一致しないバックアップに関する、Recovery Manager リポジトリの期限切れの情報を更新します。たとえば、ユーザーがオペレーティング・システム・コマンドを使用してアーカイブ・ログをディスクから削除すると、ログが実際にはディスク上に存在していない場合でも、リポジトリではディスク上に存在しているように示されます。

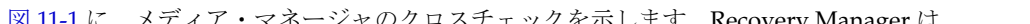
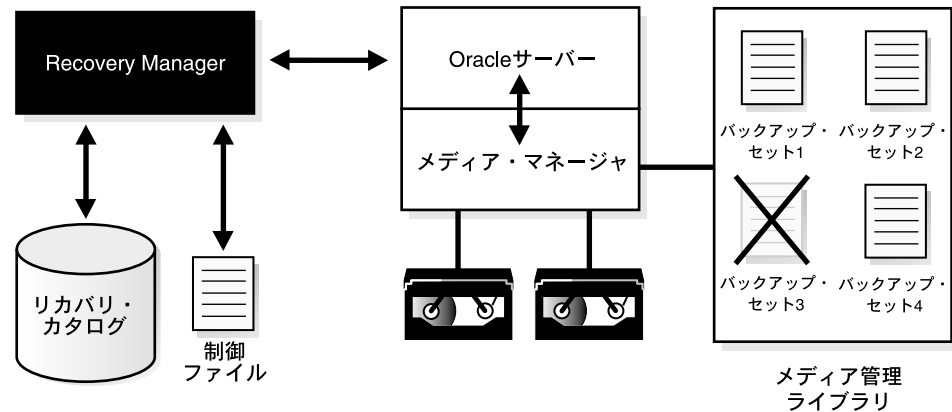
 **図 11-1** に、メディア・マネージャのクロスチェックを示します。Recovery Manager は、Recovery Manager リポジトリに対して、チェックする 4 つのバックアップ・セットの名前と場所を問い合わせます。Recovery Manager は、この情報をターゲット・データベース・サーバーに送信します。ターゲット・データベース・サーバーは、メディア管理ソフトウェアにバックアップについて問い合わせます。メディア管理ソフトウェアは、メディア・カタログを確認して、バックアップ・セット 3 が欠落していることをサーバーにレポートします。Recovery Manager は、リポジトリでバックアップ・セット 3 のステータスを EXPIRED に更新します。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 11-1 メディア・マネージャのクロスチェック



クロスチェックは、次の処理を行うため有効です。

- ディスクまたはテープから消失したバックアップや、破損したバックアップに関する以前の情報を更新できます。
- オペレーティング・システム・コマンドを使用してアーカイブ REDO ログまたはその他のファイルを削除した場合に、リポジトリを更新できます。

クロスチェック機能を使用すると、ディスクまたはテープ上のバックアップのステータスを確認できます。バックアップがディスク上に存在する場合、CROSSCHECKによって、そのファイルのヘッダーが有効かどうかを確認されます。バックアップがテープ上に存在する場合、このコマンドによって、バックアップがメディア管理ソフトウェアのカタログに存在するかどうかを確認されます。

バックアップ・ピースおよびイメージ・コピーのステータスは、AVAILABLE、EXPIRED または UNAVAILABLE となります。Recovery Manager の LIST コマンドを実行するか、V\$BACKUP\_FILES、または RC\_DATAFILE\_COPY や RC\_ARCHIVED\_LOG などのリカバリ・カタログ・ビューを問い合わせると、バックアップのステータスを表示できます。クロスチェックによって Recovery Manager リポジトリが更新されるため、これらのすべての方法で正確な情報が提供されます。Recovery Manager は、Recovery Manager 内のバックアップが使用できなくなると、そのバックアップのステータスを EXPIRED に更新します。新しいクロスチェックによって **期限切れのバックアップ** が再度使用可能であると判断された場合、Recovery Manager はそのステータスを AVAILABLE に更新します。

---

**注意：** CROSSCHECK コマンドでは、オペレーティング・システム・ファイルまたはリポジトリ・レコードは削除されません。これらを削除するには、DELETE コマンドを使用する必要があります。

---

DELETE EXPIRED コマンドを発行すると、期限切れのすべてのバックアップを削除できます。Recovery Manager は、期限切れのファイルのレコードをリポジトリから削除します。なんらかの理由でファイルがまだメディア上に存在している場合、Recovery Manager は警告を発行して、削除できない不一致のオブジェクトのリストを表示します。

#### 参照：

- CROSSCHECK の構文および使用される可能性があるステータス値については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- DELETE 構文の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## すべてのバックアップおよびコピーのクロスチェック

ターゲット・データベースとリカバリ・カタログ（使用している場合）に接続してから、必要に応じて CROSSCHECK コマンドを実行し、Recovery Manager が認識するバックアップの状態と可用性を検証します。

CROSSCHECK または DELETE コマンドを発行する前に、複数のチャンネルを構成するか、手動で割り当てます。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のタイプを持つすべてのチャンネルの各バックアップを検索します。マルチチャンネル機能は、1つのコマンドでディスクおよびテープの両方のクロスチェックまたはバックアップの削除を実行する場合に主に使用されます。たとえば、**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、ディスクおよび SBT の両方をクロスチェックできます。

```
CROSSCHECK BACKUP;
CROSSCHECK COPY;
```

Recovery Manager は、SBT チャンネルおよび事前構成済のディスク・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。出力例を次に示します。

```
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=12c5erb2_1_1 recid=32 stamp=408382820
.
.
.
```

自動 SBT チャンネルを構成していない場合は、次の例に示すように、ディスクおよびテープにメンテナンス・チャンネルを手動で割り当てることもできます。

```
RUN
{
  ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt;
  CROSSCHECK BACKUP;
  CROSSCHECK COPY;
}
```

Recovery Manager は事前構成済のディスク・チャンネルを使用するため、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。

## 特定のバックアップ・セットおよびコピーのクロスチェック

LIST コマンドを使用してバックアップをレポートしてから、CROSSCHECK コマンドを使用して LIST 出力に示された特定のバックアップが存在することを確認できます。DELETE EXPIRED コマンドは、クロスチェックの失敗の原因となるバックアップのリポジトリ・レコードを削除します。

### 指定したバックアップをクロスチェックする手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。



2. LIST コマンドを実行して、チェックするバックアップを識別します。

たとえば、次のコマンドを実行します。

```
LIST BACKUP; # lists all backup sets, proxy copies, and image copies
```

3. 目的のバックアップまたはコピーをクロスチェックします。

次のコマンド例は、様々なタイプのクロスチェックを示しています。

```
CROSSCHECK BACKUP; # checks backup sets, proxy copies, and image copies
CROSSCHECK COPY OF DATABASE;
CROSSCHECK BACKUPSET 1338, 1339, 1340;
CROSSCHECK BACKUPPIECE TAG 'nightly_backup';
CROSSCHECK BACKUP OF ARCHIVELOG ALL SPFILE;
CROSSCHECK BACKUP OF DATAFILE "?/oradata/trgt/system01.dbf"
  COMPLETED AFTER 'SYSDATE-14';
CROSSCHECK CONTROLFILECOPY '/tmp/control01.ctl';
CROSSCHECK DATAFILECOPY 113, 114, 115;
CROSSCHECK PROXY 789;
```

**参照：** 特定のファイルのバックアップを確認するための CROSSCHECK の使用方法の詳細は、『Oracle Database バックアップおよびリカバリ・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ポジトリ・ステータスの変更

この項では、バックアップおよびコピーのリポジトリ・レコードの変更方法について説明します。バックアップが一時的に使用可能または使用不可になった場合、バックアップ・ステータスを変更する必要があることがあります。たとえば、マウントされているディスクのメンテナンスを行う場合は、ディスク上のバックアップのレコードのステータスを UNAVAILABLE に更新できます。

### AVAILABLE または UNAVAILABLE へのバックアップのステータスの更新

バックアップが検出されない場合、またはサイト外に移された場合は、CHANGE ...

UNAVAILABLE コマンドを実行します。Recovery Manager は、ステータスが UNAVAILABLE のファイルは RESTORE または RECOVER コマンドでは使用しません。ファイルが後で検出された場合またはメイン・サイトに戻された場合は、CHANGE ... AVAILABLE を発行してステータスを再度更新できます。フラッシュ・リカバリ領域内のファイルを UNAVAILABLE にマークすることはできません。

### リポジトリ内のファイルのステータスを UNAVAILABLE または AVAILABLE に更新する手順

1. LIST コマンドを発行して、Recovery Manager バックアップの可用性の状態を確認します。たとえば、次のコマンドを発行してすべてのバックアップを表示します。

```
LIST BACKUP;
```

2. UNAVAILABLE または AVAILABLE キーワードを指定して CHANGE を実行し、Recovery Manager リポジトリの状態を更新します。

次の例は、CHANGE コマンドの形式を示しています。

```
CHANGE DATAFILECOPY '/tmp/control01.ctl' UNAVAILABLE;
CHANGE COPY OF ARCHIVELOG SEQUENCE BETWEEN 1000 AND 1012 UNAVAILABLE;
CHANGE BACKUPSET 12 UNAVAILABLE;
CHANGE BACKUP OF SPFILE TAG "TAG20020208T154556" UNAVAILABLE;
CHANGE DATAFILECOPY '/tmp/system01.dbf' AVAILABLE;
CHANGE BACKUPSET 12 AVAILABLE;
CHANGE BACKUP OF SPFILE TAG "TAG20020208T154556" AVAILABLE;
```

**参照：** CHANGE コマンドの構文の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## アーカイブ・バックアップのステータスの変更

8-24 ページの「[長期格納用のデータベース・バックアップの作成](#)」で説明されているように、保存方針から除外するバックアップを指定できます。この方法は、ビジネス要件に準拠するためにバックアップをアーカイブする場合に有効です。ただし、**アーカイブ・バックアップ**は完全に有効なバックアップであり、他の Recovery Manager バックアップと同様にリストアできます。

---

**注意：** 制御ファイルには Recovery Manager リポジトリの大規模なデータ・セットを無限に含めることはできないため、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。

---

CHANGE コマンドを使用すると、既存のバックアップの KEEP ステータスを変更できます。たとえば、長期バックアップを保持しないようにすることができます。BACKUP ... KEEP で使用できるオプションを CHANGE ... KEEP でも使用できます。

**フラッシュ・リカバリ領域**に格納されているバックアップ・セットまたはファイルには KEEP 属性を設定できないことに注意してください。

### アーカイブ・バックアップの KEEP ステータスを変更する手順

1. LIST コマンドを発行してバックアップを表示します。たとえば、次のコマンドを発行してすべてのバックアップを表示します。

```
LIST BACKUP;
```

2. CHANGE ... KEEP を発行してこのバックアップに別の保存期間を定義するか、または CHANGE ... NOKEEP を発行して保存方針をこのファイルに適用します。

次の例では、バックアップ・セットがバックアップの保存方針に従うことができます。

```
CHANGE BACKUPSET 231 NOKEEP;
```

次の例では、データファイルのコピーを保存方針から 180 日間除外するようにマークします。

```
CHANGE DATAFILECOPY '/tmp/system01.dbf' KEEP UNTIL 'SYSDATE+180';
```

## Recovery Manager リポジトリへのバックアップ・レコードの追加

CATALOG コマンドを使用すると、リポジトリに記録されていないアーカイブ・ログ、または Recovery Manager 以外の方法を使用して作成されたデータベース・ファイルのコピーの存在を Recovery Manager に認識させることができます。この項の内容は、次のとおりです。

- [カタログ操作](#)
- [カタログへのユーザー管理データファイルのコピーの追加](#)
- [カタログへのバックアップ・ピースの追加](#)
- [カタログへのディスクの場所内のすべてのファイルの追加](#)

### カタログ操作

ターゲット・データベースの制御ファイルには、ターゲット・データベースによって生成されたすべてのアーカイブ REDO ログおよびすべての Recovery Manager のバックアップのレコードが保持されます。CATALOG コマンドを実行すると、Recovery Manager に認識させたいファイルのレコードが存在しない場合に、メタデータをリポジトリに追加できます。



次の場合に、Recovery Manager の CATALOG コマンドを実行します。

- オペレーティング・システム・ユーティリティを使用して、データファイル、アーカイブ・ログまたはバックアップ・ピースのコピーを作成した場合。この場合、これらのレコードはリポジトリに含まれません。
- バックアップ制御ファイルを使用してリカバリを実行し、リカバリ中にアーカイブ先または形式を変更した場合。この場合、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含まれません。このため、これらのログをカタログに追加する必要があります。
- 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に追加した場合。これによって、増分バックアップ計画の基礎となるデータファイルのコピーを使用して後で増分バックアップを実行できるようになります。
- 新しいリリースに移行する前に作成した Oracle7 のデータベース・ファイルのユーザー管理コピー、または Recovery Manager を使用する前に作成した Oracle8 以上のデータベース・ファイルのユーザー管理コピーをカタログに追加する場合。移行後にデータベースがクラッシュし、移行したデータベースのバックアップを作成していない場合は、これらのデータファイルのコピーを使用してデータベースをリカバリできます。

UNIX の cp コマンドを使用したデータファイルのコピーなど、ユーザー管理コピーを作成するときには必ずカタログに追加してください。ユーザー管理コピーを作成する場合、ALTER TABLESPACE . . . BEGIN/END BACKUP 文を使用してオンライン表領域のデータファイルのコピーを作成できます。Recovery Manager はこのようなデータファイルのコピーを作成しませんが、CATALOG コマンドを使用してデータファイルのコピーをリカバリ・カタログに追加し、Recovery Manager に認識させることができます。

ユーザー管理コピーをカタログに追加するには、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可能であること
- 単一ファイルの完全なイメージ・コピーであること
- データファイル、制御ファイル、アーカイブ REDO またはバックアップ・ピースのいずれかのコピーであること

たとえば、ミラー化されたディスク・ドライブにデータファイルを格納する場合は、ミラー化を解除することによってユーザー管理コピーを作成できます。この場合、ミラー化を解除した後に CATALOG コマンドを使用してユーザー管理コピーが存在することを Recovery Manager に通知します。再度ミラー化する前に、CHANGE . . . UNCATALOG コマンドを実行してファイルのコピーが存在しないことを Recovery Manager に通知します。

## カタログへのユーザー管理データファイルのコピーの追加

CATALOG コマンドを使用して、ユーザー管理コピーに関する情報を Recovery Manager リポジトリに伝播します。ファイルがカタログに追加されたら、LIST を実行するか V\$BACKUP\_FILES を問い合せて確認します。

### データファイルのユーザー管理コピーを作成してカタログに追加する手順

1. オペレーティング・システム・ユーティリティを使用して、データファイルのコピーを作成します。バックアップ中にデータベースがオープンしており、データファイルがオンラインになっている場合は、ALTER TABLESPACE BEGIN/END BACKUP を実行する必要があります。次の例では、オペレーティング・システム・コマンドを発行するための SQL\*Plus コマンド HOST を使用して、オンラインのデータファイルをバックアップします。

```
SQL> ALTER TABLESPACE users BEGIN BACKUP;
SQL> host cp $ORACLE_HOME/oradata/trgt/users01.dbf /tmp/users01.dbf;
SQL> ALTER TABLESPACE users END BACKUP;
```

2. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。

### 3. CATALOG コマンドを実行します。

たとえば、ユーザー管理のデータファイル・コピーをカタログに追加するには、次のコマンドを実行します。

```
CATALOG DATAFILECOPY '/tmp/users01.dbf';
```

接続したターゲット・データベース以外のデータベースからデータファイル・コピーをカタログに追加しようとすると、Recovery Manager は次のようなエラーを発行します。

```
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03009: failure of catalog command on default channel at 08/29/2007 14:44:34
ORA-19563: datafile copy header validation failed for file /tmp/tools01.dbf
```

**参照：** CATALOG コマンドの構文の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## カタログへのバックアップ・ピースの追加

ディスク上のバックアップ・ピースをカタログに追加できます。この方法は、オペレーティング・システム・ユーティリティを使用して、同じホスト上のある場所から別の場所、またはあるホストから別のホストにバックアップ・ピースをコピーする場合に有効です。以前の**インカネーション**のデータベースからバックアップ・ピースのカタログを追加することもできます。Recovery Manager は、後続のリストアおよびリカバリ操作中にバックアップ・ピースを使用できるかどうかを決定できます。

### バックアップ・ピースをカタログに追加する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。

2. バックアップ・ピースのファイル名をカタログに追加します。

たとえば、次のコマンドを入力します。

```
CATALOG BACKUPPIECE '/disk2/09dtq55d_1_2', '/disk2/0bdtqdou_1_1';
```

3. 必要に応じて、LIST コマンドを実行するか、または V\$ ビューを問い合わせて変更を確認します。

ビューには、V\$BACKUP\_PIECE、V\$BACKUP\_SET、V\$BACKUP\_DATAFILE、V\$BACKUP\_REDOLOG および V\$BACKUP\_SPFILE があります。次の問合せは、バックアップ・ピース名を示しています。

```
SELECT HANDLE
FROM   V$BACKUP_PIECE;
```

**参照：** CATALOG BACKUPPIECE の制限の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## カタログへのディスクの場所内のすべてのファイルの追加

自動ストレージ管理 (ASM)、Oracle Managed Files フレームワークまたはフラッシュ・リカバリ領域を使用する場合は、ディスク管理システムによって認識されており、Recovery Manager のリポジトリには表示されていないファイルをカタログに再度追加する必要があることがあります。この状況は、メディア障害、ソフトウェアの不具合またはユーザー・エラーが原因でファイル名を追跡するメカニズムで障害が発生した場合に発生する可能性があります。

CATALOG START WITH コマンドを使用すると、ASM ディスク・グループ、Oracle Managed Files または従来のファイル・システム・ディレクトリにあるすべてのファイルを検索して、Recovery Manager リポジトリに記録されていないファイルを調べることができます。このコマンドによってファイルをカタログに追加できる場合は、ファイルがカタログに追加されます。追加できない場合は、スキップされたファイルの内容が可能なかぎり推測されます。

CATALOG RECOVERY AREA コマンドを使用すると、リカバリ領域内のすべてのファイルをカタログに追加できます。通常、このコマンドを手動で実行する必要はありません。このコマンドは、制御ファイルをリストアまたは作成する場合などに必要に応じて Recovery Manager が自動的に実行するためです。このコマンドは、オペレーティング・システム・ユーティリティを使用してファイルをフラッシュ・リカバリ領域にコピーする場合に実行できます。

### ディスクの場所内のすべてのファイルをカタログに追加する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. カタログに追加するファイルのディスクの場所を指定して、CATALOG コマンドを実行します。

たとえば、次のコマンドを入力します。

```
CATALOG START WITH '+disk'; # catalog all files from an ASM disk group
CATALOG START WITH '/fs1/datafiles/'; # catalog all files in directory
```

---

**注意：** START WITH 句では、ワイルド・カードは正しく使用できません。

---

CATALOG RECOVERY AREA コマンドを使用すると、リカバリ領域内のすべてのファイルをカタログに追加できます。この操作で、Recovery Manager リポジトリに表示されていないリカバリ領域内のすべてのファイルが追加されます。たとえば、次のように入力します。

```
CATALOG RECOVERY AREA;
```

3. LIST コマンドを実行し、ファイルがカタログに追加されていることを確認します。

## Recovery Manager リポジトリからのレコードの削除

この項では、ファイルのレコードを Recovery Manager のリポジトリから削除する方法について説明します。

### カタログからの削除操作

CHANGE ... UNCATALOG コマンドを実行すると、Recovery Manager リポジトリのレコードに次の処理を実行します。

- 制御ファイル・リポジトリのバックアップ・レコードを DELETED 状態に更新します。
- 特定のバックアップ・レコードをリカバリ・カタログ（使用している場合）から削除します。

Recovery Manager は、指定した物理ファイルは変更しません。これらのファイルのリポジトリ・レコードのみを変更します。

このコマンドは、Recovery Manager 以外の方法を使用してバックアップ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、CHANGE ARCHIVELOG ... UNCATALOG を発行することによってリポジトリからこのログのレコードを削除します。

### オペレーティング・システム・ユーティリティを使用して削除したファイルのレコードの削除

ユーザーがオペレーティング・システム・ユーティリティを使用してバックアップまたはアーカイブ REDO ログを削除している場合もあります。CROSSCHECK を実行しないかぎり、この削除は Recovery Manager では認識されません。CHANGE ... UNCATALOG コマンドを使用すると、存在しないファイルを反映して Recovery Manager リポジトリを更新できます。

**バックアップまたはアーカイブ REDO ログのカatalog・レコードを削除する手順**

- 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に対して、CHANGE ... UNCATALOG を実行します。次の例では、制御ファイルおよびデータファイル 1 のディスク・コピーへのリポジトリの参照を削除します。

```
CHANGE CONTROLFILECOPY '/tmp/control01.ct1' UNCATALOG;
CHANGE DATAFILECOPY '/tmp/system01.dbf' UNCATALOG;
```

- 任意で、関連するリカバリ・カatalog・ビュー (RC\_DATAFILE\_COPY、RC\_CONTROLFILE\_COPY など) を参照して、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、次の問合せによって、コピー 4833 のレコードが削除されたことを確認します。

```
SELECT CDF_KEY, STATUS
FROM RC_DATAFILE_COPY
WHERE CDF_KEY = 4833;
```

```
CDF_KEY    STATUS
-----
0 rows selected.
```

## Recovery Manager バックアップおよびアーカイブ REDO ログの削除

Recovery Manager の DELETE コマンドを使用すると、アーカイブ REDO ログおよび Recovery Manager バックアップを削除できます。ディスク上のバックアップの場合、バックアップを削除すると、バックアップ・ファイルがディスクから物理的に削除されます。SBT デバイス上のバックアップの場合、Recovery Manager の DELETE コマンドを使用すると、テープ上のバックアップ・ピースまたはプロキシ・コピーを削除するようにメディア・マネージャに指示が行われます。いずれの場合も、削除を反映して Recovery Manager のリポジトリが更新されます。

### Recovery Manager の削除の概要

Recovery Manager の各バックアップによって、対応するレコードが **Recovery Manager リポジトリ** に作成されます。このレコードは、制御ファイルに格納されます。リカバリ・カatalogを使用する場合、リカバリ・カatalogを制御ファイルと再同期化すると、レコードはリカバリ・カatalogにも格納されます。たとえば、完全なデータベース・バックアップ・セットを生成する場合は、このバックアップ・セットのレコードを V\$BACKUP\_SET に表示できます。リカバリ・カatalogを使用している場合は、RC\_BACKUP\_SET カatalog・ビューでレコードにアクセスすることもできます。

V\$ 制御ファイル・ビューとリカバリ・カatalog表では、情報を格納する方法が異なります。この違いが、Recovery Manager によるリポジトリ・レコードの処理方法に影響します。リカバリ・カatalogの Recovery Manager リポジトリは、実際のデータベース表に格納されますが、制御ファイルの Recovery Manager リポジトリは、制御ファイルの内部構造に格納されます。

Recovery Manager コマンドを使用してバックアップまたはアーカイブ REDO ログ・ファイルを削除する場合、Recovery Manager は次の処理を実行します。

- オペレーティング・システムから物理ファイルを削除します (ファイルが存在する場合)。
- 制御ファイル内のファイル・レコードのステータスを DELETED に更新します。
- リカバリ・カatalog表からファイル・レコードを削除します (Recovery Manager がリカバリ・カatalogに接続されている場合)。

制御ファイルのデータは内部構造に格納されているため、Recovery Manager は、制御ファイルからレコードを削除できず、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atalog表は通常のデータベース表であるため、表から行が削除される場合と同様に、カatalog表から行が削除されます。

## Recovery Manager の削除コマンド

表 11-2 に、バックアップを削除する Recovery Manager コマンドの機能を示します。

表 11-2 Recovery Manager の削除コマンド

| コマンド                            | 用途   |
|---------------------------------|--|
| DELETE                          | <p>バックアップを削除して、制御ファイル・レコードのステータスを DELETED に更新します。リカバリ・カタログが使用されている場合は、それらのレコードをリカバリ・カタログから削除します。</p> <p>DELETE によって EXPIRED または OBSOLETE のバックアップが削除されるように指定できます。存在しているバックアップに DELETE EXPIRED を実行すると、Recovery Manager は警告を発行し、バックアップの削除は行いません。オプションの FORCE キーワードを指定して DELETE コマンドを使用すると、Recovery Manager は指定されたバックアップを削除しますが、バックアップがディスクまたはテープから消失している場合に発生するエラーなど、すべての I/O エラーを無視します。その後、Recovery Manager は、ファイルが削除されたかどうかまたはファイルがすでに欠落していたかどうかに関係なく、バックアップが削除されたという事実を反映して Recovery Manager リポジトリを更新します。</p> <p>Recovery Manager は、構成されているすべてのチャンネルを使用して削除を実行します。自動チャンネル用に構成されていないデバイス上のファイルに対して DELETE を使用する場合は、ALLOCATE CHANNEL FOR MAINTENANCE を使用する必要があります。たとえば、SBT チャンネルを使用してバックアップを作成し、ディスク・チャンネルのみが構成されている場合は、DELETE に対して SBT チャンネルを手動で割り当てる必要があります。ディスクのみのファイルに対して DELETE を使用する場合は、自動的に割り当てられているメンテナンス・チャンネルまたは手動で割り当てられているメンテナンス・チャンネルが必要です。</p> |
| BACKUP ...DELETE<br>[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・セットをバックアップして、バックアップが正常に完了した後にオペレーティング・システムから入力ファイルを削除します。Recovery Manager は、削除された入力ファイルのリポジトリ・レコードも削除して更新します。</p> <p>ALL を付けずに DELETE INPUT を指定した場合、Recovery Manager は、バックアップする特定のファイルのみを削除します。ALL INPUT を指定した場合、Recovery Manager は、Recovery Manager リポジトリに記録されている、ファイルのすべてのコピーを削除します。</p>   |
| CHANGE ...<br>UNCATALOG         | <p>指定されたバックアップのリカバリ・カタログ・レコードを削除して、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 コマンドでは、バックアップの Recovery Manager リポジトリ・レコードのみが変更され、バックアップは実際には削除されないことに注意してください。</p>  |

オブジェクトの Recovery Manager リポジトリ・レコードが、そのオブジェクトの物理ステータスを反映していない場合があります。たとえば、ディスクにアーカイブ REDO ログをバックアップした後で、オペレーティング・システム・ユーティリティを使用してそのオブジェクトを削除したとします。最初に CROSSCHECK を実行せずに DELETE を実行すると、リポジトリにログが AVAILABLE と誤って表示されます。Recovery Manager リポジトリと物理メディアの間で不一致が発生した場合の DELETE の動作の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

Recovery Manager を対話形式で実行すると、ファイルを削除する前に確認を求められます。DELETE コマンドのいずれの形式でも、NOPROMPT キーワードを使用することによって、これらの確認をしないようにできます。

```
DELETE NOPROMPT ARCHIVELOG ALL;
```

### アーカイブ REDO ログの削除

11-2 ページの「バックアップおよびリポジトリのメンテナンスの基本的な概念」で説明されているように、メンテナンス計画では、フラッシュ・リカバリ領域、バックアップの保存方針およびアーカイブ REDO ログの削除方針を構成することをお勧めします。デフォルトでは、削除方針は NONE に構成されています。つまり、ログは、ディスクまたはテープに 1 回以上バックアップされた場合、またはバックアップ保存方針に従って不要になった場合に削除対象となります。

表 11-2 に示されているように、アーカイブ REDO ログは、データベースによって自動的に削除されるか、またはユーザーが実行した Recovery Manager コマンドによって削除されます。リカバリ領域内のログは、データベースによってできるかぎり保存され、ディスク領域が必要になると対象となるログが削除されます。対象となるログは、BACKUP ... DELETE INPUT または DELETE ARCHIVELOG を使用して、リカバリ領域の内部または外部のすべての場所から削除できます。FORCE が指定されていない場合、これらのコマンドはアーカイブ・ログの削除方針に従います。FORCE が指定されている場合、これらのコマンドはアーカイブ・ログの削除方針を無視します。

**参照：**

- 「アーカイブ REDO ログの削除方針の構成」 (5-27 ページ)
- 方針のオプションの詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』の CONFIGURE ARCHIVELOG DELETION POLICY エントリを参照してください。

## すべてのバックアップおよびコピーの削除

状況によっては、データベースに関連付けられているすべてのバックアップ・セット、プロキシ・コピーおよびイメージ・コピーを削除する必要がある場合があります。たとえば、データベースが不要になったため、関連するすべてのファイルをシステムから削除する場合などです。イメージ・コピーは、BACKUP AS COPY で生成されるファイル、データベースによってアーカイブされるログまたは CATALOG コマンドでカタログに追加されるファイルであることに注意してください。

### すべてのバックアップおよびコピーを削除する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. 必要に応じて、削除するバックアップが含まれているデバイス用のメンテナンス・チャンネルを割り当てます。

表 11-2 に示されているように、Recovery Manager は、構成されているすべてのチャンネルを使用して削除を実行します。チャンネルがすでに構成されている場合、メンテナンス・チャンネルを手動で割り当てる必要はありません。

3. バックアップおよびコピーをクロスチェックして、論理レコードが物理メディアと同期化されていることを確認します。

```
CROSSCHECK BACKUP;
CROSSCHECK COPY;
```

4. バックアップおよびコピーを削除します。

たとえば、次のコマンドを入力し、プロンプトが表示されたら YES と入力します。

```
DELETE BACKUP;
DELETE COPY;
```

ディスクおよびテープのチャンネルが構成されている場合、Recovery Manager は、構成されている SBT チャンネルおよび事前構成されているディスク・チャンネルの両方を削除時に使用します。Recovery Manager は、ファイルを削除する前に確認を求めるプロンプトを表示します。

## 指定したバックアップおよびコピーの削除

DELETE および BACKUP ... DELETE コマンドのいずれを使用しても、特定のバックアップおよびコピーを削除できます。BACKUP ... DELETE コマンドを実行すると、ファイルはまず通常はテープにバックアップされ、その後入力ファイルが削除されます。

### 指定したバックアップおよびコピーの削除

DELETE コマンドでは、削除するオブジェクトを識別するための様々なオプションがサポートされています。これらのオプションの詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。Recovery Manager は、アーカイブ REDO ログを削除する場合、構成済の設定を使用して、ログを削除できるかどうかを判断します (5-27 ページの「[アーカイブ REDO ログの削除方針の構成](#)」を参照)。

### 指定したバックアップおよびコピーを削除する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ (使用している場合) に接続します。
2. 必要に応じて、削除するバックアップが含まれているデバイス用のメンテナンス・チャンネルを割り当てます。

表 11-2 に示されているように、Recovery Manager は、構成されているすべてのチャンネルを使用して削除を実行します。チャンネルがすでに構成されている場合、メンテナンス・チャンネルを手動で割り当てる必要はありません。

3. 指定したバックアップおよびコピーを削除します。

次の例は、DELETE コマンドを使用して削除するバックアップおよびアーカイブ・ログを指定する一般的な方法を示しています。

- LIST 出力の主キーを使用してバックアップを削除します。

```
DELETE BACKUPPIECE 101;
```

- ディスク上のファイル名を指定してバックアップを削除します。

```
DELETE CONTROLFILECOPY '/tmp/control01.ct1';
```

- アーカイブ REDO ログを削除します。

```
DELETE NOPROMPT ARCHIVELOG UNTIL SEQUENCE 300;
```

- タグに基づいてバックアップを削除します。

```
DELETE BACKUP TAG 'before_upgrade';
```

- バックアップされているオブジェクトおよびバックアップが格納されているメディアまたはディスクの場所に基づいてバックアップを削除します。

```
DELETE BACKUP OF TABLESPACE users DEVICE TYPE sbt; # delete only from tape
DELETE COPY OF CONTROLFILE LIKE '/tmp/%';
```

- テープにバックアップされているかどうかに基づいて、ディスクからバックアップおよびアーカイブ REDO ログを削除します。

```
DELETE ARCHIVELOG ALL
BACKED UP 3 TIMES TO sbt;
```

### BACKUP ... DELETE を使用した指定ファイルの削除

BACKUP ... DELETE を使用すると、アーカイブ REDO ログ、データファイルのコピー、またはバックアップ・セットをバックアップし、バックアップに成功した後で入力ファイルを削除できます。DELETE INPUT オプションを指定すると、入力ファイルに対して DELETE コマンドを発行した場合と同じ処理が実行されます。5-27 ページの「[アーカイブ REDO ログの削除方針の構成](#)」で説明されているように、Recovery Manager は、アーカイブ REDO ログを削除する場合、構成済の設定を使用して、ログを削除できるかどうかを判断します。DELETE ALL



INPUT 句の ALL オプションは、アーカイブ REDO ログにのみ適用されます。BACKUP ... DELETE ALL INPUT を実行すると、BACKUP コマンドの選択基準を満たすアーカイブ REDO ログまたはデータファイルのすべてのコピーが削除されます。

## 期限切れの Recovery Manager バックアップおよびコピーの削除

CROSSCHECK を実行し、Recovery Manager がファイルの場所を特定できない場合、Recovery Manager リポジトリのレコードは 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、Recovery Manager リポジトリから期限切れのバックアップおよびコピーのレコードを削除できます。

EXPIRED とマークされたファイルが実際に存在する場合は、DELETE EXPIRED コマンドによって警告が発行されます。まれに、ファイルが存在する場合でも、リポジトリによってファイルが EXPIRED とマークされる場合があります。たとえば、ファイルが含まれているディレクトリがクロスチェック中に破損し、その後、修復された場合や、メディア・マネージャが適切に構成されておらず、実際には存在するバックアップが存在しないとレポートされる場合などです。

### 期限切れのリポジトリ・レコードを削除する手順

1. 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、次のように入力します。

```
CROSSCHECK BACKUP;
```

2. 期限切れのバックアップを削除します。たとえば、次のように入力します。

```
DELETE EXPIRED BACKUP;
```

## 保存方針に基づく不要な Recovery Manager バックアップの削除

Recovery Manager の DELETE コマンドでは、OBSOLETE オプションがサポートされています。このオプションによって、指定されているリカバリ可能性の要件を満たす必要がなくなったバックアップが削除されます。構成済みのデフォルトの保存方針、または DELETE OBSOLETE コマンドにオプションとして指定した他の保存方針に従って、不要とされるファイルを削除できます。DELETE コマンドの他の形式と同様に、削除されたファイルはバックアップ・メディアおよびリカバリ・カタログから削除され、制御ファイル内で DELETED としてマーク付けされます。

引数を使用せずに DELETE OBSOLETE コマンドを指定すると、Recovery Manager は、構成されている保存方針で定義されているすべての不要なバックアップを削除します。たとえば、次のように入力します。

```
DELETE OBSOLETE;
```

### KEEP UNTIL TIME に指定した期限が切れた場合の DELETE OBSOLETE の動作

**アーカイブ・バックアップ**に対して KEEP UNTIL TIME に指定した期限が切れていないかぎり、バックアップは不要とはみなされません。ただし、KEEP UNTIL に指定した期限が切れるとすぐに、構成されている**バックアップの保存方針**に関係なく、バックアップはすぐに OBSOLETE とみなされます。このため、KEEP に指定した期限が切れた場合、DELETE OBSOLETE を使用すると、BACKUP ... KEEP UNTIL TIME で作成されたすべてのバックアップが削除されます。

**参照：** *keepOption* 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。



## データベースの削除

データベースをオペレーティング・システムから削除する必要がある場合があります。このような場合は、**Recovery Manager** で **DROP DATABASE** コマンドを使用できます。**Recovery Manager** は、ターゲット・データベースに属しているすべてのデータファイル、オンライン REDO ログおよび制御ファイルを削除します。

**DROP DATABASE** では、マウントされたターゲット・データベースに **Recovery Manager** が接続されている必要があります。リカバリ・カタログに接続している必要はありません。**Recovery Manager** をリカバリ・カタログに接続した状態でオプション **INCLUDE COPIES AND BACKUPS** を指定すると、**Recovery Manager** はデータベースも登録解除します。

### データベースを削除する手順

1. **Recovery Manager** を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。

2. データベースに関連付けられているすべてのバックアップをカタログに追加します。たとえば、次のコマンドを実行すると、フラッシュ・リカバリ領域内のファイルがカタログに追加され、次にセカンダリ・アーカイブ先内のファイルがカタログに追加されます。

```
CATALOG START WITH '+disk1'; # all files from recovery area on ASM disk
CATALOG START WITH '/arch_dest2'; # all files from second archiving location
```

3. データベースに関連付けられているすべてのバックアップおよびコピーを削除します。たとえば、次のように入力します。

```
DELETE BACKUPSET; # deletes all backups
DELETE COPY; # delete all image copies (including archived logs)
```

4. データベースをオペレーティング・システムから削除します。

次のコマンドを実行すると、データベースが削除され、リカバリ・カタログ（使用している場合）から自動的に登録解除されます。**Recovery Manager** によって構成を求められません。

```
DROP DATABASE;
```

### 参照：

- SQL コマンド **DROP DATABASE** の使用方法については、29-17 ページの「[SQL\\*Plus でのデータベースの削除](#)」を参照してください。
- **Recovery Manager** の **DROP DATABASE** 構文については、『**Oracle Database バックアップおよびリカバリ・リファレンス**』を参照してください。



---

## リカバリ・カタログの管理

この章では、Recovery Manager リカバリ・カタログを管理する方法について説明します。このカタログは、1 つ以上のターゲット・データベースの Recovery Manager リポジトリ・データが含まれているデータベース・スキーマです。この章の内容は、次のとおりです。

- リカバリ・カタログの概要
- リカバリ・カタログの作成
- リカバリ・カタログへのデータベースの登録
- リカバリ・カタログへのバックアップの追加
- 仮想プライベート・カタログの作成および管理
- リカバリ・カタログの保護
- スタアド・スクリプトの管理
- リカバリ・カタログのメンテナンス
- リカバリ・カタログの削除

### 参照：

- リカバリ・カタログを使用せずに、制御ファイルに格納されている Recovery Manager リポジトリを管理する方法については、[第 11 章「Recovery Manager バックアップおよびリポジトリ・レコードのメンテナンス」](#)を参照してください。
- サポートされている相互運用の例は、『Oracle Database バックアップおよびリカバリ・リファレンス』の互換性の表を参照してください。

## リカバリ・カタログの概要

この項では、リカバリ・カタログの管理に関連する基本的な概念について説明します。

### リカバリ・カタログの目的

**リカバリ・カタログ**とは、1つ以上の Oracle Database に関するメタデータを格納するために Recovery Manager で使用されるデータベース・スキーマのことです。通常、このカタログは専用のデータベースに格納します。リカバリ・カタログには、次のメリットがあります。

- リカバリ・カタログによって、各**ターゲット・データベース**の制御ファイルに格納されている Recovery Manager リポジトリに対して冗長性が作成されます。リカバリ・カタログは、セカンダリ・メタデータ・リポジトリとして機能します。ターゲット制御ファイルおよびすべてのバックアップが消失した場合でも、Recovery Manager メタデータはリカバリ・カタログ内に存在します。
- リカバリ・カタログによって、すべてのターゲット・データベースのメタデータが集中化されます。メタデータを1つの場所に格納することによって、レポート・タスクおよび管理タスクの実行が簡単になります。
- リカバリ・カタログには、制御ファイルより長期のメタデータ履歴を格納できます。この機能は、制御ファイルの履歴より前の時点にリカバリする必要がある場合に有効です。リカバリ・カタログ・データベースを使用することによって管理はより複雑になりますが、より長期のバックアップ履歴を使用できるというメリットがあります。

一部の Recovery Manager 機能は、リカバリ・カタログを使用する場合にのみ機能します。たとえば、リカバリ・カタログには Recovery Manager スクリプトを格納することができます。**ストアド・スクリプト**の主なメリットは、ターゲット・データベースおよびリカバリ・カタログに接続可能な Recovery Manager クライアントで使用できることです。コマンド・ファイルは、それらが格納されているファイル・システムに Recovery Manager クライアントがアクセスできる場合にのみ使用できます。

Data Guard 環境で Recovery Manager を使用する場合は、リカバリ・カタログが必要です。すべてのプライマリ・データベースおよびスタンバイ・データベースのバックアップ・メタデータを格納すると、カタログによって、バックアップ・タスクを1つのスタンバイ・データベースにオフロードし、その環境内のその他のデータベースにバックアップをリストアできます。

### リカバリ・カタログの基本的な概念

リカバリ・カタログには、登録されている各ターゲット・データベースの Recovery Manager 操作に関するメタデータが含まれています。Recovery Manager は、リカバリ・カタログに接続されると、カタログから排他的にそのメタデータを取得します。このカタログには、次のタイプのメタデータが含まれています。

- データファイルとアーカイブ REDO ログのバックアップ・セットおよびバックアップ・ピース
- データファイルのコピー
- アーカイブ REDO ログおよびそのコピー
- データベース構造（表領域およびデータファイル）
- ストアド・スクリプト（ユーザーが作成した一連の名前付き Recovery Manager コマンド）
- Recovery Manager 構成の永続設定

## データベースの登録

Recovery Manager で使用するために、リカバリ・カタログにデータベースを記載することを登録と呼びます。すべてのターゲット・データベースを環境内の単一のリカバリ・カタログに登録することをお勧めします。たとえば、データベース prod1、prod2 および prod3 を、データベース catdb の catowner が所有する 1 つのカタログに登録できます。

参照: 「リカバリ・カタログへのデータベースの登録」(12-7 ページ)

## ベース・リカバリ・カタログでのメタデータの集中化

集中化されたリカバリ・カタログ (**ベース・リカバリ・カタログ**とも呼ばれる) の所有者は、他のデータベース・ユーザーに対してカタログへの制限付きアクセス権限を付与および取り消すことができます。制限付きユーザーは、それぞれ**仮想プライベート・カタログ**と呼ばれる独自のメタデータへの完全な読取り / 書込み権限を持っています。Recovery Manager メタデータは、仮想プライベート・カタログ所有者のスキーマに格納されます。ベース・リカバリ・カタログの所有者は、各仮想プライベート・カタログ・ユーザーがアクセスできるオブジェクトを決定します。

リカバリ・カタログは、様々なバージョンの Oracle Database を使用している環境、または使用してきた環境で使用できます。その結果、様々なバージョンの Recovery Manager クライアント、リカバリ・カタログ・データベース、リカバリ・カタログ・スキーマおよびターゲット・データベースが環境内に存在する可能性があります。複数のリカバリ・カタログ・スキーマを 1 つにマージする方法については、12-32 ページの「リカバリ・カタログのインポートおよび移動」を参照してください。

参照: 「仮想プライベート・カタログの作成および管理」(12-10 ページ)

## リカバリ・カタログの再同期化

バックアップ、リストア、クロスチェックなどの Recovery Manager 操作では、Recovery Manager によって常に最初に制御ファイルが更新され、次にメタデータがリカバリ・カタログに伝播されます。マウントされた制御ファイルからリカバリ・カタログへのメタデータのこのフローは、リカバリ・カタログの**再同期化**と呼ばれます。これによって、Recovery Manager が制御ファイルから取得するメタデータが現行のものになります。

参照: 「リカバリ・カタログの再同期化」(12-22 ページ)

## ストアド・スクリプト

**ストアド・スクリプト**は、制御ファイルの代替として、頻繁に使用する一連の Recovery Manager コマンドの管理に使用できます。このスクリプトは、ファイル・システムではなくリカバリ・カタログに格納されます。

ローカル・ストアド・スクリプトは、作成時に Recovery Manager が接続していたターゲット・データベースに関連付けられます。スクリプトは、関連付けられたターゲット・データベースに接続している場合のみ実行できます。グローバル・ストアド・スクリプトは、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に対して実行できます。仮想プライベート・カタログ・ユーザーは、グローバル・スクリプトへの読取り専用アクセス権限を持っています。グローバル・スクリプトの作成または更新は、ベース・リカバリ・カタログへの接続中に実行する必要があります。

ヒント: 「ストアド・スクリプトの管理」(12-16 ページ)

## Data Guard 環境でのリカバリ・カタログ

3-8 ページの「Data Guard 環境での Recovery Manager」で説明されているように、Data Guard 環境内のすべての物理データベース (プライマリ・データベースとスタンバイ・データベースの両方) の Recovery Manager メタデータを管理するには、リカバリ・カタログを使用する必要があります。Recovery Manager は、Data Guard 環境の実情の単一のソースとしてリカバリ・カタログを使用します。

Recovery Manager は、**逆再同期化**で、リカバリ・カタログを使用してプライマリまたはスタンバイ制御ファイルを更新できます。この場合、メタデータのフローの方向は、制御ファイルからカタログではなく、カタログから制御ファイルになります。Recovery Manager は、再同期化が必要なほぼすべての場合に、再同期化を自動的に実行します。そのため、RESYNC コマンドを使用して手動で再同期化する必要がある機会はあまり多くありません。

**参照：** スタンバイ・データベースで使用するために Recovery Manager 環境を構成する方法については、『Oracle Data Guard 概要および管理』を参照してください。

## リカバリ・カタログの管理の基本手順

リカバリ・カタログを Recovery Manager で使用できるように設定する基本手順は次のとおりです。

1. リカバリ・カタログを作成します。  
このタスクの実行方法については、12-5 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. リカバリ・カタログにターゲット・データベースを登録します。  
この手順によって、Recovery Manager はターゲット・データベースに関するメタデータをリカバリ・カタログに格納できます。このタスクについては、12-7 ページの「[リカバリ・カタログへのデータベースの登録](#)」を参照してください。
3. 必要に応じて、ターゲット制御ファイルに格納されていないレコードが含まれている古いバックアップをカタログに追加します。  
このタスクの実行方法については、12-9 ページの「[リカバリ・カタログへのバックアップの追加](#)」を参照してください。
4. 必要に応じて、特定のユーザー用の仮想プライベート・カタログを作成し、そのユーザーがアクセスできるメタデータを指定します。  
このタスクの実行方法については、12-10 ページの「[仮想プライベート・カタログの作成および管理](#)」を参照してください。
5. バックアップおよびリカバリ計画に含めることによって、リカバリ・カタログを保護します。  
カタログをバックアップおよびリカバリする方法および可用性を向上させる方法については、12-22 ページの「[リカバリ・カタログの保護](#)」を参照してください。

この章の残りでは、リカバリ・カタログの使用開始後、そのリカバリ・カタログを管理する方法について説明します。次のタスクを実行できます。

- Recovery Manager スクリプトをリカバリ・カタログに格納して管理する方法については、12-16 ページの「[ストアド・スクリプトの管理](#)」を参照してください。
- 第 10 章「[Recovery Manager 操作に関するレポート](#)」では、Recovery Manager 操作に関してレポートする方法について説明します。LIST および REPORT コマンドは、リカバリ・カタログを使用しているかどうかに関係なく使用できます。リカバリ・カタログ内の固定ビューによって Recovery Manager 操作に関してレポートする方法については、10-18 ページの「[リカバリ・カタログ・ビューの問合せ](#)」を参照してください。
- 実行中のリカバリ・カタログのメンテナンスでの様々なタスク（1 つのリカバリ・カタログを別のリカバリ・カタログにインポートするタスクなど）については、12-22 ページの「[リカバリ・カタログのメンテナンス](#)」を参照してください。

リカバリ・カタログをメンテナンスしない場合は、12-34 ページの「[リカバリ・カタログの削除](#)」を参照してください。

## リカバリ・カタログの作成

この項では、リカバリ・カタログ作成時の各フェーズについて説明します。この項の内容は、次のとおりです。

- リカバリ・カタログ・データベースの構成
- リカバリ・カタログのスキーマ所有者の作成
- CREATE CATALOG コマンドの実行

## 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・スキーマを保持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。SYS はリカバリ・カタログ所有者に設定できないことに注意してください。

リカバリ・カタログ・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、そのデータベースのバックアップ方法も決定します。また、カタログ・データベースを ARCHIVELOG モードで操作するかどうかを決定します。カタログ・データベースは ARCHIVELOG モードで操作することをお勧めします。

---

**注意：** ターゲット・データベースを使用して、リカバリ・カタログのデータベースとしてバックアップしないでください。リカバリ・カタログは、ターゲット・データベースの消失から保護される必要があります。

---

## リカバリ・カタログ・スキーマのサイズの計画

カタログ・スキーマによって使用される領域を割り当てる必要があります。リカバリ・カタログ・スキーマのサイズは、カタログによって監視されるデータベースの数によって異なります。スキーマは、アーカイブ REDO ログ・ファイルおよび各データベースのバックアップの増加とともに増大します。カタログに格納されている Recovery Manager ストアド・スクリプトを使用する場合は、これらのスクリプトに対して領域を割り当てる必要があります。

たとえば、trgt データベースにファイルが 100 個あり、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 個のバックアップ・セットを作成すると想定します。バックアップ・ピース表内の各行が領域を最大限に使用すると想定すると、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 未満です。この日次バックアップを 1 年間行くと、この期間中に使用される合計記憶域は約 62MB です。これとほぼ同じ量の記憶域がアーカイブ・ログに使用されると想定されます。したがって、最悪の場合はメタデータの格納に 1 年間で約 120MB が必要になります。バックアップ・ピースの行領域の一部のみを使用する通常の場合、現実的な値は年間 15MB です。

リカバリ・カタログに複数のデータベースの登録を予定している場合、前述の計算に従って各データベースに必要な領域を足し、リカバリ・カタログ・スキーマのデフォルト表領域の合計サイズを算出してください。

## リカバリ・カタログ・データベースのディスク領域の割当て

リカバリ・カタログを既存のデータベース内に作成する場合、デフォルト表領域を保持するために十分な空き領域をリカバリ・カタログ・スキーマに追加します。リカバリ・カタログを格納するための新しいデータベースを作成する場合、リカバリ・カタログ・スキーマ自体の領域に加えて、次に示すリカバリ・カタログ・データベース内の他のファイル用の領域も考慮します。

- SYSTEM および SYSAUX 表領域
- 一時表領域
- UNDO 表領域
- オンライン REDO ログ・ファイル

リカバリ・カタログ・データベース内の領域の大部分は、SYSTEM 表領域、一時表領域、UNDO 表領域などの表領域をサポートするために使用されます。表 12-1 に、一般的な領域要件を示します。

表 12-1 1 年間での一般的なリカバリ・カタログ領域要件

| 領域タイプ                 | 領域要件                             |
|-----------------------|----------------------------------|
| SYSTEM 表領域            | 90MB                             |
| 一時表領域                 | 5MB                              |
| ロールバック表領域または UNDO 表領域 | 5MB                              |
| リカバリ・カタログ表領域          | リカバリ・カタログに登録されたデータベースごとに 15MB    |
| オンライン REDO ログ         | ログごとに 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存在していないことを確認してください。リカバリ・カタログとターゲット・データベースの両方にハード・ディスク障害が発生した場合、リカバリ・プロセスがより困難になります。可能な場合、他の手段を利用して、リカバリ・カタログ・データベースとバックアップ対象のデータベースの共通の致命的な障害箇所を排除してください。

## リカバリ・カタログのスキーマ所有者の作成

リカバリ・カタログ・データベースを選択して必要な領域を作成した後、リカバリ・カタログの所有者を作成し、そのユーザーに必要な権限を付与します。後続の項に示す手順では、次の状況を想定しています。

- ユーザー SYS は、catdb リカバリ・カタログ・データベースに対する SYSDBA 権限を持っています。
- catdb リカバリ・カタログ・データベース内の tools という表領域にリカバリ・カタログが格納されています。Recovery Manager の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、大文字を使用する必要があります。(Recovery Manager の予約語のリストは、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。)
- temp という表領域が、リカバリ・カタログ・データベース内に存在します。

### リカバリ・カタログ・データベース内にリカバリ・カタログ・スキーマを作成する手順

1. SQL\*Plus を起動し、リカバリ・カタログを含むデータベースに管理者権限で接続します。この例では、データベースは catdb です。
2. リカバリ・カタログのユーザーおよびスキーマを作成します。たとえば、次の SQL 文を入力できます (password はユーザー定義のパスワードに置き換えます)。

```
CREATE USER rman IDENTIFIED BY password
TEMPORARY TABLESPACE temp
DEFAULT TABLESPACE tools
QUOTA UNLIMITED ON tools;
```

**注意：** 安全なパスワードを作成してください。詳細は、『Oracle Database セキュリティ・ガイド』を参照してください。



- スキーマ所有者に RECOVERY\_CATALOG\_OWNER ロールを付与します。このロールによって、リカバリ・カタログのメンテナンスおよび問合せに必要なすべての権限がユーザーに付与されます。

```
GRANT RECOVERY_CATALOG_OWNER TO rman;
```

## CREATE CATALOG コマンドの実行

カタログ所有者の作成後、Recovery Manager の CREATE CATALOG コマンドを使用してカタログ表を作成します。このコマンドを実行すると、カタログ所有者のデフォルト表領域内にカタログが作成されます。

### リカバリ・カタログを作成する手順

- Recovery Manager を起動し、カタログを格納するデータベースに接続します。データベースには、リカバリ・カタログ所有者として接続します。
- CREATE CATALOG コマンドを実行してカタログを作成します。カタログの作成には数分かかります。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、次のコマンドを実行できます。

```
RMAN> CREATE CATALOG;
```

CREATE CATALOG コマンドでカタログの表領域名を指定できます。たとえば、次のように入力します。

```
RMAN> CREATE CATALOG TABLESPACE cat_tbs;
```

---

**注意：** リカバリ・カタログに使用する表領域名が Recovery Manager の予約語である場合は、その予約語を大文字にして引用符で囲む必要があります。たとえば、次のように入力します。

```
CREATE CATALOG TABLESPACE 'CATALOG';
```

---

- 結果を確認するには、SQL\*Plus を使用し、リカバリ・カタログを問い合わせで作成された表を表示します。

```
SQL> SELECT TABLE_NAME FROM USER_TABLES;
```

**参照：** GRANT 文および CREATE USER 文の SQL 構文については『Oracle Database SQL 言語リファレンス』、CREATE CATALOG のコマンド構文については『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## リカバリ・カタログへのデータベースの登録

この項では、リカバリ・カタログでターゲット・データベース・レコードをメンテナンスする方法について説明します。この章の内容は、次のとおりです。

- リカバリ・カタログへのデータベースの登録
- REGISTER DATABASE コマンドを使用したデータベースの登録

## 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**ターゲット・データベース**を記載することを**登録**と呼びます。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ない場合、Recovery Manager は、カタログを使用して操作に関するメタデータをそのデータベースに格納できません。登録されていないデータベースでも、Recovery Manager 操作を実行することはできます。Recovery Manager は、そのメタデータを常にターゲット・データベースの制御ファイルに格納するためです。

Data Guard 環境でリカバリ・カタログを使用していない場合は、REGISTER コマンドを使用して各データベースを登録します。各データベースには、一意の **DBID** が必要です。DUPLICATE コマンドまたは SQL の CREATE DATABASE 文を使用すると、データベースに一意の DBID が自動的に割り当てられます。その他の方法でデータベースを作成すると、コピーされたデータベースの DBID とソース・データベースの DBID が同じになる場合があります。ソース・データベースとコピー・データベースを同じカタログに登録できるように、DBNEWID ユーティリティを使用して DBID を変更できます。

UNREGISTER コマンドを使用すると、リカバリ・カタログからデータベースの登録を解除できます。

## スタンバイ・データベースの登録

Data Guard 環境では、プライマリ・データベースとスタンバイ・データベースは同じ DBID およびデータベース名を共有します。Data Guard 環境のデータベースをリカバリ・カタログに登録できるようにするには、各データベースに異なる DB\_UNIQUE\_NAME 値が必要です。データベースの DB\_UNIQUE\_NAME パラメータは、その初期化パラメータ・ファイルに設定されています。

Data Guard 環境で Recovery Manager を使用する場合は、プライマリ・データベースに対してのみ REGISTER DATABASE コマンドを使用できます。次の方法を使用すると、スタンバイ・データベースをリカバリ・カタログに登録できます。

- TARGET としてスタンバイ・データベースに接続すると、Recovery Manager によってデータベースがリカバリ・カタログに自動的に登録されます。
- リカバリ・カタログで認識されないスタンバイ・データベースに対して CONFIGURE DB\_UNIQUE\_NAME コマンドを実行する場合、Recovery Manager は、対応するプライマリ・データベースが登録されているかぎり、このスタンバイ・データベースを自動的に登録します。

### 参照：

- 「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の解除」(12-26 ページ)
- DUPLICATE 構文については、『Oracle Database バックアップおよびリカバリ・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 を変更する方法については、『Oracle Database ユーティリティ』を参照してください。
- Data Guard 環境での Recovery Manager の使用方法については、『Oracle Data Guard 概要および管理』を参照してください。

## REGISTER DATABASE コマンドを使用したデータベースの登録

ターゲット・データベースでリカバリ・カタログを使用するには、まずそのターゲット・データベースをリカバリ・カタログに登録します。Data Guard 環境でカタログを使用する場合は、この方法でプライマリ・データベースのみを登録できます。

次の手順を実行します。

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログに接続します。リカバリ・カタログ・データベースはオープンされている必要があります。

たとえば、ネット・サービス名 catdb を使用してカタログ・データベースにユーザー rman (カタログ・スキーマ所有者) として接続するには、次のコマンドを実行します。

```
% rman TARGET / CATALOG rman@catdb
```

2. ターゲット・データベースをマウントしていない場合、マウントまたはオープンします。

```
STARTUP MOUNT;
```

3. 接続しているリカバリ・カタログに、ターゲット・データベースを登録します。

```
REGISTER DATABASE;
```

Recovery Manager は、カタログ表内に、ターゲット・データベースに関する情報を格納する行を作成し、ターゲット・データベースに関連するすべてのデータを制御ファイルからカタログにコピーし、カタログと制御ファイルを同期化します。

4. REPORT SCHEMA を実行し、正常に登録されたことを確認します。

```
REPORT SCHEMA;
```

```
Report of database schema
File Size(MB)  Tablespace      RB segs  Datafile Name
-----
1             307200  SYSTEM          NO       /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
```

## リカバリ・カタログへのバックアップの追加

データファイルのコピー、バックアップ・ピースまたはアーカイブ・ログがディスクに存在する場合、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。リカバリ・カタログを使用する場合、制御ファイルからエージ・アウトされたバックアップをカタログに追加すると、リストア操作中に、その古いバックアップが使用されます。次のコマンドは、この方法を示しています。

```
CATALOG DATAFILECOPY '/disk1/old_datafiles/01_01_2003/users01.dbf';
CATALOG ARCHIVELOG '/disk1/arch_logs/archive1_731.dbf',
                  '/disk1/arch_logs/archive1_732.dbf';
CATALOG BACKUPPIECE '/disk1/backups/backup_820.bkp';
```

また、CATALOG START WITH コマンドを使用して、1つのディレクトリ内の複数のバックアップ・ファイルを一度にカタログに追加できます。次に例を示します。

```
CATALOG START WITH '/disk1/backups/';
```

Recovery Manager によって、Recovery Manager リポジトリに追加されるファイルが表示され、それらのバックアップが追加される前に確認するように求められます。CATALOG START WITH で接頭辞を指定する場合は注意が必要です。Recovery Manager は、ディスク上で、指定された接頭辞が付いたすべてのパスのすべてのファイルをスキャンします。接頭辞は、通常のディレクトリ名ではありません。不適切な接頭辞を使用すると、不適切なファイル・セットがカタログに追加される場合があります。

たとえば、/disk1/backups、/disk1/backups-year2003、/disk1/backupsets、/disk1/backupsets/test などの一連のディレクトリにバックアップ・ファイルが含まれているとします。次のコマンドを実行すると、これらのすべてのディレクトリ内のすべてのファイルがカタログに追加されます。/disk1/backups が、これらのすべてのディレクトリへのパスの接頭辞であるためです。

```
CATALOG START WITH '/disk1/backups';
```

/disk1/backups ディレクトリ内のバックアップのみをカタログに追加するには、次のコマンドが適切です。

```
CATALOG START WITH '/disk1/backups/';
```

**参照：**

- REGISTER 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- データベースの移行に関する問題については、『Oracle Database アップグレード・ガイド』を参照してください。

## 仮想プライベート・カタログの作成および管理

すべてのデータベースの集中 Recovery Manager リポジトリとして機能する 1 つのリカバリ・カタログを作成することをお勧めします。このリカバリ・カタログは、全体として**ベース・リカバリ・カタログ**とも呼ばれます。このベース・カタログに、0（ゼロ）個以上の仮想プライベート・カタログを含めることができます。**仮想プライベート・カタログ**は、ベース・リカバリ・カタログを参照するシノニムおよびビューのセットです。

### 仮想プライベート・カタログ

デフォルトでは、ベース・リカバリ・カタログの所有者のみがそのメタデータへのアクセス権限を持っています。ベース・リカバリ・カタログの所有者は、Recovery Manager の GRANT コマンドを使用して、リカバリ・カタログへの制限付きアクセス権限を他のデータベース・ユーザーに付与できます。ベース・リカバリ・カタログの所有者は、リカバリ・カタログを共有できるデータベース・ユーザーおよびアクセスできるデータベースを決定します。

カタログ・ユーザーに制限付きアクセス権限を付与する場合は、そのユーザーにそのユーザー独自の Recovery Manager メタデータ（仮想プライベート・カタログ）への完全な読み取り / 書き込み権限を付与します。仮想プライベート・カタログ所有者は、ローカル・ストアド・スクリプトは作成できますが、グローバル・**ストアド・スクリプト**に対して持っているアクセス権限は読み取りアクセス権限のみであることに注意してください。仮想プライベート・カタログを構成するビューおよびシノニムのセットは、仮想プライベート・カタログ所有者のスキーマに格納されます。仮想プライベート・カタログのメカニズムは、リカバリ・カタログ・スキーマ自体に存在します。セキュリティは、Recovery Manager 実行可能ファイルではなく、リカバリ・カタログ・データベースによって提供されます。

仮想プライベート・カタログを作成する基本的な手順は次のとおりです。

1. 仮想プライベート・カタログを所有するデータベース・ユーザーを作成し（このユーザーが存在しない場合）、このユーザーにアクセス権限を付与します。

このタスクについては、12-11 ページの「[仮想プライベート・カタログ所有者の作成および権限の付与](#)」を参照してください。

2. 仮想プライベート・カタログを作成します。

このタスクについては、12-12 ページの「[仮想プライベート・カタログの作成](#)」を参照してください。

仮想プライベート・カタログの作成後に、必要に応じてカタログ・アクセス権限を取り消すことができます。このタスクについては、12-12 ページの「[仮想プライベート・カタログ所有者からの権限の取消し](#)」を参照してください。仮想プライベート・カタログを削除する方法については、12-13 ページの「[仮想プライベート・カタログの削除](#)」を参照してください。

リカバリ・カタログが仮想プライベート・カタログの場合、これに接続している Recovery Manager クライアントは、パッチ・レベル 10.1.0.6 または 10.2.0.3 である必要があります。Oracle9i の Recovery Manager クライアントは、仮想プライベート・カタログに接続できません。ベース・カタログに仮想プライベート・カタログ・ユーザーが含まれている場合でも、Oracle Database 11g のベース・リカバリ・カタログへの Recovery Manager クライアントの接続は、このバージョン制限の影響を受けません。

**参照：** Recovery Manager のバージョン互換性の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## 仮想プライベート・カタログ所有者の作成および権限の付与

この項では、ベース・リカバリ・カタログは作成済であると想定しています。

データベース `prod1`、`prod2` および `prod3` がベース・リカバリ・カタログに登録されていると想定します。ベース・リカバリ・カタログを所有するデータベース・ユーザーは `catowner` です。データベース・ユーザー `vpc1` を作成し、このユーザーに `prod1` および `prod2` のみへのアクセス権限を付与します。デフォルトでは、仮想プライベート・カタログ所有者にベース・リカバリ・カタログへのアクセス権限はありません。

### 仮想プライベート・カタログ所有者を作成し、権限を付与する手順

1. SQL\*Plus を起動し、管理者権限でリカバリ・カタログ・データベースに接続します。

2. 仮想プライベート・カタログを所有するユーザーが作成されていない場合は作成します。

たとえば、カタログを所有するデータベース・ユーザー `vpc1` を作成する場合は、次のコマンドを実行します (`password` はユーザー定義のパスワードに置き換えます)。

```
SQL> CREATE USER vpc1 IDENTIFIED BY password
2  DEFAULT TABLESPACE vpcusers
3  QUOTA UNLIMITED ON vpcusers;
```

---

**注意：** 安全なパスワードを作成してください。詳細は、『Oracle Database セキュリティ・ガイド』を参照してください。

---

3. 仮想カタログを所有するデータベース・ユーザーに `RECOVERY_CATALOG_OWNER` ロールを付与し、SQL\*Plus を終了します。

次の例では、ユーザー `vpc1` にロールを付与します。

```
SQL> GRANT recovery_catalog_owner TO vpc1;
SQL> EXIT;
```

4. Recovery Manager を起動し、(仮想プライベート・カタログ所有者ではなく) ベース・リカバリ・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。

次の例では、`catowner` としてベース・リカバリ・カタログに接続します。

```
% rman
RMAN> CONNECT CATALOG catowner@catdb;

recovery catalog database Password: password
connected to recovery catalog database
```

5. 仮想プライベート・カタログ所有者に目的の権限を付与します。

次の例では、`prod1` および `prod2` (`prod3` は対象外) に関するメタデータへのアクセス権限をユーザー `vpc1` に付与します。

```
RMAN> GRANT CATALOG FOR DATABASE prod1 TO vpc1;
RMAN> GRANT CATALOG FOR DATABASE prod2 TO vpc1;
```

データベース名ではなく `DBID` を使用することもできます。仮想プライベート・カタログ・ユーザーには、リカバリ・カタログに登録されている他のデータベースに関するメタデータへのアクセス権限はありません。

新しいターゲット・データベースをリカバリ・カタログに登録する機能をユーザーに付与することもできます。たとえば、次のように入力します。

```
RMAN> GRANT REGISTER DATABASE TO vpc1;
```

## 仮想プライベート・カタログの作成

この項では、仮想プライベート・カタログ所有者に `RECOVERY_CATALOG_OWNER` データベース・ロールが付与されていると想定しています。また、ベース・リカバリ・カタログ所有者が `GRANT` コマンドを使用して、ベース・リカバリ・カタログ内のメタデータへのアクセス権限を仮想プライベート・カタログ所有者に付与しています。

### 仮想プライベート・カタログを作成する手順

1. Recovery Manager を起動し、(ベース・リカバリ・カタログ所有者ではなく) 仮想プライベート・リカバリ・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。

次の例では、`vpc1` としてリカバリ・カタログに接続します。

```
% rman
RMAN> CONNECT CATALOG vpc1@catdb;
```

2. 仮想プライベート・カタログを作成します。

次のコマンドによって仮想プライベート・カタログが作成されます。

```
RMAN> CREATE VIRTUAL CATALOG;
```

3. この仮想プライベート・カタログで 10.2 以前のリリースの Recovery Manager を使用する場合は、次の PL/SQL プロシージャを実行します (`base_catalog_owner` は、ベース・リカバリ・カタログを所有するデータベース・ユーザーです)。

```
SQL> EXECUTE base_catalog_owner.DBMS_RVCAT.CREATE_VIRTUAL_CATALOG;
```

## 仮想プライベート・カタログ所有者からの権限の取消し

この項では、仮想プライベート・カタログをすでに作成していると想定しています。

2 つのデータベース `prod1` および `prod2` がベース・リカバリ・カタログに登録されていると想定します。ベース・リカバリ・カタログの所有者として、`vpc1` ユーザーに `prod1` へのアクセス権限を付与しています。また、このユーザーには、仮想プライベート・カタログにデータベースを登録する権限も付与しています。ここで、`vpc1` から権限を取り消します。

### 仮想プライベート・カタログ所有者から権限を取り消す手順

1. Recovery Manager を起動し、(仮想プライベート・カタログ所有者ではなく) リカバリ・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。

次の例では、`catowner` としてリカバリ・カタログに接続します。

```
% rman
RMAN> CONNECT CATALOG catowner@catdb;
```

2. 指定した権限を仮想プライベート・カタログ所有者から取り消します。

次のコマンドによって、`prod1` に関するメタデータへのアクセス権限が仮想プライベート・カタログ所有者 `vpc1` から取り消されます。

```
REVOKE CATALOG FOR DATABASE prod1 FROM vpc1;
```

データベース名ではなく `DBID` を指定することもできます。`vpc1` は、付与されているその他のすべてのカタログ権限を保持します。

新しいターゲット・データベースをリカバリ・カタログに登録する権限を取り消すこともできます。たとえば、次のように入力します。

```
REVOKE REGISTER DATABASE FROM vpc1;
```

## 仮想プライベート・カタログの削除

この項では、仮想プライベート・カタログをすでに作成しており、この時点でこれを削除すると想定しています。仮想プライベート・カタログを削除する場合、ベース・リカバリ・カタログ自体は削除しません。ベース・リカバリ・カタログを参照するシノニムおよびビューのみを削除します。

### 仮想プライベート・カタログを削除する手順

1. Recovery Manager を起動し、(ベース・リカバリ・カタログ所有者ではなく) 仮想プライベート・リカバリ・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。

次の例では、ユーザー `vpc1` としてリカバリ・カタログに接続します。

```
% rman
RMAN> CONNECT CATALOG vpc1@catdb;
```

2. カタログを削除します。

Oracle Database 11g 以上の Recovery Manager 実行可能ファイルを使用している場合は、`DROP CATALOG` コマンドを使用して仮想プライベート・カタログを削除します。

```
RMAN> DROP CATALOG;
```

Oracle Database 10g 以前の Recovery Manager 実行可能ファイルを使用している場合は、`DROP CATALOG` コマンドは使用できません。代わりに、仮想プライベート・カタログ・ユーザーとして `SQL*Plus` をカタログ・データベースに接続してから、次の PL/SQL プロシージャを実行します (`base_catalog_owner` は、ベース・リカバリ・カタログを所有するデータベース・ユーザーです)。

```
SQL> EXECUTE base_catalog_owner.DBMS_RCVCAT.DELETE_VIRTUAL_CATALOG;
```

## リカバリ・カタログの保護

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・データベースも含める必要があります。リカバリ・カタログをバックアップしないと、ディスク障害が発生してリカバリ・カタログ・データベースが破損した場合、カタログ内のメタデータが消失する場合があります。リカバリ・カタログの内容がない場合は、その他のデータベースのリカバリがより困難になる可能性があります。

## リカバリ・カタログのバックアップ

1つのリカバリ・カタログには、複数のターゲット・データベースに関するメタデータを格納できます。したがって、リカバリ・カタログが消失すると、重大な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。この項では、リカバリ・カタログを保護する計画を作成するための一般的なガイドラインについて説明します。

### リカバリ・カタログの頻繁なバックアップ

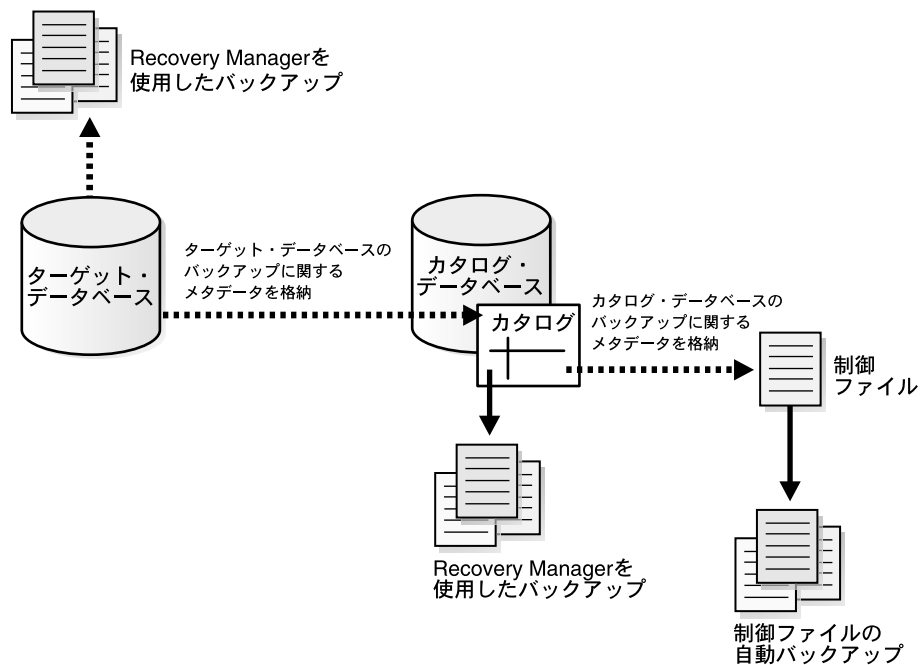
リカバリ・カタログ・データベースは他のデータベースと同様のデータベースであり、バックアップおよびリカバリ計画に含める必要があります。リカバリ・カタログは、データベースの他の部分と同様にバックアップして保護してください。リカバリ・カタログ・データベースのバックアップは、全体的なバックアップおよびリカバリの一環として計画する必要があります。

リカバリ・カタログのバックアップは、ターゲット・データベースのバックアップと同じ頻度で行います。たとえば、ターゲット・データベース全体のバックアップを週1回行う場合、ターゲット・データベースのバックアップ後にリカバリ・カタログをバックアップします。リカバリ・カタログのこのバックアップは、障害リカバリに役立ちます。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・データベースをリストアする必要がある場合も、リストアしたリカバリ・カタログ・データベース内の完全なバックアップ・レコードを使用すると、ターゲット・データベースをリストアできます。

## 適切な物理バックアップ方法の選択

リカバリ・カタログ・データベースのバックアップは、Recovery Manager を使用して実行できます。図 12-1 に示すように、Recovery Manager のリポジトリがカタログ・データベースの制御ファイルになるように、NOCATALOG オプションを指定して Recovery Manager を起動します。

図 12-1 リカバリ・カタログのバックアップのリポジトリとしての制御ファイルの使用



Recovery Manager を使用したリカバリ・カタログ・データベースのバックアップを計画する場合、次のガイドラインに従います。

- 必要に応じて Point-in-Time リカバリを実行できるように、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。
- 保存方針として、REDUNDANCY 値を 1 より大きい値に設定します。
- データベースを 2 つの異なるメディア（ディスクとテープなど）にバックアップします。
- BACKUP DATABASE PLUS ARCHIVELOG を、使用可能な場合はメディア・マネージャに、メディア・マネージャが使用不可の場合はディスクのみに定期的に行います。
- 別のリカバリ・カタログをバックアップのリポジトリに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 に設定します。

この方法では、制御ファイルの自動バックアップ機能によって、この機能が使用可能なかぎり確実にリカバリ・カタログ・データベースがリカバリされます。

**参照：** 制御ファイルの自動バックアップを使用したリカバリの詳細は、19-9 ページの「障害リカバリの実行」を参照してください。



## ターゲット・データベースからのリカバリ・カタログの分離

リカバリ・カタログは、そのカタログが保護するデータとは別の場所に格納されている場合のみ有効です。そのため、データベースの Recovery Manager リポジトリを含むリカバリ・カタログは、ターゲット・データベースと同じデータベースには格納しないでください。また、カタログ・データベースはターゲット・データベースと同じディスクに格納しないでください。

データの分離をお勧めする理由を説明するために、データベース prod1 のカタログを prod1 に格納したとします。prod1 にメディア全体の障害が発生したときに、prod1 のリカバリ・カタログも prod1 に格納されていた場合、データベースが消失すると、リカバリ・カタログも同様に消失します。この場合の対応策は1つのみです。リカバリ・カタログに格納されている情報を使用せずに、prod1 の制御ファイルの自動バックアップをリストアして使用して、データベースのリストアおよびリカバリを実行する必要があります。

## 論理バックアップ用のリカバリ・カタログ・データのエクスポート

データ・ポンプ・エクスポート・ユーティリティを使用して Recovery Manager リカバリ・カタログの論理バックアップを作成しておく、物理バックアップの有効な補助バックアップとなります。リカバリ・カタログ・データベースが破損した場合、データ・ポンプ・インポートを使用して、エクスポートされたリカバリ・カタログ・データを別のデータベースに再インポートすることで、カタログを迅速に再構築できます。

## リカバリ・カタログのリカバリ

リカバリ・カタログ・データベースのリストアおよびリカバリは、Recovery Manager を使用して他のデータベースをリストアおよびリカバリする場合とほぼ同様です。リカバリ・カタログ・データベースの制御ファイルおよびサーバー・パラメータ・ファイルを自動バックアップからリストアし、その後、データベースの残りの部分のリストアおよび完全リカバリを実行できます。複数のリカバリ・カタログを使用している場合、別のリカバリ・カタログを使用して、このリカバリ・カタログ・データベースのバックアップに関するメタデータを記録することもできます。

通常の Oracle のリカバリ手順によるリカバリ・カタログ・データベースのリカバリが実行不可能である場合は、カタログを再作成する必要があります。最悪の場合の例を次に示します。

- リカバリ・カタログ・データベースを一度もバックアップしていない。
- リカバリ・カタログ・データベースをバックアップしているが、データファイルのバックアップまたはアーカイブ・ログが使用不可であるためリカバリできない。

欠落しているリカバリ・カタログの内容を部分的に再作成するには、次の方法があります。

- RESYNC CATALOG コマンドを使用して、ターゲット・データベースの制御ファイルまたは制御ファイルのコピーからの Recovery Manager リポジトリ情報でリカバリ・カタログを更新します。制御ファイルからエージ・アウトされた制御ファイル・レコードに含まれているメタデータは消失することに注意してください。
- CATALOG START WITH ... コマンドを発行して、使用可能なバックアップをカタログに再度追加します。

この最悪の状況が発生する可能性を最小限に抑えるには、バックアップ計画に少なくともリカバリ・カタログのバックアップを含める必要があります。この方法については、12-13 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。

### 参照：

- CATALOG コマンドについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- CROSSCHECK コマンドについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## ストアド・スクリプトの管理

12-16 ページの「ストアド・スクリプト」で説明するように、リカバリ・カタログにスクリプトを格納できます。この項では、ストアド・スクリプトを作成および管理する方法について説明します。

### ストアド・スクリプト

**ストアド・スクリプト**は、制御ファイルの代替として、頻繁に使用する一連の **Recovery Manager** コマンドの管理に使用できます。このスクリプトは、ファイル・システムではなくリカバリ・カタログに格納されます。

ストアド・スクリプトには、ローカルとグローバルの 2 種類があります。ローカル・スクリプトは、作成時に **Recovery Manager** が接続していたターゲット・データベースに関連付けられます。このスクリプトは、関連付けられたターゲット・データベースに接続している場合のみ実行できます。グローバル・ストアド・スクリプトは、**Recovery Manager** クライアントがリカバリ・カタログおよびターゲット・データベースに接続している場合、リカバリ・カタログに登録されたすべてのデータベースに対して実行できます。

**CREATE SCRIPT** コマンドの大カッコ内で使用できるコマンドは、**RUN** ブロック内でサポートされているコマンドと同じです。**RUN** コマンド内で有効なコマンドは、ストアド・スクリプトで使用できます。**RUN**、**@**および**@@** コマンドは、ストアド・スクリプト内で有効ではありません。

スクリプト名を指定する場合、ストアド・スクリプト名を引用符で囲むことはできますが、必須ではありません。ただし、名前が数値で始まる場合や **Recovery Manager** の予約語である場合、その名前をストアド・スクリプト名として使用するには、引用符で囲む必要があります。ストアド・スクリプトには、英字以外の文字で始まる名前、または **Recovery Manager** の予約語と同じ名前を付けないことをお勧めします。

グローバル・ストアド・スクリプトとローカル・ストアド・スクリプトが混同されないように、ネーミング規則を使用することをお勧めします。**EXECUTE SCRIPT**、**DELETE SCRIPT** および **PRINT SCRIPT** コマンドで引数として渡されたスクリプト名が、接続しているターゲット・インスタンス用に定義されたスクリプトの名前ではない場合、**Recovery Manager** は、指定した名前が付いたグローバル・スクリプトを検索します。たとえば、リカバリ・カタログに **global\_backup** という名前のグローバル・スクリプトが存在し、ターゲット・データベース用に定義された **global\_backup** というローカル・ストアド・スクリプトが存在しない場合に次のコマンドを実行すると、このグローバル・スクリプトは削除されます。

```
DELETE SCRIPT global_backup;
```

ストアド・スクリプト（グローバル・スクリプトの場合も）に関連するコマンドを使用するには、リカバリ・カタログとターゲット・データベース・インスタンスの両方に接続する必要があります。ことに注意してください。

### ストアド・スクリプトの作成

**CREATE SCRIPT** コマンドを使用すると、ストアド・スクリプトを作成できます。**GLOBAL** を指定する場合は、この名前のグローバル・スクリプトがリカバリ・カタログ内に存在していない必要があります。**GLOBAL** を指定しない場合は、同じターゲット・データベースに対して同じ名前のローカル・スクリプトが存在していない必要があります。また、**REPLACE SCRIPT** を使用すると、新しいスクリプトの作成または既存のスクリプトの更新を行うこともできます。

#### ストアド・スクリプトを作成する手順

1. **Recovery Manager** を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。

2. CREATE SCRIPT コマンドを実行します。

次に、ローカル・スクリプトを作成する例を示します。

```
CREATE SCRIPT full_backup
{
  BACKUP DATABASE PLUS ARCHIVELOG;
  DELETE OBSOLETE;
}
```

グローバル・スクリプトの場合、次のとおり、同様の構文を実行します。

```
CREATE GLOBAL SCRIPT global_full_backup
{
  BACKUP DATABASE PLUS ARCHIVELOG;
  DELETE OBSOLETE;
}
```

必要に応じて、COMMENT を使用して説明を追加できます。

```
CREATE GLOBAL SCRIPT global_full_backup
COMMENT 'use only with ARCHIVELOG mode databases'
{
  BACKUP DATABASE PLUS ARCHIVELOG;
  DELETE OBSOLETE;
}
```

テキスト・ファイルから内容を読み取ってスクリプトを作成することもできます。このファイルは、左中カッコ ({} で始まり、次に有効な一連のコマンドが含まれている RUN ブロックを含み、最後に右中カッコ (}) で終わる必要があります。そうでない場合、キーボードからコマンドを入力した場合と同様の構文エラーが発生します。

```
CREATE SCRIPT full_backup
FROM FILE '/tmp/my_script_file.txt';
```

3. 出力を確認します。

エラーが表示されていない場合、Recovery Manager は、スクリプトを正常に作成し、リカバリ・カタログに格納しています。

**参照：** Recovery Manager の予約語のリストは、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## ストアド・スクリプトの置換え

ストアド・スクリプトを更新するには、REPLACE SCRIPT コマンドを使用します。ローカル・スクリプトを置き換える場合は、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ります。スクリプトが存在しない場合、Recovery Manager によってスクリプトが作成されます。

### ストアド・スクリプトを置き換える手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. REPLACE SCRIPT を実行します。

次の例では、スクリプト full\_backup を新しい内容で更新します。

```
REPLACE SCRIPT full_backup
{
  BACKUP DATABASE PLUS ARCHIVELOG;
}
```

GLOBAL キーワードを次のように指定することによって、グローバル・スクリプトを更新できます。

```
REPLACE GLOBAL SCRIPT global_full_backup
COMMENT 'A script for full backup to be used with any database'
{
    BACKUP AS BACKUPSET DATABASE PLUS ARCHIVELOG;
}
```

CREATE SCRIPT と同様に、ローカルまたはグローバル・ストアド・スクリプトをテキスト・ファイルから更新できます。その場合、次の形式のコマンドを使用します。

```
REPLACE GLOBAL SCRIPT global_full_backup
FROM FILE '/tmp/my_script_file.txt';
```

**参照：** REPLACE SCRIPT コマンドの構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## ストアド・スクリプトの実行

ストアド・スクリプトを実行するには、EXECUTE SCRIPT コマンドを使用します。GLOBAL を指定する場合は、この名前のグローバル・スクリプトがリカバリ・カタログ内に存在している必要があります。存在していない場合、Recovery Manager はエラー RMAN-06004 を戻します。GLOBAL を指定しない場合、Recovery Manager は、現在のターゲット・データベースに対して定義されたローカル・ストアド・スクリプトを検索します。この名前のローカル・スクリプトが検出されなかった場合、Recovery Manager は、同じ名前でもグローバル・スクリプトを検索し、検出された場合は実行します。

### ストアド・スクリプトを実行する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。
2. 必要に応じて、SHOW を使用して構成済のチャンネルを確認します。

スクリプトは、実行時に構成されていた自動チャンネルを使用します。構成済のチャンネルを変更する必要がある場合、スクリプトに ALLOCATE CHANNEL を指定します。RUN ブロックが使用されているため、スクリプト内のある Recovery Manager コマンドが正常に実行されなかった場合、スクリプト内の後続の Recovery Manager コマンドは実行されません。

3. EXECUTE SCRIPT を実行します。このコマンドには RUN ブロックが必要です。次に例を示します。

```
RUN
{
    EXECUTE SCRIPT full_backup;
}
```

このコマンドによって、指定した名前が付いたローカル・スクリプトが起動されます。該当するローカル・スクリプトが存在しないが、指定した名前が付いたグローバル・スクリプトが存在する場合、Recovery Manager はそのグローバル・スクリプトを実行します。

EXECUTE GLOBAL SCRIPT を使用して、ローカル・スクリプトとグローバル・スクリプトの名前が同じである場合に起動するスクリプトを制御することもできます。global\_full\_backup というローカル・スクリプトが存在しない場合、次の2つのコマンドによって実行される操作は同じになります。

```
RUN
{
    EXECUTE GLOBAL SCRIPT global_full_backup;
}
```

```

RUN
{
EXECUTE SCRIPT global_full_backup;
}

```

**参照：** EXECUTE SCRIPT コマンドの構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## 動的ストアド・スクリプトの作成および実行

CREATE SCRIPT コマンドに置換変数を指定できます。コマンドラインで Recovery Manager を起動する場合、USING 句によって、コマンド・ファイル内の置換変数で使用される 1 つ以上の値が指定されます。SQL\*Plus の場合と同様に、&1 は最初の値を配置する場所を示し、&2 は 2 番目の値を配置する場所を示します。それ以降も同様です。

### 動的ストアド・スクリプトを作成および使用する手順

1. 動的に更新する必要がある値用の置換変数が指定されている CREATE SCRIPT 文を含むコマンド・ファイルを作成します。

次の例では、テープ・セットの名前、FORMAT 指定内の文字列およびリストア・ポイントの名前の置換変数が使用されています。

```

CREATE SCRIPT quarterly
{
ALLOCATE CHANNEL c1
DEVICE TYPE sbt
PARMS 'ENV=(OB_MEDIA_FAMILY=&1)';
BACKUP
TAG &2
FORMAT '/disk2/bck/&1%U.bck'
KEEP FOREVER
RESTORE POINT &3
DATABASE;
}

```

2. Recovery Manager をターゲット・データベース（マウントまたはオープンされている必要があります）およびリカバリ・カタログに接続して、リカバリ・カタログ・スクリプトの初期値を指定します。

たとえば、次のコマンドを入力します。

```
% rman TARGET / CATALOG rman@catdb USING arc_backup bck0906 FY06Q3
```

リカバリ・カタログは、KEEP FOREVER には必要ですが、その他の KEEP オプションには必要ありません。

3. 最初の手順で作成したコマンド・ファイルを実行して、ストアド・スクリプトを作成します。

たとえば、/tmp/catscript.rman コマンド・ファイルを次のように実行します。

```
RMAN> @/tmp/catscript.rman
```

この手順では、ストアド・スクリプトは作成はされますが、実行はされないことに注意してください。

4. 四半期ごとに、ストアド・スクリプトを実行して、置換変数の値を渡します。

次の例では、quarterly というリカバリ・カタログ・スクリプトを実行します。この例では、メディア・ファミリー（テープのセット）の名前として arc\_backup、FORMAT 文字列の一部として bck1206、リストア・ポイントの名前として FY06Q4 を指定します。

```

RUN
{
  EXECUTE SCRIPT quarterly
  USING arc_backup
      bck1206
      FY06Q4;
}

```

**参照:** 「長期格納用のデータベース・バックアップの作成」(8-24 ページ)

## ストアド・スクリプトの出力

ストアド・スクリプトを表示またはファイルに出力するには、PRINT SCRIPT コマンドを使用します。

### ストアド・スクリプトを出力する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログに接続します。
2. PRINT SCRIPT コマンドを次のように実行します。

```
PRINT SCRIPT full_backup;
```

スクリプトの内容をファイルに送信するには、次の形式のコマンドを使用します。

```
PRINT SCRIPT full_backup
  TO FILE '/tmp/my_script_file.txt';
```

グローバル・スクリプトの場合、次の構文では同じ操作が実行されます。

```
PRINT GLOBAL SCRIPT global_full_backup;
PRINT GLOBAL SCRIPT global_full_backup
  TO FILE '/tmp/my_script_file.txt';
```

**参照:** PRINT SCRIPT コマンドの構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## ストアド・スクリプト名の表示

リカバリ・カタログに定義されたスクリプトの名前を表示するには、LIST ... SCRIPT NAMES コマンドを使用します。Recovery Manager がターゲット・インスタンスに接続せずにリカバリ・カタログに接続している場合に使用できるコマンドは、LIST GLOBAL SCRIPT NAMES および LIST ALL SCRIPT NAMES のみです。その他の形式の LIST ... SCRIPT NAMES コマンドでは、リカバリ・カタログ接続が必要です。

### ストアド・スクリプト名を表示する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログに接続します。
2. LIST ... SCRIPT NAMES コマンドを実行します。

たとえば、現在接続しているターゲット・データベースに対して実行可能なすべてのグローバル・スクリプトおよびローカル・スクリプトの名前を表示するには、次のコマンドを実行します。

```
LIST SCRIPT NAMES;
```

次の例では、グローバル・スクリプトの名前のみを表示します。

```
LIST GLOBAL SCRIPT NAMES;
```

現行のリカバリ・カタログに格納されたすべてのスクリプト（リカバリ・カタログに登録されたすべてのターゲット・データベースのグローバル・スクリプトとローカル・スクリプトを含む）の名前を表示するには、次の形式のコマンドを使用します。

```
LIST ALL SCRIPT NAMES;
```

出力では、表示された各スクリプトに定義されたターゲット・データベース（または各スクリプトがグローバルであるかどうか）が示されます。

**参照：** LIST SCRIPT NAMES コマンドの構文については、『Oracle Database バックアップおよびリカバリ・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

リカバリ・カタログからストアド・スクリプトを削除するには、DELETE GLOBAL SCRIPT コマンドを使用します。

### ストアド・スクリプトを削除する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログに接続します。
2. DELETE SCRIPT コマンドを入力します。

GLOBAL を指定せずに DELETE SCRIPT を使用し、指定した名前のストアド・スクリプトがそのターゲット・データベースに存在しない場合、指定した名前が付いたグローバル・ストアド・スクリプトが検索され、該当するスクリプトが削除されます。たとえば、次のコマンドを入力するとします。

```
DELETE SCRIPT 'global_full_backup';
```

この場合、Recovery Manager は、接続したターゲット・データベース用に定義された global\_full\_backup スクリプトを検索します。該当するスクリプトが検出されない場合、グローバル・スクリプト内で global\_full\_backup という名前のスクリプトを検索し、該当するスクリプトを削除します。

グローバル・ストアド・スクリプトを削除するには、DELETE GLOBAL SCRIPT を使用します。

```
DELETE GLOBAL SCRIPT 'global_full_backup';
```

**参照：** DELETE SCRIPT コマンドの構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## Recovery Manager の起動時のストアド・スクリプトの実行

Recovery Manager クライアントの起動時にリカバリ・カタログ内のストアド・スクリプトを実行するには、Recovery Manager クライアントの起動時に SCRIPT 引数を指定します。たとえば、次のコマンドを入力してスクリプト /tmp/fbkp.cmd を実行できます。

```
% rman TARGET / CATALOG rman@catdb SCRIPT '/tmp/fbkp.cmd';
```

Recovery Manager クライアントの起動時には、（ストアド・スクリプトを含む）リカバリ・カタログおよび（スクリプトの実行先の）ターゲット・データベースに接続する必要があります。

ローカル・スクリプトとグローバル・ストアド・スクリプトが同じ名前で定義されている場合、Recovery Manager は常にローカル・スクリプトを実行します。

**参照：** Recovery Manager クライアントのコマンドライン構文の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## リカバリ・カタログのメンテナンス

この項では、様々な管理タスクおよびメンテナンス・タスクについて説明します。この項の内容は、次のとおりです。

- [リカバリ・カタログのメンテナンス](#)
- [リカバリ・カタログの再同期化](#)
- [DB\\_UNIQUE\\_NAME を変更した後のリカバリ・カタログの更新](#)
- [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の解除](#)
- [リカバリ・カタログのデータベース・インカネーションの再設定](#)
- [リカバリ・カタログのアップグレード](#)
- [リカバリ・カタログのインポートおよび移動](#)

## リカバリ・カタログのメンテナンス

リカバリ・カタログを作成してターゲット・データベースに登録した後、このカタログをメンテナンスする必要があります。たとえば、[Recovery Manager のメンテナンス・コマンド](#) (第 11 章「[Recovery Manager バックアップおよびリポジトリ・レコードのメンテナンス](#)」を参照) を実行して、バックアップ・レコードの更新および不要なバックアップの削除を行う必要があります。このタイプのメンテナンスは、リカバリ・カタログで Recovery Manager を使用するかどうかに関係なく実行する必要があります。リカバリ・カタログ・スキーマの更新などのその他のタイプのメンテナンスは、リカバリ・カタログでの Recovery Manager の使用に固有です。

Data Guard 環境でリカバリ・カタログを使用する場合は、カタログに記録されるバックアップおよびデータベース・ファイルに、特別な考慮事項が適用されます。Recovery Manager でバックアップにアクセスできるタイミング、およびアクセス可能なバックアップに対する Recovery Manager メンテナンス・コマンドの動作については、3-8 ページの「[Data Guard 環境での Recovery Manager によるファイル管理](#)」を参照してください。

## リカバリ・カタログの再同期化

Recovery Manager は、[再同期化](#)を実行する際に、リカバリ・カタログと、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を比較し、欠落したメタデータまたは変更されたメタデータを追加してカタログを更新します。ほとんどの Recovery Manager コマンドは、ターゲット制御ファイルがマウントされ、カタログが使用可能になると、再同期化を自動的に実行します。Data Guard 環境では、Recovery Manager は[逆再同期化](#)を実行して、カタログのメタデータでデータベース制御ファイルを更新できます。

### リカバリ・カタログの再同期化

リカバリ・カタログを再同期化すると、Recovery Manager が制御ファイルから取得するメタデータが常に最新に保たれます。再同期化には、完全再同期化と部分再同期化があります。

[部分再同期化](#)では、Recovery Manager はターゲット・データベースの現行の制御ファイルを読み取って、新しいバックアップ、新しいアーカイブ REDO ログなどに関する変更されたメタデータを更新します。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。

[完全再同期化](#)では、Recovery Manager は、データベース・スキーマのレコードを含め、変更されたすべてのレコードを更新します。Recovery Manager は、データベースに構造変更 (データベース・ファイルの追加または削除、新しいインカネーションの作成など) を行った後または Recovery Manager の永続的な構成を変更した後に完全再同期化を実行します。



Recovery Manager は、完全再同期化の実行時に、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** を作成します。データベースでは、あるスナップショット制御ファイルにアクセスする Recovery Manager セッションが一度に 1 つのみに制限されます。Recovery Manager は、ターゲット・データベース・ホスト上のオペレーティング・システム固有の場所にスナップショット制御ファイルを作成します。スナップショット制御ファイルの名前および場所は、6-11 ページの「**スナップショット制御ファイルの場所の構成**」の説明に従って指定できます。

このスナップショット制御ファイルによって、Recovery Manager は制御ファイルの一貫性ビューを持つことができます。制御ファイルは短期の使用を目的としているため、カタログには登録されません。Recovery Manager は、制御ファイルのチェックポイントをリカバリ・カタログに記録して、そのカタログが記録された時点を示します。

**参照：** RESYNC コマンドの詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

**Data Guard 環境でのリカバリ・カタログの再同期化** Recovery Manager は、データベースに TARGET として接続すると、リカバリ・カタログをそのデータベースとのみ自動的に再同期化します。つまり、Recovery Manager は、Data Guard 環境内の 1 つのデータベースに TARGET として接続しても、Data Guard 環境内のすべてのデータベースは自動的に再同期化しません。RESYNC CATALOG FROM DB\_UNIQUE\_NAME コマンドを使用すると、リカバリ・カタログを Data Guard 環境内のデータベースと手動で再同期化できます。

手動での再同期化の例では、Recovery Manager が本番データベース prod に TARGET として接続していること、および CONFIGURE を使用して dgprod3 用の構成を作成していることを想定しています。RESYNC CATALOG FROM DB\_UNIQUE\_NAME dgprod3 を実行すると、Recovery Manager は、リカバリ・カタログを dgprod3 制御ファイルと再同期化します。この場合、Recovery Manager は、通常の再同期化（メタデータのフローが dgprod3 制御ファイルからカタログの方向）と **逆再同期化** の両方が実行されます。逆再同期化では、Recovery Manager は、リカバリ・カタログの永続的な構成を使用して、dgprod3 制御ファイルを更新します。

**参照：** 『Oracle Data Guard 概要および管理』

## リカバリ・カタログを再同期化するタイミングの決定

Recovery Manager がターゲット・データベースおよびリカバリ・カタログに接続しているときに Recovery Manager コマンドを実行すると、Recovery Manager は自動的にリカバリ・カタログを再同期化します。そのため、RESYNC CATALOG コマンドを手動で実行する必要がある機会はあまり多くありません。次の項では、カタログの再同期化を手動で実行する必要がある場合について説明します。

**リカバリ・カタログが使用不可の場合の再同期化** 部分再同期化を実行する Recovery Manager コマンドの発行時にリカバリ・カタログが使用不可であった場合、後でカタログ・データベースをオープンして、RESYNC CATALOG コマンドを使用して手動で再同期化します。

たとえば、ターゲット・データベースがニューヨークに存在し、リカバリ・カタログが日本に存在するとします。地理的に離れたデータベースの可用性に依存しないようにするため、CATALOG モードのターゲット・データベースの日次バックアップは行わないことにします。このような場合、実行可能な頻度でカタログに接続し、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** ターゲット・データベースが ARCHIVELOG モードで実行されていると想定します。また、次の操作を実行すると想定します。

- データベースを低い頻度でバックアップします（たとえば、REDO ログが 100 個アーカイブされるごとにデータベースをアーカイブします）。
- 毎日多数のログ・スイッチが発生します（たとえば、スイッチが 1000 回発生するごとにカタログを再同期化します）。

この場合、REDO ログ・スイッチの発生時または REDO ログのアーカイブ時にリカバリ・カタログが自動的に更新されないため、リカバリ・カタログを定期的に手動で再同期化する必要があります。データベースでは、REDO ログ・スイッチおよびアーカイブ REDO ログに関するメタデータは制御ファイルにのみ格納されます。定期的に再同期化を実行して、この情報をリカバリ・カタログに伝播する必要があります。

リカバリ・カタログの再同期化の頻度は、データベースが REDO ログをアーカイブする頻度によって異なります。この操作のコストは、制御ファイル内の、最後の再同期化以降に挿入または変更されたレコードの数に比例します。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ます。多くのレコードが挿入または変更された場合、再同期化にかかる時間が長くなります。

**スタンバイ・データベースの構成後の再同期化** スタンバイ・データベースに TARGET として接続されていない場合でも、このデータベースに対して Recovery Manager 構成の作成または変更を行うことができます。このタスクは、CONFIGURE DB\_UNIQUE\_NAME または CONFIGURE ... FOR DB\_UNIQUE\_NAME コマンドを使用して実行します。12-25 ページの「リカバリ・カタログの手動での再同期化」で説明されているように、スタンバイ・データベースを手動で再同期化して、スタンバイ・データベースの制御ファイルを更新できます。

**制御ファイル・レコードがエージ・アウトされる前のリカバリ・カタログの再同期化** リカバリ・カタログ内のメタデータは、常に最新の状態にしておく必要があります。リカバリ・カタログはターゲット制御ファイルからメタデータを取得するため、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決まります。制御ファイル内のバックアップ・メタデータが、新しい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。

CONTROL\_FILE\_RECORD\_KEEP\_TIME 初期化パラメータは、レコードが制御ファイルに保持される最小日数を指定します。この日数後は、レコードは上書き対象になります。このため、これらのレコードが消去される前に、リカバリ・カタログを制御ファイルのレコードと再同期化する必要があります。CONTROL\_FILE\_RECORD\_KEEP\_TIME 設定より短い間隔で、次の処理のいずれかを実行する必要があります。

- バックアップを実行することによって、リカバリ・カタログの再同期化を暗黙的に実行します。
- 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ます。

CONTROL\_FILE\_RECORD\_KEEP\_TIME がバックアップまたは再同期化の間隔より長くなるようにします。そうしないと、制御ファイルがリカバリ・カタログに伝播される前に再利用される場合があります。通常、予定の間隔に 1 週間を追加することをお勧めします。

---

**注意：** CONTROL\_FILE\_RECORD\_KEEP\_TIME は 0 に設定しないでください。0 に設定すると、制御ファイル内のバックアップ・レコードが、Recovery Manager によってカタログに追加される前に上書きされる場合があります。

---

制御ファイルのサイズが大きくなりすぎることは問題です。ターゲット・データベースの制御ファイルのサイズは、次のものの数に応じて大きくなります。

- 実行するバックアップ操作
- データベースが生成するアーカイブ REDO ログ
- 前述の情報が制御ファイルに保持される日数

制御ファイルのサイズが大きくなり、ブロック数またはレコード数が上限に達して制御ファイルのサイズがそれ以上大きくならない場合、最も古いレコードは、CONTROL\_FILE\_RECORD\_KEEP\_TIME の設定値に達していなくても上書きされる場合があります。この場合、データベースはアラート・ログにメッセージを書き込みます。この状況が頻繁に発生する場合、CONTROL\_FILE\_RECORD\_KEEP\_TIME の値を減らし、再同期化の頻度を高めます。

**参照:**

- CONTROL\_FILE\_RECORD\_KEEP\_TIME パラメータの詳細は、『Oracle Database リファレンス』を参照してください。
- 制御ファイル管理の詳細は、『Oracle Database 管理者ガイド』を参照してください。
- 制御ファイル・レコードの上書きを監視する方法については、11-5 ページの「[制御ファイル・レコードの消失の防止](#)」を参照してください。

**リカバリ・カタログの手動での再同期化**

ターゲット・データベースの制御ファイルを使用してリカバリ・カタログの完全再同期化を強制実行するには、RESYNC CATALOG を使用します。Data Guard 環境で特定のデータベースを再同期化するか、またはすべてのデータベースを再同期化するかに応じて、RESYNC FROM DB\_UNIQUE\_NAME または ALL を使用してデータベースの一意の名前を指定できます。通常、この操作は、スタンバイ・データベースに対する CONFIGURE コマンドを実行した後、そのスタンバイ・データベースに接続していない場合に実行します。

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログに接続します。
2. ターゲット・データベースがマウントまたはオープンされていない場合は、マウントまたはオープンします。

```
STARTUP MOUNT;
```

3. リカバリ・カタログを再同期化します。

Recovery Manager プロンプトで、RESYNC CATALOG コマンドを次のように実行します。

```
RESYNC CATALOG;
```

次の例では、standby1 の制御ファイルを再同期化します。

```
RESYNC CATALOG FROM DB_UNIQUE_NAME standby1;
```

次の例では、Data Guard 環境内ですべてのデータベースの制御ファイルを再同期化します。

```
RESYNC CATALOG FROM DB_UNIQUE_NAME ALL;
```

**参照:**

- RESYNC CATALOG コマンドの構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- スタンバイ・データベースで使用するために Recovery Manager 環境を構成する方法については、『Oracle Data Guard 概要および管理』を参照してください。

**DB\_UNIQUE\_NAME を変更した後のリカバリ・カタログの更新**

Data Guard 環境でデータベースの DB\_UNIQUE\_NAME を変更する必要がある場合があります。この場合は、CHANGE DB\_UNIQUE\_NAME コマンドを実行して、リカバリ・カタログに格納されている古い DB\_UNIQUE\_NAME に関するメタデータを新しい DB\_UNIQUE\_NAME に関連付けることができます。CHANGE DB\_UNIQUE\_NAME コマンドによって、データベース自体の DB\_UNIQUE\_NAME が実際に変更されることはありません。かわりに、一意の名前が変更された（または変更される）データベースに関するカタログ・メタデータが更新されます。

次の手順では、プライマリ・データベースの DB\_UNIQUE\_NAME が prodny であること、およびスタンバイ・データベースの DB\_UNIQUE\_NAME を prodsf1 から prodsf2 に変更したことを想定しています。プライマリ・データベースの DB\_UNIQUE\_NAME を変更した後に、同じ手順を使用できます。ただし、手順 1 で、プライマリ・データベースではなく、スタンバイ・データベースに Recovery Manager を TARGET として接続する点は異なります。

**DB\_UNIQUE\_NAME を変更した後にリカバリ・カタログを更新する手順**

1. Recovery Manager を TARGET としてプライマリ・データベースに接続し、リカバリ・カタログにも接続します。たとえば、次のコマンドを入力します。

```
% rman
RMAN> CONNECT CATALOG catowner@catdb

recovery catalog database Password: password
connected to recovery catalog database

RMAN> CONNECT TARGET SYS@prodny

target database Password: password
connected to target database: PRODNY (DBID=39525561)
```

2. リカバリ・カタログで認識される DB\_UNIQUE\_NAME 値を表示します。

次の LIST コマンドを実行します。

```
RMAN> LIST DB_UNIQUE_NAME OF DATABASE;
```

3. Recovery Manager メタデータ内の DB\_UNIQUE\_NAME を変更します。

次の例では、データベースの一意の名前をスタンバイ・データベース prodsf1 から prodsf2 に変更します。

```
RMAN> CHANGE DB_UNIQUE_NAME FROM prodsf1 TO prodsf2;
```

**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の解除**

UNREGISTER DATABASE コマンドを使用すると、リカバリ・カタログからデータベースの登録を解除できます。データベースがリカバリ・カタログから登録解除されると、Recovery Manager のすべてのリポジトリ・レコードが消失します。データベースは再度登録できますが、このデータベースのリカバリ・カタログ・レコードは、再登録時の制御ファイルの内容に基づきます。ターゲット・データベースの制御ファイル内の CONTROLFILE\_RECORD\_KEEP\_TIME 設定より古いレコードは消失します。また、制御ファイルに格納されていないストアド・スクリプトも消失します。

**Data Guard 環境外のターゲット・データベースの登録の解除**

この例では、リカバリ・カタログを使用せずにプライマリ・データベースとスタンバイ・データベースのメタデータを格納すると想定しています。

**データベースの登録を解除する手順**

1. Recovery Manager を起動し、登録を解除するデータベースに TARGET として接続します。リカバリ・カタログにも接続します。

ターゲット・データベースに接続する必要はありませんが、接続しない場合は UNREGISTER コマンドにターゲット・データベースの名前を指定する必要があります。リカバリ・カタログ内に同じ名前が付いた複数のデータベースが存在する場合、RUN ブロックに UNREGISTER コマンドを含め、SET DBID を使用して、該当するデータベースの DBID を設定します。

2. 起動時に、Recovery Manager によって表示された DBID を書き留めます。

たとえば、Recovery Manager は、オープンしているターゲット・データベースに接続した場合、次の形式の行を出力します。

```
connected to target database: PROD (DBID=39525561)
```

- 念のため、LIST BACKUP SUMMARY および LIST COPY SUMMARY を使用して、リカバリ・カタログに記録されているすべてのバックアップを表示すると役に立つ場合があります。この場合、後でデータベースを再登録するときに、制御ファイルで認識されないバックアップをカタログに再度追加できます。
- データベースのすべてのバックアップを実際に削除する場合は、DELETE 文を実行して、既存のすべてのバックアップを削除します。データベースをリカバリ・カタログから削除し、制御ファイルを使用してこのデータベースの Recovery Manager メタデータを格納する場合は、すべてのバックアップは削除しないでください。

次のコマンドは、バックアップの削除方法を示しています。

```
DELETE BACKUP DEVICE TYPE sbt;
DELETE BACKUP DEVICE TYPE DISK;
DELETE COPY;
```

Recovery Manager によって、削除されるバックアップが表示され、それらの削除前に確認するように求められます。

- UNREGISTER DATABASE コマンドを実行します。たとえば、次のように入力します。

```
UNREGISTER DATABASE;
```

Recovery Manager にデータベース名および DBID が表示され、確認するように求められます。

```
database name is "RDBMS" and DBID is 931696259
```

```
Do you really want to unregister the database (enter YES or NO)? yes
```

プロセスが完了すると、Recovery Manager は次のメッセージを出力します。

```
database unregistered from the recovery catalog
```

## スタンバイ・データベースの登録の解除

UNREGISTER コマンドでは、Data Guard 環境での使用のために DB\_UNIQUE\_NAME 句がサポートされています。この句を使用すると、特定のデータベースのメタデータを削除できます。

リカバリ・カタログによって、バックアップは特定のデータベースに関連付けられます。データベースの登録を解除すると、Recovery Manager は、これらのバックアップ・ファイルのデータベース名を NULL に更新します。そのため、バックアップは記録されていますが、バックアップの所有者は存在しなくなります。CHANGE ... RESET DB\_UNIQUE\_NAME コマンドを実行すると、現在所有者が存在しないバックアップの所有権を別のデータベースに関連付けることができます。UNREGISTER コマンドに INCLUDING BACKUPS を指定した場合、Recovery Manager は、登録を解除されたデータベースのバックアップ・メタデータも削除します。

この例では、プライマリ・データベース lnx3 に、関連付けられたスタンバイ・データベース standby1 があると想定しています。このスタンバイ・データベースの登録を解除します。

## スタンバイ・データベースの登録を解除する手順

- Recovery Manager を起動し、TARGET としてソース・データベースに接続します。また、リカバリ・カタログにも Recovery Manager を接続します。

たとえば、次のコマンドを入力します。

```
% rman
RMAN> CONNECT TARGET SYS@lnx3

target database Password: password
connected to target database: LNX3 (DBID=781317675)

RMAN> CONNECT CATALOG rman@catdb
```

2. データベースの一意の名前を表示します。

たとえば、LIST DB\_UNIQUE\_NAME コマンドを次のように実行します。

```
RMAN> LIST DB_UNIQUE_NAME OF DATABASE;
```

```
List of Databases
DB Key  DB Name DB ID          Database Role  Db_unique_name
-----
1       LNX3    781317675      STANDBY       STANDBY
1       LNX3    781317675      PRIMARY       LNX3
```

3. UNREGISTER DB\_UNIQUE\_NAME コマンドを実行します。

たとえば、データベース standby の登録を解除するには、UNREGISTER コマンドを次のように実行します。

```
RMAN> UNREGISTER DB_UNIQUE_NAME standby;
```

Recovery Manager にデータベース名および DBID が表示され、確認するように求められます。

```
database db_unique_name is "standby", db_name is "LNX3" and DBID is 781317675
```

```
Do you really want to unregister the database (enter YES or NO)? yes
```

プロセスが完了すると、Recovery Manager は次のメッセージを出力します。

```
database with db_unique_name standby unregistered from the recovery catalog
```

## リカバリ・カタログのデータベース・インカーネーションの再設定

13-6 ページの「データベース・インカーネーション」で説明されているように、RESETLOGS オプションでデータベースをオープンする際にデータベースの新しいインカーネーションを作成します。新しいインカーネーションのレコードには、V\$DATABASE\_INCARNATION ビューでアクセスできます。

RESETLOGS オプションでデータベースをオープンすると、新しいデータベース・インカーネーション・レコードがリカバリ・カタログ内に自動的に作成されます。また、データベースによって暗黙的かつ自動的に RESET DATABASE コマンドが発行され、このデータベースの新しいインカーネーションが現行のインカーネーションであることが指定されます。ターゲット・データベースによって実行される後続のすべてのバックアップおよびログのアーカイブは、新しいデータベース・インカーネーションに関連付けられます。

Recovery Manager で、RESTORE と RECOVER、または FLASHBACK DATABASE のいずれかを使用してデータベースを現行の RESETLOGS SCN より前の SCN に戻す場合は、常に RESET DATABASE TO INCARNATION コマンドを実行する必要があります。ただし、Recovery Manager ではフラッシュバックで RESET DATABASE TO INCARNATION が暗黙的に実行されるため、次の場合はこのコマンドを明示的に実行する必要はありません。

- FLASHBACK DATABASE を使用して、データベースを直系祖先パス内の SCN に戻す場合 (直系祖先パスについては、13-6 ページの「データベース・インカーネーション」を参照)
- FLASHBACK DATABASE を使用して、データベースをリストア・ポイントに戻す場合

次の手順では、RESETLOGS を使用してリカバリする場合にデータベース・インカーネーションをリセットする方法について説明します。

**リカバリ・カタログをメディア・リカバリの古いインカーネーションにリセットする手順**

1. 必要なデータベース・インカーネーションのインカーネーション・キーを確認します。インカーネーション・キー値を取得するには、LIST コマンドを発行します。

```
LIST INCARNATION OF DATABASE trgt;
```

```
List of Database Incarnations
```

| DB Key | Inc Key | DB Name | DB ID      | STATUS  | Reset SCN | Reset Time |
|--------|---------|---------|------------|---------|-----------|------------|
| 1      | 2       | TRGT    | 1224038686 | PARENT  | 1         | 02-JUL-02  |
| 1      | 582     | TRGT    | 1224038686 | CURRENT | 59727     | 10-JUL-02  |

インカーネーション・キーは、Inc Key 列に表示されます。

2. データベースを以前のインカーネーションに再設定します。たとえば、次のように入力します。

```
RESET DATABASE TO INCARNATION 2;
```

3. 以前のインカーネーションの制御ファイルが使用可能でマウントされている場合、手順 6 にスキップします。そうでない場合、データベースを停止して、NOMOUNT モードで起動します。たとえば、次のように入力します。

```
SHUTDOWN IMMEDIATE  
STARTUP NOMOUNT
```

4. 以前のインカーネーションから制御ファイルをリストアします。タグ付きの制御ファイルの場合、タグを指定します。そうでない場合、次のとおり、SET UNTIL コマンドを実行できます。

```
RUN  
{  
  SET UNTIL 'SYSDATE-45';  
  RESTORE CONTROLFILE; # only if current control file is not available  
}
```

5. リストアされた制御ファイルをマウントします。

```
ALTER DATABASE MOUNT;
```

6. RESTORE コマンドおよび RECOVER コマンドを実行して以前のインカーネーションからデータベース・ファイルをリストアおよびリカバリし、RESETLOGS オプションを指定してデータベースをオープンします。たとえば、次のように入力します。

```
RESTORE DATABASE;  
RECOVER DATABASE;  
ALTER DATABASE OPEN RESETLOGS;
```

**参照：** RESET DATABASE 構文については『Oracle Database バックアップおよびリカバリ・リファレンス』、LIST 構文については『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。



## リカバリ・カタログのアップグレード

この項では、リカバリ・カタログのアップグレードおよびアップグレードの実行が必要となるタイミングについて説明します。

### リカバリ・カタログのアップグレード

Recovery Manager クライアントで要求されるバージョンより古いリカバリ・カタログ・スキーマを使用している場合、そのカタログをアップグレードする必要があります。Recovery Manager のバージョンと互換性があるスキーマ・バージョンについては、『Oracle Database バックアップおよびリカバリ・リファレンス』の互換性の表を参照してください。たとえば、Oracle Database 11g の Recovery Manager クライアントでリリース 10.2 のリカバリ・カタログ・スキーマを使用する場合、そのカタログをアップグレードする必要があります。

Oracle Database 10gR1 バージョンのリカバリ・カタログ・スキーマには、CREATE TYPE 権限が必要であることに注意してください。10gR1 より前のリリースでリカバリ・カタログの所有者を作成し、CREATE TYPE 権限が含まれていない RECOVERY\_CATALOG\_OWNER ロールを付与した場合は、カタログのアップグレードの前に、明示的にこの所有者に CREATE TYPE を付与する必要があります。

Recovery Manager クライアントで要求され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、UPGRADE CATALOG を発行するとエラーが表示されます。Recovery Manager では、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必要ない場合でも UPGRADE CATALOG コマンドを実行できます。このコマンドを実行すると、必要に応じていつでもパッケージを再作成できます。アップグレード中に生成されたエラー・メッセージを、メッセージ・ログで確認してください。

**Data Guard 環境での特別な考慮事項** Data Guard 環境で、リカバリ・カタログ・スキーマを Oracle Database 11g にアップグレードすると想定します。Recovery Manager は、スタンバイ・データベースに接続する際に、新しいデータベース情報を自動的に登録し、再同期化してファイル名を制御ファイルから取得します。

再同期化中に、Recovery Manager は名前をターゲット・データベース名に関連付けます。リカバリ・カタログには、履歴メタデータが含まれているため、制御ファイルで認識されないレコードもあります。たとえば、standby1 制御ファイルでは、primary1 に作成されたすべてのバックアップについては認識されません。これらの古いレコードのデータベースの一意の名前は NULL になります。12-22 ページの「リカバリ・カタログのメンテナンス」で説明されているように、CROSSCHECK を使用してこれらのレコードを修正できます。

### リカバリ・カタログのスキーマ・バージョンの確認

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存在し、カタログのスキーマ・バージョンが特定のターゲット・データベース・バージョンで使用可能かどうかを確認する必要がある場合に重要です。

#### リカバリ・カタログのスキーマ・バージョンを確認する手順

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。
2. RCVER 表を問い合せてスキーマ・バージョンを取得します。次に、コマンドと出力の例を示します。

```
SELECT *
FROM rcver;

VERSION
-----
10.02.00
```



rcver 表に複数の行が表示される場合、この表内で最上位のバージョンが、現行のカタログ・スキーマ・バージョンです。この表に格納されるのはメジャー・バージョン番号のみであり、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たと想定します。

```
VERSION
-----
08.01.07
09.00.01
10.02.00
```

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によって作成され、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 (10.2) にアップグレードされたことを示しています。カタログ・スキーマの現行バージョンは 2 (10.2) です。

**参照：** Recovery Manager 環境で適用される互換性規則の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## UPGRADE CATALOG コマンドの使用

この例では、リカバリ・カタログ・スキーマを現行のバージョンにアップグレードすると想定しています。

### リカバリ・カタログをアップグレードする手順

1. 10gR1 より前のリリースでリカバリ・カタログの所有者を作成し、RECOVERY\_CATALOG\_OWNER ロールに CREATE TYPE 権限が含まれていなかった場合は、これを付与します。

たとえば、SQL\*Plus を起動し、管理者権限でリカバリ・カタログ・データベースに接続します。これによって、次の GRANT 文を実行できます。

```
SQL> GRANT CREATE TYPE TO rman;
SQL> EXIT;
```

2. Recovery Manager を起動し、リカバリ・カタログ・データベースに Recovery Manager を接続します。
3. UPGRADE CATALOG コマンドを実行します。

```
RMAN> UPGRADE CATALOG;

recovery catalog owner is rman
enter UPGRADE CATALOG command again to confirm catalog upgrade
```

4. 確認のために UPDATE CATALOG コマンドを再度実行します。

```
RMAN> UPGRADE CATALOG;

recovery catalog upgraded to version 11.01.00
DBMS_RCVMAN package upgraded to version 11.01.00
DBMS_RCVCAT package upgraded to version 11.01.00
```

### 参照：

- UPGRADE CATALOG コマンドの構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- リカバリ・カタログの互換性については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- 互換性および移行の詳細は、『Oracle Database アップグレード・ガイド』を参照してください。

## リカバリ・カタログのインポートおよび移動

Recovery Manager で `IMPORT CATALOG` コマンドを使用して、1 つのリカバリ・カタログ・スキーマを別のリカバリ・カタログ・スキーマにマージできます。このコマンドは、次の場合に有効です。

- 様々なバージョンのデータベースに対して複数のリカバリ・カタログ・スキーマが存在する場合。バックアップ・メタデータを失わずに、既存のすべてのスキーマを 1 つにマージする必要があります。
- リカバリ・カタログをデータベース間で移動する場合。

### リカバリ・カタログのインポート

**ソース・カタログ・スキーマ**とは、`IMPORT CATALOG` を使用して別のスキーマにインポートするカタログ・スキーマのことです。**宛先カタログ・スキーマ**とは、ソース・カタログ・スキーマのインポート先となるカタログ・スキーマのことです。

デフォルトでは、Recovery Manager は、ソース・リカバリ・カタログに登録されているすべてのターゲット・データベースのメタデータをインポートします。必要に応じて、ソース・カタログ・スキーマからインポートするデータベースの ID のリストを指定できます。

デフォルトでは、Recovery Manager は、正常なインポートの後で、インポートしたデータベースをソース・カタログ・スキーマから登録解除します。登録解除が正常に実行されたかどうかを示すために、Recovery Manager は、マージされたデータベースの登録解除の前後にメッセージを出力します。NO `UNREGISTER` オプションを指定して、データベースが宛先カタログから登録解除されないように指定することもできます。

**ストアド・スクリプト**は、グローバルまたはローカルのいずれかです。グローバル・スクリプトでは、宛先スキーマにすでにスクリプト名が含まれているために、インポート中に名前の競合が発生する可能性があります（ローカル・スクリプトでは発生しません）。この場合、Recovery Manager は、グローバル・スクリプト名を `COPY OF script_name` に変更します。たとえば、Recovery Manager は、`bp_cmd` を `COPY OF bp_cmd` に変更します。

名前が変更されたグローバル・スクリプトも一意でない場合、Recovery Manager は、その名前を `COPY(2) OF script_name` に変更します。このスクリプト名も存在する場合、Recovery Manager は、スクリプトの名前を `COPY(3) OF script_name` に変更します。Recovery Manager は、スクリプト名が一意になるまで `COPY(n) OF` のパターンを繰り返します。

### リカバリ・カタログのインポートの前提条件

『Oracle Database バックアップおよびリカバリ・リファレンス』の互換性の表に示されているように、ターゲット・データベース、リカバリ・カタログ・データベースおよびリカバリ・カタログ・スキーマのデータベース・バージョンが異なっている場合があります。最新のバージョンのリカバリ・カタログ・スキーマで、既存のすべてのリカバリ・カタログを 1 つのリカバリ・カタログにインポートすることをお勧めします。カタログのバージョンを確認する方法については、12-30 ページの「[リカバリ・カタログのスキーマ・バージョンの確認](#)」を参照してください。互換性の表を確認し、使用している環境内で互換性があるスキーマ・バージョンを判別します。

`IMPORT CATALOG` を使用する場合、ソース・リカバリ・カタログ・スキーマのバージョンは、このコマンドの実行に使用する Recovery Manager 実行可能ファイルの現行のバージョンと一致する必要があります。ソース・カタログ・スキーマのバージョンのほうが高い場合は、スキーマをインポートする前に現行のバージョンにアップグレードします。アップグレード方法については、12-30 ページの「[リカバリ・カタログのアップグレード](#)」を参照してください。ソース・リカバリ・カタログ・スキーマのバージョンのほうが高い場合は、より高いバージョンの Recovery Manager 実行可能ファイルを使用してインポートを再試行します。

ソース・カタログ・スキーマと宛先カタログ・スキーマの両方にデータベースを登録することはできません。現在、データベースが両方のカタログ・スキーマに登録されている場合は、インポートを実行する前にソース・カタログ・スキーマからデータベースを登録解除します。

## リカバリ・カタログのインポート

リカバリ・カタログを別のリカバリ・カタログにインポートする場合、ターゲット・データベースへの接続は必要ありません。Recovery Manager は、ソース・カタログおよび宛先カタログにのみ接続する必要があります。

次の例では、データベース `srcdb` にユーザー `102cat` が所有する 10.2 リカバリ・カタログ・スキーマが含まれており、データベース `destdb` にユーザー `111cat` が所有する 11.1 リカバリ・カタログ・スキーマが含まれています。

### リカバリ・カタログをインポートする手順

1. Recovery Manager を起動し、CATALOG として宛先リカバリ・カタログ・スキーマに接続します。たとえば、次のように入力します。

```
% rman
RMAN> CONNECT CATALOG 111cat@destdb;
```

2. ソース・カタログの接続文字列を指定して、ソース・リカバリ・カタログ・スキーマをインポートします。

たとえば、次のコマンドを入力して、データベース `srcdb` の `102cat` が所有するカタログをインポートします。

```
IMPORT CATALOG 102cat@srcdb;
```

前述の例を少し変更して、ソース・カタログに登録されたターゲット・データベースのサブセットに関するメタデータをインポートすることもできます。DBID またはデータベース名によってデータベースを指定できます。次に例を示します。

```
IMPORT CATALOG 102cat@srcdb DBID=1423241, 1423242;
IMPORT CATALOG 102cat@srcdb DB_NAME=prod3, prod4;
```

3. 必要に応じて、ターゲット・データベースに接続して、メタデータが正常にインポートされたことを確認します。たとえば、次のコマンドは、データベース `prod1` に TARGET として接続し、このデータベースのすべてのバックアップを表示します。

```
LIST BACKUP;
```

## リカバリ・カタログの移動

データベース間でリカバリ・カタログを移動する手順は、カタログをインポートする手順に類似しています。次の例では、ソース・データベースは既存のリカバリ・カタログを含むデータベースで、移動先データベースには移動されたリカバリ・カタログが含まれます。

ソース・データベースから移動先データベースにリカバリ・カタログを移動する手順

1. 移動先データベースでリカバリ・カタログを作成しますが、新しいカタログにデータベースは登録しません。

このタスクの実行方法については、12-5 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. 前の手順で作成したカタログにソース・カタログをインポートします。

このタスクの実行方法については、12-33 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。

## リカバリ・カタログの削除

DROP CATALOG コマンドを使用すると、CREATE CATALOG コマンドの結果として作成されたオブジェクトが削除されます。リカバリ・カタログを所有するユーザーが、CREATE CATALOG で作成されなかったオブジェクトも所有している場合、これらのオブジェクトは DROP CATALOG コマンドでは削除されません。

リカバリ・カタログ・スキーマのバックアップが存在しない場合にリカバリ・カタログを削除すると、このカタログに登録されているすべてのターゲット・データベースのバックアップが使用できなくなる可能性があります。ただし、すべてのターゲット・データベースの制御ファイルには、そのデータベースの最近のバックアップのレコードが保持されます。

DROP CATALOG コマンドは、複数のターゲット・データベースが登録されたリカバリ・カタログから 1 つのデータベースを登録解除する場合には使用しないことをお勧めします。リカバリ・カタログを削除すると、そのカタログに登録されたすべてのターゲット・データベースのバックアップのリカバリ・カタログ・レコードが削除されます。

### リカバリ・カタログ・スキーマを削除する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログに接続します。削除するカタログ・スキーマの所有者として、リカバリ・カタログに接続します。

次の例では、ユーザー catowner としてリカバリ・カタログに接続します。

```
% rman TARGET / CATALOG catowner@catdb
```

2. DROP CATALOG コマンドを実行します。

```
DROP CATALOG;
```

```
recovery catalog owner is catowner  
enter DROP CATALOG command again to confirm catalog removal
```

3. 確認のために DROP CATALOG コマンドを再度実行します。

```
DROP CATALOG;
```

---

**注意：** リカバリ・カタログを削除しても、制御ファイルにはバックアップに関するレコードが含まれたままです。Recovery Manager リポジトリ・レコードを制御ファイルから消去するには、制御ファイルを再作成します。

---

**参照：** DROP CATALOG コマンドの構文については『Oracle Database バックアップおよびリカバリ・リファレンス』、カタログからデータベースを登録解除する方法については 12-26 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の解除」を参照してください。

# 第 V 部

---

## 障害の診断および対応

第 V 部では、メディア障害およびデータ破損に対する診断および対応方法について説明します。第 V 部に含まれる章は次のとおりです。

- 第 13 章「Recovery Manager のデータ修復の概要」
- 第 14 章「データ・リカバリ・アドバイザーを使用した障害の診断および修復」
- 第 15 章「データベース・ファイルおよびバックアップの検証」
- 第 16 章「フラッシュバックおよびデータベースの Point-in-Time リカバリの実行」
- 第 17 章「データベースの完全リカバリの実行」
- 第 18 章「ブロック・メディア・リカバリの実行」
- 第 19 章「Recovery Manager のリカバリの実行: 高度な例」
- 第 20 章「Recovery Manager の表領域の Point-in-Time リカバリ (TSPITR) の実行」



---

## Recovery Manager のデータ修復の概要

この章では、[データ修復](#)を実行するために理解しておく必要がある一般的な概念について説明します。この章の内容は、次のとおりです。

- [Recovery Manager のデータ修復の概要](#)
- [Recovery Manager のリストア操作](#)
- [Recovery Manager のメディア・リカバリ](#)

## Recovery Manager のデータ修復の概要

1-2 ページの「[データ保護](#)」で説明されているように、バックアップおよびリカバリ計画の主な目的はデータの保護です。効果的で効率的な計画を立案するために重要な点は、データ修復の基本的な方法を理解することです。

### データ修復が必要な問題

Oracle Database の正常な稼働を停止させたり、データベースの I/O 処理に影響を与える問題はいくつかありますが、通常、ユーザー・エラー、アプリケーション・エラーおよびメディア障害が発生した場合にのみ DBA が介入する必要があります。

#### ユーザー・エラー

ユーザー・エラーは、アプリケーション・ロジックのエラーまたは誤操作のため、データベースのデータが誤って変更または削除された場合に発生します。たとえば、ユーザーが間違ったデータベースにログインしてデータベースの表を削除した場合などです。ユーザー・エラーは、データベースの停止時間を発生させる最も大きな原因であるとみなされています。

#### アプリケーション・エラー

ソフトウェアの障害によってデータ・ブロックが破損する場合があります。[物理的な破損](#)（メディア破損とも呼ばれる）の場合、ブロックがデータベースで認識されません。

#### メディア障害

[メディア障害](#)は、通常の操作中に、データベースの外部の問題によってファイルに対する読み取りまたは書き込みができなくなった場合に発生します。典型的なメディア障害としては、ディスク障害、データベース・ファイルの削除などがあります。メディア障害は、ユーザー・エラーまたはアプリケーション・エラーほど一般的ではありませんが、バックアップおよびリカバリ計画では、メディア障害に対する準備をしておく必要があります。

## Recovery Manager のデータ修復方法

予測する状況に基づいて、データの消失に対処するために次の各オプションを組み込むことを検討した後、これらのオプションを使用できるようにデータベースを設定します。

- [データ・リカバリ・アドバイザー](#)

この Oracle Database インフラストラクチャを使用すると、障害を診断して対応する方法を提示し、自動的にその障害を修復できます。

データ・リカバリ・アドバイザーの基本的な概念については、14-2 ページの「[データ・リカバリ・アドバイザーの概要](#)」を参照してください。

- [論理フラッシュバック機能](#)

この [Oracle フラッシュバック技術](#)機能のサブセットを使用すると、過去の時点の個々のデータベース・オブジェクトまたはトランザクションを表示したり、個々のデータベース・オブジェクトまたはトランザクションを過去の時点まで巻き戻すことができます。この機能では、Recovery Manager を使用する必要はありません。

論理フラッシュバック機能の基本的な概念および（必要に応じた）参照先については、16-2 ページの「[フラッシュバック技術およびデータベースの Point-in-Time リカバリの概要](#)」を参照してください。

- [Oracle Flashback Database](#)

フラッシュバック・データベースとは、メディア・リカバリに類似したブロックレベルのリカバリ・メカニズムのことですが、通常、高速で、リストアするバックアップは必要としません。事前にフラッシュバック・ロギングを有効にしているかぎり、データファイルの古いコピーをバックアップからリストアせずに、データベース全体を前の状態に戻すことができます。データベースのフラッシュバック・ロギング用または保証付きリストア・ポイント用にフラッシュ・リカバリ領域を構成しておく必要があります。



フラッシュバック・データベースの基本的な概念については、16-2 ページの「[Point-in-Time リカバリおよびフラッシュバック機能の基本的な概念](#)」を参照してください。

- **データファイルのメディア・リカバリ**

この形式の**メディア・リカバリ**では、データファイルのバックアップをリストアし、アーカイブ REDO ログまたは増分バックアップを適用して、消失した変更をリカバリできます。データベース全体またはデータベースのサブセットのいずれかをリカバリすることができます。データファイルのメディア・リカバリは、最も汎用的な形式のリカバリで、物理的な破損および論理的な破損の両方から保護できます。

データファイルのメディア・リカバリの基本的な概念については、この章で説明します。方法については、第 17 章「[データベースの完全リカバリの実行](#)」および 16-15 ページの「[データベースの Point-in-Time リカバリの実行](#)」を参照してください。

- **ブロック・メディア・リカバリ**

この形式の**メディア・リカバリ**では、データファイル全体ではなくデータファイル内の個々のブロックをリカバリできます。

ブロック・メディア・リカバリの基本的な概念については、18-2 ページの「[ブロック・メディア・リカバリの概要](#)」を参照してください。

- **表領域の Point-in-Time リカバリ**

これは特別な形式の **Point-in-Time リカバリ** で、1 つ以上の表領域をデータベースの残りの部分より前の時点までリカバリします。

TSPITR の基本的な概念については、20-2 ページの「[Recovery Manager の TSPITR の概要](#)」を参照してください。

通常、前述の修復方法を使用するために必要な概念は、その方法とともに説明されています。この章では、Recovery Manager のいくつかのデータ修復方法に共通の概念について説明します。

## Recovery Manager のリストア操作

Recovery Manager の **リストア** 操作では、リストアされるファイルを選択してから RESTORE コマンドを実行します。通常は、メディア・リカバリの準備としてファイルをリストアします。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（すべてのデータファイル）
- 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

リストアされるデータファイルおよび制御ファイルには、デフォルトの場所または新しい場所を指定できます。デフォルトの場所にリストアする場合、Recovery Manager は、現在その場所に存在する同じ名前のすべてのファイルを上書きします。また、SET NEWNAME コマンドを使用して、リストアされるデータファイルに対して新しい場所を指定することもできます。その後、SWITCH コマンドを実行して、新しい場所にあるリストアされたファイルが現行のデータファイルであることを示すように制御ファイルを更新できます。

**参照：** RESTORE 構文および前提条件は『Oracle Database バックアップおよびリカバリ・リファレンス』、SET NEWNAME 構文は『Oracle Database バックアップおよびリカバリ・リファレンス』、および SWITCH 構文は『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## バックアップの選択

Recovery Manager は、Recovery Manager リポジトリに存在する使用可能なバックアップ・セットまたはイメージ・コピーのレコードを使用して、リストア操作に使用する最適なバックアップを選択します。使用可能な最新のバックアップ、または RESTORE コマンドで指定した UNTIL 句を満たす最新のバックアップが選択されます。同じ時点の 2 つのバックアップが存在する場合、Recovery Manager はバックアップ・セットよりイメージ・コピーを選択します。これは、バックアップ・セットよりイメージ・コピーからのリストアの方が高速に実行できるためです（特に、テープ上に格納されている場合）。

Recovery Manager でバックアップをリストアする前に、RESTORE コマンドのすべての仕様を満たす必要があります。DEVICE TYPE 句で制限していないかぎり、RESTORE コマンドでは、構成されたチャンネルのすべてのデバイス・タイプでバックアップが検索されます。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がリポジトリに存在しない場合は、ファイルがリストアできないことを示すエラーが戻されます。

手動で割り当てたチャンネルのみを使用する場合、チャンネルを割り当てたメディアに使用可能なバックアップが存在しないと、バックアップ・ジョブが失敗する場合があります。自動チャンネルを構成すると、指定した基準を満たすバックアップが Recovery Manager によって検索およびリストアされる可能性が高くなります。

バックアップ・セットが**バックアップの暗号化**で保護されている場合、Recovery Manager は、バックアップ・セットの内容をリストアする際にバックアップ・セットを自動的に復号化します。透過的に暗号化されたバックアップは、Oracle ウォレットがオープンして使用可能な場合にかぎり、ユーザーが介入せずにリストアできます。パスワード暗号化バックアップをリストアするには、正しいパスワードを入力する必要があります。

**参照:** 「高度なチャンネル・オプションの構成」 (6-2 ページ)

## リストア・フェイルオーバー

Recovery Manager は、自動的に**リストア・フェイルオーバー**を使用し、破損したバックアップまたはアクセスできないバックアップをスキップして使用可能なバックアップを検索します。バックアップが検出されなかった場合、または破損したデータがバックアップに含まれている場合、Recovery Manager は、目的のファイルをリストアするための別のバックアップを自動的に検索します。

Recovery Manager は、実行しているフェイルオーバーのタイプを示すメッセージを生成します。たとえば、Recovery Manager は、同じファイルの別のバックアップにフェイルオーバーする場合、次のようなメッセージを表示します。

```
failover to piece handle=/u01/backup/db_1 tag=BACKUP_031009
```

Recovery Manager は、使用可能なコピーが存在しない場合、以前のバックアップを検索します。次に、表示されるメッセージの例を示します。

```
ORA-19624: operation failed, retry possible
ORA-19505: failed to identify file "/u01/backup/db_1"
ORA-27037: unable to obtain file status
SVR4 Error: 2: No such file or directory
Additional information: 3
failover to previous backup
```

Recovery Manager は、すべての使用可能なバックアップを使い切るまでリストア・フェイルオーバーを繰り返し実行します。すべてのバックアップを使用できない場合またはバックアップが存在しない場合、Recovery Manager はデータファイルの再作成を試行します。リストア・フェイルオーバーは、RECOVER、RECOVER ... BLOCK および FLASHBACK DATABASE コマンドの実行中、アーカイブ REDO ログのリストアでエラーが発生した場合にも使用されます。

## リストアの最適化

Recovery Manager は、可能な場合、バックアップからデータファイルがリストアされないように **リストアの最適化** を使用します。データファイルが正しい場所にあり、そのヘッダーに必要な情報が含まれている場合、バックアップからデータファイルはリストアされません。

---

**注意：** リストアの最適化では、データファイルのヘッダーのみがチェックされます。データファイルのボディに破損ブロックがあるかどうかはスキャンされません。

---

RESTORE コマンドの FORCE オプションを使用すると、この動作を無効にして、目的のファイルを無条件でリストアできます。

リストアの最適化は、複数のデータファイルをリストアする操作が中断された場合に特に有効です。たとえば、データベース全体のリストアで、1つ以外のすべてのデータファイルがリストアされた後に停電が発生したと想定します。同じ RESTORE コマンドを再度実行した場合、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をリストアします。

リストアの最適化は、データベースの複製時にも使用されます。複製のデータファイルが正しい場所にあり、ヘッダーの内容も適切な場合、このデータファイルは複製されません。ただし、RESTORE とは異なり、DUPLICATE では FORCE オプションがサポートされていません。リストアの最適化によってスキップされたデータファイルを Recovery Manager で強制的に複製するには、複製からこのデータファイルを削除した後、DUPLICATE コマンドを実行します。

**参照：** Oracle RAC 構成の RESTORE の動作の詳細は、『Oracle Real Application Clusters 管理およびデプロイメント・ガイド』を参照してください。

## Recovery Manager のメディア・リカバリ

**メディア・リカバリ** では、Recovery Manager は、リストアされたデータに変更を適用してそのデータを任意の時点にロールフォワードします。Recovery Manager では、**データファイルのメディア・リカバリ** または **ブロック・メディア・リカバリ** のいずれかを実行できます。

データファイルのメディア・リカバリとは、リストアされたデータファイルを現在の時点、または他の特定の時点の状態に更新するために、REDO ログまたは増分バックアップをファイルに適用することです。『Oracle Database 概要』で説明されているように、Recovery Manager を使用して **完全リカバリ**、**データベースの Point-in-Time リカバリ** または **表領域の Point-in-Time リカバリ** を実行できます。RESTORE コマンドを使用して、消失または破損したデータファイルまたは制御ファイルのバックアップをリストアし、RECOVER コマンドを使用して、メディア・リカバリを実行することができます。

ブロック・メディア・リカバリとは、データファイル全体ではなく、個々のデータ・ブロックのリカバリのことです。ここでは、データファイルのメディア・リカバリについてのみ説明します。メディア・リカバリの特殊な形式であるブロック・メディア・リカバリについては、18-2 ページの「**ブロック・メディア・リカバリの概要**」を参照してください。

## 増分バックアップおよびアーカイブ REDO ログの選択

メディア・リカバリは、Recovery Manager によって自動化されています。Recovery Manager は、増分バックアップおよびアーカイブ REDO ログを最適な組合せで自動的にリストアし、適用します。

必要なログ順序番号のコピーがディスク上に存在しないことが Recovery Manager リポジトリに示されている場合、Recovery Manager は、必要なログをバックアップから自動的にリストアします。デフォルトでは、いずれかのアーカイブ先が USE\_DB\_RECOVERY\_FILE\_DEST に設定されている場合、Recovery Manager は、アーカイブ・ログをフラッシュ・リカバリ領域にリストアします。その他の場合は、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。

**参照：** CROSSCHECK 構文の詳細は、『Oracle Database バックアップおよびリカバリ・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

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、データベース・**インカネーション**が作成されます。リカバリが完了すると、OPEN RESETLOGS を使用せずに通常の操作を再開できます。ただし、バックアップ制御ファイルを使用した DBPITR またはリカバリの後は、RESETLOGS オプションを使用してデータベースをオープンする必要があります。これによって、データベースの新しいインカネーションが作成されます。異なる時刻に発生した2つの異なる REDO ストリームの SCN が同じになっている場合は、混同を回避するために、データベースに新しいインカネーションが必要となります。間違った REDO をデータベースに適用すると、データベースは破損します。

1つのデータベースに複数のインカネーションが存在するかどうかによって、Recovery Manager による現行のインカネーション・パスに存在しないバックアップの処理方法が決まります。ほとんどの場合、使用に適したインカネーションは現行のデータベース・インカネーションです。ただし、データベースを以前のインカネーションにリセットする方法が最適な場合もあります。たとえば、すでに実行した Point-in-Time リカバリの結果が適切でなく、データベースを RESETLOGS より前の時点に戻す必要がある場合があります。データベース・インカネーションを理解することによって、そのような状況に備えることができます。

### OPEN RESETLOGS 操作

RESETLOGS オプションを指定してデータベースをオープンすると、データベースによって次の処理が実行されます。

- 現行のオンライン REDO ログがアーカイブされ (アクセス可能な場合)、オンライン REDO ログの内容が消去されて、ログ順序番号が 1 にリセットされます。

たとえば、現行のオンライン REDO ログの順序が 1000 および 1001 である場合、RESETLOGS でデータベースをオープンすると、ログ 1000 および 1001 がアーカイブされ、オンライン・ログは順序 1 および 2 にリセットされます。

- 現在存在していない場合は、オンライン REDO ログ・ファイルが作成されます。
- 制御ファイル内の REDO スレッド・レコードおよびオンライン REDO ログ・レコードが、新しいデータベース・インカネーションの開始時点の状態に初期化されます。

具体的には、REDO スレッドのステータスが CLOSED に、REDO スレッド・レコード内の現行のスレッド順序が 1 に、各 REDO スレッドのスレッド・チェックポイントがログ順序の最初である 1 に設定され、各スレッドから REDO ログが 1 つ選択され、その順序が 1 に初期化されます。

- すべての現行のデータファイル、オンライン REDO ログおよび後続のアーカイブ REDO ログが、新しい RESETLOGS SCN およびタイムスタンプによって更新されます。

RESETLOGS SCN とタイムスタンプが一致しないかぎり、データファイルにアーカイブ REDO ログは適用されません。そのため、RESETLOGS 要件を指定すると、現行のインカネーションの直接の親インカネーションのものではないアーカイブ・ログによってデータファイルが破損されることを回避できます。インカネーション間の関係については、後続の項で説明します。

以前のリリースでは、OPEN RESETLOGS の直後にデータベースをバックアップすることが推奨されていました。今回のリリースでは、他のバックアップと同様に RESETLOGS の実行前のバックアップを簡単にリカバリできるため、新しいデータベースのバックアップを作成するかどうかは任意です。RESETLOGS を介したリカバリを実行するには、最後のバックアップ以降に生成されたすべてのアーカイブ・ログおよび 1 つ以上の制御ファイル (現行のファイル、バックアップまたは作成したファイル) が必要です。

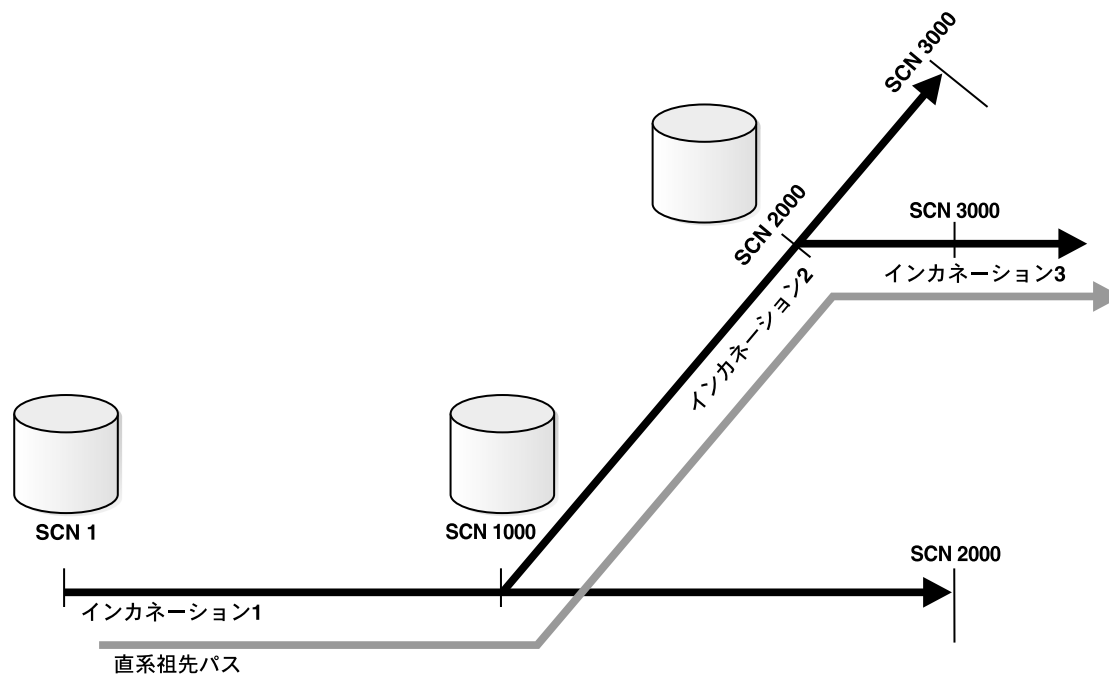
## データベース・インカネーション間の関係

データベース・インカネーションは、相互に次の関係を持つことができます。

- **現行のインカネーション**は、データベースで現在操作されているインカネーションです。
- OPEN RESETLOGS 操作によって、あるインカネーションから現行のインカネーションが分岐している場合、そのインカネーションを、現行のインカネーションの**親インカネーション**といいます。
- 親インカネーションの親は、**祖先インカネーション**です。また、祖先インカネーションのすべての親も現行インカネーションの祖先です。
- 現行インカネーションの**直系祖先パス**は、最も古いインカネーションで始まり、現行インカネーションの祖先、親インカネーションまたは現行インカネーションへのブランチのみを含みます。

REDO のストリームを一意にタグ付けして識別するために、インカネーション番号が使用されます。図 13-1 に、複数のインカネーションを持つデータベースを示します。各インカネーションに、異なるインカネーション番号が付けられています。

図 13-1 データベース・インカネーションの履歴



データベースのインカネーション 1 は SCN 1 で始まり、SCN 1000 から SCN 2000 まで続いています。インカネーション 1 の SCN 2000 で、Point-in-Time リカバリを実行して SCN 1000 まで戻り、RESETLOGS オプションを指定してデータベースをオープンするとします。これによって、インカネーション 2 は SCN 1000 で始まり、SCN 3000 まで続きます。この例では、インカネーション 1 はインカネーション 2 の親となります。

インカネーション 2 の SCN 3000 で、Point-in-Time リカバリを実行して SCN 2000 まで戻り、RESETLOGS オプションを指定してデータベースをオープンするとします。この場合、インカネーション 2 はインカネーション 3 の親となり、インカネーション 1 はインカネーション 3 の祖先となります。

データベースで DBPITR またはフラッシュバック・データベースが実行されると、いずれのインカネーションが現行インカネーションであるかによって、SCN が複数の時点を示す場合があります。たとえば、図 13-1 の SCN 1500 は、インカネーション 1 または 2 での SCN を示す場合があります。

RESET DATABASE TO INCARNATION コマンドを使用すると、SCN が特定のデータベース・インカーネーションの参照の範囲にあると解釈されます。FLASHBACK、RESTORE または RECOVER を使用して現行以外のデータベース・インカーネーションに戻る場合は、RESET DATABASE TO INCARNATION コマンドが必要です。ただし、12-28 ページの「リカバリ・カタログのデータベース・インカーネーションの再設定」で説明するように、Recovery Manager はフラッシュバックで RESET DATABASE TO INCARNATION コマンドを暗黙的に実行します。

**参照：**

- 「祖先インカーネーションへのデータベースのリカバリ」 (16-20 ページ)
- RESET DATABASE コマンドの詳細は、『Oracle Database バックアップおよびリカバリ・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

データベースが複数のインカーネーションを持つ場合、一部のバックアップは**孤立したバックアップ**の可能性があります。孤立したバックアップとは、直系祖先パス内にはないデータベースのインカーネーションで作成されたバックアップのことです。

13-7 ページの図 13-1 の例を想定します。インカーネーション 3 が現行インカーネーションの場合、次のバックアップが孤立します。

- SCN 1000 より後のインカーネーション 1 のすべてのバックアップ
- SCN 2000 より後のインカーネーション 2 のすべてのバックアップ

これに対して、次のバックアップは、直系祖先パス内にあるため孤立しません。

- SCN 1000 より前のインカーネーション 1 のすべてのバックアップ
- SCN 2000 より前のインカーネーション 2 のすべてのバックアップ
- インカーネーション 3 のすべてのバックアップ

直系祖先パス内にはない SCN までデータベースをリストアする場合は、孤立したバックアップを使用できます。Recovery Manager は、OPEN RESETLOGS 操作が実行された場合でも、最も古いバックアップからリカバリする時点までのアーカイブ・ログが連続して存在しているかぎり、親および祖先のインカーネーションからバックアップをリストアして、現在の時点までリカバリすることができます。また、Recovery Manager は、バックアップに示されている変更が取り消されたことがないインカーネーションから制御ファイルをリストアする場合、孤立したバックアップを使用してリストアおよびリカバリを実行できます。

---

## データ・リカバリ・アドバイザーを使用した 障害の診断および修復

この章では、Recovery Manager のデータ・リカバリ・アドバイザー・ツールを使用して、データベースの障害を診断および修復する方法について説明します。この章の内容は、次のとおりです。

- データ・リカバリ・アドバイザーの概要
- 障害の表示
- データベースの検証によるブロック破損の確認
- 修復オプションの決定
- 障害の修復
- 障害のステータスおよび優先順位の変更



## データ・リカバリ・アドバイザの概要

この項では、[データ・リカバリ・アドバイザ](#)の目的および基本的な概念について説明します。

### データ・リカバリ・アドバイザの目的

データ・リカバリ・アドバイザは、データ障害を自動的に診断し、適切な修復オプションを判断して提示し、ユーザーの要求に応じて修復を実行する Oracle Database ツールです。このコンテキストでは、データ障害とはディスク上の永続的なデータの破損または損失のことです。データの自動修復用の集中化されたツールであるデータ・リカバリ・アドバイザを使用すると、Oracle Database の管理性および信頼性が向上し、**MTTR** を短縮できます。

データ障害を診断し、修復に最適な計画を立案するには、高度なトレーニングおよび経験が必要です。データ・リカバリ・アドバイザには、従来の修復手法と比較して、次のようなメリットがあります。

- データ・リカバリ・アドバイザは、データベース・プロセスで破損が検出されてエラーが通知される前に、データ障害の検出、分析および修復を行うことができます。早い時期に警告することによって、破損による損害を最小限に抑えることができます。
- データ障害の症状を手動で評価し、症状を関連付けて問題陳述文にまとめる作業は、複雑で時間のかかる、エラーが発生しやすい作業です。データ・リカバリ・アドバイザは、自動的に障害を診断してその影響を評価し、発見した内容をユーザーにレポートします。
- 通常、ユーザーは修復の影響とともに修復オプションを手動で判断する必要があります。複数の障害が発生している場合、ユーザーは修復を実行する正しい順序を判断し、修復を統合する必要があります。一方、データ・リカバリ・アドバイザは、最適な修復オプションを自動的に判断し、それらのオプションが環境に適していることを確認するためのチェックを実行します。
- データの修復は、複雑でエラーが発生しやすい作業です。自動修復オプションを選択すると、データ・リカバリ・アドバイザによって修復が実行され、修復に成功したことが確認されます。

### データ・リカバリ・アドバイザの基本的な概念

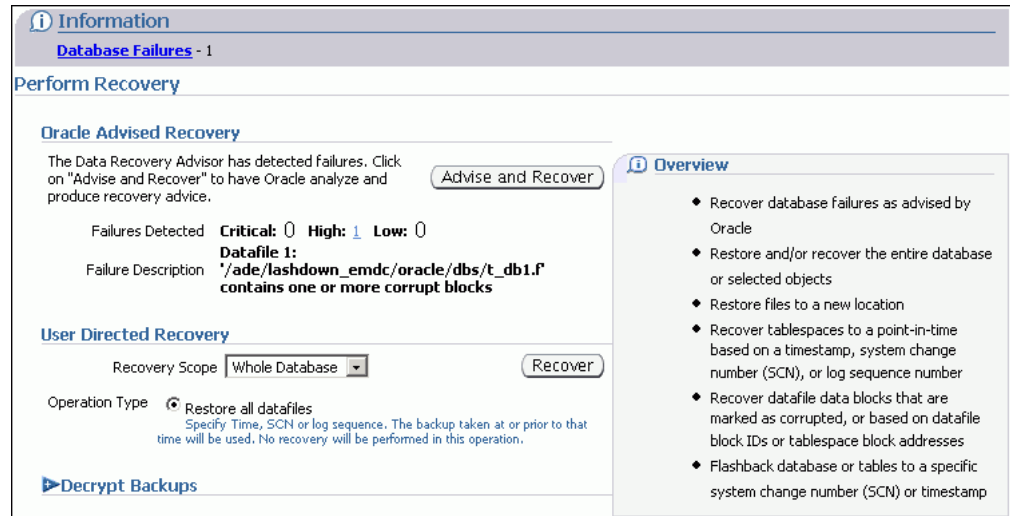
この項では、データ・リカバリ・アドバイザを使用する前に理解しておく必要がある概念について説明します。

#### データ・リカバリ・アドバイザのユーザー・インタフェース

データ・リカバリ・アドバイザには、コマンドラインおよび GUI の両方のインタフェースがあります。GUI インタフェースは、Oracle Enterprise Manager Database Control および Grid Control で使用できます。データベースのホームページの「可用性」タブにある「[リカバリの実行](#)」をクリックすると、[図 14-1](#) に示すページにナビゲートできます。



図 14-1 データ・リカバリ・アドバイザ



Recovery Manager のコマンドライン・インタフェースでのデータ・リカバリ・アドバイザ・コマンドは、LIST FAILURE、ADVISE FAILURE、REPAIR FAILURE および CHANGE FAILURE です。

**障害**は、データベースによって自動的に検出されるか、または VALIDATE コマンドなどの手動チェックによって検出されます。LIST FAILURE コマンドを使用すると、障害に関する問題陳述文およびデータベースの操作に対するそれらの障害の影響を表示できます。各障害は、障害番号によって一意に識別されます。同じ Recovery Manager セッションで ADVISE FAILURE コマンドを使用すると、修復オプションを表示できます。通常、これらのオプションには、自動オプションと手動オプションの両方があります。

ADVISE FAILURE を実行した後、障害を手動で修復するか、または REPAIR FAILURE コマンドを実行して障害を自動的に修復することができます。**修復**とは、1つ以上の障害を修復する操作のことです。修復の例には、**ブロック・メディア・リカバリ**、**データファイルのメディア・リカバリ**および **Oracle Flashback Database** があります。自動**修復オプション**を選択すると、データ・リカバリ・アドバイザは修復に成功したかどうかを検証し、該当する修復済の障害をクローズします。

**参照：** Enterprise Manager によるバックアップおよびリカバリの実行方法は、『Oracle Database 2 日でデータベース管理者』を参照してください。

## データ整合性チェック

チェッカーとは、データベースまたはデータベース・コンポーネントの状態を評価するために状態モニターに登録された診断操作またはプロシージャのことです。状態の評価は**データ整合性チェック**と呼ばれ、事後対応的または事前対応的に実行できます。

通常、障害は事後対応的に検出されます。破損したデータに対するデータベース操作でエラーが発生すると、データ整合性チェックが自動的に開始され、データベースでエラーに関連する障害が調査されます。障害が診断されると、その障害は**自動診断リポジトリ**に記録されます。ADR は、データベースの外部に格納されているディレクトリ構造です。データベースによって障害が検出され、ADR に格納された後のみ、データ・リカバリ・アドバイザを使用して、修復アドバイスを生成し、障害を修復できます。

データ整合性チェックを事前対応的に実行することもできます。チェックは状態モニターを介して実行します。状態モニターによって、事後対応的にチェックが実行された場合と同じ方法で障害が検出され、格納されます。また、14-9 ページの「**データベースの検証によるブロック破損の確認**」で説明するように、VALIDATE および BACKUP VALIDATE コマンドを使用してブロック破損をチェックすることもできます。

**参照：** 状態モニターを使用する方法の詳細は、『Oracle Database 管理者ガイド』を参照してください。

## 障害

**障害**とは、データ整合性チェックによって検出される永続的なデータの破損のことです。障害は、エラー・メッセージやアラートなどの目に見える症状として示される場合がありますが、診断された**問題**を表すため、症状とは異なります。データベースによって問題が障害として診断された後、データ・リカバリ・アドバイザを使用して、障害に関する情報を取得し、障害の修復を行うことができます。

障害情報はデータベース自体には格納されていないため、この障害情報にアクセスするためにデータベースをオープンまたはマウントする必要はありません。障害は、データベースが NOMOUNT モードで起動されている場合に表示できます。したがって、制御ファイルおよび**リカバリ・カタログ**の可用性が、検出された障害の表示に影響することはありません。ただし、修復の可能性に影響を与えることはあります。

データ・リカバリ・アドバイザでは、次のような障害を診断できます。

- 存在しない、適切なアクセス権限がない、オフラインになっているなどの理由でアクセスできないデータファイルや制御ファイルなどのコンポーネント
- ブロック・チェックサム障害、無効なブロック・ヘッダー・フィールド値などの物理的な破損
- 他のデータベース・ファイルより古いデータファイルなどの矛盾
- ハードウェア・エラー、オペレーティング・システムのドライバの障害、オペレーティング・システムのリソース制限（たとえば、オープンしているファイルの数）の超過などの I/O 障害

データ・リカバリ・アドバイザが、論理的な破損の一部を検出または処理することがあります。ただし、一般的に、このタイプの破損の場合は、Oracle サポート・サービスに連絡する必要があります。

**障害ステータス** 各障害には、OPEN または CLOSED の**障害ステータス**があります。障害ステータスは、適切な修復操作が行われるまで OPEN のままです。障害が修復されると、ステータスは CLOSED に変更されます。

LIST FAILURE を実行するたびに、データ・リカバリ・アドバイザは、ステータスが OPEN のすべての障害を再検証し、存在しなくなった障害をクローズします。したがって、別の手順の一部として障害を修復した場合、または障害が自然に消滅した一時的な問題であった場合、LIST FAILURE を実行すると、障害は自動的にクローズされます。

CHANGE FAILURE を使用すると、ステータスが OPEN の障害を手動で修復した場合にそのステータスを CLOSED に変更できます。ただし、CHANGE FAILURE ... CLOSED は、なんらかの理由で障害が自動的にクローズされなかった場合にのみ使用することをお勧めします。CHANGE を使用して手動でクローズした障害が存在したままになっている場合、データ・リカバリ・アドバイザは、適切なデータ整合性チェックを実行する際に別の障害 ID でこの障害を再作成します。

**障害優先順位** 各障害には、CRITICAL、HIGH または LOW の**障害優先順位**があります。データ・リカバリ・アドバイザは、診断された障害に対して CRITICAL または HIGH の優先順位のみを割り当てます。

優先順位が CRITICAL の障害は、データベース全体を使用できなくするため、すぐに対応する必要があります。たとえば、現行の制御ファイルを格納するディスクに障害が発生した場合などがあります。優先順位が HIGH の障害は、データベースの一部を使用できなくなったり、リカバリできなくするため、通常は迅速に修復する必要があります。これらの障害の例としては、ブロック破損やアーカイブ REDO ログの欠落などがあります。

障害に優先順位 HIGH が割り当てられていても、データベースの可用性およびリカバリ可能性にはほとんど影響がない場合は、優先順位を LOW に下げることができます。優先順位 LOW は、より重要な障害が修復されるまで、この障害を無視できることを示します。

デフォルトでは、LIST FAILURE を実行すると、優先順位が CRITICAL および HIGH の障害のみが表示されます。CHANGE コマンドを使用して、LOW および HIGH の障害のステータスは変更できますが、CRITICAL の障害のステータスは変更できません。優先順位を LOW に変更する主な理由としては、LIST FAILURE の出力を削減することがあげられます。別の障害などのためにこの時点で障害を再検証できない場合、LIST FAILURE には障害が OPEN として表示されます。

**障害のグループ化** わかりやすくするために、データ・リカバリ・アドバイザによって、関連する障害がグループ化されます。たとえば、1つのファイル内の 20 の異なるブロックが破損している場合、これらの障害は 1つの親障害にまとめられます。デフォルトでは、データ・リカバリ・アドバイザは、障害のグループに関する情報を表示します。ただし、DETAIL オプションを使用して、個々の下位の障害に関する情報を表示することもできます。

下位の障害の形式は通常の障害と同じです。下位の障害に関するアドバイスの取得およびその修復は、個別に行うことも、他の障害と組み合わせて行うこともできます。

## 手動処理および自動修復オプション

ADVISE FAILURE コマンドでは、手動および自動の両方の修復オプションを表示できます。データ・リカバリ・アドバイザは、手動処理を必須またはオプションのいずれかとしてカテゴリ化します。

場合によっては、手動処理のみが可能であることがあります。消失した制御ファイルのバックアップが存在しないとします。この場合、手動処理では、CREATE CONTROLFILE 文を使用して制御ファイルを再作成します。自動修復が利用可能ではないため、データ・リカバリ・アドバイザでは、この手動処理を必須として表示します。これに対して、欠落したデータファイルに対して Recovery Manager バックアップが存在するとします。この場合、REPAIR FAILURE コマンドを実行すると、データファイルをリストアおよびリカバリすることによって、修復を自動的に実行できます。データファイルが誤っての名前変更されたり移動された場合は、オプションの手動処理として、データファイルをリストアします。データ・リカバリ・アドバイザは、データファイルのリストアやリカバリなどの大規模な修復を回避できる場合に、オプションの手動処理を推奨します。

手動処理とは異なり、自動修復はデータ・リカバリ・アドバイザによって実行されます。ADVISE FAILURE コマンドによって、各自動修復オプションのオプション ID が表示され、処理の要約が示されます。

データ・リカバリ・アドバイザでは、自動修復を推奨する前に、実行可能性チェックを実行します。たとえば、データ・リカバリ・アドバイザは、メディア・リカバリに必要なすべてのバックアップおよびアーカイブ REDO ログが存在しており、一貫性があるかどうかをチェックします。データ・リカバリ・アドバイザでは、特定のバックアップおよびアーカイブ REDO ログを必要とすることがあります。リカバリに必要なファイルが利用できない場合、リカバリを行うことはできません。

---

**注意：** パフォーマンス上の理由から、データ・リカバリ・アドバイザは、全ファイルの全バイトを完全にチェックするわけではありません。このため、バックアップまたはアーカイブ REDO ログの破損が原因で、実行可能な修復が失敗する場合があります。

---

**統合された修復** データ・リカバリ・アドバイザは、可能なかぎり複数の障害が 1回で修復されるように修復を統合します。統合された修復では、手順が複数にわたる場合があります。

修復は統合できないこともあります。1つの障害によって、他の障害の修復を作成できない場合があるためです。たとえば、制御ファイルが欠落している場合、データファイルの修復の可能性を判断することはできません。このような場合、データ・リカバリ・アドバイザは、修復可能な障害に対して 1つの修復オプションを生成し、この時点では修復できない障害が存在することを示すメッセージをユーザーに出力します。提示された修復を実行した後、LIST、ADVISE および REPAIR を繰り返し実行すると、残りの障害を修復できます。

**スクリプトの修復** データ・リカバリ・アドバイザは、自動修復オプションの生成時に常に、Recovery Manager で障害の修復に使用されるコマンドに関するスクリプトを作成します。データ・リカバリ・アドバイザは、このスクリプトの場所を出力します。このスクリプトは、オペレーティング・システムに存在するテキスト・ファイルです。例 14-1 に、修復スクリプトの例を示します。この例は、データ・リカバリ・アドバイザによる消失したデータファイル 27 の修復計画を示しています。

#### 例 14-1 修復スクリプトのサンプル

```
# restore and recover datafile
sql 'alter database datafile 27 offline';
restore datafile 27;
recover datafile 27;
sql 'alter database datafile 27 online';
```

データ・リカバリ・アドバイザで障害を自動的に修復しない場合は、スクリプトをコピーして編集し、手動で実行できます。

## サポートされているデータベースの構成

現行のリリースでは、データ・リカバリ・アドバイザはシングル・インスタンス・データベースのみをサポートしています。Oracle Real Application Clusters (Oracle RAC) データベースはサポートしていません。

すべての Oracle RAC インスタンスを停止するデータ障害が発生した場合は、データベースをシングル・インスタンス・モードでマウントして、データ・リカバリ・アドバイザを使用し、制御ファイル、SYSTEM データファイルおよびデータ・ディクショナリの障害を検出および修復できます。また、データ・リカバリ・チェックを事前対応的に実行して、他のデータベース・コンポーネントのデータ障害をテストすることもできます。この方法では、データファイルにアクセスできないなど、他のクラスタ・インスタンスのローカルのデータ障害を検出することはできません。

データ・リカバリ・アドバイザは、**フィジカル・スタンバイ・データベース**から転送されたブロックまたはファイルを使用して、プライマリ・データベース上の障害を修復することはできません。また、データ・リカバリ・アドバイザは、スタンバイ・データベース上の障害を診断および修復することもできません。ただし、Enterprise Manager Grid Control を Data Guard 構成で使用している場合は、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て、古いプライマリ・データベースの修復を行うことができます。

**参照：** Data Guard 構成での Recovery Manager の使用方法については、『Oracle Data Guard 概要および管理』を参照してください。

## 障害の診断および修復の基本手順

データ・リカバリ・アドバイザのワークフローは、障害の疑いが発生したときまたは障害が検出されたときに始まります。障害は、エラー・メッセージ、アラート、トレース・ファイル、データ整合性チェックの失敗などの多くの方法で検出できます。14-3 ページの「**データ整合性チェック**」で説明されているように、データベースでは、エラーが発生した場合に障害を自動的に診断できます。

障害に対応する場合の基本的なプロセスでは、Recovery Manager セッションを開始し、その同じセッションで次の手順をすべて実行します。

1. LIST FAILURE コマンドを実行して、障害を表示します。

このタスクの詳細は、14-7 ページの「**障害の表示**」を参照してください。

2. データベースによって自動的に診断されていない障害の存在が疑われる場合は、VALIDATE DATABASE を実行して、破損ブロックおよび欠落しているファイルを確認します。

VALIDATE によって問題が検出されると、Recovery Manager が障害の評価の実行をトリガーします。障害が検出されると、データ・リカバリ・アドバイザからアクセスできる自動診断リポジトリにログが記録されます。

このタスクの詳細は、14-9 ページの「データベースの検証によるブロック破損の確認」を参照してください。

3. ADVISE FAILURE コマンドを実行して、修復オプションを決定します。

このタスクの詳細は、14-10 ページの「修復オプションの決定」を参照してください。

4. 修復オプションを選択します。障害は、手動で修復するか、または REPAIR FAILURE コマンドを実行して自動的に修復することができます。

このタスクの詳細は、14-13 ページの「障害の修復」を参照してください。

5. 最初の手順に戻り、すべての障害が修復されていることを確認するか、または残っている障害を特定します。

必要に応じて、データ・リカバリ・アドバイザーのワークフロー内の任意の時点で CHANGE FAILURE コマンドを使用して、障害優先順位を LOW から HIGH に、または HIGH から LOW に変更するか、あるいは手動で修復された障害をクローズすることができます。このタスクの詳細は、14-15 ページの「障害のステータスおよび優先順位の変更」を参照してください。

## 障害の表示

1 つ以上のデータベースの障害が発生した疑いがあるか、または発生したことがわかっている場合は、LIST FAILURE を使用して、これらの障害に関する情報を取得します。障害のすべてまたはサブセットを表示し、様々な方法で出力を制限できます。障害は、障害番号によって一意に識別されます。これらの番号は連続していないため、障害番号が離れていても問題はありません。

LIST FAILURE コマンドは、新しい障害を診断するためのデータ整合性チェックは実行しません。かわりに、以前に実行された評価の結果を表示します。したがって、LIST FAILURE を繰り返して実行すると、コマンドを実行してから次にコマンドを実行するまでの間に発生したエラーに対応してデータベースで新しい障害が自動的に診断された場合にのみ、その障害が検出されます。ただし、LIST FAILURE を実行すると、データ・リカバリ・アドバイザーは既存のすべての障害を再検証します。ユーザーが障害を手動で修復した場合または一時的な障害が消滅した場合、データ・リカバリ・アドバイザーは、これらの障害を LIST FAILURE の出力から削除します。別の障害などのためにこの時点で障害を再検証できない場合、LIST FAILURE によって障害が OPEN として表示されます。

## すべての障害の表示

データベースで発生する問題を確認する最も簡単な方法は、LIST FAILURE コマンドを使用する方法です。

### すべての障害を表示する手順

1. Recovery Manager を起動し、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・インスタンスが起動されている必要があります。
2. LIST FAILURE コマンドを実行します。

次の例では、データ・リカバリ・アドバイザーで認識されるすべての障害をレポートします (ページに収まるように出力の形式が再設定されています)。

```
RMAN> LIST FAILURE;
```

```
List of Database Failures
```

```
=====
```

| Failure ID | Priority | Status | Time Detected | Summary   |
|------------|----------|--------|---------------|---|
| 142        | HIGH     | OPEN   | 23-APR-07     | One or more non-system datafiles are missing  |
| 101        | HIGH     | OPEN   | 23-APR-07     | Datafile 1:<br>'/disk1/oradata/prod/system01.dbf' contains one or more corrupt blocks |

この例では、Recovery Manager は、欠落しているデータファイルのグループおよび破損ブロックが含まれているデータファイルの2つの異なる障害をレポートしています。出力には、各障害の一意の識別子（12 および 5）、優先順位、ステータスおよび検出時刻が示されています。

- 必要に応じて、LIST FAILURE ... DETAIL コマンドを実行して、障害を個々に表示します。

14-5 ページの「[障害のグループ化](#)」で説明されているように、データ・リカバリ・アドバイザは可能な限り障害を統合します。障害を個々に表示するには、DETAIL オプションを指定します。たとえば、ファイル内で複数のブロックが破損している場合、DETAIL オプションを指定すると、各ブロックの破損が表示されます。次の例では、障害 101 に関する詳細情報を表示します。

```
RMAN> LIST FAILURE 101 DETAIL;
```

```
List of Database Failures
```

```
=====
```

```
Failure ID Priority Status    Time Detected Summary
-----
101          HIGH     OPEN      23-APR-07   Datafile 1:
'/disk1/oradata/prod/system01.dbf' contains one or more corrupt blocks
List of child failures for parent failure ID 101
Failure ID Priority Status    Time Detected Summary
-----
104          HIGH     OPEN      23-APR-07   Block 56416 in datafile 1:
'/disk1/oradata/prod/system01.dbf' is media corrupt
Impact: Object BLKTEST owned by SYS might be unavailable
```

- 14-10 ページの「[修復オプションの決定](#)」に進み、LIST FAILURE コマンドによって表示された障害の修復方法を決定します。

## 障害のサブセットの表示

LIST FAILURE では、さらに詳しい出力を表示できるのみでなく、出力を制限することもできます。たとえば、CRITICAL、HIGH、LOW または CLOSED オプションを使用して LIST FAILURE を実行すると、特定のステータスまたは優先順位を持つ障害のみを表示できます。また、EXCLUDE FAILURE を指定して、指定した障害を出力から除外することもできます。

### 障害のサブセットを表示する手順

- Recovery Manager を起動し、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・インスタンスが起動されている必要があります。
- 必要なオプションを指定して LIST FAILURE を実行します。

次の例は、いくつかの LIST FAILURE コマンドを示しています。

```
LIST FAILURE LOW;
LIST FAILURE CLOSED;
LIST FAILURE EXCLUDE FAILURE 234234;
```

**参照：** LIST FAILURE コマンドについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。



## データベースの検証によるブロック破損の確認

14-3 ページの「データ整合性チェック」で説明されているように、破損したデータへのアクセスをユーザーのトランザクションで試行すると、データベースはそれに対応してデータ整合性チェックを実行します。遅延障害は検出されない場合もあります。たとえば、データ・ブロック破損エラーが発生すると、データベースは、エラーが発生したブロックおよびそのブロックに隣接している他のブロックを検証する事後対応型データ整合性チェックを実行します。ただし、エラーが発生したブロックに隣接していないブロックが破損している場合もあります。また、データベースによって読み取られない破損したブロックは、事後対応型データ整合性チェックでは検出されません。

事前対応型データ整合性チェックを効果的に実行する方法の1つとして、Recovery Manager で VALIDATE または BACKUP VALIDATE コマンドを実行する方法があります。これらのコマンドによって、データファイルおよび制御ファイルに物理的な破損および論理的な破損がないかどうかを確認できます。Recovery Manager によってブロック破損が検出された場合、自動診断リポジトリにログが記録され、1つ以上の障害が作成されます。その後、データ・リカバリ・アドバイザーを使用すると、これらの障害に関する情報を表示して、これらの障害を修復することができます。

### データベースを検証する手順

1. Recovery Manager を起動し、ターゲット・データベースに接続します。ターゲット・データベースはマウントされている必要があります。
2. 目的のデータベース・ファイルを検証します。

次の例では、VALIDATE DATABASE を使用して、データベース全体の物理的および論理的な破損を確認します（出力例も示します）。14-7 ページの「障害の表示」には一部のデータファイルが消失していることが示されているため、SKIP INACCESSIBLE 句が指定されています。出力には、system01.dbf データベース・ファイルに1つの新しい破損ブロック (Blocks Failing) があり、以前にデータベースによって破損とマークされたブロック (Marked Corrupt) がないことが示されています。

```
RMAN> VALIDATE CHECK LOGICAL SKIP INACCESSIBLE DATABASE;
```

```
Starting validate at 23-APR-07
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=103 device type=DISK
could not access datafile 28
skipping inaccessible file 28
RMAN-06060: WARNING: skipping datafile compromises tablespace USERS recoverability
RMAN-06060: WARNING: skipping datafile compromises tablespace USERS recoverability
channel ORA_DISK_1: starting validation of datafile
channel ORA_DISK_1: specifying datafile(s) for validation
input datafile file number=00001 name=/disk1/oradata/prod/system01.dbf
input datafile file number=00002 name=/disk1/oradata/prod/sysaux01.dbf
input datafile file number=00022 name=/disk1/oradata/prod/undotbs01.dbf
input datafile file number=00023 name=/disk1/oradata/prod/cwmlite01.dbf
input datafile file number=00024 name=/disk1/oradata/prod/drsys01.dbf
input datafile file number=00025 name=/disk1/oradata/prod/example01.dbf
input datafile file number=00026 name=/disk1/oradata/prod/indx01.dbf
input datafile file number=00027 name=/disk1/oradata/prod/tools01.dbf
channel ORA_DISK_1: validation complete, elapsed time: 00:00:25
List of Datafiles
=====
File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
1 FAILED 0 3536 57600 637711
```

```

File Name: /disk1/oradata/prod/system01.dbf
Block Type Blocks Failing Blocks Processed
-----
Data      1          41876
Index     0          7721
Other     0          4467
.
.
.
File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
27  OK      0          1272          1280          400914
File Name: /disk1/oradata/prod/tools01.dbf
Block Type Blocks Failing Blocks Processed
-----
Data      0           0
Index     0           0
Other     0           8

validate found one or more corrupt blocks
See trace file /disk1/oracle/log/diag/rdbms/prod/prod/trace/prod_ora_2596.trc
for details
channel ORA_DISK_1: starting validation of datafile
channel ORA_DISK_1: specifying datafile(s) for validation
including current control file for validation
including current SPFILE in backup set
channel ORA_DISK_1: validation complete, elapsed time: 00:00:01
List of Control File and SPFILE
=====
File Type      Status Blocks Failing Blocks Examined
-----
SPFILE        OK      0           2
Control File OK      0          512
Finished validate at 23-APR-07

```

**参照：**

- [第 15 章「データベース・ファイルおよびバックアップの検証」](#)
- VALIDATE コマンドの詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- Oracle Database が診断データを管理する方法については、『Oracle Database 管理者ガイド』を参照してください。

## 修復オプションの決定

Recovery Manager セッションで LIST FAILURE を実行した後、ADVISE FAILURE コマンドを使用して修復オプションを表示します。このコマンドによって、障害のサマリーが出力され、すでに修復されているステータスが OPEN のすべての障害がクローズされます。

必要に応じて、ADVISE FAILURE コマンドによって手動および自動の修復オプションのリストが表示されます。最初に、必須またはオプションのいずれかとしてカテゴリ化された手動オプションが表示されます。オプションの手動の修復によって、データファイルのリストアやリカバリなどの大規模な処理を回避できる場合もあります。通常は、データベースに及ぶ影響が最も少なく、エラーが発生する可能性が最も少ない修復手法を使用します。



## すべての障害に対する修復オプションの決定

1 つ以上の障害が存在する場合は、通常、LIST FAILURE を使用して障害の情報を表示してから、同じ Recovery Manager セッションで ADVISE FAILURE を使用して修復オプションのレポートを取得します。

### すべての障害に対する修復オプションを決定する手順

- 14-7 ページの「すべての障害の表示」の説明に従って障害を表示します。
- 同じ Recovery Manager セッションで、ADVISE FAILURE を実行します。

次の例では、データ・リカバリ・アドバイザーで認識されるすべての障害に対する修復オプションを要求し、出力例（ページに収まるように形式が再設定されています）を示します。

```
RMAN> ADVISE FAILURE;
```

```
List of Database Failures
```

```
=====
```

| Failure ID | Priority | Status | Time Detected | Summary   |
|------------|----------|--------|---------------|---|
| 142        | HIGH     | OPEN   | 23-APR-07     | One or more non-system datafiles are missing  |
| 101        | HIGH     | OPEN   | 23-APR-07     | Datafile 1:<br>'/disk1/oradata/prod/system01.dbf' contains one or more corrupt blocks |

```
analyzing automatic repair options; this may take some time
using channel ORA_DISK_1
analyzing automatic repair options complete
```

```
Mandatory Manual Actions
```

```
=====
```

```
no manual actions available
```

```
Optional Manual Actions
```

```
=====
```

1. If file /disk1/oradata/prod/users01.dbf was unintentionally renamed or moved, restore it

```
Automated Repair Options
```

```
=====
```

```
Option Repair Description
```

```
-----
```

- 1 Restore and recover datafile 28; Perform block media recovery of block 56416 in file 1  
Strategy: The repair includes complete media recovery with no data loss  
Repair script: /disk1/oracle/log/diag/rdbms/prod/prod/hm/reco\_660500184.hm

前述の例では、ADVISE FAILURE によって、欠落しているデータファイルおよび破損ブロックが含まれているデータファイルの 2 つの障害がレポートされています。このコマンドでは必須の手動処理は表示されませんが、欠落しているデータファイルの名前が誤って変更されていないこと、または欠落しているデータファイルが誤って削除されていないことを確認するように指示されています。この修復オプションは、**ブロック・メディア・リカバリ**および欠落したデータファイルのリストアとリカバリに関連しています。ADVISE FAILURE では、修復スクリプトの場所が表示されます。

次の類似した例では、自動修復に必要な Recovery Manager バックアップまたはアーカイブ REDO ログが利用できない場合の出力を示します。ADVISE FAILURE で必須の手動処理が表示されるようになったことに注意してください。

```

RMAN> ADVISE FAILURE;

List of Database Failures
=====

Failure ID Priority Status    Time Detected Summary
-----
142          HIGH    OPEN      23-APR-07    One or more non-system datafiles
are missing
101          HIGH    OPEN      23-APR-07    Datafile 1:
'/disk1/oradata/prod/system01.dbf' contains one or more corrupt blocks

analyzing automatic repair options; this may take some time
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=103 device type=DISK
analyzing automatic repair options complete

Mandatory Manual Actions
=====
1. If file /disk1/oradata/prod/users01.dbf was unintentionally renamed or moved,
restore it
2. Contact Oracle Support Services if the preceding recommendations cannot be used,
or if they do not fix the failures selected for repair

Optional Manual Actions
=====
no manual actions available

Automated Repair Options
=====
Option Repair Description
-----
1          Perform block media recovery of block 56416 in file 1
Strategy: The repair includes complete media recovery with no data loss
Repair script: /disk1/oracle/log/diag/rdbms/prod/prod/hm/reco_1863891774.hm

3. 14-13 ページの「障害の修復」に進み、LIST FAILURE 出力によって表示された障害の修復方法を決定します。

```

## 障害のサブセットに対する修復オプションの決定

特定の障害に対する修復オプションを要求することもできます。障害は、ステータス (CRITICAL、HIGH、LOW) または障害番号で指定できます。EXCLUDE FAILURE を使用して、1 つ以上の障害をレポートから除外することもできます。

### 障害のサブセットに対する修復オプションを決定する手順

1. 14-7 ページの「すべての障害の表示」の説明に従って障害を表示します。
2. 同じ Recovery Manager セッションで、必要なオプションを指定して ADVISE FAILURE を実行します。

次の例では、障害 101 のみに対する修復オプションを要求します。

```
RMAN> ADVISE FAILURE 101;
```

```

List of Database Failures
=====

```

```

Failure ID Priority Status      Time Detected Summary
-----
101          HIGH    OPEN      23-APR-07   Datafile 1:
'/disk1/oradata/prod/system01.dbf' contains one or more corrupt blocks

analyzing automatic repair options; this may take some time
using channel ORA_DISK_1
analyzing automatic repair options complete

Mandatory Manual Actions
=====
no manual actions available

Optional Manual Actions
=====
no manual actions available

Automated Repair Options
=====
Option Repair Description
-----
1          Perform block media recovery of block 56416 in file 1
Strategy: The repair includes complete media recovery with no data loss
Repair script: /disk1/oracle/log/diag/rdbms/prod/prod/hm/reco_708819503.hm

```

- 14-13 ページの「障害の修復」に進み、LIST FAILURE コマンドによって表示された障害の修復方法を決定します。

**参照：** ADVISE FAILURE コマンドの詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## 障害の修復

この項では、データ・リカバリ・アドバイザーを使用して、障害を自動的に修復する方法について説明します。

### 障害の修復の概要

ADVISE FAILURE で手動の修復が提示された場合は、これらの修復を最初に試行します。手動の修復が不可能でない場合、または手動の修復ではすべての障害が修復されない場合は、REPAIR FAILURE を使用して、現行の Recovery Manager セッションでの最新の ADVISE FAILURE コマンドで提示された障害の修復を自動的に実行することができます。

デフォルトでは、REPAIR FAILURE の実行を開始する前に確認を求められます。NOPROMPT オプションを指定して、確認のプロンプトが表示されないようにすることができます。コマンドの実行が開始されると、現行の修復フェーズが示されます。状況に応じて、Recovery Manager は、ユーザーに操作を求める場合があります。修復を実行した後、Recovery Manager は、この修復中に修復された可能性がある既存のすべての障害を再評価します。

通常は、修復を実行する前に、PREVIEW オプションを指定して修復をプレビューすることをお勧めします。Recovery Manager は、修復を行わず、すべての修復操作およびコメントが含まれているスクリプトを生成します。特定の修復オプションを指定しなかった場合、Recovery Manager は、現行セッションの最新の ADVISE FAILURE コマンドの最初の修復オプションを使用します。デフォルトでは、修復スクリプトは標準出力に表示されます。SPOOL コマンドを使用すると、編集可能なファイルにスクリプトを書き込むことができます。

**参照：**

- REPAIR FAILURE コマンドについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- SPOOL コマンドの詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## 障害の修復

デフォルトでは、スクリプトは標準出力に表示されます。SPOOL コマンドを使用すると、編集可能なファイルにスクリプトを書き込むことができます。

### 障害を修復する手順

1. 14-7 ページの「すべての障害の表示」の説明に従って障害を表示します。
2. 14-10 ページの「修復オプションの決定」の説明に従って修復オプションを表示します。
3. 必要に応じて、REPAIR FAILURE PREVIEW を実行します。

次の例では、Recovery Manager セッションの以前の ADVISE FAILURE コマンドによって表示された最初の修復オプションをプレビューします。

```
RMAN> REPAIR FAILURE PREVIEW;
```

```
Strategy: The repair includes complete media recovery with no data loss  
Repair script: /disk1/oracle/log/diag/rdbms/prod/prod/hm/reco_475549922.hm  
contents of repair script:
```

```
# restore and recover datafile  
sql 'alter database datafile 28 offline';  
restore datafile 28;  
recover datafile 28;  
sql 'alter database datafile 28 online';  
# block media recovery  
recover datafile 1 block 56416;
```

4. REPAIR FAILURE を実行します。

次の修復では、1つのデータファイルをリストアおよびリカバリし、1つの破損ブロックでブロック・メディア・リカバリを実行します。Recovery Manager によって、修復を行うかどうかの確認が求められます。ユーザーが入力したテキストは太字で示されています。

```
RMAN> REPAIR FAILURE;
```

```
Strategy: The repair includes complete media recovery with no data loss  
Repair script: /disk1/oracle/log/diag/rdbms/prod/prod/hm/reco_475549922.hm  
contents of repair script:
```

```
# restore and recover datafile  
sql 'alter database datafile 28 offline';  
restore datafile 28;  
recover datafile 28;  
sql 'alter database datafile 28 online';  
# block media recovery  
recover datafile 1 block 56416;
```

```
Do you really want to execute the above repair (enter YES or NO)? YES  
executing repair script
```

```
sql statement: alter database datafile 28 offline
```

```
Starting restore at 23-APR-07  
using channel ORA_DISK_1
```

```
channel ORA_DISK_1: starting datafile backup set restore
```

```
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
channel ORA_DISK_1: restoring datafile 00028 to /disk1/oradata/prod/users01.dbf
channel ORA_DISK_1: reading from backup piece /disk2/PROD/backupset/2007_04_18/o1_
mf_nnndf_TAG20070418T182042_32fjzd3z_.bkp
channel ORA_DISK_1: piece handle=/disk2/PROD/backupset/2007_04_18/o1_mf_nnndf_
TAG20070418T182042_32fjzd3z_.bkp tag=TAG20070418T182042
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:03
Finished restore at 23-APR-07
```

```
Starting recover at 23-APR-07
using channel ORA_DISK_1
```

```
starting media recovery
media recovery complete, elapsed time: 00:00:01
```

```
Finished recover at 23-APR-07
```

```
sql statement: alter database datafile 28 online
```

```
Starting recover at 23-APR-07
using channel ORA_DISK_1
searching flashback logs for block images until SCN 429690
finished flashback log search, restored 1 blocks
```

```
starting media recovery
media recovery complete, elapsed time: 00:00:03
```

```
Finished recover at 23-APR-07
repair failure complete
```

5. 必要に応じて、LIST FAILURE を実行して確認します。

## 障害のステータスおよび優先順位の変更

CHANGE FAILURE コマンドを使用して、障害のステータスまたは優先順位を変更する必要がある場合もあります。たとえば、ブロックの破損の優先順位が HIGH で、そのブロックがほとんど使用されない表領域に存在する場合は、そのブロックの破損の優先順位を一時的に LOW に変更できます。

REPAIR FAILURE コマンド以外の方法で障害を修復すると、次回 LIST FAILURE を実行した場合、データ・リカバリ・アドバイザーはこの障害を暗黙的にクローズします。このため、通常は CHANGE FAILURE ... CLOSED コマンドを実行する必要はありません。このコマンドは、自動障害再検証が正常に実行されなかったにもかかわらず、障害は存在していないと確信できる場合にのみ使用する必要があります。CHANGE FAILURE を使用して、存在したままになっている障害をクローズすると、データ・リカバリ・アドバイザーは、適切なデータ整合性チェックを実行する際に別の障害 ID でこの障害を再作成します。

通常、変更する障害は障害番号で指定します。ALL、CRITICAL、HIGH または LOW を指定して、障害を一括で変更することもできます。障害は、CLOSED、PRIORITY HIGH または PRIORITY LOW に変更できます。

### 障害のステータスまたは優先順位を変更する手順

1. 14-7 ページの「すべての障害の表示」の説明に従って障害を表示します。

次の例は、破損データ・ブロックが含まれている 1 つの障害を示しています。

```
RMAN> LIST FAILURE;

List of Database Failures
=====
```

```

Failure ID Priority Status      Time Detected Summary
-----
142          HIGH      OPEN      23-APR-07  One or more non-system datafiles
are missing
101          HIGH      OPEN      23-APR-07  Datafile 25:
'/disk1/oradata/prod/example01.dbf' contains one or more corrupt blocks

```

- 必要なオプションを使用して CHANGE FAILURE を実行します。

次の例では、ブロックの破損障害の優先順位を HIGH から LOW に変更します。

```

RMAN> CHANGE FAILURE 101 PRIORITY LOW;

```

```

List of Database Failures
=====

```

```

Failure ID Priority Status      Time Detected Summary
-----
101          HIGH      OPEN      23-APR-07  Datafile 25:
'/disk1/oradata/prod/example01.dbf' contains one or more corrupt blocks

```

```

Do you really want to change the above failures (enter YES or NO)? YES
changed 1 failures to LOW priority

```

- 必要に応じて、LIST FAILURE ALL を実行して変更内容を表示します。

ALL を指定せずに LIST FAILURE を実行すると、CRITICAL または HIGH の優先順位の障害が存在しない場合にのみ、LOW の優先順位の障害が表示されます。

```

RMAN> LIST FAILURE ALL;

```

```

List of Database Failures
=====

```

```

Failure ID Priority Status      Time Detected Summary
-----
142          HIGH      OPEN      23-APR-07  One or more non-system datafiles
are missing
101          LOW       OPEN      23-APR-07  Datafile 25:
'/disk1/oradata/prod/example01.dbf' contains one or more corrupt blocks

```

**参照：** CHANGE コマンドの詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

---

## データベース・ファイルおよびバックアップの 検証

この章では、データベース・ファイルおよびバックアップの整合性をチェックする方法について説明します。この章の内容は、次のとおりです。

- [Recovery Manager の検証の概要](#)
- [VALIDATE コマンドによるブロック破損の確認](#)
- [BACKUP VALIDATE を使用したデータベース・ファイルの検証](#)
- [リストアする前のバックアップの検証](#)

## Recovery Manager の検証の概要

この項では、Recovery Manager で検証を行う場合の基本的な概念およびタスクについて説明します。

### Recovery Manager の検証の目的

Recovery Manager の**検証**は、ブロックが破損していないか、およびファイルが欠落していないかどうかを確認することを主に目的としています。また、Recovery Manager を使用して、バックアップがリストア可能かどうかを確認することもできます。Recovery Manager の次のコマンドを使用すると、検証を実行できます。

- VALIDATE
- BACKUP ...VALIDATE
- RESTORE ...VALIDATE

#### 参照：

- VALIDATE 構文の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- RESTORE ... VALIDATE 構文の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

### Recovery Manager の検証の基本的な概念

バックアップ・ファイルが使用できなくなったり、リストアされたデータファイルが破損する可能性がある操作は防止されます。データベースは、自動的に次の処理を実行します。

- データファイルのリストアまたはリカバリ中、そのファイルへのアクセスをブロックします。
- 各データファイルのリストア操作を、一度に1つのみ許可します。
- 増分バックアップが適切な順序で適用されることを確認します。
- バックアップ・ファイルに情報を格納して、破損の検出を可能にします。
- 破損を検出するとすぐに通知できるように、ブロックが読取りまたは書き込みされるたびにブロックを確認します。

### チェックサムおよび破損ブロック

**破損ブロック**とは、変更が行われたため、Oracle Database での検出時に予測される内容とは異なる内容になっているブロックのことです。ブロックの破損は、次に示す様々な障害によって発生する可能性があります。ただし、原因はこれらに限定されるわけではありません。

- ディスクおよびディスク・コントローラの障害
- メモリー障害
- Oracle Database ソフトウェアの不具合

DB\_BLOCK\_CHECKSUM は、(バックアップ内ではなく) データベース内のデータファイルおよびオンライン REDO ログ・ファイルのブロックに対するチェックサムの書き込みを制御するデータベース初期化パラメータです。DB\_BLOCK\_CHECKSUM が typical に設定されている場合、データベースは、通常の操作中に各ブロックのチェックサムを計算し、ヘッダーにその値を格納してからブロックをディスクに書き込みます。データベースは、後でディスクからブロックを読み取る場合、チェックサムを再計算し、格納されている値と比較します。値が一致しなかった場合、そのブロックは破損しています。

デフォルトでは、BACKUP コマンドは、各ブロックのチェックサムを計算し、その値をバックアップに格納します。DB\_BLOCK\_CHECKSUM は、バックアップ内ではなくデータベース内のデータファイルに適用される初期化パラメータであるため、BACKUP コマンドではこの値は無視されます。



## 物理的および論理的なブロック破損

**物理的な破損**（メディア破損とも呼ばれます）の場合、ブロックがデータベースで認識されません。**チェックサム**が無効か、ブロック内容がすべて 0（ゼロ）か、またはブロックのヘッダーとフッターが一致していません。

---

**注意：** デフォルトでは、BACKUP コマンドは、各ブロックのチェックサムを計算し、その値をバックアップに格納します。NOCHECKSUM オプションを指定すると、バックアップの作成時に Recovery Manager はブロックのチェックサムを実行しません。

---

**論理的な破損**の場合、ブロックの内容は論理的に一貫性のない状態になっています。論理的な破損の例としては、行ピースまたは索引エントリの破損などがあります。Recovery Manager は、論理的な破損を検出すると、アラート・ログおよびサーバー・セッションのトレース・ファイルにそのブロックを記録します。

デフォルトでは、Recovery Manager は、論理的な破損を確認しません。ただし、BACKUP コマンドで CHECK LOGICAL を指定すると、Recovery Manager は、データおよび索引ブロックをテストして、行ピースまたは索引エントリの破損などの論理的な破損がないかどうかを調べ、その結果を**自動診断リポジトリ**内のアラート・ログに記録します。ファイルのバックアップ時またはリストア時に、Recovery Manager を次の構成で使用すると、Recovery Manager は、検出可能なすべてのタイプのブロック破損を検出します。

- データベースがデータファイルのチェックサムを（バックアップに対してではなく、データベースで使用中のデータファイルに対して）自動的に計算するように、データベースの初期化パラメータ・ファイルで DB\_BLOCK\_CHECKSUM=typical を設定する。
- Recovery Manager がブロック破損を許容しないように、BACKUP または RESTORE コマンドの前に SET MAXCORRUPT を使用しない。
- バックアップの書き込み時に Recovery Manager がチェックサムを計算するように、BACKUP コマンドに NOCHECKSUM オプションを指定しない。
- Recovery Manager が物理的な破損とともに論理的な破損も確認するように、BACKUP コマンドおよび RESTORE コマンドに CHECK LOGICAL オプションを指定する。

## Recovery Manager バックアップの破損ブロックの制限

SET MAXCORRUPT コマンドを使用すると、1 つのファイルで Recovery Manager バックアップに許容される破損の合計数を設定できます。デフォルトは 0（ゼロ）です。つまり、Recovery Manager はいずれの種類の破損ブロックも許容しません。

バックアップ中に破損ブロックが検出された際に MAXCORRUPT の制限を超えていると、Recovery Manager はバックアップを終了します。これ以外の場合、Recovery Manager は、ブロックが破損とマークされていることを示す特別なヘッダーを付けて、破損ブロックをバックアップに書き込みます。VALIDATE コマンドを使用すると、破損とマークされているブロックを特定できます。

Recovery Manager は、バックアップ内のブロック破損を容認できるため、ブロック破損が含まれていると認識されるデータファイルをリストアできます。リストアしたデータファイルをバックアップする場合、Recovery Manager は、MAXCORRUPT を超えているかどうかを計算するときに、すでに破損のマークが付けられているブロックは考慮しません。

**参照：** SET MAXCORRUPT 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## ブロック破損の検出

Oracle Database では、ブロックの破損を検出、修復および監視する様々な方法がサポートされています。方法は、破損が**ブロック間の破損**か**ブロック内の破損**かによって異なります。ブロック内の破損は、ブロック自体内で発生します。この破損は、物理的な破損または論理的な破損のいずれかです。ブロック間の破損の場合、論理的な破損のみがブロック間で発生します。

たとえば、V\$DATABASE\_BLOCK\_CORRUPTION ビューではブロック内の破損が記録され、**自動診断リポジトリ**ではすべてのタイプの破損が追跡されます。表 15-1 に、データベースで様々なタイプのブロック破損を処理する方法を示します。

表 15-1 ブロックの破損の検出、修復および監視

| 対策 | ブロック内の破損   | ブロック間の破損  |
|----|--|---|
| 検出 | Recovery Manager (BACKUP コマンドなど)、DBVERIFY ユーティリティなどのすべてのデータベース・ユーティリティによって、ブロック内の破損が検出されます。データベース・プロセスで ORA-1578 エラーが発生すると、破損が検出され、監視されます。   | DBVERIFY および ANALYZE 文でのみ、ブロック間の破損が検出されます。        |
| 追跡 | V\$DATABASE_BLOCK_CORRUPTION ビューに、Recovery Manager コマンド、ANALYZE、dbv、SQL 問合せなどの Oracle Database コンポーネントによって破損とマークされたブロックが表示されます。ブロック内の破損を検出したプロセスでは、このビューおよび ADR にブロック破損が記録されます。  | データベースは、ADR でこのタイプのブロック破損を監視します。                  |
| 修復 | 修復方法には、 <b>ブロック・メディア・リカバリ</b> 、データファイルのリストア、増分バックアップによるリカバリおよびブロックの新規作成などがあります。ブロック・メディア・リカバリでは、物理的な破損は修復できますが、論理的な破損は修復できません。<br><br>修復を行うかブロックが修復されたことを検出するすべての Recovery Manager コマンドによって、V\$DATABASE_BLOCK_CORRUPTION が更新されます。たとえば、Recovery Manager は、ブロック・メディア・リカバリが正常に実行された後にリポジトリを更新します。BACKUP、RESTORE または VALIDATE コマンドによって、ブロックの破損がなくなったことが検出された場合、修正済のブロックがビューから削除されます。 | ブロック間の破損は、オブジェクトの削除、索引の再構築などの手動による方法で修正する必要があります。 |

**参照：**

- 第 17 章「データベースの完全リカバリの実行」
- 第 18 章「ブロック・メディア・リカバリの実行」
- ADR については、『Oracle Database 管理者ガイド』を参照してください。

## VALIDATE コマンドによるブロック破損の確認

VALIDATE コマンドを使用すると、データベース・ファイル内の物理的および論理的な破損を手動で確認できます。このコマンドでは、BACKUP VALIDATE と同じタイプのチェックが実行されますが、VALIDATE では、より広範囲のオブジェクトをチェックできます。たとえば、VALIDATE DATAFILE ... BLOCK コマンドを使用すると、個々のブロックを検証できます。

すべてのファイルを検証する場合、Recovery Manager によって、入力ファイルのすべてのブロックが確認されます。バックアップの検証によって破損ブロックが検出された後、Recovery Manager は、破損を示している行を反映して、V\$DATABASE\_BLOCK\_CORRUPTION ビューを更新します。

バックアップ・セット内の 1 つ以上のバックアップ・ピースが欠落または破損している可能性がある場合は、VALIDATE BACKUPSET を使用します。このコマンドでは、バックアップ・セット内のすべてのブロックがチェックされ、バックアップがリストア可能かどうかを確認されます。ブロックの破損が検出されると、エラーが発行されて検証が停止されます。VALIDATE BACKUPSET では、ユーザーがチェック対象のバックアップを選択できます。これに対して、RESTORE コマンドの VALIDATE オプションでは、Recovery Manager によって選択が行われます。

**VALIDATE を使用してデータベース・ファイルおよびバックアップを確認する手順**

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. 必要なオプションを指定して VALIDATE コマンドを実行します。

たとえば、データファイル、制御ファイルおよび（使用されている場合は）サーバー・パラメータ・ファイルをすべて検証するには、次のコマンドを Recovery Manager のプロンプトで実行します。

```
RMAN> VALIDATE DATABASE;
```

また、次の例に示すコマンド形式を使用すると、特定のバックアップ・セットを検証することもできます（出力例も示します）。

```
RMAN> VALIDATE BACKUPSET 22;
```

```
Starting validate at 17-AUG-06
using channel ORA_DISK_1
allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=89 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: Oracle Secure Backup
channel ORA_DISK_1: starting validation of datafile backup set
channel ORA_DISK_1: reading from backup piece
/disk1/oracle/work/orcva/RDBMS/backupset/2007_08_16/o1_mf_nnndf_TAG20070816T153034_
2g774bt2_.bkp
channel ORA_DISK_1: piece
handle=/disk1/oracle/work/orcva/RDBMS/backupset/2007_08_16/o1_mf_nnndf_TAG20070816T
153034_2g774bt2_.bkp tag=TAG20070816T153034
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: validation complete, elapsed time: 00:00:01
Finished validate at 17-AUG-06
```

次の例に、データファイル内の個々のデータ・ブロックを調べて破損を確認する方法を示します。

```
RMAN> VALIDATE DATAFILE 1 BLOCK 10;
```

```
Starting validate at 17-AUG-06
using channel ORA_DISK_1
channel ORA_DISK_1: starting validation of datafile
channel ORA_DISK_1: specifying datafile(s) for validation
input datafile file number=00001 name=/disk1/oracle/dbs/tbs_01.f
channel ORA_DISK_1: validation complete, elapsed time: 00:00:01
List of Datafiles
=====
File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
1    OK      0          2          127          481907
File Name: /disk1/oracle/dbs/tbs_01.f
Block Type Blocks Failing Blocks Processed
-----
Data      0          36
Index     0          31
Other     0          58

Finished validate at 17-AUG-06
```

## データファイルの検証の平行化

大規模なデータファイルを検証する必要がある場合、Recovery Manager は、そのファイルを複数のセクションに分割し、各 **ファイル・セクション** を平行に処理することによって、作業を平行化できます。複数のチャンネルが構成されているか、または割り当てられている場合にチャンネルで検証を平行化するには、VALIDATE コマンドの SECTION SIZE パラメータを指定します。

ファイルのサイズより大きいセクション・サイズを指定した場合、Recovery Manager はファイル・セクションを作成しません。小さなセクション・サイズを指定した結果、セクションの数が 256 を超えると、Recovery Manager は、正確に 256 になる値までセクション・サイズを増やします。

### データファイルの検証を平行化する手順

1. Recovery Manager を起動し、ターゲット・データベースに接続します。ターゲット・データベースは、マウントまたはオープンされている必要があります。
2. SECTION SIZE パラメータを指定して VALIDATE を実行します。

次の例では、2 つのチャンネルを割り当て、大規模なデータファイルを検証します。セクション・サイズは 1200MB です。

```
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE DISK;
  ALLOCATE CHANNEL c2 DEVICE TYPE DISK;
  VALIDATE DATAFILE 1 SECTION SIZE 1200M;
}
```

#### 参照:

- 「セクションへの大規模なデータファイルのバックアップの分割」 (9-3 ページ)
- VALIDATE コマンドの詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## BACKUP VALIDATE を使用したデータベース・ファイルの検証

BACKUP VALIDATE コマンドを使用すると、次の操作を実行できます。

- データファイルのブロックの物理的および論理的な破損の確認
- すべてのデータベース・ファイルが存在するかどうか、および格納場所が正しいかどうかの確認

BACKUP VALIDATE を実行すると、Recovery Manager は、実際のバックアップ時と同様に、バックアップするファイル全体を読み取ります。ただし、実際には、バックアップ・セットもイメージ・コピーも作成しません。

BACKUPSET パラメータ、MAXCORRUPT パラメータまたは PROXY パラメータは、BACKUP VALIDATE を指定して使用することはできません。特定のバックアップ・セットを検証するには、VALIDATE コマンドを実行します。

### BACKUP VALIDATE コマンドを使用してファイルを検証する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ（使用している場合）に接続します。

## 2. BACKUP VALIDATE コマンドを実行します。

たとえば、次の例のようにコマンドを実行すると、すべてのデータベース・ファイルおよびアーカイブ・ログがバックアップ可能かどうかを検証できます。このコマンドは物理的な破損のみ確認します。

```
BACKUP VALIDATE
  DATABASE
  ARCHIVELOG ALL;
```

物理的な破損に加えて論理的な破損も確認するには、前述のコマンドを次のように少し変更して実行します。

```
BACKUP VALIDATE
  CHECK LOGICAL
  DATABASE
  ARCHIVELOG ALL;
```

前述の例では、**Recovery Manager** クライアントは、実際にファイルをバックアップしたときと同じ出力を表示します。**Recovery Manager** は、ファイルを1つもバックアップできない場合、エラー・メッセージを発行します。たとえば、次のように表示されます。

```
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of backup command at 08/29/2007 14:33:47
ORA-19625: error identifying file /oracle/oradata/trgt/arch/archive1_6.dbf
ORA-27037: unable to obtain file status
SVR4 Error: 2: No such file or directory
Additional information: 3
```

### 参照:

- BACKUP 構文の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- BACKUP VALIDATE コマンドで検出される破損ブロックの修復の詳細は、第18章「ブロック・メディア・リカバリの実行」を参照してください。

## リストアする前のバックアップの検証

RESTORE ... VALIDATE を実行すると、**Recovery Manager** で特定のファイルまたはファイル・セットをバックアップからリストアできるかどうかテストされます。**Recovery Manager** によって、使用するバックアップが選択されます。

このコマンドでは、データベースがマウントまたはオープンされている必要があります。データファイルのリストアの検証時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。データファイルのバックアップの検証は、バックアップを読み取るのみで、本番データファイルには影響しないためです。

ディスクまたはテープのファイルを検証する場合、**Recovery Manager** は、バックアップ・ピースまたはイメージ・コピー内のすべてのブロックを読み取ります。また、**Recovery Manager** は、オフサイトのバックアップの検証も行います。この検証は、**Recovery Manager** によって出力ファイルが書き込まれないことを除き、実際のリストア操作と同じです。

---

**注意:** RECOVER ... TEST コマンドを使用して、**試行リカバリ**を実行するテスト方法もあります。試行リカバリでは、通常のリカバリと同じ方法で REDO が適用されますが、これはメモリー内のみで行われ、変更は試行後にロールバックされます。

---

### RESTORE コマンドを使用してバックアップを検証する手順

1. VALIDATE オプションを指定して RESTORE コマンドを実行します。

次に、データベースおよびすべてのアーカイブ REDO ログのリストアの検証例を示します。

```
RESTORE DATABASE VALIDATE;  
RESTORE ARCHIVELOG ALL VALIDATE;
```

Recovery Manager のエラー・スタックが表示されない場合は、後続の手順をスキップします。エラー・メッセージが表示されないということは、実際のリストアおよびリカバリ中にこれらのバックアップを正常に使用できることが Recovery Manager で確認されたということを示しています。

2. 出力内にエラー・メッセージおよび RMAN-06026 メッセージが表示された場合は、問題の原因を調査します。可能な場合は、Recovery Manager によるバックアップの検証を妨げている問題を解決し、検証を再実行します。

次のエラーは、Recovery Manager が、指定した 1 つ以上のファイルを使用可能なバックアップからリストアできないことを意味します。

```
RMAN-06026: some targets not found - aborting restore
```

次の出力例は、指定したバックアップの読取りで Recovery Manager が問題を検出したことを示しています。

```
RMAN-03009: failure of restore command on c1 channel at 12-DEC-06 23:22:30  
ORA-19505: failed to identify file "oracle/dbs/1fafv9gl_1_1"  
ORA-27037: unable to obtain file status  
SVR4 Error: 2: No such file or directory  
Additional information: 3
```

**参照:** RESTORE ... VALIDATE コマンドの詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

---

## フラッシュバックおよびデータベースの Point-in-Time リカバリの実行

この章では、不要なデータベース変更について調べ、Oracle のフラッシュバック技術およびデータベース・バックアップに基づいた適切なリカバリ計画を選択して実行する方法について説明します。この章では、次の項目について説明します。

- [フラッシュバック技術およびデータベースの Point-in-Time リカバリの概要](#)
- [フラッシュバック表を使用した表の巻戻し](#)
- [フラッシュバック・ドロップを使用した DROP TABLE 操作の巻戻し](#)
- [フラッシュバック・データベースを使用したデータベースの巻戻し](#)
- [データベースの Point-in-Time リカバリの実行](#)
- [フラッシュバックおよびデータベースの Point-in-Time リカバリの例](#)

# フラッシュバック技術およびデータベースの Point-in-Time リカバリの概要

この項では、フラッシュバック技術およびデータベースの Point-in-Time リカバリの目的および基本的な概念について説明します。

## フラッシュバックおよびデータベースの Point-in-Time リカバリの目的

通常、フラッシュバック機能または Point-in-Time リカバリは、次のような場合に実行します。

- ユーザー・エラーまたは破損によって、必要なデータが削除されるか、またはデータが破損された場合。たとえば、ユーザーまたは DBA が誤って 1 つ以上の表の内容を削除または更新したり、アプリケーションの更新中にまだ必要なデータベース・オブジェクトを削除したり、大規模なバッチ更新を実行して途中で障害が発生した場合などです。
- データベースのアップグレードに失敗するか、アップグレード・スクリプトが消失した場合。
- すべての必要な REDO ログまたは増分バックアップがないため、メディア障害後の完全なデータベース・リカバリを正常に実行できない場合。

いずれの場合も、Point-in-Time リカバリまたはフラッシュバック機能を使用して、データベースまたはデータベース・オブジェクトを前の時点の状態に戻すことができます。

## Point-in-Time リカバリおよびフラッシュバック機能の基本的な概念

不要なデータベース変更に対する最も基本的な解決方法は、Recovery Manager の **データベースの Point-in-Time リカバリ** です。DBPITR は、使用可能なすべての REDO を使用することも、データベースに対するすべての変更をリカバリすることもないため、**不完全リカバリ**とも呼ばれます。この場合は、データベース全体のバックアップをリストアしてから、REDO ログまたは増分バックアップを適用して、不要な変更の前の時点までのすべての変更を再作成します。

不要なデータベース変更が大規模ではあるが、特定の表領域に制限されている場合は、**表領域の Point-in-Time リカバリ**を使用して、影響を受けない表領域を使用可能な状態にしたまま、影響を受けている表領域を以前の SCN に戻すことができます。TSPITR は高度な手法です。詳細は、**第 20 章「Recovery Manager の表領域の Point-in-Time リカバリ (TSPITR) の実行」**を参照してください。

また、Oracle Database には、フラッシュバック技術と呼ばれる一連の機能も備えられています。この技術によって、バックアップからデータベースをリストアせずに、データの過去の状態を表示したり、データを任意の時点に戻したり、進めることができます。データベースに対する変更によっては、フラッシュバック技術で、データベースの可用性に与える影響を抑えてより迅速に不要な変更を無効にすることができます。

### データベースの Point-in-Time リカバリの基本的な概念

DBPITR は、物理レベルで動作し、データファイルを過去の目標時点の状態に戻します。Recovery Manager の DBPITR 操作では、目的の SCN、ログ順序、リストア・ポイントまたは時刻を指定します。Recovery Manager は、目標時点以前に作成されたバックアップからデータベースをリストアしてから、増分バックアップおよびログを適用して、データファイルをバックアップした時点とリカバリの終了時点の間のすべての変更を再作成します。終了時点が SCN として指定されている場合、データベースは、REDO ログを適用し、各 REDO スレッドの終了時点または指定した SCN のうちの早い方の時点で停止します。終了時点が時刻として指定されている場合、データベースは、指定した時刻に適した SCN を内部的に判別した後、この SCN までリカバリします。

バックアップ計画が適切に設計され、データベースが ARCHIVELOG モードで実行されている場合は、ほぼすべての状況で DBPITR を実行できます。28-14 ページの「**データベースの不完全リカバリの実行**」で説明されているように、Recovery Manager の DBPITR は、ユーザー管理の DBPITR と比較すると簡略化されています。目的の SCN を指定すると、データファイルがバックアップからリストアされ、ユーザーの介入なしで効率的にリカバリが行われます。ただし、Recovery Manager の DBPITR には、次のデメリットがあります。



- 以前の状態に戻すことができるのはデータベース全体のみであり、選択したオブジェクトを以前の状態に戻すことはできません。
- DBPITR の実行中、データベース全体が使用不可になります。
- Recovery Manager はすべてのデータファイルをリストアする必要があるため、DBPITR に時間がかかる可能性があります。また、Recovery Manager は、データファイルをリカバリするために REDO ログおよび増分バックアップをリストアする必要がある場合もあります。バックアップがテープ上にある場合は、このプロセスにさらに時間がかかる可能性があります。

## Oracle のフラッシュバック技術の基本的な概念

Oracle のフラッシュバック機能が使用可能なほとんどの状況では、フラッシュバック機能の方がメディア・リカバリより効率的です。フラッシュバック機能を使用すると、データベースの過去の状態を調べることができます。

**バックアップおよびリカバリで役立つ物理フラッシュバック機能** 16-11 ページの「[フラッシュバック・データベースを使用したデータベースの巻戻し](#)」で説明されているように、DBPITR の最も効率的な代替機能として、**Oracle Flashback Database** があります。他のフラッシュバック機能とは異なり、この機能は物理レベルで動作して、現行のデータファイルが過去の内容に戻されます。結果は、DBPITR (OPEN RESETLOGS を含む) の結果に類似していますが、フラッシュバック・データベースではデータファイルをリストアする必要はなく、メディア・リカバリと比較して REDO の適用が限られているため、通常、はるかに短時間でリカバリが行われます。

5-14 ページの「[フラッシュ・リカバリ領域の構成](#)」で説明するとおり、フラッシュバック・データベースには[フラッシュ・リカバリ領域](#)が必要です。Oracle Flashback Database のロギングを有効にするには、DB\_FLASHBACK\_RETENTION\_TARGET 初期化パラメータを設定し、ALTER DATABASE FLASHBACK ON 文を発行する必要があります。

通常の操作中、データベースによって、[フラッシュバック・ログ](#)にデータファイル・ブロックの古いイメージが定期的書き込まれます。フラッシュバック・ログは順次書き込まれ、バルク書込みが行われることも多くあります。フラッシュバック・ロギングは、継続バックアップに類似しています。データベースでは、リカバリ領域に対するフラッシュバック・ログの書込み、削除およびサイズ変更が自動的に行われます。フラッシュバック・ログはアーカイブされません。フラッシュバック・ログは、パフォーマンスを監視したり、フラッシュ・リカバリ領域に対するディスク領域の割当てを決定する場合にのみ確認する必要があります。

フラッシュバック・データベース操作を行う際に、データベースは、過去のバージョンのデータ・ブロックへのアクセスにフラッシュバック・ログを使用し、アーカイブ REDO ログの一部のデータも使用します。したがって、障害が検出された後はフラッシュバック・データベースを有効にできなくなるため、フラッシュバック・データベースを使用してこの障害発生の前まで巻き戻します。保証付きリストア・ポイントの関連機能を使用すると、危険を伴うデータベースの変更の直前などの特定の時点のデータベースの内容を保護できます。

**バックアップおよびリカバリで役立つ論理フラッシュバック機能** 残りのフラッシュバック機能は、論理レベルで動作します。この章で説明する論理機能は、次のとおりです。

- Oracle Flashback Table

データベースのいずれの部分もオフラインにすることなく、表または表のセットを過去の指定した時点にリカバリできます。ほとんどの場合、フラッシュバック表を使用すると、より複雑な Point-in-Time リカバリ操作を行う必要がなくなります。フラッシュバック表では、表がリストアされ、同時に現在の索引、トリガーおよび制約などの関連する属性が自動的にメンテナンスされるため、ユーザーはアプリケーション固有のプロパティを検索してリストアする必要がありません。

この機能の使用方法については、16-4 ページの「[フラッシュバック表を使用した表の巻戻し](#)」を参照してください。

### ■ Oracle Flashback Drop

DROP TABLE 文の結果を無効にできます。

この機能の使用方法については、16-7 ページの「[フラッシュバック・ドロップを使用した DROP TABLE 操作の巻戻し](#)」を参照してください。

---

**注意：** 論理フラッシュバック機能の用途はバックアップおよびリカバリに特化したものではないため、これらの機能に関する一部のドキュメントはドキュメント・セット内の他の場所に存在しています。

---

フラッシュバック・ドロップ以外のすべての論理フラッシュバック機能では、**UNDO データ**が使用されます。UNDO レコードは、主に、SQL 問合せでの読取り一貫性の提供およびトランザクションのロールバックのために使用されます。このため、UNDO レコードには、過去の時点のデータの再構築、およびその過去の時点以降の変更のレコードの調査に必要な情報が含まれています。

フラッシュバック・ドロップでは、**ごみ箱**と呼ばれるメカニズムが使用されます。Oracle は、削除されたデータベース・オブジェクトの占める領域が新しいデータのために必要となるまで、ごみ箱を使用してそれらのオブジェクトを管理します。ごみ箱に割り当てられる領域の量は固定されていません。また、削除されたオブジェクトがごみ箱に残される期間については保証されていません。システム・アクティビティに応じて、削除されたオブジェクトがごみ箱に残される期間は、数秒または数か月になる場合があります。

#### 参照：

- UNDO データおよび自動 UNDO 管理の詳細は、『Oracle Database 概要』および『Oracle Database 管理者ガイド』を参照してください。
- 論理フラッシュバック機能を使用する方法については、『Oracle Database アドバンスト・アプリケーション開発者ガイド』を参照してください。
- フラッシュバック・データベースを使用するためのデータベースの設定および関連するリストア・ポイント機能の詳細は、5-28 ページの「[Oracle Flashback Database およびリストア・ポイントの構成](#)」を参照してください。

## フラッシュバック表を使用した表の巻戻し

フラッシュバック表では、リストアされたバックアップではなく UNDO 表領域の情報を使用して、表を取得します。フラッシュバック表操作を行うと、新しい行が削除され、古い行が再度挿入されます。表のフラッシュバックの実行中、データベースの残りの部分は使用可能のままになります。

**参照：** 自動 UNDO 管理の詳細は、『Oracle Database 管理者ガイド』を参照してください。

## フラッシュバック表の前提条件

1 つ以上の表で Oracle Flashback Table 機能を使用するには、目標時点または SCN で FLASHBACK TABLE SQL 文を使用します。

フラッシュバック表機能を使用するには、次の権限を所有している必要があります。

- FLASHBACK ANY TABLE システム権限を付与されているか、または表に対する FLASHBACK オブジェクト権限を所有している必要があります。
- 表に対する SELECT、INSERT、DELETE および ALTER 権限を持っている必要があります。
- 表をリストア・ポイントまでフラッシュバックするには、SELECT ANY DICTIONARY または FLASHBACK ANY TABLE システム権限または SELECT\_CATALOG\_ROLE ロールを所有している必要があります。

オブジェクトをフラッシュバックするには、そのオブジェクトが次の前提条件を満たしている必要があります。

- オブジェクトが、クラスタの一部である表、マテリアライズド・ビュー、アドバンスト・キューイング (AQ) 表、静的データ・ディクショナリ表、システム表、リモート表、オブジェクト表、ネストした表、個々の表パーティションまたは個々の表サブパーティションなどのカテゴリに含まれていないこと。
- 現在の時刻とフラッシュバックの目標時点の間に、表の構造が変更されていないこと。

表の構造は、表の更新、移動または切捨て、表への制約の追加、クラスタへの表の追加、列の変更または削除、パーティションまたはサブパーティションの追加、削除、マージ、分割、結合または切捨て（レンジ・パーティションの追加を除く）などの DDL 操作によって変更されます。

- 行の移動が、表で有効化されていること（フラッシュバックの発生後に行 ID が変更されることを意味します）。

フラッシュバック前の行 ID がアプリケーションによって保存された場合、その行 ID がフラッシュバック後の同じ行に対応することは保証されないため、この制限が存在します。アプリケーションが行 ID に依存している場合は、フラッシュバック表を使用することができません。

- **UNDO 表領域**の UNDO データの範囲が、フラッシュバック目標時点または SCN を含む十分な範囲になっていること。

フラッシュバック表を実行することができる時点は、**UNDO 保存期間**（UNDO データが再利用されるまで保存される最小時間）および表領域の特性によって決定されます。UNDO データには、変更前のデータ・ブロックに関する情報が含まれています。フラッシュバック操作では、UNDO を使用して元のデータが再作成されます。

フラッシュバック表操作で UNDO 情報が確実に保持されるように、UNDO 表領域の UNDO\_RETENTION パラメータを 86400 秒（24 時間）以上に設定することをお勧めします。

---

**注意：** FLASHBACK TABLE ... TO BEFORE DROP では、フラッシュバック表ではなくフラッシュバック・ドロップ機能を使用されるため、これらの前提条件は適用されません。詳細は、16-7 ページの「[フラッシュバック・ドロップを使用した DROP TABLE 操作の巻戻し](#)」を参照してください。

---

## フラッシュバック表操作の実行

次の例では、ユーザーが更新を誤って複数回実行した後、hr.temp\_employees 表のフラッシュバックを実行すると想定しています。

### temp\_employees のフラッシュバックを実行する手順

1. SQL\*Plus をターゲット・データベースに接続し、現行の SCN を識別します。

FLASHBACK TABLE 文はロールバックできませんが、別の FLASHBACK TABLE 文を発行し、現在の時刻の直前の時刻を指定することができます。このため、現行の SCN を記録することをお勧めします。現行の SCN は、V\$DATABASE を次のように問い合わせることによって取得できます。

```
SELECT CURRENT_SCN
FROM   V$DATABASE;
```

2. 表を戻す時刻、SCN またはリストア・ポイントを識別します。

リストア・ポイントを作成した場合は、次の問合せを実行して、使用可能なリストア・ポイントを表示できます。

```
SELECT NAME, SCN, TIME
FROM   V$RESTORE_POINT;
```

- 指定した目標時点に表を巻戻すための十分な UNDO データが存在していることを確認します。

UNDO\_RETENTION 初期化パラメータが設定されており、UNDO 保存保証が有効になっている場合は、次の問合せを使用して、UNDO データが保持される期間を確認できます。

```
SELECT NAME, VALUE/60 MINUTES_RETAINED
FROM   V$PARAMETER
WHERE  NAME = 'undo_retention';
```

- 行の移動が、フラッシュバック表を使用して巻き戻しているすべてのオブジェクトに対して有効になっていることを確認します。

次の SQL 文を使用して、表の移動を有効にすることができます。ここで、*table* は、巻き戻す表の名前です。

```
ALTER TABLE table ENABLE ROW MOVEMENT;
```

- フラッシュバック対象の表に、他の表への依存性があるかどうかを確認します。依存性が存在する場合は、それらの表をフラッシュバックするかどうかも決定します。

次の SQL 文を発行し、依存性を確認します。ここで、*schema\_name* はフラッシュバック対象の表のスキーマ、*table\_name* は表の名前です。

```
SELECT other.owner, other.table_name
FROM   sys.all_constraints this, sys.all_constraints other
WHERE  this.owner = schema_name
AND    this.table_name = table_name
AND    this.r_owner = other.owner
AND    this.r_constraint_name = other.constraint_name
AND    this.constraint_type='R';
```

- フラッシュバック対象のオブジェクトに対して、FLASHBACK TABLE 文を実行します。

次の SQL 文を実行すると、hr.temp\_employees 表が、temp\_employees\_update というリストア・ポイントまでリストアされます。

```
FLASHBACK TABLE hr.temp_employees
TO RESTORE POINT temp_employees_update;
```

次の SQL 文を実行すると、hr.temp\_employees 表の状態が、SCN で指定された時点のデータベースの状態まで巻き戻されます。

```
FLASHBACK TABLE hr.temp_employees
TO SCN 123456;
```

次の例に示すように、TO\_TIMESTAMP を使用して目標時点を指定することもできます。

```
FLASHBACK TABLE hr.temp_employees
TO TIMESTAMP TO_TIMESTAMP('2007-10-17 09:30:00', 'YYYY-MM-DD HH:MI:SS');
```

---

**注意：** タイムスタンプと SCN のマッピングは、常に正確であるとはかぎりません。FLASHBACK TABLE 文でタイムスタンプを使用した場合、表がフラッシュバックされる時刻と TO\_TIMESTAMP で指定される時刻の間で最大 3 秒異なることがあります。過去の正確な時点が必要な場合は、時刻ではなく SCN を使用します。

---

- 必要に応じて、表を問い合わせてデータを確認します。

## フラッシュバック表中のトリガーの有効状態の保持

デフォルトでは、データベースによって、FLASHBACK TABLE 操作の実行前に、影響を受ける表のトリガーが無効にされます。このトリガーの状態は、操作の終了後、操作前の状態（有効または無効）に戻されます。表のフラッシュバック中にトリガーを有効な状態のままにするには、16-6 ページの手順 6 で FLASHBACK TABLE 文に、ENABLE TRIGGERS 句を追加します。

たとえば、HR 管理者が 17 時に、hr.temp\_employees 表からある従業員が欠落していることに気付いたとします。この従業員は、レポートが最後に実行された 14 時には表に含まれていました。したがって、この従業員のレコードが誤って削除されたのは、14 時から 17 時の間です。HR 管理者は、フラッシュバック表を使用して、hr.temp\_employees 表に設定されているすべてのトリガーを有効にして、表を午後 2 時の状態に戻します。この場合、次の例の SQL 文を使用します。

```
FLASHBACK TABLE temp_employees
  TO TIMESTAMP TO_TIMESTAMP('2005-03-03 14:00:00' , 'YYYY-MM-DD HH:MI:SS')
  ENABLE TRIGGERS;
```

### 参照:

- フラッシュバック表機能を使用して表をリカバリする方法については、『Oracle Database 管理者ガイド』を参照してください。
- Oracle Flashback Table の簡単な使用例は、『Oracle Database SQL 言語 リファレンス』を参照してください。

## フラッシュバック・ドロップを使用した DROP TABLE 操作の巻戻し

この項では、FLASHBACK TABLE ... TO BEFORE DROP 文を使用して、ごみ箱からオブジェクトを取得する方法について説明します。

### フラッシュバック・ドロップ

フラッシュバック・ドロップによって、DROP TABLE 操作の結果を無効にできます。フラッシュバック・ドロップは、この状況で使用できる他のリカバリ・メカニズム（Point-in-Time リカバリなど）より高速です。また、フラッシュバック・ドロップによって、停止時間または最新のトランザクションの消失は発生しません。

表を削除した場合、その表に関連付けられている領域はデータベースによってすぐには削除されません。かわりに、表は、名前を変更され、関連付けられているすべてのオブジェクトとともにごみ箱に配置されます。システム生成のごみ箱のオブジェクト名は一意です。他のオブジェクトを問い合わせる場合と同様に、ごみ箱内のオブジェクトを問い合わせることができます。

フラッシュバック操作では、ごみ箱から表が取り出されます。削除されたテーブルを取得する場合は、ユーザーが指定した表の元の名前またはシステム生成の名前を使用できます。

表を削除すると、表およびそれに依存するすべてのオブジェクトが同時にごみ箱に移動されます。同様に、フラッシュバック・ドロップを実行すると、通常、すべてのオブジェクトが同時に取得されます。表をごみ箱からリストアすると、索引などの依存オブジェクトの名前は、元の名前に戻されず、ごみ箱内のシステム生成の名前のままになります。Oracle Database では、表に定義されたすべての索引（ビットマップ結合索引を除く）、および表に定義されたすべてのトリガーと制約（他の表を参照する参照整合性制約を除く）が取得されます。

索引などの依存オブジェクトの一部は、領域圧迫のため再利用される場合があります。この場合、再利用された依存オブジェクトは、ごみ箱から取得できません。

## フラッシュバック・ドロップの前提条件

次に、フラッシュバック・ドロップおよびごみ箱に関連する操作に必要なユーザー権限の概要を示します。

- DROP  
オブジェクトに対して DROP 権限を持つすべてのユーザーは、オブジェクトを削除し、ごみ箱に配置することができます。
- FLASHBACK TABLE ... TO BEFORE DROP  
この文に対する権限は、DROP 権限に関連付けられています。つまり、オブジェクトを削除できるすべてのユーザーは、フラッシュバック・ドロップを実行して、削除済オブジェクトをごみ箱から取得できます。
- PURGE  
ごみ箱の消去に対する権限は、DROP 権限に関連付けられています。DROP TABLE または DROP ANY TABLE 権限を持つユーザーは、オブジェクトをごみ箱から消去できます。
- ごみ箱内のオブジェクトに対する SELECT  
ごみ箱のオブジェクトの間合せを行う場合、ユーザーは、ごみ箱のオブジェクトに対して SELECT および FLASHBACK 権限を所有している必要があります。削除前にオブジェクトに対して SELECT 権限を所有していたすべてのユーザーは、ごみ箱内のそのオブジェクトに対して SELECT 権限を継続して所有しています。ごみ箱内のオブジェクトは過去の状態のデータベースのオブジェクトであるため、これらのオブジェクトを問い合わせるには、ユーザーは FLASHBACK 権限が必要です。

ごみ箱からオブジェクトを取得するには、オブジェクトが次の要件を満たしている必要があります。

- システム表領域ではないローカル管理表領域に対してのみごみ箱が使用可能であること。表がシステム表領域ではないローカル管理表領域に含まれていて、その依存セグメント (オブジェクト) がディクショナリ管理表領域に含まれている場合、これらのオブジェクトはごみ箱によって保護されます。
- ファイングレイイン監査ポリシー (FGA) および Virtual Private Database (VPD) ポリシーが定義されている表が、ごみ箱によって保護されていないこと。
- パーティション化された索引構成表が、ごみ箱によって保護されていないこと。
- 領域再利用操作の結果として、ユーザーまたは Oracle Database のいずれかによって表が削除されていないこと。

## フラッシュバック・ドロップ操作の実行

FLASHBACK TABLE ... TO BEFORE DROP 文を使用すると、ごみ箱からオブジェクトをリカバリできます。ごみ箱内の表の名前または元の表の名前のいずれかを指定できます。

この項では、誤った表を削除した例を想定しています。テスト・データベースで表の削除を複数回要求されましたが、この場合、かわりに本番データベースに誤って接続し、hr.employee\_demo を削除したとします。FLASHBACK TABLE を使用し、削除されたオブジェクトを取得します。

### 削除された表を取得する手順

1. SQL\*Plus をターゲット・データベースに接続し、ごみ箱内の削除された表の名前を取得します。

SQL\*Plus コマンド SHOW RECYCLEBIN は、次のように使用できます。

```
SHOW RECYCLEBIN;
```

| ORIGINAL NAME | RECYCLEBIN NAME                    | TYPE  | DROP TIME           |
|---------------|------------------------------------|-------|---------------------|
| EMPLOYEE_DEMO | BIN\$gk31sj/3akk5hg3j2lk15j3d==\$0 | TABLE | 2005-04-11:17:08:54 |

ORIGINAL\_NAME 列には、オブジェクトの元の名前が表示され、RECYCLEBIN\_NAME 列には、ごみ箱内でのオブジェクトの名前が表示されます。

また、USER\_RECYCLEBIN または DBA\_RECYCLEBIN を問い合せて表の名前を取得することもできます。次の例では、ビューを問い合せて、削除されたオブジェクトの元の名前を確認します。

```
SELECT object_name AS recycle_name, original_name, type
FROM recyclebin;
```

| RECYCLE_NAME                       | ORIGINAL_NAME | TYPE  |
|------------------------------------|---------------|-------|
| BIN\$gk31sj/3akk5hg3j21k15j3d==\$0 | EMPLOYEE_DEMO | TABLE |
| BIN\$JKS983293M1dsab4gsz/I249==\$0 | I_EMP_DEMO    | INDEX |

依存オブジェクトの元の名前を手動でリストアする場合は、表をリストアする前に、各依存オブジェクトのごみ箱内のシステム生成の名前を書き留めておいてください。

---

**注意：** DBA\_TABLES などのオブジェクト・ビューには、ごみ箱オブジェクトは表示されません。

---

## 2. 必要に応じて、ごみ箱内の表を問い合せます。

問合せでは、オブジェクトの元の名前ではなく、オブジェクトのごみ箱内の名前を使用する必要があります。次の例では、BIN\$KSD8DB9L345KLA==\$0 というごみ箱内の名前を持つ表を問い合せます。

```
SELECT *
FROM "BIN$gk31sj/3akk5hg3j21k15j3d==$0";
```

ごみ箱内の名前に特殊文字が含まれているため、引用符が必要です。

---

**注意：** 必要な権限を所有している場合は、ごみ箱内の表に対して Oracle Flashback Query を使用できますが、表の元の名前ではなく、ごみ箱内の名前を使用する必要があります。ごみ箱内のオブジェクトに対して、DDL または DDL 文を使用することはできません。

---

## 3. 削除された表を取得します。

FLASHBACK TABLE ...TO BEFORE DROP 文を使用します。次の例では、BIN\$gk31sj/3akk5hg3j21k15j3d==\$0 表をリストアし、その名前を hr.employee\_demo に戻して、そのエントリをごみ箱から消去します。

```
FLASHBACK TABLE "BIN$gk31sj/3akk5hg3j21k15j3d==$0" TO BEFORE DROP;
```

ごみ箱内のオブジェクト名に特殊文字が含まれる可能性があるため、表の名前が引用符で囲まれていることに注意してください。

また、次のようにして、表の元の名前を使用することもできます。

```
FLASHBACK TABLE HR.EMPLOYEE_DEMO TO BEFORE DROP;
```

RENAME TO 句を指定することによって、リストアされた表に新しい名前を割り当てることもできます。たとえば、次のように入力します。

```
FLASHBACK TABLE "BIN$KSD8DB9L345KLA==$0" TO BEFORE DROP
RENAME TO hr.emp_demo;
```

- 必要に応じて、すべての依存オブジェクトがごみ箱内のシステム生成の名前を保持していることを確認します。

次の問合せによって、取得された hr.employee\_demo 表の索引の名前を確認します。

```
SELECT INDEX_NAME
FROM   USER_INDEXES
WHERE  TABLE_NAME = 'EMPLOYEE_DEMO';
```

```
INDEX_NAME
-----
BIN$JKS983293M1dsab4gsz/I249==$0
```

- 必要に応じて、取得された索引を元の名前に変更します。

次の文を実行すると、索引がその元の名前である i\_emp\_demo に変更されます。

```
ALTER INDEX "BIN$JKS983293M1dsab4gsz/I249==$0" RENAME TO I_EMP_DEMO;
```

- 取得された表に、ごみ箱に配置される前に参照制約が含まれていた場合は、その参照制約を再作成します。

ごみ箱では表に対する参照制約が保存されないため、この手順は手動で実行する必要があります。

## 元の名前が同じオブジェクトが複数存在する場合のオブジェクトの取出し

複数のオブジェクトを同じ元の名前で作成し、その後削除できます。削除されたすべてのオブジェクトは、ごみ箱に保存されます。たとえば、次の例の SQL 文について考えてみます。

### 例 16-1 同じ名前の複数オブジェクトの削除

```
CREATE TABLE temp_employees ( ...columns ); # temp_employees version 1
DROP TABLE temp_employees;
```

```
CREATE TABLE temp_employees ( ...columns ); # temp_employees version 2
DROP TABLE temp_employees;
```

```
CREATE TABLE temp_employees ( ...columns ); # temp_employees version 3
DROP TABLE temp_employees;
```

例 16-1 では、temp\_employees 表が削除されるたびに、ごみ箱の中でそれぞれに一意の名前が割り当てられています。次の例に示すように、FLASHBACK TABLE ... TO BEFORE DROP 文は、表の元の名前を指定して使用できます。

```
FLASHBACK TABLE temp_employees TO BEFORE DROP;
```

この元の名前で最後に削除された表は、その元の名前でごみ箱から取得されます。例 16-2 では、前の例で削除された 3 つすべての temp\_employees 表が、それぞれ新しい名前を割り当てられてごみ箱から取得されています。

### 例 16-2 削除された表の名前の変更

```
FLASHBACK TABLE temp_employees TO BEFORE DROP
  RENAME TO temp_employees_VERSION_3;
FLASHBACK TABLE temp_employees TO BEFORE DROP
  RENAME TO temp_employees_VERSION_2;
FLASHBACK TABLE temp_employees TO BEFORE DROP
  RENAME TO temp_employees_VERSION_1;
```

FLASHBACK TABLE での元の名前は、この名前ですべて最後に削除された表を参照するため、最後に削除された表が最初に取得されます。



また、元の名前が競合している場合でも、ごみ箱内の一意の名前を使用して、すべての表をごみ箱から取得することができます。たとえば、ごみ箱を次のように問い合わせます（出力例も示します）。

```
SELECT object_name, original_name, createtime
FROM recyclebin;
```

| OBJECT_NAME                      | ORIGINAL_NAME  | CREATETIME          |
|----------------------------------|----------------|---------------------|
| BIN\$yrmKlZaLMhfgNAgAIMenRA==\$0 | TEMP_EMPLOYEES | 2007-02-05:21:05:52 |
| BIN\$yrmKlZaVMhfgNAgAIMenRA==\$0 | TEMP_EMPLOYEES | 2007-02-05:21:25:13 |
| BIN\$yrmKlZaQMhfgNAgAIMenRA==\$0 | TEMP_EMPLOYEES | 2007-02-05:22:05:53 |

次のコマンドを使用すると、中間の表を取得できます。

```
FLASHBACK TABLE BIN$yrmKlZaVMhfgNAgAIMenRA==$0 TO BEFORE DROP;
```

#### 参照：

- フラッシュバック・ドロップを使用する方法およびごみ箱を管理する方法については、『Oracle Database 管理者ガイド』を参照してください。
- FLASHBACK TABLE 文の詳細は、『Oracle Database SQL 言語リファレンス』を参照してください。

## フラッシュバック・データベースを使用したデータベースの巻戻し

この項では、フラッシュバック・データベースを使用して、データベースに対する不要な変更を巻き戻す最も一般的な例について説明します。

### フラッシュバック・データベースの前提条件

FLASHBACK DATABASE コマンドを使用してデータベースの内容をフラッシュバック・ウィンドウ内の時点まで戻すには、「[Oracle Flashback Database およびリストア・ポイントの構成](#)」の説明に従って、フラッシュバック・ロギングを行うようにデータベースを事前に構成しておく必要があります。保証付きリストア・ポイントまでデータベースを戻すには、5-35 ページの「[通常のリストア・ポイントと保証付きリストア・ポイントの作成](#)」の説明に従って、保証されたリストア・ポイントを事前に定義しておく必要があります。

フラッシュバック・データベースでは、コマンド実行時に存在した、データファイルに対する変更が取り消されます。重要な前提条件は、次のとおりです。

- 現行のデータファイルが消失または破損していないこと。FLASHBACK DATABASE は、Oracle Database で行われたデータファイルに対する変更の巻戻しにのみ使用できます。メディア障害の修復またはデータファイルの誤った削除からのリカバリには使用できません。
- 制御ファイルのリストアまたは再作成前の時点に戻すために FLASHBACK DATABASE を使用しようとしていないこと。データベース制御ファイルがバックアップからリストアされるか、または再作成されると、すべての累積フラッシュバック・ログ情報が破棄されます。
- データファイルのサイズ変更操作を取り消すために FLASHBACK DATABASE を使用しようとしていないこと。表などのデータベース・オブジェクトの縮小は、FLASHBACK DATABASE を使用する機能には影響しません。

**参照：** FLASHBACK DATABASE のコマンドの前提条件および使用方法の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## フラッシュバック・データベース操作の実行

この項では、時刻書式、通常のリストア・ポイントまたは保証付きリストア・ポイントの名前、あるいは SCN を使用して目的の目標時点を指定し、ほぼすべての場合でデータベースのフラッシュバックを実行する基本的な方法について説明します。

次の例では、現行のデータベースの **インカーネーション** 内の時点までデータベースを巻戻すと想定しています。最後の OPEN RESETLOGS の直前の時点までデータベースを戻す場合は、16-17 ページの「[フラッシュバック・データベースを使用した OPEN RESETLOGS 操作の巻戻し](#)」を参照してください。

デフォルトでは、FLASHBACK DATABASE コマンドで使用される SCN は、データベース・インカーネーションの **直系祖先パス** 内の SCN を参照します。13-6 ページの「[データベース・インカーネーション](#)」で説明されているように、データベースを以前 RESETLOGS オプションでオープンした後で取り消さなかった場合、インカーネーションはこのパス内にあります。取り消されたインカーネーション内の変更を取得する場合は、16-18 ページの「[取り消されたインカーネーション・ブランチ内の SCN へのデータベースの巻戻し](#)」を参照してください。

### フラッシュバック・データベース操作を行う手順

1. SQL\*Plus をターゲット・データベースに接続し、FLASHBACK DATABASE コマンドに必要な SCN、リストア・ポイントまたは時点を確認します。

次のように入力して、フラッシュバック・データベース・ウィンドウ内の最も古い SCN を取得します。

```
SELECT OLDEST_FLASHBACK_SCN, OLDEST_FLASHBACK_TIME
FROM   V$FLASHBACK_DATABASE_LOG;
```

フラッシュバック・データベースで到達できる最新の SCN は、データベースの現行の SCN です。次の問合せを実行すると、現行の SCN が戻されます。

```
SELECT CURRENT_SCN
FROM   V$DATABASE;
```

使用可能な保証付きリストア・ポイントは、次のように問い合わせることができます（出力例も示します）。

```
SELECT NAME, SCN, TIME, DATABASE_INCARNATION#,
       GUARANTEE_FLASHBACK_DATABASE
FROM   V$RESTORE_POINT
WHERE  GUARANTEE_FLASHBACK_DATABASE='YES';
```

| NAME           | SCN     | TIME                  | DATABASE_INCARNATION# | GUA |
|----------------|---------|-----------------------|-----------------------|-----|
| BEFORE_CHANGES | 5753126 | 04-MAR-05 12.39.45 AM | 2                     | YES |

---

**注意：** フラッシュバックの期間が目的の目標時点に達する過去の時点まで提供されておらず、その目的の時点での保証付きリストア・ポイントが存在しない場合は、同じ結果を得るために、データベースの Point-in-Time リカバリを使用できます。詳細は、16-15 ページの「[データベースの Point-in-Time リカバリの実行](#)」を参照してください。

---

2. データベースを、一貫性のある状態で停止し、インスタンスによってオープンされていないことを確認してからマウントします。

```
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
```

## 3. 最初の手順で実行した問合せを繰り返します。

データベースが停止されている場合は、一部のフラッシュバック・ロギング・データが生成されます。フラッシュバック・リカバリ領域での領域圧迫が原因でフラッシュバック・ログが削除された場合は、目的の SCN が取得不可になっている可能性があります。

---

**注意：** 目的の SCN がフラッシュバック・ウィンドウの範囲外にある場合に FLASHBACK DATABASE を実行すると、FLASHBACK DATABASE は失敗し、ORA-38729 エラーが戻されます。この場合、データベースは変更されません。

---

## 4. Recovery Manager を起動し、ターゲット・データベースに接続します。

## 5. SHOW コマンドを実行して、事前構成済のチャンネルを確認します。

フラッシュバック操作中、Recovery Manager は、バックアップからアーカイブ REDO ログをリストアする必要がある場合があります。次のコマンドを入力し、自動チャンネルが構成されているかどうかを確認します（出力例も示します）。

```
SHOW ALL;
```

```
RMAN configuration parameters for database with db_unique_name PROD1 are:
```

```
.
.
.
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
CONFIGURE DEVICE TYPE SBT_TAPE PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMS "SBT_
LIBRARY=/usr/local/oracle/backup/lib/libobk.so";
```

必要なデバイスおよびチャンネルがすでに構成されている場合、操作は必要ありません。そうでない場合は、CONFIGURE コマンドを使用して自動チャンネルを構成するか、または RUN ブロック内に ALLOCATE CHANNEL コマンドを含めます。

## 6. Recovery Manager の FLASHBACK DATABASE コマンドを実行します。

次の例に示すいずれかの形式のコマンドを使用して、目標時点を指定できます。

```
FLASHBACK DATABASE TO SCN 46963;
```

```
FLASHBACK DATABASE
  TO RESTORE POINT BEFORE_CHANGES;
```

```
FLASHBACK DATABASE TO TIME
  "TO_DATE('09/20/05','MM/DD/YY')";
```

FLASHBACK DATABASE コマンドが完了すると、データベースはマウントされたまま、指定した目標時点にリカバリされています。

## 7. SQL\*Plus を使用してデータベースを読取り専用でオープンし、いくつかの問合せを実行して、データベースの内容を検証します。

次のように入力して、データベースを読取り専用でオープンします。

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

データベースの状態が適切である場合は、手順 8 で手順を終了します。データベースの状態が適切でない場合は、手順 9 に進みます。

8. 結果が適切である場合は、相互に排他的な次の処理のいずれかを実行します。

- RESETLOGS オプションを指定してデータベースをオープンし、データベースを更新可能な状態にします。データベースが現在読取り専用でオープンされている場合は、SQL\*Plus で次のコマンドを実行します。

```
SHUTDOWN IMMEDIATE
STARTUP MOUNT
ALTER DATABASE OPEN RESETLOGS;
```

---

**注意：** この OPEN RESETLOGS 操作を実行した後、FLASHBACK DATABASE の目的の SCN 以降に行われたデータベースへの変更はすべて破棄されます。ただし、16-18 ページの「[取り消されたインカネーション・ブランチ内の SCN へのデータベースの巻戻し](#)」に示す方法を使用すると、複数の SCN がフラッシュバックの期間内にあるかぎり、データベースをそれぞれの SCN の範囲まで戻すことができます。

---

- Oracle Data Pump Export を使用し、破損状態のオブジェクトの論理バックアップを行います。その後、Recovery Manager を使用して、データベースを現在の時刻までリカバリします。

```
RECOVER DATABASE;
```

この手順では、REDO ログ内のすべての変更をデータベースに適用してフラッシュバック・データベースの影響を取り消し、データベースを最新の SCN まで戻します。

データベースを読取り / 書込みでオープンした後、データ・ポンプ・インポート・ユーティリティを使用してエクスポートされたオブジェクトをインポートできます。データ・ポンプを使用する方法の詳細は、『Oracle Database ユーティリティ』を参照してください。

9. フラッシュバックで誤ったリストア・ポイント、時刻または SCN を使用した場合は、データベースをマウントし、次の相互に排他的ないずれかのオプションを実行します。

- 選択した目標時点が十分に前の時点でなかった場合は、別の FLASHBACK DATABASE コマンドを使用し、データベースをさらに巻き戻します。

```
FLASHBACK DATABASE TO SCN 42963; #earlier than current SCN
```

- 選択した目的の SCN が前過ぎる場合は、RECOVER DATABASE UNTIL を使用してデータベースを必要な SCN まで進めます。

```
RECOVER DATABASE UNTIL SCN 56963; #later than current SCN
```

- FLASHBACK DATABASE コマンドの影響を完全に取り消す必要がある場合は、UNTIL 句または SET UNTIL コマンドを指定しないで RECOVER DATABASE コマンドを使用することによって、データベースの完全リカバリを実行できます。

```
RECOVER DATABASE;
```

RECOVER DATABASE コマンドを実行すると、すべての変更がデータベースに再適用され、データベースが最新の SCN まで戻されます。

## データベースの Point-in-Time リカバリの実行

Recovery Manager の DBPITR を実行すると、リカバリの目標時点より前のバックアップからデータベースがリストアされ、増分バックアップおよび REDO を使用してデータベースが目標時点までロールフォワードされます。SCN、時刻、ログ順序番号またはリストア・ポイントまでのリカバリを行うことができます。必要になった場合に Point-in-Time リカバ리를より管理しやすい状態にしておくために、重要な時刻にリストア・ポイントを作成しておくことをお勧めします。

可能な場合は、データベースの Point-in-Time リカバリではなく、フラッシュバック・データベースを実行することをお勧めします。バックアップを使用したメディア・リカバリは、最後に行った変更を取り消すためにフラッシュバックを使用できない場合の最後の手段として使用してください。

### データベースの Point-in-Time リカバリの前提条件

データベースの Point-in-Time リカバリの前提条件は、次のとおりです。

- データベースは ARCHIVELOG モードで実行されている必要があります。
- DBPITR の目的の SCN より前のすべてのデータファイルのバックアップ、およびバックアップの SCN と目的の SCN の間の期間のアーカイブ・ログが存在している必要があります。

コマンドの前提条件および使用上の注意の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』の RECOVER エントリに関する項を参照してください。

### データベースの Point-in-Time リカバリの実行

この項では、DBPITR の基本手順について説明します。この手順では、次のことを想定しています。

- 現行のデータベースの **インカネーション** 内で、DBPITR を実行します。目標時点が現行のインカネーション内に存在しない場合は、祖先インカネーションへの DBPITR について、16-20 ページの「祖先インカネーションへのデータベースのリカバリ」を参照してください。
- 制御ファイルは現行のもので、バックアップ制御ファイルをリストアする必要がある場合は、19-5 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」を参照してください。
- データベースでは、現行のサーバー・パラメータ・ファイルが使用されています。バックアップ・サーバーのパラメータ・ファイルをリストアする必要がある場合は、19-2 ページの「サーバー・パラメータ・ファイルのリストア」を参照してください。

DBPITR を実行する場合は、RESTORE および RECOVER コマンドで個々に UNTIL 句を指定するのではなく、SET UNTIL コマンドを使用して目標時点を手順の最初に設定することによって、エラーを回避できます。これによって、バックアップからリストアされたデータファイルに十分に古いタイムスタンプが設定され、後続の RECOVER 操作に使用できます。

#### DBPITR を実行する手順

1. リカバ리를終了する SCN、リストア・ポイントまたはログ順序番号を決定します。

フラッシュバック問合せ機能を使用すると、論理的な破損が発生した時点特定できます。表に対して **フラッシュバック・データ・アーカイブ** が有効になっている場合は、非常に古いデータを問い合わせることができます。

また、アラート・ログを使用して、リカバリする必要があるイベントの時刻を特定することもできます。

さらに、SQL 問合せを使用して、目的の SCN が含まれているログ順序番号を特定し、このログを使用してリカバリすることもできます。たとえば、次の問合せを実行して、現行のデータベース・インカーネーションのログを表示します（出力例も示します）。

```
SELECT RECID, STAMP, THREAD#, SEQUENCE#, FIRST_CHANGE#
       FIRST_TIME, NEXT_CHANGE#
FROM   V$ARCHIVED_LOG
WHERE  RESETLOGS_CHANGE# =
      ( SELECT RESETLOGS_CHANGE#
        FROM   V$DATABASE_INCARNATION
        WHERE  STATUS = 'CURRENT');
```

| RECID | STAMP     | THREAD# | SEQUENCE# | FIRST_CHAN | FIRST_TIM | NEXT_CHANG |
|-------|-----------|---------|-----------|------------|-----------|------------|
| 1     | 344890611 | 1       | 1         | 20037      | 24-SEP-05 | 20043      |
| 2     | 344890615 | 1       | 2         | 20043      | 24-SEP-05 | 20045      |
| 3     | 344890618 | 1       | 3         | 20045      | 24-SEP-05 | 20046      |

たとえば、ユーザーが誤って表領域を午前 9 時 2 分に削除したことがわかった場合は、削除が発生する直前の午前 9 時までリカバリを行うことができます。この時刻より後にデータベースに対して行われたすべての変更は消失します。

2. 目的の SCN ではなく、目標の時刻書式を使用する場合は、**Recovery Manager** の起動前に時刻書式の環境変数が適切であることを確認します。

グローバルゼーション・サポート設定の例を示します。

```
NLS_LANG = american_america.us7ascii
NLS_DATE_FORMAT="Mon DD YYYY HH24:MI:SS"
```

3. **Recovery Manager** をターゲット・データベースに接続し、必要に応じてリカバリ・カタログにも接続します。データベースをマウント状態にします。

```
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
```

4. RUN ブロック内で、次の操作を実行します。

- a. **SET UNTIL** を使用して、**DBPITR** の目標時点、リストア・ポイント、SCN またはログ順序番号を指定します。時刻を指定する場合は、環境変数 **NLS\_LANG** および **NLS\_DATE\_FORMAT** で指定した日付書式を使用します。
- b. 自動チャネルを構成していない場合は、必要に応じて、ディスク・チャネルおよびテープ・チャネルを手動で割り当てます。
- c. データベースをリストアおよびリカバリします。

次の例では、ターゲット・データベースで SCN 1000 まで **DBPITR** を実行します。

```
RUN
{
  SET UNTIL SCN 1000;
  RESTORE DATABASE;
  RECOVER DATABASE;
}
```

また、次の例に示すように、時刻書式、リストア・ポイントまたはログ順序番号を使用して、**SET UNTIL** 時刻を指定することもできます。

```
SET UNTIL TIME 'Nov 15 2004 09:00:00';
SET UNTIL SEQUENCE 9923;
SET UNTIL RESTORE POINT before_update;
```

操作がエラーを戻さずに完了した場合、**DBPITR** は正常に終了しています。

- SQL\*Plus を使用してデータベースを読取り専用でオープンし、必要に応じて問合せを実行して、論理的な破損の影響が無効にされたことを確認します。

次のように入力して、データベースを読取り専用でオープンします。

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

データベースの状態が適切である場合は、手順 6 で手順を終了します。適切でない場合は、目的の SCN を誤って選択した可能性があります。この場合は、不要な変更についてさらに調査し、新しい目的の SCN を特定してから、DBPITR の手順を繰り返します。

- 結果が適切である場合は、相互に排他的な次の処理のいずれかを実行します。
  - データベースを読取り / 書込みでオープンし、目的の SCN 以降のすべての変更を取り消します。この場合は、データベースを停止し、マウントしてから、次のコマンドを実行する必要があります。

```
ALTER DATABASE OPEN RESETLOGS;
```

データファイルがオフラインになっていると、データファイルが正常にオフラインにされているか、または読取り専用である場合を除き、OPEN RESETLOGS 操作が失敗することに注意してください。RESETLOGS の実行後は REDO が不要になるため、読取り専用または NORMAL モードでオフラインにされた表領域のファイルをオンラインにすることができます。

- データ・ポンプ・エクスポートを使用してデータベースから 1 つ以上のオブジェクトをエクスポートします。次に、データベースを現在の時点までリカバリして、エクスポートされたオブジェクトを再インポートします。これによって、他のすべての変更を取り消さずにこれらのオブジェクトを不要な変更前の状態に戻すことができます。

## フラッシュバックおよびデータベースの Point-in-Time リカバリの例

この項では、16-11 ページの「フラッシュバック・データベースを使用したデータベースの巻戻し」および 16-15 ページの「データベースの Point-in-Time リカバリの実行」で説明されている基本例を少し変更した例について説明します。

### フラッシュバック・データベースを使用した OPEN RESETLOGS 操作の巻戻し

フラッシュバック・データベースを使用して不要な ALTER DATABASE OPEN RESETLOGS 文の結果を前の状態に戻すための手順は、16-12 ページの「フラッシュバック・データベース操作の実行」で説明する通常の場合と類似しています。ただし、FLASHBACK DATABASE コマンドで特定の SCN または時点を指定するかわりに、FLASHBACK DATABASE TO BEFORE RESETLOGS を使用します。

#### OPEN RESETLOGS 操作を取り消す手順

- SQL\*Plus をターゲット・データベースに接続し、フラッシュバック・ウィンドウの開始時点が最新の OPEN RESETLOGS より前であることを確認します。

次の問合せを実行します。

```
SELECT RESETLOGS_CHANGE#
FROM   V$DATABASE;
```

```
SELECT OLDEST_FLASHBACK_SCN
FROM   V$FLASHBACK_DATABASE_LOG;
```

V\$DATABASE.RESETLOGS\_CHANGE# が V\$FLASHBACK\_DATABASE\_LOG.OLDEST\_FLASHBACK\_SCN より大きい場合は、Oracle Flashback Database を使用して OPEN RESETLOGS を前の状態に戻すことができます。

- データベースを停止してマウントし、フラッシュバック・ウィンドウを再確認します。RESETLOGS SCN がまだフラッシュバック・ウィンドウ内にある場合は、次の手順に進みます。

3. Recovery Manager をターゲット・データベースに接続し、RESETLOGS の直前の SCN までフラッシュバックを実行します。

次の形式の FLASHBACK DATABASE コマンドを使用します。

```
FLASHBACK DATABASE TO BEFORE RESETLOGS;
```

FLASHBACK DATABASE の他の使用例と同様に、ターゲット SCN がフラッシュバック・データベース・ウィンドウの開始時点より前である場合は、エラーが戻され、データベースは変更されません。コマンドが正常終了した場合、データベースはマウントされたまま、以前のインカネーションで OPEN RESETLOGS が実行される前の最新の SCN までリカバリされます。

4. SQL\*Plus を使用してデータベースを読取り専用でオープンし、必要に応じて問合せを実行して、論理的な破損の影響が無効にされたことを確認します。

次のように入力して、データベースを読取り専用でオープンします。

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

5. データベースを再度更新できるようにするには、データベースを停止してマウントしてから、次のコマンドを実行します。

```
ALTER DATABASE OPEN RESETLOGS;
```

## フラッシュバック・データベースを使用したスタンバイ・データベースでの OPEN RESETLOGS の取消し

OPEN RESETLOGS をまたぐフラッシュバック・データベースは、Dataguard 環境で次の機能を実現するために使用できます。

- 論理スタンバイ・スイッチオーバーを取り消すためのフラッシュバック

この場合、データベースは、フラッシュバック・データベース操作の目標時点での役割（プライマリまたはスタンバイ）に戻されます。

- フィジカル・スタンバイのアクティブ化の取消し

一時的にフィジカル・スタンバイ・データベースをアクティブ化し、テストまたはレポートの目的で使用してから、フラッシュバック・データベースを使用してフィジカル・スタンバイとしての役割に戻すことができます。

- テスト目的でのスタンバイ・データベースの継続的な使用

フラッシュバック・データベースを使用すると、ストレージ・スナップショットを使用する必要がなくなります。

**参照：** Data Guard でのフラッシュバック・データベースのこれらの高度な適用の詳細は、『Oracle Data Guard 概要および管理』を参照してください。

## 取り消されたインカネーション・ブランチ内の SCN へのデータベースの巻戻し

フラッシュバック・データベースまたは DBPITR の後に OPEN RESETLOGS を使用すると、データベースが以前の SCN に戻され、その時点以降の変更が取り消されます。したがって、その時点以降の一部の SCN は、取り消された変更またはデータベースの現行の履歴にある変更のいずれかを参照できます。この方法では、FLASHBACK DATABASE で指定された目的の SCN が不明確になる可能性があります。

SCN とは異なり、時刻書式およびリストア・ポイントは不明確にはなりません。時刻書式は、常に、その時点の現行インカネーションに関連付けられています。リストア・ポイントは、常に、作成時の現行インカネーションに関連付けられています。これは、取り消されたデータベース・インカネーションに対応する時刻およびリストア・ポイントにも該当します。データベース・インカネーションは、指定した時刻またはリストア・ポイントの作成時の現行インカネーションに自動的に再設定されます。



フラッシュバック・データベースを使用して、現行のインカネーション・パスが古いインカネーションからブランチした時点の OPEN RESETLOGS の SCN より後の親インカネーション内の時点までデータベースを巻き戻す必要がある場合があります。OPEN RESETLOGS で新しいインカネーションを作成した後でも、インカネーション・ブランチで SCN を生成できる方法については、13-7 ページの図 13-1 「データベース・インカネーションの履歴」を参照してください。この図に示されているように、インカネーション 1 内の取り消された SCN 1500 まで戻る必要がある場合、データベースはインカネーション 3 内の SCN 3000 にある可能性があります。

巻き戻している SCN が直系祖先パスにある場合、またはデータベースをリストア・ポイントまで巻き戻している場合は、明示的な RESET DATABASE はフラッシュバック・データベースに必要ありません。ただし、明示的な RESET DATABASE TO INCARNATION コマンドは、FLASHBACK DATABASE を使用してデータベースを取り消されたデータベース・インカネーションの SCN に巻き戻す場合は必須です。

### 取り消されたインカネーション・ブランチ内の SCN までデータベースを巻き戻す手順

1. SQL\*Plus を使用してターゲット・データベースに接続し、SCN までフラッシュバックするための十分な情報がフラッシュバック・ログに含まれていることを確認します。

たとえば、次の問合せを実行します。

```
SELECT OLDEST_FLASHBACK_SCN
FROM   V$FLASHBACK_DATABASE_LOG;
```

2. フラッシュバック・データベース操作のターゲット・インカネーション番号（親インカネーションのインカネーション・キー）を特定します。

たとえば、次の問合せを実行します。

```
SELECT PRIOR_INCARNATION#
FROM   V$DATABASE_INCARNATION
WHERE  STATUS = 'CURRENT';
```

3. Recovery Manager を起動し、ターゲット・データベースに接続します。
4. 次のように入力して、データベースを停止してからマウントします。

```
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
```

5. データベース・インカネーションを親インカネーションに設定します。

たとえば、次のコマンドを使用してインカネーション 1 に戻ります。

```
RESET DATABASE TO INCARNATION 1;
```

6. FLASHBACK DATABASE コマンドを実行し、目的の SCN を指定します。

たとえば、次のコマンドを使用して、データベースを SCN 1500 まで巻き戻します。

```
FLASHBACK DATABASE TO SCN 1500;
```

7. SQL\*Plus を使用してデータベースを読取り専用でオープンし、必要に応じて問合せを実行して、論理的な破損の影響が無効にされたことを確認します。

次のように入力して、データベースを読取り専用でオープンします。

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

8. データベースを再度更新できるようにするには、データベースを停止してマウントしてから、次のコマンドを実行します。

```
ALTER DATABASE OPEN RESETLOGS;
```

**参照：**

- データベースのインカネーション、取り消された変更、および ALTER DATABASE OPEN RESETLOGS の影響に関する有効な基礎知識については、13-6 ページの「データベース・インカネーション」を参照してください。
- RESET DATABASE コマンドの詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## 祖先インカネーションへのデータベースのリカバリ

現行のインカネーション内での DBPITR の手順は、現行以外のインカネーション内の SCN への DBPITR とは異なります。後者の場合は、明示的に RESET DATABASE を実行して、目的の SCN での現行インカネーションにデータベースを再設定する必要があります。また、目的の SCN が含まれているデータベース・インカネーションから制御ファイルをリストアする必要があります。

Recovery Manager がリカバリ・カタログに接続されている場合に RESTORE CONTROLFILE コマンドを実行すると、UNTIL 句で指定された時刻に最も近い時刻の現行のデータベース・インカネーションのみが検索されます。現行以外のインカネーションから制御ファイルをリストアするには、LIST INCARNATION を実行し、ターゲット・データベースのインカネーションを特定して、RESET DATABASE TO INCARNATION コマンドにこのインカネーションを指定する必要があります。

Recovery Manager がリカバリ・カタログに接続されている場合は、データベースをマウントする前に RESET DATABASE TO INCARNATION コマンドを実行することはできません。したがって、SET UNTIL を実行し、制御ファイルを自動バックアップからリストアしてマウントする必要があります。

次の状況を想定しています。

- Recovery Manager がリカバリ・カタログに接続されています。
- 2007 年 10 月 2 日にターゲット・データベース trgt のバックアップを実行しています。
- 2007 年 10 月 10 日にこのデータベースに対して DBPITR を実行し、以前のエラーを修正しています。DBPITR の最後に OPEN RESETLOGS 操作を実行して、新しいインカネーションを開始しています。

10 月 25 日、2007 年 10 月 8 日午前 8 時にデータベースから削除された重要なデータが必要であることが判明したとします。これは、現行のインカネーションの開始点より前の時刻です。

### DBPITR を現行以外のインカネーションに実行する手順

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログに接続します。
2. バックアップの時点での現行のデータベース・インカネーションを特定します。

LIST INCARNATION を使用して、目標時点での現行のインカネーションの主キーを取得します。

```
LIST INCARNATION OF DATABASE trgt;
```

```
List of Database Incarnations
```

| DB Key | Inc Key | DB Name | DB ID      | STATUS  | Reset SCN | Reset Time |
|--------|---------|---------|------------|---------|-----------|------------|
| 1      | 2       | TRGT    | 1224038686 | PARENT  | 1         | 02-OCT-06  |
| 1      | 582     | TRGT    | 1224038686 | CURRENT | 59727     | 10-OCT-06  |

Reset SCN および Reset Time 列を参照し、正しいインカネーションを特定して、Inc Key 列内のインカネーション・キーを書き留めます。この例では、バックアップは 2007 年 10 月 2 日に実行されました。この場合、インカネーション・キーの値は 2 です。

3. データベースは起動されているが、マウントされていないことを確認します。

```
STARTUP FORCE NOMOUNT
```

4. ターゲット・データベースを、手順 2 で取得したインカネーションに再設定します。

この例では、10月2日のバックアップ時点での現行インカネーションを指定します。  
Inc Key 列の値を使用して、インカネーションを識別します。

```
RESET DATABASE TO INCARNATION 2;
```

5. RUN コマンドで次の操作を実行して、データベースをリストアおよびリカバリします。

- 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。
- 構成されていない必要なチャンネルを割り当てます。
- 10月2日のバックアップから制御ファイルをリストアし、マウントします。
- データファイルをリストアし、データベースをリカバリします。RECOVER DATABASE ... UNTIL コマンドを使用して DBPITR を実行し、データが消失する直前の10月8日午前7時55分の目標時点にデータベースを戻します。

次の例では、この場合に必要なすべての手順を示しています。

```
RUN
{
  SET UNTIL TIME 'Oct 8 2007 07:55:00';
  RESTORE CONTROLFILE;
  # without recovery catalog, use RESTORE CONTROLFILE FROM AUTOBACKUP
  ALTER DATABASE MOUNT;
  RESTORE DATABASE;
  RECOVER DATABASE;
}
ALTER DATABASE OPEN RESETLOGS;
```

**参照：** RESET DATABASE コマンドの詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。



---

## データベースの完全リカバリの実行

この章では、1つ以上のデータファイルが消失した後、Recovery Manager を使用してデータベースを通常の稼働状態に戻す方法について説明します。この章の内容は、次のとおりです。

- データベースの完全リカバリの概要
- データベースの完全リカバリの準備
- データベースの完全リカバリの実行

## データベースの完全リカバリの概要

この項では、データベースの完全リストアおよびリカバリの目的およびこの章の概要について説明します。

### データベースの完全リカバリの目的

この章では、データファイルの一部またはすべてが消失または破損したと想定しています。通常、この状況は、メディア障害または誤った削除によって発生します。破損したファイルを Recovery Manager バックアップからリストアし、すべてのデータベース変更をリカバリすることによって、データベースを通常の稼働状態に戻す必要があります。

### この章の概要

この章では、完全リカバリを使用して最も一般的なデータベースの問題を解決する方法について説明します。この章では、次のことを想定しています。

- 一部またはすべてが消失したデータファイルがあり、すべての変更をリカバリすることが目標です。ただし、現行のすべての制御ファイルまたはオンライン REDO ログ・グループ全体は消失していません。

すべてのデータベースの変更ではなく、一部の変更をリカバリする方法については、[第 16 章「フラッシュバックおよびデータベースの Point-in-Time リカバリの実行」](#)を参照してください。[第 29 章「ユーザー管理のリカバリの実行: 高度な例」](#)では、現行のすべての制御ファイルではなく一部の制御ファイル、またはオンライン REDO ログ・グループのメンバーを消失したときに対応する方法について説明します。[19-5 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」](#)では、すべての制御ファイルを消失したときにデータベースをリカバリする方法について説明します。

- データベースでは、現行のサーバー・パラメータ・ファイルが使用されています。バックアップ・サーバー・パラメータ・ファイルをリストアするには、[19-2 ページの「サーバー・パラメータ・ファイルのリストア」](#)を参照してください。
- データファイルのバックアップのリカバリに必要なすべてのアーカイブ REDO ログおよび増分バックアップが存在します。すべてのデータファイルには、バックアップ、またはバックアップなしでデータファイルを作成した時点からのすべてのオンラインおよびアーカイブ REDO ログが存在します。

Recovery Manager は、リストアおよびリカバリ時に、消失したデータファイルを処理できます。ユーザーが手動で操作を行う必要はありません。データファイルが消失している場合、次のような状況が考えられます。

- 制御ファイルにデータファイルのレコードが含まれている（データファイルの作成後に制御ファイルをバックアップしているが、データファイル自体はバックアップしていない）。データファイルのレコードが制御ファイルに含まれている場合、RESTORE を実行すると、元の場所またはユーザーが指定する場所にデータファイルが作成されます。その後、RECOVER コマンドを実行すると、必要なログをデータファイルに適用できます。
  - 制御ファイルにデータファイルのレコードが含まれていない（データファイルの作成後に制御ファイルをバックアップしていない）。リカバリ時に、データベースで欠落したデータファイルが検出され、Recovery Manager に報告されます。これにより、新しいデータファイルが作成され、残りのログが適用されてリカバリが続行されます。データファイルが親インカネーション内に作成されている場合は、必要に応じて、リストアまたはリカバリ・フェーズ中にデータファイルが作成されます。
- 暗号化された表領域は、リストアおよびリカバリしません。

暗号化された表領域でメディア・リカバリを実行する場合は、この表領域のメディア・リカバリの実行時に、Oracle ウォレットをオープンする必要があります。暗号化された表領域については、『Oracle Database 管理者ガイド』を参照してください。

- データベースは、シングル・インスタンス構成で実行されています。  
Recovery Manager は、Oracle RAC および Data Guard 構成でデータベースをリストアおよびリカバリできますが、このマニュアルではこれらの例については説明しません。
- Oracle Enterprise Manager ではなく Recovery Manager クライアントを使用しています。  
Enterprise Manager では、一連のウィザードによって Recovery Manager にアクセスできます。これらのウィザードは、データベース、使用可能なバックアップおよびデータ・リカバリの目的の分析に基づいて、様々なリカバリ手順を提供します。  
Enterprise Manager を使用することによって、この章で説明する簡単なリストアおよびリカバリの例を実行できます。また、Point-in-Time リカバリやデータベースのフラッシュバックなどのより高度なリストアおよびリカバリ方法を使用して、メディア障害およびユーザー・エラーを効率的に修復することもできます。ほとんどの場合、Enterprise Manager のほうが Recovery Manager コマンドライン・クライアントを直接使用するより簡単です。

#### 参照：

- Oracle RAC での Recovery Manager の使用の詳細は、『Oracle Real Application Clusters 管理およびデプロイメント・ガイド』を参照してください。
- Data Guard での Recovery Manager の使用の詳細は、『Oracle Data Guard 概要および管理』を参照してください。
- Enterprise Manager のリストアおよびリカバリ機能の詳細は、『Oracle Database 2 日でデータベース管理者』を参照してください。

## データベースの完全リカバリの準備

Recovery Manager によってデータベースのほとんどのリストアおよびリカバリ・タスクは簡単になりますが、データベースのリストアおよびリカバリ計画は、消失したデータベース・ファイルおよびリカバリの目標に基づいて作成する必要があります。この項の内容は、次のとおりです。

- リストアまたはリカバリするデータファイルの識別
- データベースの DBID の確認
- リストア操作で使用されるバックアップのプレビュー
- リストアする前のバックアップの検証
- リカバリに必要なアーカイブ REDO ログのリストア

### リストアまたはリカバリするデータファイルの識別

リストアまたはリカバリが必要なファイルを判別する方法は、消失したファイルのタイプによって異なります。

#### 消失した制御ファイルの識別

通常、データベースの制御ファイルが消失した日時は明らかです。多重制御ファイルのいずれかがアクセス不可能になると、データベースは即時に停止します。また、CONTROL\_FILES 初期化パラメータに指定された有効な制御ファイルがそれぞれの場所に存在しない状態でデータベースを起動しようとした場合も、データベースによってエラーがレポートされます。

制御ファイルのコピーのすべてではなく一部が消失した場合は、バックアップから制御ファイルをリストアする必要はありません。影響を受けていない制御ファイルが1つ以上存在する場合は、影響を受けていない制御ファイルを破損または消失した制御ファイルにコピーするか、または破損または消失した制御ファイルを参照しないように初期化パラメータ・ファイルを更新することができます。存在し、影響を受けていない制御ファイルのコピーのみを CONTROL\_FILES パラメータが参照するようにした後で、データベースを再起動できます。

制御ファイルをバックアップからリストアする場合は、データファイルをリストアする必要がない場合でも、データベース全体のメディア・リカバリを実行してから、OPEN RESETLOGS オプションを使用してデータベースをオープンする必要があります。この方法については、19-5 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」を参照してください。

## メディア・リカバリが必要なデータファイルの識別

リカバリを実行するタイミングと方法は、データベースの状態とデータファイルの場所によって異なります。

**Recovery Manager を使用したデータファイルの識別** 欠落しているデータファイルは、VALIDATE DATABASE コマンドを実行すると簡単に確認できます。このコマンドでは、指定したすべてのデータファイルの読取りが試行されます。たとえば、Recovery Manager クライアントを起動し、次のコマンドを実行してデータベースを検証します（出力例も示します）。

### 例 17-1 BACKUP VALIDATE DATABASE

```
RMAN> VALIDATE DATABASE;
```

```
Starting validate at 20-OCT-06
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=90 device type=DISK
could not read file header for datafile 7 error reason 4
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of backup command at 10/20/2007 13:05:43
RMAN-06056: could not access datafile 7
```

例 17-1 の出力は、データファイル 7 がアクセス不可能であることを示しています。この場合は、REPORT SCHEMA コマンドを次のように実行して、データファイル 7 の表領域名およびファイル名を取得できます（出力例も示します）。

```
RMAN> REPORT SCHEMA;
```

```
Report of database schema for database with db_unique_name RDBMS
```

```
List of Permanent Datafiles
```

```
=====
```

| File | Size (MB) | Tablespace | RB segs | Datafile Name         |
|------|-----------|------------|---------|-----------------------|
| 1    | 450       | SYSTEM     | ***     | +DATAFILE/tbs_01.f    |
| 2    | 86        | SYSAUX     | ***     | +DATAFILE/tbs_ax1.f   |
| 3    | 15        | UD1        | ***     | +DATAFILE/tbs_undol.f |
| 4    | 2         | SYSTEM     | ***     | +DATAFILE/tbs_02.f    |
| 5    | 2         | TBS_1      | ***     | +DATAFILE/tbs_11.f    |
| 6    | 2         | TBS_1      | ***     | +DATAFILE/tbs_12.f    |
| 7    | 2         | TBS_2      | ***     | +DATAFILE/tbs_21.f    |

```
List of Temporary Files
```

```
=====
```

| File | Size (MB) | Tablespace | Maxsize (MB) | Tempfile Name        |
|------|-----------|------------|--------------|----------------------|
| 1    | 40        | TEMP       | 32767        | +DATAFILE/tbs_tmp1.f |



**SQLを使用したデータファイルの識別** VALIDATE DATABASE はファイルがアクセス不可能かどうかを確認する場合の最適な方法ですが、SQL 問合せを使用してより詳細な情報を取得する必要がある場合があります。

### データファイルにメディア・リカバリが必要かどうかを確認する手順

1. SQL\*Plus を起動し、管理者権限でターゲット・データベース・インスタンスに接続します。

2. 次の SQL 問合せを実行して、データベースのステータスを確認します。

```
SELECT STATUS FROM V$INSTANCE;
```

状態が OPEN の場合、データベースはオープン状態です。ただし、一部のデータファイルにメディア・リカバリが必要な場合があります。

3. V\$DATAFILE\_HEADER を問い合せて、データファイルのステータスを確認します。次の SQL 文を実行して、データファイルのヘッダーを確認します。

```
SELECT FILE#, STATUS, ERROR, RECOVER, TABLESPACE_NAME, NAME
FROM V$DATAFILE_HEADER
WHERE RECOVER = 'YES'
OR (RECOVER IS NULL AND ERROR IS NOT NULL);
```

戻される各行は、メディア・リカバリが必要なデータファイルまたはリストアが必要なエラー発生しているデータファイルを表します。RECOVER および ERROR 列を確認します。RECOVER には、ファイルにメディア・リカバリが必要かどうかを示され、ERROR には、データファイルのヘッダーの読取り時および検証時にエラーが発生したかどうかを示されます。

ERROR が NULL でない場合は、データファイルのヘッダーの読取りおよび検証は実行できません。エラーの原因となる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がないかどうかを確認します。問題がない場合は、ファイルをリストアしてコピーに切り替える必要があります。

ERROR 列が NULL で、RECOVER 列が YES の場合、ファイルにはメディア・リカバリが必要です（バックアップからリストアする必要がある場合もあります）。

---

**注意：** V\$DATAFILE\_HEADER は各データファイルのヘッダー・ブロックのみを読み取るため、データファイルのリストアが必要となるすべての問題を検出するわけではありません。たとえば、このビューではデータファイルに破損したデータ・ブロックが含まれているかどうかは確認できません。

---

4. 必要に応じて、V\$RECOVER\_FILE を問い合せて、リカバリが必要なデータファイルをそのステータス情報およびエラー情報とともにデータファイル番号で表示します。たとえば、次の問合せを実行します。

```
SELECT FILE#, ERROR, ONLINE_STATUS, CHANGE#, TIME
FROM V$RECOVER_FILE;
```

---

**注意：** V\$RECOVER\_FILE は、バックアップからリストアされた制御ファイル、またはデータファイルに影響を及ぼすメディア障害の発生後に再作成された制御ファイルには使用できません。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。

---

データファイルおよび表領域の名前を検出するために、データファイル番号、V\$DATAFILE および V\$TABLESPACE ビューを使用して、有効な結合を実行することもできます。たとえば、次のように入力します。

```
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#;
```

ERROR 列で、リカバリが必要な各ファイルに関する問題を識別します。

**参照：** V\$ ビューの詳細は、『Oracle Database リファレンス』を参照してください。

## データベースの DBID の確認

自動バックアップからのサーバー・パラメータ・ファイルまたは制御ファイルのリカバリが必要な状況では、DBID がわかっている必要があります。DBID は、データベースに関する他の基本情報とともに記録する必要があります。

データベースの DBID の記録がない場合は、データベースをオープンせずに次の場所で参照することができます。

- DBID は、制御ファイルの自動バックアップのファイル名の一部に使用されています。制御ファイルを検索した後、5-8 ページの「**制御ファイルの自動バックアップ書式の構成**」を参照して、ファイル名のどこに DBID が表示されるかを確認します。
- Recovery Manager セッションからの出力を保持するテキスト・ファイルがある場合、DBID は、Recovery Manager クライアントが起動してデータベースに接続する際に表示されます。通常の出力を次に示します。

```
% rman TARGET /
```

```
Recovery Manager: Release 11.1.0.6.0 - Production on Wed Jul 11 17:51:30 2007
```

```
Copyright (c) 1982, 2007, Oracle. All rights reserved.
```

```
connected to target database: PROD (DBID=36508508)
```

## リストア操作で使用されるバックアップのプレビュー

RESTORE ... PREVIEW を任意の RESTORE 操作に適用すると、要求された RESTORE 操作で使用するすべてのバックアップ、および RESTORE 操作完了後のリカバリに必要な目的の SCN の詳細なリストを作成できます。このコマンドは、Recovery Manager リポジトリにアクセスしてバックアップ・メタデータを問い合わせますが、実際にバックアップ・ファイルを読み取ってリストア可能であることは確認しません。

RESTORE ... PREVIEW の代替として、RESTORE ... VALIDATE HEADER コマンドを使用できます。RESTORE ... VALIDATE HEADER コマンドは、リストアおよびリカバリに必要なファイルを一覧表示する以外に、バックアップ・ファイル・ヘッダーを検証して、ディスク上またはメディア管理カタログ内のファイルが、Recovery Manager リポジトリのメタデータに対応しているかどうかを確認します。

リストアおよびリカバリ操作を計画する場合は、RESTORE ... PREVIEW または RESTORE ... VALIDATE HEADER を使用して、必要なすべてのバックアップが使用可能であることを確認するか、または特定のバックアップの使用または回避を Recovery Manager に指示する必要があります。

**リストア操作で使用されるバックアップをプレビューする手順**

1. PREVIEW オプションを指定して RESTORE コマンドを実行します。

たとえば、次のいずれかのコマンドを実行します。

```
RESTORE DATABASE PREVIEW;
RESTORE ARCHIVELOG FROM TIME 'SYSDATE-7' PREVIEW;
```

RESTORE ... PREVIEW によって生成されるレポートの情報が多すぎる場合は、次の例に示すように SUMMARY オプションを指定します。

```
RESTORE DATABASE PREVIEW SUMMARY;
```

出力が適切である場合は、ここで停止します。一時的に使用できないことがわかっているテープからのバックアップを Recovery Manager が要求することが出力に示された場合は、この手順を続行します。バックアップがオフサイトに格納されていることが出力に示された場合は、17-7 ページの「[オフサイト・バックアップのリコール](#)」に進みます。

2. 必要に応じて、CHANGE コマンドを使用して、一時的に使用できないバックアップのバックアップ・ステータスを UNAVAILABLE に設定します。

このタスクの実行方法については、11-15 ページの「[AVAILABLE または UNAVAILABLE へのバックアップのステータスの更新](#)」を参照してください。

3. 必要に応じて、RESTORE ... PREVIEW を再度実行して、使用できないバックアップがリストアで使用されないことを確認します。

**参照：** RESTORE ... PREVIEW の出力（LIST コマンドの出力と同じ書式）の解釈の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

**オフサイト・バックアップのリコール**

一部のメディア・マネージャは、オフサイトにあるバックアップに関するステータス情報を Recovery Manager に提供します。[オフサイトのバックアップ](#)は、安全なストレージ設備などのリモートの場所に格納され、メディア・マネージャがメディアを取り込まないかぎりリストアできません。

バックアップをリストアする前にストレージからメディアを取り込む必要がある場合でも、オフサイト・バックアップは、Recovery Manager リポジトリで AVAILABLE とマークが付けられます。Recovery Manager がオフサイト・バックアップのリストアを試みると、リストア・ジョブに失敗します。

RESTORE ... PREVIEW を使用すると、オフサイト・バックアップを確認できます。[例 17-2](#)のサンプル出力の最後にあるテキストに示すように、コマンドの出力には、バックアップがオフサイトに格納されたかどうかを示されます。

**例 17-2 RESTORE ... PREVIEW の出力**

List of Backup Sets

=====

| BS Key   | Size  | Device Type | Elapsed Time | Completion Time |
|--|-------|-------------|--------------|-----------------|
| 9  | 2.25M | SBT_TAPE    | 00:00:00     | 21-MAY-07       |
| BP Key: 9 Status: AVAILABLE Compressed: NO Tag: TAG20070521T144258 |       |             |              |                 |
| Handle: 0aai9k7i_1_1 Media: 0aai9k7i_1_1                           |       |             |              |                 |

List of Archived Logs in backup set 9

| Thrd | Seq | Low SCN | Low Time  | Next SCN | Next Time |
|------|-----|---------|-----------|----------|-----------|
| 1    | 1   | 392314  | 21-MAY-07 | 392541   | 21-MAY-07 |
| 1    | 2   | 392541  | 21-MAY-07 | 392545   | 21-MAY-07 |
| 1    | 3   | 392545  | 21-MAY-07 | 392548   | 21-MAY-07 |

```

1    4    392548    21-MAY-07 395066    21-MAY-07
1    5    395066    21-MAY-07 395095    21-MAY-07
1    6    395095    21-MAY-07 395355    21-MAY-07

```

List of remote backup files

=====

```

      Handle: aii9k7i_1_1    Media: 0aii9k7i_1_1
validation succeeded for backup piece
Finished restore at 21-MAY-07
released channel: dev1

```

RESTORE ... PREVIEW RECALL を使用すると、オフサイト・バックアップを使用可能にするようにメディア・マネージャに指示できます。

### オフサイト・バックアップをリコールする手順

1. バックアップがオフサイトに格納されている場合、RECALL オプションを指定して RESTORE ... PREVIEW コマンドを実行します。

次の例では、例 17-2 に示されているオフサイト・アーカイブ・ログのバックアップのリコールを開始します（出力例も示します）。

```
RESTORE ARCHIVELOG ALL PREVIEW RECALL;
```

次のサンプル出力は、Recovery Manager がリコールを開始したことを示しています。

List of Backup Sets

=====

```

BS Key  Size      Device Type Elapsed Time Completion Time
-----
9       2.25M    SBT_TAPE    00:00:00    21-MAY-07
      BP Key: 9    Status: AVAILABLE Compressed: NO Tag: TAG20070521T144258
      Handle: VAULT0aii9k7i_1_1    Media: /tmp,VAULT0aii9k7i_1_1

```

List of Archived Logs in backup set 9

```

Thrd Seq    Low SCN    Low Time    Next SCN    Next Time
-----
1    1        392314    21-MAY-07 392541    21-MAY-07
1    2        392541    21-MAY-07 392545    21-MAY-07
1    3        392545    21-MAY-07 392548    21-MAY-07
1    4        392548    21-MAY-07 395066    21-MAY-07
1    5        395066    21-MAY-07 395095    21-MAY-07
1    6        395095    21-MAY-07 395355    21-MAY-07

```

Initiated recall for the following list of remote backup files

=====

```

      Handle: VAULT0aii9k7i_1_1    Media: /tmp,VAULT0aii9k7i_1_1
validation succeeded for backup piece
Finished restore at 21-MAY-07
released channel: dev1

```

2. RESTORE ... PREVIEW コマンドを実行します。必要に応じて、リストアに必要なバックアップがオフサイトとしてレポートされなくなるまで、前の手順に戻ります。

## リストアする前のバックアップの検証

17-6 ページの「リストア操作で使用するバックアップのプレビュー」の手順では、リストアされるバックアップは示されますが、それらのバックアップが実際に使用可能であることは検証されません。Recovery Manager コマンドを実行して、RESTORE 操作で使用可能なバックアップの可用性または RESTORE 操作で使用する特定のバックアップの内容をテストできます。バックアップの内容が実際に読み取られ、破損が確認されます。次の検証オプションがあります。

- RESTORE ... VALIDATE を実行すると、Recovery Manager で特定のオブジェクトをバックアップからリストアできるかどうかをテストされます。Recovery Manager によって、使用するバックアップが選択されます。
- VALIDATE BACKUPSET は、指定するバックアップ・セットの妥当性をテストします。

参照: [第 15 章「データベース・ファイルおよびバックアップの検証」](#)

## リカバリに必要なアーカイブ REDO ログのリストア

Recovery Manager は、リカバリを実行するために、必要に応じてアーカイブ REDO ログ・ファイルをバックアップから自動的にリストアします。また、後で RECOVER コマンド中にこれらのファイルのリストアに必要な時間を節約するために、またはリストアされたアーカイブ REDO ログ・ファイルを新しい場所に格納する場合に、アーカイブ REDO ログを手動でリストアすることもできます。

デフォルトでは、Recovery Manager は、ターゲット・データベースの LOG\_ARCHIVE\_FORMAT および LOG\_ARCHIVE\_DEST\_1 パラメータを使用して構成された名前アーカイブ REDO ログをリストアします。これらのパラメータがプラットフォーム固有の形式で組み合わせられ、リストアされたアーカイブ・ログの名前が構成されます。

### 新しい場所へのアーカイブ REDO ログのリストア

SET ARCHIVELOG DESTINATION コマンドを使用すると、リストアしたアーカイブ REDO ログのデフォルトの名前を上書きできます。このコマンドによって、アーカイブ・ログは、データベースのリストア中に別の場所に手動でステージングされます。リカバリ中、Recovery Manager は新しくリストアされたアーカイブ・ログを検索する場所を認識するため、初期化パラメータ・ファイルで指定された場所にアーカイブ・ログが存在している必要はありません。

### アーカイブ REDO ログを新しい場所へリストアする手順

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. 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。
3. RUN コマンド内で、次の操作を実行します。
  - a. SET ARCHIVELOG DESTINATION を使用して、リストアするアーカイブ REDO ログの新しい場所を指定します。
  - b. アーカイブ REDO ログを明示的にリストアするか、またはログを自動的にリストアするコマンドを実行します。

次の例の RUN コマンドは、すべてのバックアップ・アーカイブ・ログを新しい場所に明示的にリストアします。

```
RUN
{
  SET ARCHIVELOG DESTINATION TO '/oracle/temp_restore';
  RESTORE ARCHIVELOG ALL;
  # restore and recover datafiles as needed
  .
  .
  .
}
```

次の例では、アーカイブ・ログの転送先を設定し、RECOVER DATABASE を使用してアーカイブ・ログをこの転送先から自動的にリストアします。

```
RUN
{
  SET ARCHIVELOG DESTINATION TO '/oracle/temp_restore';
  RESTORE DATABASE;
  RECOVER DATABASE; # restores and recovers logs automatically
}
```

### 複数の場所へのアーカイブ REDO ログのリストア

アーカイブ・ログのリストア先を1つの **RUN ブロック** に複数回指定して、リストアしたログを複数のリストア先に分散できます。(ただし、同時に複数のリストア先を指定して、リストア操作時に同じログの複数のコピーを生成することはできません。) この機能を使用して、リストアしたログを格納するために使用するディスク領域を管理できます。

次の例では、バックアップから 300 のアーカイブ REDO ログをリストアし、ディレクトリ /fs1/tmp、/fs2/tmp および /fs3/tmp に分散します。

```
RUN
{
  # Set a new location for logs 1 through 100.
  SET ARCHIVELOG DESTINATION TO '/fs1/tmp';
  RESTORE ARCHIVELOG FROM SEQUENCE 1 UNTIL SEQUENCE 100;
  # Set a new location for logs 101 through 200.
  SET ARCHIVELOG DESTINATION TO '/fs2/tmp';
  RESTORE ARCHIVELOG FROM SEQUENCE 101 UNTIL SEQUENCE 200;
  # Set a new location for logs 201 through 300.
  SET ARCHIVELOG DESTINATION TO '/fs3/tmp';
  RESTORE ARCHIVELOG FROM SEQUENCE 201 UNTIL SEQUENCE 300;
  # restore and recover datafiles as needed
  .
  .
}
```

RECOVER コマンドを発行すると、リストアした必要なアーカイブ・ログがリストア先で自動的に検出され、データファイルに適用されます。

## データベースの完全リカバリの実行

この項では、17-3 ページの「データベースの完全リカバリの準備」のタスクをすでに実行していると想定しています。この項では、広範囲の様々な例に対応することを目的としているデータベースの完全リカバリの概要について説明します。

### データベースの完全リカバリ

データベースをリストアおよびリカバリするには、RESTORE および RECOVER コマンドを使用します。リカバリ中、Recovery Manager は、必要なアーカイブ REDO ログのバックアップを自動的にリストアします。バックアップがメディア・マネージャに格納されている場合は、チャンネルを事前に構成しておくか、または ALLOCATE CHANNEL コマンドを含む RUN ブロックを使用して、そこに格納されているバックアップへのアクセスを有効にする必要があります。

Recovery Manager は、リカバリ中にアーカイブ REDO ログをフラッシュ・リカバリ領域にリストアする場合、リストアしたログを、データファイルに適用した後で自動的に削除します。そうでない場合は、リストアしたアーカイブ REDO ログがリカバリに必要ななくなった後、DELETE ARCHIVELOG コマンドを使用して、それらのアーカイブ REDO ログをディスクから削除できます。たとえば、次のコマンドを入力できます。

```
RECOVER DATABASE DELETE ARCHIVELOG;
```

## デフォルト以外の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は、データファイルの新しい場所が反映されるように、制御ファイルを更新する必要があります。Recovery Manager の SET NEWNAME コマンドを RUN コマンド内で使用して、新しいファイル名を指定します。その後、SQL 文 ALTER DATABASE RENAME FILE と同じ動作をする SWITCH コマンドを使用し、制御ファイル内のデータファイルの名前を更新します。SWITCH DATAFILE ALL を実行すると、制御ファイルに、RUN コマンドで SET NEWNAME が発行されているすべてのデータファイルの新しい名前が反映されます。

**参照：** SWITCH 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## バックアップの復号化

Recovery Manager は、暗号化バックアップをリストアする場合、バックアップ・セットの内容をリストアする際にバックアップ・セットを自動的に復号化します。透過的に暗号化されたバックアップは、Oracle ウォレットがオープンしていて使用可能な場合にかぎり、ユーザーが介入せずにリストアできます。

パスワード暗号化バックアップをリストアするには、正しいパスワードを入力する必要があります。SET DECRYPTION コマンドを使用して、暗号化パスワードを入力する必要があります。異なるパスワードを使用して作成されたバックアップのグループからリストアを行う場合は、SET DECRYPTION コマンドで、必要なすべてのパスワードを指定する必要があります。Recovery Manager では、各バックアップ・セットに対して正しいパスワードが自動的に使用されます。

**参照：**

- バックアップの透過的暗号化を構成する方法の詳細は、6-7 ページの「バックアップの暗号化の構成」を参照してください。
- 暗号化バックアップを作成する方法については、9-11 ページの「Recovery Manager バックアップの暗号化」を参照してください。

## データベース全体の完全リカバリの実行

この例では、データベース trgt のほとんどまたはすべてのデータファイルが消失したと想定しています。また、データベースでフラッシュ・リカバリ領域が使用されているとも想定しています。

データベース全体のリストアおよびリカバリの後で、データベースがオープンされると、制御ファイルに記録された欠落している一時表領域は、以前の作成サイズ、AUTOEXTEND および MAXSIZE 属性で再作成されます。欠落している一時表領域のみが再作成されます。Recovery Manager リポジトリに記録された場所に存在する一時ファイルのヘッダーが無効な場合、Recovery Manager は一時ファイルを再作成しません。

一時ファイルが Oracle 管理ファイルとして作成されていた場合は、現在の DB\_CREATE\_FILE\_DEST の場所に再作成されます。そうでない場合は、以前の場所で再作成されます。Recovery Manager が I/O エラーまたは他の原因でファイルを再作成できない場合、エラーがアラート・ログにレポートされ、データベースのオープン操作が続行されます。

### データベース全体をリストアおよびリカバリする手順

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、次のコマンドを入力します。

```
% rman
RMAN> CONNECT TARGET /
```

Recovery Manager は、接続時に not started、not mounted、not open (データベースはマウントされているがオープンされていない場合) または none (データベースがオープンされている場合) のいずれかのデータベース・ステータスを表示します。

2. データベースがマウントされていない場合は、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。

たとえば、次のコマンドを入力します。

```
STARTUP MOUNT;
```

3. SHOW コマンドを使用して、事前構成済のチャネルを確認します。

たとえば、次のコマンドを入力します（出力例も示します）。

```
SHOW ALL;
```

```
RMAN configuration parameters for database with db_unique_name PROD1 are:
```

```
.
.
.
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
CONFIGURE DEVICE TYPE SBT_TAPE PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMS "SBT_
LIBRARY=/usr/local/oracle/backup/lib/libobk.so";
```

必要なデバイスおよびチャネルがすでに構成されている場合、操作は必要ありません。そうでない場合は、CONFIGURE コマンドを使用して自動チャネルを構成するか、または RUN ブロック内に ALLOCATE CHANNEL コマンドを含めます。

4. パスワード保護された暗号化バックアップをリストアする場合は、パスワードを指定します。

次の例に示すように、SET DECRYPTION IDENTIFIED BY コマンドを使用して、パスワード保護されたバックアップのパスワードを指定します (*password* は実際に入力するパスワードを表します)。

```
SET DECRYPTION IDENTIFIED BY password;
```

異なるパスワードでバックアップを作成した場合は、バックアップのリストアに必要な可能性のあるすべてのパスワードを指定して SET DECRYPTION IDENTIFIED BY *password* コマンドを複数回実行できます。

5. データベースをリストアおよびリカバリします。次のいずれかの処理を実行します。
  - すべてのデータファイルを元の場所にリストアする場合は、Recovery Manager プロンプトで RESTORE DATABASE と RECOVER DATABASE を順に実行します。

たとえば、自動チャネルが構成されている場合は、次のコマンドを入力します（出力例も示します）。

```
RMAN> RESTORE DATABASE;
```

```
Starting restore at 20-JUN-06
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=35 device type=DISK
allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=34 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: Oracle Secure Backup

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
channel ORA_DISK_1: restoring datafile 00001 to /disk1/oracle/dbs/tbs_01.f
channel ORA_DISK_1: restoring datafile 00002 to /disk1/oracle/dbs/tbs_ax1.f
.
.
.
```



```
Finished restore at 20-JUN-06
```

```
RMAN> RECOVER DATABASE;
```

```
Starting recover at 20-JUN-06
```

```
using channel ORA_DISK_1
```

```
allocated channel: ORA_SBT_TAPE_1
```

```
channel ORA_SBT_TAPE_1: SID=34 device type=SBT_TAPE
```

```
channel ORA_SBT_TAPE_1: Oracle Secure Backup
```

```
starting media recovery
```

```
channel ORA_DISK_1: starting archived log restore to default destination
```

```
channel ORA_DISK_1: restoring archived log
```

```
archived log thread=1 sequence=5
```

```
channel ORA_DISK_1: restoring archived log
```

```
archived log thread=1 sequence=6
```

```
.
```

```
.
```

```
.
```

```
channel ORA_DISK_1: reading from backup piece
```

```
/disk1/oracle/work/orcva/TKRM/backupset/2007_06_20/o1_mf_anmnn_
```

```
TAG20070620T113128_29jhr197_.bkp
```

```
channel ORA_DISK_1: piece handle=/disk1/oracle/work/orcva/TKRM/backupset/2007_
```

```
06_20/o1_mf_anmnn_TAG20070620T113128_29jhr197_.bkp tag=TAG20070620T113128
```

```
channel ORA_DISK_1: restored backup piece 1
```

```
channel ORA_DISK_1: restore complete, elapsed time: 00:00:02
```

```
archived log file name=/disk1/oracle/work/orcva/TKRM/archivelog/2007_06_20/o1_
```

```
mf_1_5_29jhr47k_.arc thread=1 sequence=5
```

```
channel default: deleting archived log(s)
```

```
.
```

```
.
```

```
.
```

```
media recovery complete, elapsed time: 00:00:15
```

```
Finished recover at 20-JUN-06
```

チャンネルを手動で割り当てる場合は、RESTORE および RECOVER コマンドを RUN ブロック内で同時に発行する必要があります。次に例を示します。

```
RUN
```

```
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt;
  RESTORE DATABASE;
  RECOVER DATABASE;
}
```

- 一部のデータファイルを新しい場所にリストアする場合は、RUN コマンドで RESTORE DATABASE と RECOVER DATABASE を順に実行します。17-11 ページの「[デフォルト以外の場所へのデータファイルのリストア](#)」で説明するように、SET NEWNAME を使用して、データファイルの名前を変更します。

次の例では、データベースをリストアし、3つのデータファイルに新しい名前を指定してから、データベースをリカバリします。

```
RUN
```

```
{
  SET NEWNAME FOR DATAFILE 2 TO '/disk2/df2.dbf';
  SET NEWNAME FOR DATAFILE 3 TO '/disk2/df3.dbf';
  SET NEWNAME FOR DATAFILE 4 TO '/disk2/df4.dbf';
  RESTORE DATABASE;
  SWITCH DATAFILE ALL;
  RECOVER DATABASE;
}
```

- 出力を調べて、メディア・リカバリが正常に実行されたかどうかを確認します。正常に実行された場合は、データベースをオープンします。

たとえば、次のコマンドを入力します。

```
ALTER DATABASE OPEN;
```

## 表領域の完全リカバリの実行

基本的な例では、データベースはオープンされており、データファイルのすべてではなく一部が破損しています。データベースの残りの部分を使用可能にしておくために、データベースはオープンしたままで、破損した表領域をリストアおよびリカバリします。この例では、データベース `trgt` の表領域 `users` が消失したと想定しています。

### 表領域をリストアおよびリカバリする手順

- Recovery Manager を起動し、ターゲット・データベースに接続します。
- データベースがオープンされている場合は、リカバリが必要なデータファイルをオフラインにします。

たとえば、次のコマンドを入力して `users` をオフラインにします。

```
SQL "ALTER TABLESPACE users OFFLINE IMMEDIATE";
```

- SHOW コマンドを使用して、事前構成済のチャネルを確認します。

たとえば、次のコマンドを入力します（出力例も示します）。

```
SHOW ALL;
```

```
RMAN configuration parameters for database with db_unique_name PROD1 are:
```

```
.  
.
.
```

```
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default  
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default  
CONFIGURE DEVICE TYPE SBT_TAPE PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default  
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMS "SBT_  
LIBRARY=/usr/local/oracle/backup/lib/libobk.so";
```

必要なデバイスおよびチャネルがすでに構成されている場合、操作は必要ありません。そうでない場合は、CONFIGURE コマンドを使用して自動チャネルを構成するか、または RUN ブロック内に ALLOCATE CHANNEL コマンドを含めます。

- パスワード保護された暗号化バックアップをリストアする場合は、パスワードを指定します。

次の例に示すように、SET DECRYPTION IDENTIFIED BY コマンドを使用して、パスワード保護されたバックアップのパスワードを指定します (`password` は実際に入力するパスワードを表します)。

```
SET DECRYPTION IDENTIFIED BY password;
```

- 表領域をリストアおよびリカバリします。次のいずれかの処理を実行します。

- データファイルを元の場所にリストアする場合は、Recovery Manager プロンプトで RESTORE TABLESPACE と RECOVER TABLESPACE コマンドを実行します。

たとえば、自動チャネルが構成されている場合は、次のコマンドを入力します（出力例も示します）。

```
RMAN> RESTORE TABLESPACE users;
```

```
Starting restore at 20-JUN-06  
allocated channel: ORA_DISK_1  
channel ORA_DISK_1: SID=37 device type=DISK  
allocated channel: ORA_SBT_TAPE_1
```

```
channel ORA_SBT_TAPE_1: SID=38 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: Oracle Secure Backup

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
channel ORA_DISK_1: restoring datafile 00012 to /disk1/oracle/dbs/users01.f
channel ORA_DISK_1: restoring datafile 00013 to /disk1/oracle/dbs/users02.f
channel ORA_DISK_1: restoring datafile 00021 to /disk1/oracle/dbs/users03.f
channel ORA_DISK_1: reading from backup piece
/disk1/oracle/work/orcva/TKRM/backupset/2007_06_20/o1_mf_nnndf_
TAG20070620T105435_29jflwor_.bkp
channel ORA_DISK_1: piece handle=/disk1/oracle/work/orcva/TKRM/backupset/2007_
06_20/o1_mf_nnndf_TAG20070620T105435_29jflwor_.bkp tag=TAG20070620T105435
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished restore at 20-JUN-06

RMAN> RECOVER TABLESPACE users;

Starting recover at 20-JUN-06
using channel ORA_DISK_1
using channel ORA_SBT_TAPE_1

starting media recovery

archived log for thread 1 with sequence 27 is already on disk as file
/disk1/oracle/work/orcva/TKRM/archivelog/2007_06_20/o1_mf_1_27_29jjmtc9_.arc
archived log for thread 1 with sequence 28 is already on disk as file
/disk1/oracle/work/orcva/TKRM/archivelog/2007_06_20/o1_mf_1_28_29jjnc5x_.arc
.
.
.
channel ORA_DISK_1: starting archived log restore to default destination
channel ORA_DISK_1: restoring archived log
archived log thread=1 sequence=5
channel ORA_DISK_1: restoring archived log
archived log thread=1 sequence=6
channel ORA_DISK_1: restoring archived log
archived log thread=1 sequence=7
.
.
.
channel ORA_DISK_1: reading from backup piece
/disk1/oracle/work/orcva/TKRM/backupset/2007_06_20/o1_mf_anmn_
TAG20070620T113128_29jhr197_.bkp
channel ORA_DISK_1: piece handle=/disk1/oracle/work/orcva/TKRM/backupset/2007_
06_20/o1_mf_anmn_TAG20070620T113128_29jhr197_.bkp tag=TAG20070620T113128
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:02
archived log file name=/disk1/oracle/work/orcva/TKRM/archivelog/2007_06_20/o1_
mf_1_5_29jkdvjq_.arc thread=1 sequence=5
channel default: deleting archived log(s)
archived log file name=/disk1/oracle/work/orcva/TKRM/archivelog/2007_06_20/o1_
mf_1_5_29jkdvjq_.arc RECID=91 STAMP=593611179
archived log file name=/disk1/oracle/work/orcva/TKRM/archivelog/2007_06_20/o1_
mf_1_6_29jkdvbz_.arc thread=1 sequence=6
channel default: deleting archived log(s)
.
.
.
media recovery complete, elapsed time: 00:00:01
Finished recover at 20-JUN-06
```

- 一部のデータファイルを新しい場所にリストアする場合は、RUN コマンドで RESTORE TABLESPACE と RECOVER TABLESPACE を実行します。17-11 ページの「[デフォルト以外の場所へのデータファイルのリストア](#)」で説明するように、SET NEWNAME を使用して、データファイルの名前を変更します。

次の例では、表領域 users 内の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。古いデータファイルは /disk1 パスに格納されており、新しいデータファイルは /disk2 パスに格納されると想定しています。

```
RUN
{
  # specify the new location for each datafile
  SET NEWNAME FOR DATAFILE '/disk1/oracle/dbs/users01.f' TO
    '/disk2/users01.f';
  SET NEWNAME FOR DATAFILE '/disk1/oracle/dbs/users02.f' TO
    '/disk2/users02.f';
  SET NEWNAME FOR DATAFILE '/disk1/oracle/dbs/users03.f' TO
    '/disk2/users03.f';

  RESTORE TABLESPACE users;
  SWITCH DATAFILE ALL; # update control file with new filenames
  RECOVER TABLESPACE users;
}
```

6. 出力を調べて、リカバリが正常に実行されたことを確認します。正常に実行された場合は、リカバリした表領域をオンラインに戻します。

たとえば、次のコマンドを入力します。

```
SQL "ALTER TABLESPACE users ONLINE";
```

## コピーへの切替え後の完全リカバリの実行

アクセスできないデータファイルのイメージ・コピーがフラッシュ・リカバリ領域に存在する場合、SWITCH DATAFILE ... TO COPY コマンドを使用して制御ファイルをデータファイルのコピーにポイントしてから RECOVER を使用すると、消失した変更をリカバリできます。SWITCH DATABASE TO COPY コマンドを使用して、制御ファイルをデータベース全体のコピーにポイントすることもできます。バックアップをリストアする必要がないため、このリカバリ方法は従来のリストアおよびリカバリより時間がかかりません。

---

---

**注意：** SWITCH TABLESPACE ... TO COPY コマンドは、表領域内のすべてのデータファイルが消失し、すべてのデータファイルのコピーが存在する場合でもサポートされます。SWITCH DATABASE TO COPY についても同じ制限があります。

---

---

### データファイルのコピーへの切替え

基本的な例では、データベースはオープンされており、データファイルのすべてではなく一部が破損しています。1 日の中で、あるデータファイルがストレージ障害によって欠落します。このファイルを修復する必要がありますが、バックアップからリストアおよびリカバリする時間はありません。最新のイメージ・コピー・バックアップを新しいファイルとして使用して、リストア時間をなくします。この例では、データベース trgt のデータファイル 4 が消失したと想定しています。

#### データファイルのコピーに切り替えてリカバリを実行する手順

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. データベースがオープンされている場合は、リカバリが必要な表領域をオフラインにします。

次のコマンドを入力してデータファイル 4 をオフラインにします。

```
SQL "ALTER DATABASE DATAFILE 4 OFFLINE";
```

3. オフラインのデータファイルを最新のコピーに切り替えます。

次のコマンドを入力して、制御ファイルをデータファイル 4 の最新のイメージ・コピーにポイントします。

```
SWITCH DATAFILE 4 TO COPY;
```

4. RECOVER DATAFILE コマンドで、データファイルをリカバリします。

次のコマンドを入力します。

```
RECOVER DATAFILE 4;
```

Recovery Manager は、アーカイブ REDO ログおよび増分バックアップを自動的にリストアします。データベースではフラッシュ・リカバリ領域が使用されるため、Recovery Manager は、これらを、適用した後に自動的に削除します。

5. 出力を調べて、リカバリが正常に実行されたことを確認します。正常に実行された場合は、リカバリしたデータファイルをオンラインに戻します。

次のコマンドを入力してデータファイル 4 をオンラインにします。

```
SQL "ALTER DATABASE DATAFILE 4 ONLINE";
```

## データベースのコピーへの切替え

この例では、データベースは停止しており、すべてのデータファイルが破損しています。破損したすべてのデータファイルには、イメージ・コピーがあります。既存のイメージ・コピーを新しいデータファイルとして使用して、リストア時間をなくします。

### データベースのコピーに切り替えてリカバリを実行する手順

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. データベースをマウントします。

3. データベースを最新のコピーに切り替えます。

次のコマンドを入力して、制御ファイルをデータベースの最新のイメージ・コピーにポイントします。

```
SWITCH DATABASE TO COPY;
```

4. RECOVER DATABASE コマンドで、データベースをリカバリします。

次のコマンドを入力します。

```
RECOVER DATABASE;
```

Recovery Manager は、アーカイブ REDO ログおよび増分バックアップを自動的にリストアします。データベースではフラッシュ・リカバリ領域が使用されるため、Recovery Manager は、これらを、適用した後に自動的に削除します。

5. 出力を調べて、リカバリが正常に実行されたことを確認します。正常に実行された場合は、データベースをオープンします。

データベースをオープンするには、次のコマンドを入力します。

```
ALTER DATABASE OPEN;
```



---

## ブロック・メディア・リカバリの実行

この章では、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する方法について説明します。この章の内容は、次のとおりです。

- [ブロック・メディア・リカバリの概要](#)
- [ブロック・メディア・リカバリの前提条件](#)
- [個々のブロックのリカバリ](#)
- [V\\$DATABASE\\_BLOCK\\_CORRUPTION 内のすべてのブロックのリカバリ](#)

**参照：**

- RECOVER 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
- V\$DATABASE\_BLOCK\_CORRUPTION ビューの詳細は、『Oracle Database リファレンス』を参照してください。

## ブロック・メディア・リカバリの概要

この項では、ブロック・メディア・リカバリの目的および基本的な概念について説明します。

### ブロック・メディア・リカバリの目的

**ブロック・メディア・リカバリ**を使用すると、データファイル内の1つ以上の破損したデータ・ブロックをリカバリできます。ブロック・メディア・リカバリには、**データファイルのメディア・リカバリ**にはない次のようなメリットがあります。

- リカバリが必要なブロックのみがリストアおよびリカバリされるため、**平均リカバリ時間**が小さくなります。
- リカバリ中、影響を受けるデータファイルをオンラインのままにしておくことができます。

ブロック・メディア・リカバリを使用しないと、1つのブロックが破損した場合でも、データファイルをオフラインにしてデータファイルのバックアップをリストアする必要があります。バックアップの作成後にデータファイルに対して生成されたすべての REDO を適用する必要があります。メディア・リカバリが完了するまで、ファイル全体が使用不可となります。ブロック・メディア・リカバリを使用すると、実際にリカバリされているブロックのみがリカバリ中に使用不可となります。

ブロック・メディア・リカバリは、ブロックが少数でその数がわかっている場合に発生する**物理的な破損**の問題に特に有効です。ブロックレベルのデータ消失は、通常、広範囲なデータ消失は引き起こさない断続的でランダムな I/O エラーや、破損したメモリーがディスクに書き込まれることが原因で発生します。ブロック・メディア・リカバリは、データの消失または破損の程度が不明で、データファイル全体のリカバリが必要な場合には適していません。このような場合は、データファイルのメディア・リカバリが最適です。

### ブロック・メディア・リカバリの基本的な概念

ほとんどの場合、データベースは、破損が最初に発生したときに、ブロックをメディア破損とマークし、ディスクに書き込みます。それ以降のブロックの読取りは、ブロックがリカバリされるまで正常に実行できません。ブロック・リカバリは、破損とマークされたブロックまたは破損チェックが失敗したブロックに対してのみ実行できます。

ブロック・メディア・リカバリは、RECOVER ... BLOCK コマンドを使用して実行します。デフォルトでは、Recovery Manager は、ブロックの完全な状態に近いコピーのフラッシュバック・ログを検索し、次に全体またはレベル 0 の増分バックアップ内のブロックを検索します。Recovery Manager は、完全な状態に近いコピーを検出すると、それらをリストアし、ブロックに対してメディア・リカバリを実行します。ブロック・メディア・リカバリでは、レベル 1 の増分バックアップではなく、メディア・リカバリの REDO ログのみを使用できます。

**参照：** RECOVER ... BLOCK 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

### 破損ブロックの識別

V\$DATABASE\_BLOCK\_CORRUPTION ビューに、Recovery Manager、ANALYZE、dbv、SQL 問合せなどのデータベース・コンポーネントによって破損とマークされたブロックが表示されます。次のタイプの破損が、このビューに行として追加されます。

- 物理的な破損（メディア破損とも呼ばれる）  
ブロックがデータベースで認識されません。**チェックサム**が無効か、ブロックの内容がすべて 0（ゼロ）か、またはブロック・ヘッダーが分裂しています。  
物理的な破損のチェックは、デフォルトで有効になっています。BACKUP コマンドの NOCHECKSUM オプションを指定することによってチェックサムのチェックを無効にすることはできますが、ブロック・ヘッダーやフッターのチェックなどの他の物理的な一貫性チェックを無効にすることはできません。



■ 論理的な破損

ブロックのチェックサムが有効で、ヘッダーとフッターが一致している点などは正常ですが、内容が論理的に一貫性のない状態になっています。ブロック・メディア・リカバリでは、論理的なブロック破損は修復できません。

論理的な破損のチェックは、デフォルトでは無効になっています。BACKUP、RESTORE、RECOVER および VALIDATE コマンドの CHECK LOGICAL オプションを指定することによって、有効にすることができます。

データベースは、ブロックとセグメント間の関係を検証することによって一部の破損を検出することはできますが、個々のブロックのチェックによって検出することはできません。V\$DATABASE\_BLOCK\_CORRUPTION ビューに、このタイプの破損は記録されません。

**ブロック・リカバリ時に欠落している REDO**

データファイルのメディア・リカバリと同様に、ブロック・メディア・リカバリでも通常は、アーカイブ・ログが欠落しているか、またはアクセスできない場合はリカバリを実行できません。ただし、アーカイブ REDO ログ・ファイルの使用可能なコピーが検出された場合は、リストア・フェイルオーバーが試行されます (13-4 ページの「リストア・フェイルオーバー」を参照)。また、ブロック・メディア・リカバリでは、**チェックサム**障害となる物理的な REDO の破損がある場合はリカバリを実行できません。ただし、ブロック・メディア・リカバリでは、REDO ストリームが連続していない場合に、欠落または破損している REDO レコードがリカバリするブロックに影響しなければ、リカバリを実行できます。データファイルのリカバリには、リカバリの開始から終了まで連続した一連の REDO 変更が必要ですが、ブロック・メディア・リカバリでは、リカバリするブロックに対してのみ、連続した一連の REDO 変更が必要です。

---

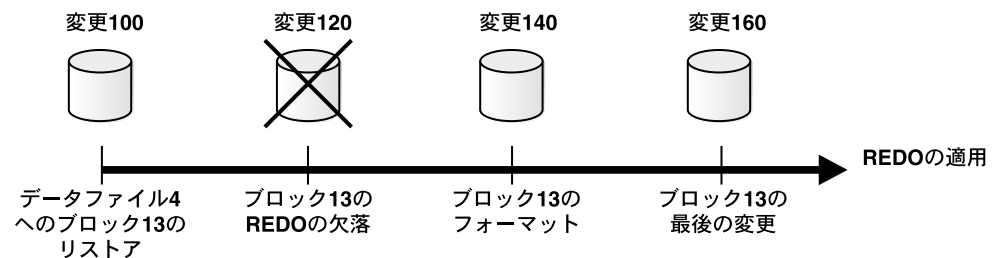
**注意：** ブロック・メディア・リカバリ中、各ブロックは別々にリカバリされるため、一部のブロックでのみリカバリが正常に行われる場合があります。

---

Recovery Manager は、ブロック・メディア・リカバリ中に欠落または破損している REDO レコードを初めて検出した際、すぐにはエラーを通知しません。これは、リカバリ中のブロックがこの後の REDO ストリームで**更新ブロック**になる場合があるためです。ブロックが更新されると、それ以前の REDO はブロックの古いインカンネーションに適用されていたことになるため、それまでのすべての REDO は無関係になります。たとえば、ユーザーが表を削除するか、または切り捨ててから、ブロックを他のデータに対して使用した場合、データベースはブロックを更新できます。

次の図に示すように、メディア・リカバリがブロック 13 で実行されるとします。

**図 18-1 Recovery Manager のメディア・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 が REDO ストリームから欠落していることを検出します。ログ・ブロックが破損しているか、またはログを検出できないためです。ブロック 13 はこの後の REDO ストリームで更新される可能性があるため、Recovery Manager はリカバリを続行します。変更 140 で、ユーザーが、ブロック 13 に格納されている employees 表を削除し、このブロックで新しい表を割り当て、新しい表にデータを挿入したとします。この時点で、データベースはブロック 13 を新しいブロックとしてフォーマットします。これで、更新操作の前の一部の REDO が欠落していた場合でも、このブロックでリカバリを続行できるようになります。

## ブロック・メディア・リカバリの前提条件

次の前提条件が RECOVER ... BLOCK コマンドに適用されます。

- ターゲット・データベースが ARCHIVELOG モードで実行されており、現行の制御ファイルを使用してオープンまたはマウントされている必要があります。
- ターゲット・データベースがスタンバイ・データベースの場合は、データベースが一貫性のある状態で、リカバリが開始されておらず、バックアップは破損したファイルよりも古いものである必要があります。
- 破損ブロックが含まれているデータファイルのバックアップは、プロキシ・コピーではなく全体またはレベル 0 のバックアップである必要があります。

**プロキシ・コピー**・バックアップのみが存在する場合は、それらをディスク上のデフォルト以外の場所にリストアできます。この場合、Recovery Manager は、プロキシ・バックアップをデータファイルのコピーとみなして、ブロック・メディア・リカバリ中にプロキシ・バックアップ内のブロックを検索します。

- Recovery Manager は、リカバリでアーカイブ REDO ログのみを使用できます。

Recovery Manager は、レベル 1 の増分バックアップを使用できません。ブロック・メディア・リカバリでは、アーカイブ REDO ログが欠落しているか、またはアクセスできない場合はリカバリを実行できません。ただし、REDO レコードが欠落している場合はリカバリを実行できる場合もあります。

- Recovery Manager がフラッシュバック・ログで破損ブロックの完全な状態に近いコピーを検索するには、フラッシュバック・データベースがターゲット・データベースで有効になっている必要があります。

破損していない古いバージョンの破損ブロックが含まれている場合にフラッシュバック・ロギングが有効になっていると、Recovery Manager は、それらのブロックを使用し、リカバリに必要な時間を短縮できます。

## 個々のブロックのリカバリ

通常、ブロック破損は次の場所に表示されます。

- LIST FAILURE、VALIDATE または BACKUP ... VALIDATE コマンドの結果
- V\$DATABASE\_BLOCK\_CORRUPTION ビュー
- 標準出力のエラー・メッセージ
- アラート・ログ
- 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・ファイルに次のメッセージが表示される場合があります。

```
ORA-01578: ORACLE data block corrupted (file # 7, block # 3)
ORA-01110: data file 7: '/oracle/oradata/trgt/tools01.dbf'
ORA-01578: ORACLE data block corrupted (file # 2, block # 235)
ORA-01110: data file 2: '/oracle/oradata/trgt/undotbs01.dbf'
```

次の手順では、リカバリが必要なブロックを識別し、使用可能なすべてのバックアップを使用してこれらのブロックのリストアおよびリカバリを実行します。

**特定のデータ・ブロックをリカバリする手順**

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。

トレース・ファイルおよびアラート・ログを検索する最も簡単な方法は、SQL\*Plus をターゲット・データベースに接続し、次の問合せを実行する方法です。

```
SELECT NAME, VALUE
FROM   V$DIAG_INFO;
```

2. Recovery Manager を起動し、ターゲット・データベースに接続します。ターゲット・データベースは、マウントまたはオープンされている必要があります。
3. SHOW ALL コマンドを実行して、適切なチャンネルが事前構成されていることを確認します。
4. Recovery Manager プロンプトで、ファイルおよび破損ブロックのブロック番号を指定して RECOVER ... BLOCK コマンドを実行します。

次の例では、2つのブロックをリカバリします。

```
RECOVER
  DATAFILE 8 BLOCK 13
  DATAFILE 2 BLOCK 19;
```

Recovery Manager の動作を制御する様々なオプションを指定することもできます。次の例は、ブロックの検索時にタグ mondayam が含まれているバックアップのみが使用されることを示しています。FROM BACKUPSET オプションを使用して、Recovery Manager によって検索されるバックアップのタイプを制限できます。または、EXCLUDE FLASHBACK LOG を使用して、Recovery Manager によるフラッシュバック・ログの検索を制限できます。

```
RECOVER
  DATAFILE 8 BLOCK 13
  DATAFILE 2 BLOCK 199
  FROM TAG mondayam;
```

## V\$DATABASE\_BLOCK\_CORRUPTION 内のすべてのブロックのリカバリ

この例では、Recovery Manager が V\$DATABASE\_BLOCK\_CORRUPTION ビューに表示されているすべてのブロックを自動的にリカバリします。

**V\$DATABASE\_BLOCK\_CORRUPTION に記録されているすべてのブロックをリカバリする手順**

1. SQL\*Plus を起動し、ターゲット・データベースに接続します。
2. V\$DATABASE\_BLOCK\_CORRUPTION を問い合わせ、破損ブロックが存在するかどうかを確認します。たとえば、次の文を実行します。

```
SQL> SELECT * FROM V$DATABASE_BLOCK_CORRUPTION;
```

3. Recovery Manager を起動し、ターゲット・データベースに接続します。
4. V\$DATABASE\_BLOCK\_CORRUPTION 内の破損とマークされたすべてのブロックをリカバリします。

次のコマンドを実行すると、ビューに記録された物理的に破損しているすべてのブロックが修復されます。

```
RMAN> RECOVER CORRUPTION LIST;
```

ブロックは、修復された後、データベースによって V\$DATABASE\_BLOCK\_CORRUPTION から削除されます。

**参照：** RECOVER ... BLOCK コマンドについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。



---

## Recovery Manager のリカバリの実行： 高度な例

第 V 部「障害の診断および対応」の前述の章は、最も基本的なリカバリの例についての一般的な説明となっています。この章では、基本的な例ほど一般的ではないか、または基本的な例より複雑である高度な例について説明します。

この章の内容は、次のとおりです。

- 増分バックアップを使用した NOARCHIVELOG モードのデータベースのリカバリ
- 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
- 障害リカバリの実行
- 新しいホストへのデータベースのリストア

## 増分バックアップを使用した NOARCHIVELOG モードのデータベースのリカバリ

NOARCHIVELOG モードで実行されているデータベースのリストアは、ARCHIVELOG モードのデータベースのリストアと類似しています。主な違いを次に示します。

- NOARCHIVELOG モードでのデータベースのリストアでは、一貫性バックアップのみを使用できます。
- アーカイブ REDO ログが存在しないため、メディア・リカバリは実行できません。

増分バックアップを適用することによって、制限された変更のリカバリを NOARCHIVELOG モードで実行されているデータベースに実行できます。NOARCHIVELOG モードで実行されるデータベースのすべてのバックアップと同様に、増分バックアップは一貫性バックアップである必要があります。そのため、データベースのオープン時にデータベースのバックアップは作成できません。

NOARCHIVELOG データベースをリカバリする場合は、RECOVER コマンドで NOREDO オプションを指定し、Recovery Manager がアーカイブ REDO ログの適用を試行しないように指定します。そうしない場合、Recovery Manager によってエラーが戻されます。

### 増分バックアップを使用して NOARCHIVELOG データベースをリカバリする手順

1. trgt およびカタログ・データベースに接続した後、データベースをマウント状態にします。

```
STARTUP FORCE MOUNT
```

2. データベースをリストアおよびリカバリします。

たとえば、次のコマンドで不完全リカバリを実行できます。

```
RESTORE DATABASE
FROM TAG "consistent_whole_backup";
RECOVER DATABASE NOREDO;
```

3. RESETLOGS オプションを指定してデータベースをオープンします。

たとえば、次のコマンドを入力します。

```
ALTER DATABASE OPEN RESETLOGS;
```

## サーバー・パラメータ・ファイルのリストア

サーバー・パラメータ・ファイルが消失した場合は、Recovery Manager を使用して、デフォルトの場所または選択した場所にそのファイルをリストアできます。制御ファイルが消失した場合は異なり、サーバー・パラメータ・ファイルが消失した場合、インスタンスはすぐに停止しません。インスタンスは継続して実行できます。ただし、サーバー・パラメータ・ファイルをリストアした後、インスタンスを停止して再起動する必要があります。

サーバー・パラメータ・ファイルをリストアする場合は、次の考慮事項に注意してください。

- インスタンスがサーバー・パラメータ・ファイルを使用してすでに起動されている場合、既存のサーバー・パラメータ・ファイルを上書きすることはできません。
- TO 句を使用せず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起動すると、Recovery Manager によってサーバー・パラメータ・ファイルがデフォルトの場所にリストアされます。デフォルトの場所はプラットフォーム固有です。たとえば、Linux では ?/dbs/spfile.ora です。
- リカバリ・カタログを使用すると、DBID を記録および記憶する手順を省略できるため、リカバリ手順が簡単になります。次の手順では、リカバリ・カタログを使用していないと想定しています。

**自動バックアップからサーバー・パラメータ・ファイルをリストアする手順****1. Recovery Manager を起動し、次のいずれかを実行します。**

- サーバー・パラメータ・ファイルが消失したときにデータベース・インスタンスが起動されていた場合は、ターゲット・データベースに接続します。
- サーバー・パラメータ・ファイルが消失したときにデータベース・インスタンスが起動されておらず、リカバリ・カタログを使用していない場合は、SET DBID を実行してターゲット・データベースの DBID を設定します。DBID を決定する方法については、17-6 ページの「データベースの DBID の確認」を参照してください。

**2. データベース・インスタンスを停止し、マウントせずに再起動します。**

サーバー・パラメータ・ファイルを使用できない場合、Recovery Manager では仮のパラメータ・ファイルを使用してインスタンスが起動されます。たとえば、次のコマンドを入力します。

```
STARTUP FORCE NOMOUNT;
```

**3. RUN コマンドを実行してサーバー・パラメータ・ファイルをリストアします。**

状況によっては、RUN コマンドで複数のコマンドを実行する必要がある場合もあります。次のことに注意してください。

- テープからリストアする場合は、ALLOCATE CHANNEL を使用して SBT チャンネルを手動で割り当てます。ディスクからリストアする場合は、Recovery Manager によってデフォルトのディスク・チャンネルが使用されます。
- 自動バックアップがデフォルトの書式 (%F) で生成されなかった場合は、SET CONTROLFILE AUTOBACKUP FOR DEVICE TYPE コマンドを使用して、自動バックアップの実行時に有効であった書式を指定します。
- 最新の自動バックアップが今日作成されなかった場合は、SET UNTIL を使用して検索を開始する日付を指定します。
- Recovery Manager がリカバリ・カタログに接続されていない場合は、SET DBID を使用してターゲット・データベースの DBID を設定する必要があります。
- サーバー・パラメータ・ファイルをデフォルト以外の場所にリストアする場合は、RESTORE SPFILE コマンドで TO 句または TO PFILE 句を指定します。
- Recovery Manager によって 1 日に  $n$  個を超える自動バックアップが生成されないことがわかっている場合は、RESTORE SPFILE FROM AUTOBACKUP ... MAXSEQ パラメータに  $n$  を設定して、検索時間を短縮することができます。MAXSEQ はデフォルトで 255 に設定されており、RESTORE は MAXSEQ から逆算して、その日最後のバックアップを検索します。現在の日付（または指定した日付）の自動バックアップが検出されない場合にリストア操作を終了するには、RESTORE コマンドで MAXDAYS 1 を設定します。

次の例では、テープ上の自動バックアップからサーバー・パラメータ・ファイルをリストアする RUN コマンドを示します。

```
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt PARMS ...;
  SET UNTIL TIME 'SYSDATE-7';
  SET CONTROLFILE AUTOBACKUP FORMAT
    FOR DEVICE TYPE sbt TO '/disk1/control_files/autobackup_%F';
  SET DBID 123456789;
  RESTORE SPFILE
    TO '/tmp/spfileTEMP.ora'
    FROM AUTOBACKUP MAXDAYS 10;
}
```

4. リストアしたファイルを使用してデータベース・インスタンスを再起動します。

デフォルト以外の場所にあるサーバー・パラメータ・ファイルを使用して再起動する場合は、行 `SPFILE=new_location` を含む新しい初期化パラメータ・ファイルを作成します。ここで、`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1行を含む `/tmp/init.ora` ファイルを作成します。

```
SPFILE=/tmp/spfileTEMP.ora
```

次の `Recovery Manager` コマンドを使用すると、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再起動できます。

```
STARTUP FORCE PFILE=/tmp/init.ora;
```

## 制御ファイルの自動バックアップからのサーバー・パラメータ・ファイルのリストア

制御ファイルの自動バックアップが構成されている場合は、自動バックアップが実行されるたびに、制御ファイルとともにサーバー・パラメータ・ファイルが常にバックアップされます。

制御ファイルの自動バックアップからサーバー・パラメータ・ファイルのリストアするには、まずデータベースの `DBID` を設定し、次に `RESTORE SPFILE FROM AUTOBACKUP` コマンドを使用する必要があります。自動バックアップがデフォルト以外の書式である場合は、最初に `SET CONTROLFILE AUTOBACKUP FORMAT` コマンドを使用して書式を指定します。

**例 19-1** では、`DBID` を設定し、デフォルト以外の場所にある制御ファイルの自動バックアップから、サーバー・パラメータをリストアします。

### 例 19-1 制御ファイルの自動バックアップからのサーバー・パラメータ・ファイルのリストア

```
SET DBID 320066378;
RUN
{
  SET CONTROLFILE AUTOBACKUP FORMAT
    FOR DEVICE TYPE DISK TO 'autobackup_format';
  RESTORE SPFILE FROM AUTOBACKUP;
}
```

`Recovery Manager` では、自動バックアップの書式および `DBID` を使用して、制御ファイルの自動バックアップが検索されます。制御ファイルの自動バックアップが検出されると、`Recovery Manager` によってサーバー・パラメータ・ファイルがバックアップからデフォルトの場所にリストアされます。

`autobackup_format` の適切な値を判断する方法については、『Oracle Database バックアップおよびリカバリ・リファレンス』の `CONFIGURE` 用のエントリ内の `CONFIGURE CONTROLFILE AUTOBACKUP FORMAT` についての説明を参照してください。

**参照：** `DBID` を決定する方法については、17-6 ページの「データベースの `DBID` の確認」を参照してください。

## Recovery Manager を使用した初期化パラメータ・ファイルの作成

サーバー・パラメータ・ファイルは、`TO PFILE 'filename'` 句を使用して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することもできます。指定するファイル名は、`Recovery Manager` クライアントが実行されているホストからアクセス可能なファイル・システム上にあることを示している必要があります。このファイルは、インスタンスが実行されているホストから直接アクセス可能である必要はありません。



次の Recovery Manager コマンドを実行すると、Recovery Manager クライアントが実行されているシステム上に /tmp/initTEMP.ora という名前の初期化パラメータ・ファイルが作成されます。

```
RESTORE SPFILE TO PFILE '/tmp/initTEMP.ora';
```

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するには、次のコマンドを使用して、同じクライアント・ホスト上で Recovery Manager を再度実行します。

```
STARTUP FORCE PFILE='/tmp/initTEMP.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

この項では、現行のすべての制御ファイルが消失したため、制御ファイルのバックアップをリストアする必要がある場合に行う操作について説明します。

### バックアップ制御ファイルを使用したりカバリの実行

現行のすべての制御ファイルのコピーが消失したか、または破損した場合は、バックアップ制御ファイルをリストアおよびマウントする必要があります。リストアされたデータファイルがない場合でも、RECOVER コマンドを実行し、RESETLOGS オプションを指定してデータベースをオープンする必要があります。ただし、現行の制御ファイルのいくつかのコピーを使用できる場合は、29-2 ページの「[現行の制御ファイルのサブセットが消失した場合の対応](#)」の手順を実行し、リカバリおよび RESETLOGS の実行を回避できます。

Recovery Manager は、リカバリ時に Recovery Manager リポジトリに記録されていないオンライン・ログおよびアーカイブ・ログを自動的に検索し、検出したログをカタログに追加します。Recovery Manager は、現行のアーカイブ先で、現行のログ・フォーマットを持つすべての有効なアーカイブ REDO ログの検索を試行します。現行のフォーマットは、このインスタンス（または Oracle RAC 構成のすべてのインスタンス）の起動に使用される初期化パラメータで指定されます。同様に、Recovery Manager は、制御ファイルに示されたファイル名を使用して、オンライン REDO ログの検索を試行します。

リカバリ時にアーカイブ先またはフォーマットを変更したり、制御ファイルのバックアップの後に新しいオンライン・ログ・メンバーを追加すると、Recovery Manager で、必要なオンライン・ログまたはアーカイブ・ログを自動的にカタログに追加できなくなる場合があります。Recovery Manager では、UNTIL 時間を指定しなかった場合にオンライン REDO ログを検出できないと、次のようなエラーが通知されます。

```

RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of recover command at 08/29/2007 14:23:09
RMAN-06054: media recovery requesting unknown log: thread 1 scn 86945

```

この場合、リカバリを続行するには、CATALOG コマンドを使用して必要な REDO ログをリポジトリに手動で追加する必要があります。

**参照：** 様々な例（リカバリ・カタログの使用または特定のバックアップからのリストアなど）で RESTORE CONTROLFILE を使用する場合の制限の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』の RESTORE CONTROLFILE に関する項を参照してください。

### 制御ファイルの場所

制御ファイルをリストアする場合、デフォルトのリストア先は、CONTROL\_FILES 初期化パラメータに定義されているすべての場所です。CONTROL\_FILES 初期化パラメータを設定しない場合、データベースは、CONTROL\_FILES パラメータが設定されていないときに制御ファイルを作成する場合と同じ規則を使用して、リストアされた制御ファイルの格納先を決定します。これらの規則については、『Oracle Database SQL 言語リファレンス』の CREATE CONTROLFILE 文の説明を参照してください。

1つ以上の新しい場所に制御ファイルをリストアするには、CONTROL\_FILES 初期化パラメータを変更した後、引数を指定せずに RESTORE CONTROLFILE を実行して制御ファイルをデフォルトの場所にリストアする方法があります。たとえば、CONTROL\_FILES に定義されている場所の一部がディスク障害によって使用できなくなった後に制御ファイルをリストアする場合は、CONTROL\_FILES で障害ディスクへの参照を別のディスクへのパス名に置き換えた後、引数を指定せずに RESTORE CONTROLFILE を実行します。

RESTORE CONTROLFILE TO '*filename*' [FROM AUTOBACKUP] という書式を使用して、CONTROL\_FILES に定義されている以外の任意の場所に制御ファイルをリストアすることもできます。

```
RESTORE CONTROLFILE TO '/tmp/my_controlfile';
```

現在使用中の制御ファイルは上書きされないため、この操作は、NOMOUNT、MOUNT または OPEN の状態のデータベースで実行できます。'*filename*' という名前のすべての既存ファイルが上書きされます。制御ファイルを新しい場所にリストアした後、CONTROL\_FILES 初期化パラメータを更新して新しい場所を追加定義できます。

**参照：** RESTORE CONTROLFILE 構文については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## リカバリ・カタログを使用したリカバリおよび使用しないリカバリ

Recovery Manager がリカバリ・カタログに接続している場合、バックアップ制御ファイルを使用したリカバリ手順は、現行の制御ファイルを使用したリカバリと同様になります。バックアップ制御ファイルで欠落している Recovery Manager メタデータは、リカバリ・カタログで入手可能です。唯一の例外は、データベース名がカタログ内で一意でない場合です。この場合は、SET DBID コマンドを使用してから制御ファイルをリストアする必要があります。

リカバリ・カタログを使用していない場合は、自動バックアップから制御ファイルをリストアする必要があります。自動バックアップから制御ファイルをリストアするには、データベースが NOMOUNT 状態である必要があります。例 19-2 に示すように、まずデータベースの DBID を設定し、次に RESTORE CONTROLFILE FROM AUTOBACKUP コマンドを使用する必要があります。

### 例 19-2 DBID の設定および自動バックアップからの制御ファイルのリストア

```
SET DBID 320066378;
RUN
{
  SET CONTROLFILE AUTOBACKUP FORMAT
  FOR DEVICE TYPE DISK TO 'autobackup_format';
  RESTORE CONTROLFILE FROM AUTOBACKUP;
}
```

Recovery Manager では、自動バックアップの書式および DBID を使用して、制御ファイルの自動バックアップを検索する場所が決定されます。制御ファイルの自動バックアップが検出されると、Recovery Manager によって、CONTROL\_FILES 初期化パラメータに示されているすべての制御ファイルの場所に制御ファイルがリストアされます。

**参照：** 自動バックアップの書式の正しい値を確認する方法については、『Oracle Database バックアップおよびリカバリ・リファレンス』の CONFIGURE 用のエントリの CONFIGURE CONTROLFILE AUTOBACKUP FORMAT に関する項を参照してください。DBID を決定する方法については、17-6 ページの「データベースの DBID の確認」を参照してください。

## フラッシュ・リカバリ領域を使用している場合のリカバリ

制御ファイルをリストアするコマンドは、データベースでフラッシュ・リカバリ領域が使用されているかどうかに関係なく同じです。データベースでリカバリ領域が使用されている場合、Recovery Manager は、すべてのディスクベースのバックアップおよび制御ファイルに記録されているイメージ・コピーをクロスチェックして、バックアップからリストアされた制御ファイルを更新します。Recovery Manager は、リカバリ領域内の記録されていないすべてのバックアップをカタログに追加します。この結果、リストアされた制御ファイルには、リカバリ領域内のすべてのバックアップ、およびバックアップの時点で制御ファイルに認識されたその他のバックアップの完全で正確なレコードが含まれます。

Recovery Manager は、制御ファイルをリストアした後、テープ・バックアップのクロスチェックを自動的に実行しません。テープ・バックアップを使用している場合は、制御ファイルをリストアおよびマウントできます。また、必要に応じてテープ上のバックアップをクロスチェックできます。次に例を示します。

```
CROSSCHECK BACKUP DEVICE TYPE sbt;
```

## 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、制御ファイルの Recovery Manager バックアップを使用し、リカバリ・カタログは使用しないと想定しています。また、ターゲット・データベースの制御ファイルの自動バックアップ機能を有効にして、制御ファイルの自動バックアップをリストアできると想定しています。

自動バックアップでは標準的な書式が使用されるため、使用可能なバックアップを表示するリポジトリが存在しなくても、Recovery Manager は自動バックアップをリストアできます。自動バックアップはデフォルトの場所または新しい場所にリストアできます。Recovery Manager は、CONTROL\_FILES で指定されたすべての場所に制御ファイルを自動的にレプリケートします。

---

**注意：**（たとえば、メディア・マネージャによって、またはバックアップ・ピースがディスク上に存在するため）制御ファイルを含むバックアップ・ピースの名前がわかっている場合は、RESTORE CONTROLFILE FROM '*filename*' コマンドを使用してバックアップ・ピースの名前を指定できます。データベースは、すべての自動バックアップの場所をアラート・ログに記録します。

---

リカバリ・カタログに接続していないため、Recovery Manager リポジトリには制御ファイルのバックアップの時点で使用可能なバックアップに関する情報のみが含まれます。他の使用可能なバックアップ・セットまたはイメージ・コピーの場所がわかっている場合は、CATALOG コマンドを使用して制御ファイルの Recovery Manager リポジトリに追加します。

### NOCATALOG モードで制御ファイルの自動バックアップを使用してデータベースをリカバリする手順

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. データベースをマウントせずにターゲット・データベース・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. SET DBID を使用して、ターゲット・データベースのデータベース識別子を設定します。

ターゲット・データベースに接続するたびに、DBID が表示されます。保存された Recovery Manager のログ・ファイルを調べるか、カタログを問い合わせるか、または制御ファイルの自動バックアップのファイル名を検索することによって DBID を取得することもできます。たとえば、次のコマンドを実行します。

```
SET DBID 676549873;
```

4. 自動バックアップの制御ファイルをリストアしてリカバリを実行するための **Recovery Manager** コマンド・ファイルを記述します。

このコマンド・ファイルには、次の手順が含まれている必要があります。

- 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、リストアする制御ファイルの自動バックアップの検索時に、このタイムスタンプを使用できます。
- b. 制御ファイルの自動バックアップの作成時に制御ファイルの自動バックアップの異なる書式が有効になっていたことがわかっている場合は、制御ファイルのリストアにデフォルト以外の書式を指定します。
- c. **SBT** チャンネルによって制御ファイルの自動バックアップが作成された場合は、1 つ以上の **SBT** チャンネルを割り当てます。使用可能なリカバリ・カタログがないため、事前構成されたチャンネルは使用できません。
- d. 必要に応じて、**Recovery Manager** が検索できる過去の最大日数および最初の日付の検索に使用する必要がある最初の順序番号を設定して、制御ファイルの自動バックアップをリストアします。
- e. 残りのリストア・プロセスに有効な構成済のチャンネルに関する情報が制御ファイルに含まれていることがわかっている場合は、**Recovery Manager** を終了し、手順 c で行ったチャンネルの手動割当てを解除できます。

**Recovery Manager** クライアントを再起動してデータベースをマウントする場合は、これらの構成済のチャンネルを使用できます。制御ファイルに含まれている構成済のチャンネルを使用しない場合は、データベースをマウントできます。

- f. この手順は、オンライン **REDO** ログが使用可能かどうかによって異なります。ログが使用可能かどうかに関係なく、バックアップ制御ファイルを使用してリカバリを行った後は、常に **OPEN RESETLOGS** が必要となることに注意してください。

オンライン **REDO** ログが使用可能な場合、**Recovery Manager** はこれらのログを検出および適用できます。完全なリストアおよびリカバリを実行します (17-10 ページの「データベースの完全リカバリの実行」を参照)。

オンライン **REDO** ログを使用できない場合は、**DBPITR** を実行します (16-15 ページの「データベースの **Point-in-Time** リカバリの実行」を参照)。オンライン **REDO** ログの最初の **SCN** より前の時点までリカバリする場合の目標時点、**SCN** またはログ順序番号を指定するには、**UNTIL** 句を使用する必要があります (この句を使用しない場合、**Recovery Manager** は **RMAN-6054** エラーを発行します)。

---

**注意：** ログ順序の指定時に、最後に作成されたアーカイブ **REDO** ログの順序が  $n$  の場合は、**Recovery Manager** が  $n$  を適用してから停止するように、**UNTIL SEQUENCE  $n+1$**  を指定します。

---

次の例では、オンライン **REDO** ログ・ファイルが消失しており、最新のアーカイブ **REDO** ログの順序番号は 13243 です。この例は、制御ファイルの自動バックアップをリストアし、最新のログを使用してリカバリする方法を示しています。

```
RUN
{
  # Optionally, set upper limit for eligible time stamps of control file
  # backups
  # SET UNTIL TIME '09/10/2007 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 PARS '...'; # allocate manually
  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;
```

```

}
# Now use automatic channels configured in restored control file
RESTORE DATABASE UNTIL SEQUENCE 13244;
RECOVER DATABASE UNTIL SEQUENCE 13244;

```

- リカバリに成功した場合は、データベースをオープンし、オンライン・ログをリセットします。

```
ALTER DATABASE OPEN RESETLOGS;
```

## 障害リカバリの実行

障害リカバリには、ターゲット・データベース全体、リカバリ・カタログ・データベース、すべての現行の制御ファイル、すべてのオンライン REDO ログ・ファイルおよびすべてのパラメータ・ファイルが消失した後のターゲット・データベースのリストアおよびリカバリが含まれます。

## 障害リカバリの前提条件

障害リカバリを実行するには、次のものがが必要です。

- すべてのデータファイルのバックアップ
- リストアする最も古いバックアップの作成時刻の後に生成されたすべてのアーカイブ REDO ログ
- 1つ以上の制御ファイルの自動バックアップ
- データベースの DBID のレコード

## 障害発生後のデータベースのリカバリ

障害リカバリの手順は、NOCATALOG モードでバックアップ制御ファイルを使用してデータベースをリカバリする手順と類似しています。新しいホストにデータベースをリストアする場合は、19-11 ページの「[新しいホストへのデータベースのリストア](#)」で説明されている考慮事項を参照してください。

この例では、データベースが実行されていた Linux サーバーに修復不可能な破損が発生していると想定しています。データベースを Oracle Secure Backup にバックアップしてあるため、そのテープを使用できます。この例では、次のことを想定しています。

- Oracle Database が新しいホストにすでにインストールされています。
- 古いホストと同じディレクトリ構造を持つ新しい Linux ホストにデータベースをリストアします。
- 1つのテープ・ドライブを使用します（このテープ・ドライブには、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、すべてのデータファイルのバックアップおよびログ 1124 までのアーカイブ REDO ログが含まれています）。
- データベースでリカバリ・カタログは使用しません。

### 新しいホストにデータベースをリカバリする手順

- 可能な場合は、tnsnames.ora、listener.ora、パスワード・ファイルなどの関連するすべてのネットワーク・ファイルをリストアまたは再生成します。
- Recovery Manager を起動し、ターゲット・データベース・インスタンスに接続します。

この時点では、初期化パラメータ・ファイルは存在しません。ORACLE\_SID および ORACLE\_HOME を設定すると、オペレーティング・システム認証を使用して SYSDBA として接続できます。たとえば、Recovery Manager を次のように起動します。

```
% rman
RMAN> CONNECT TARGET /
```

3. 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 (19-2 ページの「サーバー・パラメータ・ファイルのリストア」を参照)。

たとえば、次のコマンドを入力します。

```
SET DBID 676549873;
```

4. STARTUP NOMOUNT コマンドを実行します。

サーバー・パラメータ・ファイルを使用できない場合、Recovery Manager は仮のサーバー・パラメータ・ファイルを使用してインスタンスの起動を試行します。

5. チャンネルをメディア・マネージャに割り当て、自動バックアップからサーバー・パラメータ・ファイルのリストアします。

たとえば、Oracle Secure Backup からサーバー・パラメータ・ファイルのリストアする場合は、次のコマンドを入力します。

```
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt;
  RESTORE SPFILE FROM AUTOBACKUP;
}
```

6. リストアしたサーバー・パラメータ・ファイルを使用してインスタンスを再起動します。

```
STARTUP FORCE NOMOUNT;
```

7. リストアおよびリカバリを実行するためのコマンド・ファイルを記述し、そのコマンド・ファイルを実行します。このコマンド・ファイルには、次の手順が含まれている必要があります。

- a. チャンネルをメディア・マネージャに割り当てます。
- b. 制御ファイルの自動バックアップをリストアします (19-7 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)。
- c. リストアした制御ファイルをマウントします。
- d. CATALOG コマンドを使用して、リポジトリに記録されていないバックアップをカタログに追加します。
- e. データファイルを元の場所にリストアします。ボリューム名が変更されている場合は、リストアの前に SET NEWNAME コマンドを実行し、リストアの後に SWITCH を実行して、制御ファイルをデータファイルの新しい場所に更新します。次に例を示します。
- f. データファイルをリカバリします。指定したログ順序番号に達すると、Recovery Manager によってリカバリが停止されます。

```
RMAN> RUN
{
  # Manually allocate a channel to the media manager
  ALLOCATE CHANNEL t1 DEVICE TYPE sbt;
  # Restore 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.
  ALTER DATABASE MOUNT;
  SET UNTIL SEQUENCE 1124 THREAD 1;
  RESTORE DATABASE;
  RECOVER DATABASE;
}
```



次の RUN コマンドの例は、リストアしたデータファイルに新しいファイル名を使用することを除き、前述の例と同様です。

```

RMAN> RUN
{
  # If you need to restore the files to new locations,
  # use 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;
  ALTER DATABASE MOUNT;
  SET UNTIL SEQUENCE 124 THREAD 1;
  RESTORE DATABASE;
  SWITCH DATAFILE ALL; # Update control file with new location of datafiles.
  RECOVER DATABASE;
}

```

8. リカバリに成功した場合は、データベースをオープンし、オンライン・ログをリセットします。

```
ALTER DATABASE OPEN RESETLOGS;
```

## 新しいホストへのデータベースのリストア

障害リカバリ手順のテスト実行を行うか、またはデータベースを新しいホストに永続的に移動する場合は、この項で説明する手順を実行します。この手順では、RESTORE コマンドおよび RECOVER コマンドを使用します。

この項の手順を実行すると、リストアされるデータベースの DBID は、元のデータベースの DBID と同じになります。この方法で作成したテスト・データベースは、ソース・データベースと同じリカバリ・カタログに登録しないでください。2つのデータベースの DBID が同じであるため、テスト・データベースのメタデータが、ソース・データベースをリストアおよびリカバリする Recovery Manager の機能に影響を与える可能性があります。

新しいホストで継続して使用するためにターゲット・データベースの新しいコピーを作成する場合は、この手順ではなく、Recovery Manager の DUPLICATE コマンドを実行します。DUPLICATE コマンドを実行すると、作成されるデータベースに対して新しい DBID が割り当てられるため、このデータベースを元のデータベースと同じリカバリ・カタログに追加できます。

**参照：** データベースの複製方法については、23-2 ページの「[Recovery Manager データベースの複製の概要](#)」を参照してください。

## 新しいホストへのデータベースのリストアの準備

新しいホストへのデータベースのリストアを準備するには、次の手順を実行します。

- ソース・データベースの DBID を記録します。データベースの DBID が不明な場合は、DBID の確認方法について 17-6 ページの「[データベースの DBID の確認](#)」を参照してください。
- 新しいホスト上でソース・データベースの初期化パラメータ・ファイルをアクセス可能にします。オペレーティング・システム・ユーティリティを使用して、古いホストから新しいホストにこのファイルをコピーします。
- テスト・リストアのみを実行する場合は、Recovery Manager がリカバリ・カタログに接続されていないことを確認してください。接続されている場合は、Recovery Manager によって、リストアされたデータファイルに関するメタデータがリカバリ・カタログに記録されます。このメタデータは、プライマリ・データベースを将来リストアおよびリカバリする場合の障害となります。

リストアする必要があるすべてのバックアップの Recovery Manager リポジトリ・データを格納するには制御ファイルのサイズが十分ではないため、リカバリ・カタログを使用する必要がある場合は、Oracle Data Pump を使用してカタログをエクスポートし、別のスキーマまたはデータベースにインポートします。その後、コピーしたリカバリ・カタログをテスト・リストアに使用します。これを行わない場合、リカバリ・カタログでは、リストアされたデータベースが現行のターゲット・データベースとみなされます。

- リストアに使用されるバックアップがリストア・ホスト上でアクセス可能であることを確認します。たとえば、メディア・マネージャを使用してバックアップを作成した場合は、テープ・デバイスが新しいホストに接続されていることを確認します。ディスクのコピーを使用している場合は、次の項の手順を実行します。
- 本番データベースの試行リストアを実行する場合は、テスト環境でデータベースをリストアする前に、次の操作のいずれかを実行します。
  - 本番データベースで使用されるリカバリ領域と物理的に異なるフラッシュ・リカバリ領域をテスト・データベースで使用する場合は、テスト・データベース・インスタンスの DB\_RECOVERY\_FILE\_DEST を新しい場所に設定します。
  - 本番データベースで使用されるリカバリ領域と物理的に同じフラッシュ・リカバリ領域をテスト・データベースで使用する場合は、テスト・データベース・インスタンスの DB\_UNIQUE\_NAME を、本番データベースとは異なる名前に設定します。

前述の操作のいずれも実行しないと、Recovery Manager には本番データベースをリストアしていると想定され、フラッシュバック・ログは、使用不可とみなされるためフラッシュ・リカバリ領域から削除されます。

## 新しいホストへのディスク・バックアップのリストア

ディスク上のデータファイルのコピーまたはバックアップ・セットを使用して新しいホストにデータベースを移動するには、新しいホストにそのファイルを手動で転送する必要があります。次の例では、Recovery Manager でリカバリ・カタログが使用されていると想定しています。

### バックアップ・ファイルを新しいホストにリストアする方法

1. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログに接続します。
2. LIST コマンドを実行して、データファイルのバックアップおよび制御ファイルの自動バックアップのリストを表示します。

たとえば、次のコマンドを入力してデータファイルのコピーを表示します。

```
LIST COPY;
```

たとえば、次のコマンドを入力して制御ファイルのバックアップを表示します。

```
LIST BACKUP OF CONTROLFILE;
```

自動バックアップ・ピースの名前には、%F 置換変数を使用する必要があります。したがって、自動バックアップ・ピースの名前には、文字列 c-#####-YYYYMMDD-QQ が含まれます。ここで、##### は DBID を表し、YYYYMMDD はバックアップが生成された日のグレゴリオ暦でのタイムスタンプ、QQ は 16 進数の順序です。

3. オペレーティング・システム・ユーティリティを使用して、バックアップを新しいホストにコピーします。

次のコマンドを入力して、すべてのデータファイルのコピーを新しいホストの ?/oradata/trgt ディレクトリにコピーします。

```
% cp -r /disk1/*dbf /net/new_host/oracle/oradata/trgt
```

次のコマンドを入力して、自動バックアップのバックアップ・ピースを新しいホストの /tmp ディレクトリにコピーします。

```
% cp -r /disk1/auto_bkp_loc/c-1618370911-20070208-00 /net/new_host/tmp
```



19-4 ページの「制御ファイルの自動バックアップからのサーバー・パラメータ・ファイルのリストア」で説明されているように、デフォルト以外の場所から自動バックアップをリストアする場合は、SET CONTROLFILE AUTOBACKUP FORMAT コマンドを使用する必要があります。

## 新しいホストへのデータベースのリストアのテスト

次の例では、データベースを新しいホストにリストアすることができるかどうかをテストすると想定しています。また、ネットワークで接続された 2 台の Linux ホスト `hosta` と `hostb` があると想定しています。ターゲット・データベースが `hosta` 上に存在し、リカバリ・カタログ `catdb` に登録されています。データベース `trgta` が起動され、`hosta` 上で実行されている間に、`hostb` 上で `trgta` のリストアおよびリカバリをテストします。

わかりやすくするため、`hostb` のディレクトリ構造は `hosta` のディレクトリ構造と異なっていると想定しています。ターゲット・データベースは `/net/hosta/dev3/oracle/dbs` に存在していますが、このデータベースを `/net/hostb/oracle/oradata/test` にリストアします。データファイル、制御ファイル、アーカイブ REDO ログおよびサーバー・パラメータ・ファイルのテープ・バックアップが、両方のホストからアクセス可能なメディア・マネージャ上にあります。ターゲット・データベースの `ORACLE_SID` は `trgta` で、リストアされたデータベースでも変更されません。

---

**注意：** テストの目的でデータベースをリストアする場合は、Recovery Manager をテスト・データベースおよびリカバリ・カタログに接続しないでください。

---

### 新しいホストにデータベースをリストアする手順

1. ターゲット・データベースのバックアップが新しいホスト上でアクセス可能であることを確認します。

障害リカバリをテストするには、ターゲット・データベースのリカバリ可能なバックアップが必要です。障害リカバリ計画を作成する場合は、データファイル、制御ファイルおよびサーバー・パラメータ・ファイルのバックアップが `hostb` 上にリストア可能であることを確認します。そのため、`hostb` をメディア・マネージャのクライアントにして、`hosta` 上に作成されたバックアップ・セットを読み取ることができるように、メディア管理ソフトウェアを構成する必要があります。この問題に対するサポートについては、メディア管理ベンダーに問い合せてください。

2. `hostb` 上で `ORACLE_SID` を構成します。

この例では、Recovery Manager クライアントを `hostb` 上で起動し、オペレーティング・システムを介して自分を認証すると想定しています。ただし、ローカルで、またはネット・サービス名を介して、`hostb` に接続する必要があります。

管理者権限を使用して `hostb` にログインした後、自分が `DBA` グループに含まれるように `/etc/group` ファイルを編集します。

```
dba:*:614:<your_user_name>
```

`hostb` 上で、環境変数 `ORACLE_SID` を `hosta` 上で使用した値と同じ値に設定します。

```
% setenv ORACLE_SID trgta
```

3. `hostb` 上で Recovery Manager を起動し、リカバリ・カタログに接続せずにターゲット・データベースに接続します。

たとえば、次のコマンドを入力します。

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

- DBID を設定し、データベースをマウントせずにデータベース・インスタンスを起動します。

たとえば、SET DBID を実行して DBID を設定し、次に STARTUP NOMOUNT を実行します。

```
SET DBID 1340752057;
STARTUP NOMOUNT
```

サーバー・パラメータ・ファイルがリストアされていないため、Recovery Manager がサーバー・パラメータ・ファイルの検出に失敗しますが、仮のファイルを使用してインスタンスが起動されます。出力例を次に示します。

```
startup failed: ORA-01078: failure in processing system parameters
LRM-00109: could not open parameter file '/net/hostb/oracle/dbs/inittrgta.ora'
```

```
trying to start the Oracle instance without parameter files ...
Oracle instance started
```

- サーバー・パラメータ・ファイルをリストアおよび編集します。

バックアップの実行時に制御ファイルの自動バックアップ機能を有効にしたため、バックアップにサーバー・パラメータ・ファイルが含まれます。デフォルト以外の書式の自動バックアップをリストアする場合は、SET CONTROLFILE AUTOBACKUP FORMAT コマンドを使用して書式を指定します。

メディア・マネージャにチャンネルを割り当て、サーバー・パラメータ・ファイルをクライアント側のパラメータ・ファイルとしてリストアし、SET コマンドを使用して自動バックアップの場所を指定します（この例では、自動バックアップは /tmp にあります）。

```
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt PARMS '...';
  SET CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '/tmp/%F';
  RESTORE SPFILE
    TO PFILE '?/oradata/test/inittrgta.ora'
    FROM AUTOBACKUP;
  SHUTDOWN ABORT;
}
```

- リストアされた初期化パラメータ・ファイルを編集します。

末尾が DEST などの場所固有のすべてのパラメータを変更し、新しいディレクトリ構造を反映します。たとえば、次のパラメータを編集します。

```
- IFILE
- LOG_ARCHIVE_DEST_1
- CONTROL_FILES
```

- 編集した初期化パラメータ・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、次のコマンドを入力します。

```
STARTUP FORCE NOMOUNT PFILE='?/oradata/test/inittrgta.ora';
```

- 自動バックアップから制御ファイルをリストアし、データベースをマウントします。

たとえば、次のコマンドを入力します。

```
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt PARMS '...';
  RESTORE CONTROLFILE FROM AUTOBACKUP;
  ALTER DATABASE MOUNT;
}
```

CONTROL\_FILES 初期化パラメータで指定した場所に制御ファイルがリストアされます。

9. 新しいファイル名または CATALOG START WITH (すべてのファイルが共通の接頭辞を持つディレクトリに含まれ、CATALOG START WITH で簡単に指定できることがわかっている場合) を使用して、19-12 ページの「新しいホストへのディスク・バックアップのリストア」でコピーしたデータファイルのコピーをカタログに追加します。たとえば、次のコマンドを実行します。

```
CATALOG START WITH '/oracle/oradata/trgt/';
```

ファイルを個別に指定する場合は、CATALOG コマンドを次のように実行します。

```
CATALOG DATAFILECOPY
  '/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';
```

10. 新しいデータベース上で SQL\*Plus セッションを開始し、制御ファイルに記録されたデータベースのファイル名を問い合わせます。

制御ファイルは trgta データベースからリストアされているため、記録されたファイル名では元の hosta のファイル名が使用されます。V\$ ビューを問い合わせると、この情報を取得できます。SQL\*Plus で次の問合せを実行します。

```
COLUMN NAME FORMAT a60
SPOOL LOG '/tmp/db_filenames.out'
SELECT FILE# AS "File/Grp#", NAME
FROM V$DATAFILE
UNION
SELECT GROUP#,MEMBER
FROM V$LOGFILE;
SPOOL OFF
EXIT
```

11. Recovery Manager のリストア・スクリプトおよびリカバリ・スクリプトを記述します。このスクリプトには、次の手順が含まれている必要があります。
- ソース・ホスト上のパスとは異なるパスにリストアされた転送先ホスト上の各データファイルに対して、SET NEWNAME コマンドを使用して転送先ホスト上に新しいパスを指定します。転送先システム上のファイル・システムがソース・ホスト上のパスと同じパスを持つように設定されている場合は、ソース・ホスト上のパスと同じパスにリストアされるファイルに対して SET NEWNAME を使用しないでください。
  - ソース・ホスト上の場所とは異なる場所に作成される各オンライン REDO ログに対して、SQL の ALTER DATABASE RENAME FILE コマンドを使用して転送先ホスト上にパス名を指定します。転送先システム上のファイル・システムがソース・ホスト上のパスと同じパスを持つように設定されている場合は、ソース・ホスト上のパスと同じパスにリストアされるファイルに対して ALTER DATABASE RENAME FILE を使用しないでください。
  - SET UNTIL を実行して、リカバリをアーカイブ REDO ログの最後までに制限します。SET UNTIL が指定されていない場合、エラーが発生してリカバリが停止することに注意してください。
  - データベースをリストアおよびリカバリします。
  - SWITCH DATAFILE ALL を実行して、新しいパス名をデータファイルの正式な新しい名前として制御ファイルに認識させます。

例 19-3 に、リストアおよびリカバリを実行できる Recovery Manager のスクリプト reco\_test.rman を示します。

**例 19-3 新しいホストへのデータベースのリストア**

```
RUN
{
# allocate a channel to the tape device
ALLOCATE CHANNEL c1 DEVICE TYPE sbt PARMS '...';

# rename the datafiles and online redo logs
SET NEWNAME FOR DATAFILE 1 TO '?/oradata/test/system01.dbf';
SET NEWNAME FOR DATAFILE 2 TO '?/oradata/test/undotbs01.dbf';
SET NEWNAME FOR DATAFILE 3 TO '?/oradata/test/cwmlite01.dbf';
SET NEWNAME FOR DATAFILE 4 TO '?/oradata/test/drsys01.dbf';
SET NEWNAME FOR DATAFILE 5 TO '?/oradata/test/example01.dbf';
SET NEWNAME FOR DATAFILE 6 TO '?/oradata/test/indx01.dbf';
SET NEWNAME FOR DATAFILE 7 TO '?/oradata/test/tools01.dbf';
SET NEWNAME FOR DATAFILE 8 TO '?/oradata/test/users01.dbf';
SQL "ALTER DATABASE RENAME FILE '/dev3/oracle/dbs/redo01.log'
    TO '?/oradata/test/redo01.log' ";
SQL "ALTER DATABASE RENAME FILE '/dev3/oracle/dbs/redo02.log'
    TO '?/oradata/test/redo02.log' ";

# Do a SET UNTIL to prevent recovery of the online logs
SET UNTIL SCN 123456;
# restore the database and switch the datafile names
RESTORE DATABASE;
SWITCH DATAFILE ALL;

# recover the database
RECOVER DATABASE;
}
EXIT
```

12. 前述の手順で作成したスクリプトを実行します。

たとえば、ターゲット・データベースに対して **Recovery Manager** を起動し、@ コマンドを実行します。

```
% rman TARGET / NOCATALOG
RMAN> @reco_test.rman
```

13. **RESETLOGS** オプションを指定して、リストアしたデータベースをオープンします。

**Recovery Manager** プロンプトで、**RESETLOGS** オプションを指定してデータベースをオープンします。

```
ALTER DATABASE OPEN RESETLOGS;
```

---

---

**注意：** 次の手順でデータベースを再オープンする場合は、リカバリ・カタログに接続しないでください。接続すると、作成される新しいデータベース・インカネーションがリカバリ・カタログに自動的に登録され、本番データベースのファイル名が、スクリプトに指定されている新しいファイル名に置換されます。

---

---

14. 必要に応じて、含まれているすべてのファイルとともにテスト・データベースを削除します。

---

---

**注意：** ASM ディスク・グループを使用した場合、テスト・データベースのファイルを安全に削除する方法は **DROP DATABASE** のみです。非 ASM ストレージにリストアした場合は、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を削除することもできます。

---

---

DROP DATABASE コマンドを使用して、データベースに関連付けられているすべてのファイルを自動的に削除します。次の例では、データベース・ファイルを削除します。

```
STARTUP FORCE NOMOUNT PFILE='?/oradata/test/inittrgta.ora';  
DROP DATABASE;
```

リカバリ・カタログへの接続時にリストアおよびリカバリを実行しなかったため、リカバリ・カタログには、リストアしたファイルまたはテスト中に実行した手順のレコードは含まれません。同様に、trgta データベースの制御ファイルは、テストによる影響を受けません。



---

## Recovery Manager の表領域の Point-in-Time リカバリ (TSPITR) の実行

この章では、Recovery Manager の表領域の Point-in-Time リカバリを実行する方法について説明します。この章では、次の項目について説明します。

- [Recovery Manager の TSPITR の概要](#)
- [TSPITR の前提条件および結果](#)
- [TSPITR の計画および準備](#)
- [完全に自動化された Recovery Manager の TSPITR の実行](#)
- [Recovery Manager 管理の補助インスタンスを使用した Recovery Manager のカスタマイズ TSPITR の実行](#)
- [独自の補助インスタンスを使用した Recovery Manager の TSPITR の実行](#)
- [Recovery Manager の TSPITR のトラブルシューティング](#)

## Recovery Manager の TSPITR の概要

この項では、Recovery Manager の表領域の Point-in-Time リカバリに含まれる基本的な概念およびタスクについて説明します。

### Recovery Manager の TSPITR の目的

Recovery Manager で自動化される TSPITR を使用すると、データベースの 1 つ以上の表領域をデータベースの残りの表領域およびオブジェクトには影響を及ぼさずに、過去の時点まで迅速にリカバリできます。

Recovery Manager の TSPITR は、次の場合に最も有効です。

- 1 つの物理データベースの個別の表領域に複数の論理データベースが存在するときに、ある論理データベースを物理データベースの残りの部分とは異なる時点までリカバリする場合。たとえば、orders および personnel 表領域に論理データベースを保持している場合などです。不正なバッチ・ジョブまたは DML 文を実行すると、いずれかの表領域にのみ存在するデータが破損します。
- 表の構造を変更する DDL 操作を実行した後に消失したデータをリカバリする場合。フラッシュバック表を使用して、表の切捨て操作などで構造を変更した時点より前に表を巻き戻すことはできません。
- PURGE オプションを使用して表を削除した後、その表をリカバリする場合。
- 表の論理的な破損からリカバリする場合。

フラッシュバック・データベースを使用してデータを巻き戻すこともできますが、サブセットのみでなくデータベース全体を巻き戻す必要があります。また、TSPITR とは異なり、フラッシュバック・データベース機能ではフラッシュバック・ログを保持するオーバーヘッドが伴います。データベースのフラッシュバックが可能な期間は、TSPITR の期間より限られています。TSPITR の期間は、最も古いリカバリ可能なバックアップまでの期間となります。

### Recovery Manager の TSPITR の基本的な概念

TSPITR は、Recovery Manager の RECOVER TABLESPACE コマンドを使用して実行します。ターゲット・インスタンスには、目標時点までリカバリする表領域が含まれています。目標時点とは、TSPITR が完了した後の表領域の過去の時刻または SCN のことです。

**補助インスタンス**とは、リカバリ・プロセスで使用され、リカバリ作業を実行するデータベース・インスタンスのことです。補助インスタンスには制御ファイル、パラメータ・ファイル、オンライン・ログなどの他のファイルが関連付けられますが、これらのファイルは補助セットの一部ではありません。

**リカバリ・セット**には、リカバリする表領域のデータファイルが含まれています。**補助セット**にはリカバリ・セットの TSPITR に必要なデータファイルが含まれていますが、そのデータファイル自体はリカバリ・セットの一部ではありません。通常、補助セットには次のものが含まれます。

- SYSTEM 表領域および SYSAUX 表領域
- ターゲット・データベース・インスタンスのロールバック・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

**補助の格納場所**とは、TSPITR の実行中に補助インスタンスの補助セットのデータファイル、制御ファイルおよびオンライン REDO ログの格納に使用できる、ディスク上の任意の場所のことです。ここに格納されたファイルは、TSPITR の完了後に削除できます。



最も簡単な形式の TSPITR (20-8 ページの「完全に自動化された Recovery Manager の TSPITR の実行」を参照) では、リカバリ・セットの表領域および目標時点を指定します。Recovery Manager によって次の処理が自動的に実行されます。

1. 例 20-1 で示すように、リカバリ・セットの表領域に対して問合せ SYS.TS\_PITR\_CHECK を実行します。問合せが行を戻す場合、Recovery Manager は TSPITR を続行しません。
2. 補助インスタンスを作成し、そのインスタンスを起動して接続します (既存の補助インスタンスが存在しない場合)。
3. ターゲット・データベース内のリカバリする表領域をオフラインにします。
4. 目標時点より前の時点のバックアップの制御ファイルを、補助インスタンスにリストアします。
5. リカバリ・セットおよび補助セットのデータファイルを、補助インスタンスにリストアします。  
各ファイルに指定した場所、またはファイルの元の場所 (リカバリ・セット・ファイルの場合) か補助の格納場所 (補助セット・ファイルの場合で、RECOVER TABLESPACE に AUXILIARY DESTINATION 引数を使用したとき) のいずれかにファイルをリストアします。
6. 補助インスタンスにリストアしたデータファイルを、指定した時点までリカバリします。
7. 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。
8. リカバリした表領域内のオブジェクトに関するディクショナリ・メタデータを、ターゲット・データベースにエクスポートします。
9. 補助インスタンスを停止します。
10. リカバリ・セットのデータファイルに新しい名前を指定した場合は、ターゲット・データベース・インスタンスで SWITCH コマンドを発行します。これによって、ターゲット・データベースの制御ファイルが、補助インスタンスでリカバリされたリカバリ・セットのデータファイルを指すようになります。
11. 補助インスタンスからターゲット・インスタンスにディクショナリ・メタデータをインポートし、リカバリしたオブジェクトをアクセス可能にします。
12. すべての補助セット・ファイルを削除します。

この時点で TSPITR は完了です。リカバリ・セットのデータファイルは、指定した時点の内容に戻り、ターゲット・データベースに属します。

## Recovery Manager の TSPITR の基本手順

TSPITR を実行する前に、20-4 ページの「TSPITR の前提条件および結果」を参照して、TSPITR が実行可能かどうかを確認します。TSPITR を実行する場合は、20-5 ページの「TSPITR の計画および準備」で説明されている準備段階に進むことができます。

TSPITR を実際に実行する準備ができた場合、基本手順は使用方法によって異なります。選択できる方法の詳細は、次の項を参照してください。

- **完全に自動化された Recovery Manager の TSPITR の実行**

補助の格納場所を指定し、Recovery Manager で TSPITR のすべての手順を管理します。これが TSPITR の最も簡単な方法です。特に操作を制御する必要がないかぎり、この方法をお勧めします。

- **Recovery Manager 管理の補助インスタンスを使用した Recovery Manager のカスタマイズ TSPITR の実行**

補助の格納場所を使用して、完全に自動化された TSPITR の動作に基づいて TSPITR を構成します。ただし、動作の 1 つ以上の設定をカスタマイズします。たとえば、補助セットまたはリカバリ・セット・ファイルの場所を指定するか、または Recovery Manager によって作成および管理される補助インスタンスの初期化パラメータまたはチャンネル構成を指定できます。

- **独自の補助インスタンスを使用した Recovery Manager の TSPITR の実行**

この方法では、TSPITR で使用する補助インスタンスの設定、起動、停止およびクリーンアップを独自に行います。また、自動補助インスタンスを使用したカスタマイズ TSPITR で使用可能ないくつかの方法でも、TSPITR のプロセスを管理できます。

## TSPITR の前提条件および結果

TSPITR では解決できないデータベースの問題も多くあります。次に、TSPITR を実行できない場合の TSPITR の前提条件のリストを示します。

- アーカイブ REDO ログがない場合またはデータベースが NOARCHIVELOG モードで実行されている場合は、TSPITR を実行できません。
- 削除された表領域はリカバリできません。
- 名前が変更された表領域は、変更より前の時点までリカバリできません。名前の変更操作より前の SCN までの TSPITR を試行した場合、Recovery Manager では、その SCN の時点のリポジットで新しい表領域名を検出できません（その SCN での表領域の名前は、新しい名前とは異なっているためです）。  
この場合、表領域名が変更される前の時点までデータベース全体をリカバリする必要があります。表領域は、その時点で付けられていた名前で検出されます。
- 表領域 tbs1 の表の制約が表領域 tbs2 に含まれている場合、tbs2 をリカバリせずに tbs1 をリカバリすることはできません。
- TSPITR では、次のオブジェクトはリカバリできません。
  - 複製されたマスター表
  - 部分的な表（たとえば、パーティション表で Recovery Manager の TSPITR を実行し、複数の表領域にパーティションを分散させた場合は、表のパーティションを含むすべての表領域をリカバリする必要があります。）
  - VARRAY 列、ネストした表または外部ファイルを持つ表
  - スナップショット・ログおよび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
  - ロールバック・セグメントなどの、SYS が所有するオブジェクトを含む表領域

## TSPITR の結果

TSPITR が完了した後、Recovery Manager は、リカバリ・セットのデータファイルを目標時点までリカバリします。TSPITR の結果は、次のとおりです。

- Recovery Manager でリカバリを行う目標時点以降にデータファイルを追加すると、Recovery Manager の TSPITR の実行後に、同じ名前を持つ空のデータファイルが表領域に含まれます。
- リカバリしたオブジェクトに関する問合せオプティマイザの統計情報はリカバリされません。

TSPITR の完了後に新しい統計情報を収集する必要があります。

- 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、時刻  $t$  より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ません。

この表領域に対して、時刻  $t$  以前の時点まで TSPITR を再実行することはできません。また、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。そのため、TSPITR の完了直後にリカバリした表領域をバックアップする必要があります。

## リカバリ・カタログを使用しない場合の特別な考慮事項

TSPITR の実行時にリカバリ・カタログを使用しない場合は、次の特別な考慮事項があります。

- TSPITR の実行時の UNDO セグメントは、補助セットの一部である必要があります。Recovery Manager での UNDO の履歴レコードは制御ファイルに含まれないため、現行のロールバック・セグメントまたは UNDO セグメントは、リカバリの目標時点に存在したセグメントと同じであるとみなされます。UNDO セグメントがその時点以降に変更されると、TSPITR は失敗します。
- 必要なバックアップの制御ファイル・レコードが再利用されている場合は、TSPITR で時間を過度に戻すと失敗することがあります。(データベースの計画時に、CONTROL\_FILE\_RECORD\_KEEP\_TIME 初期化パラメータを十分に大きい値に設定して、TSPITR に必要な制御ファイル・レコードが確実に保持されるようにします。)
- 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。リカバリ・カタログを使用しない場合は、現行の制御ファイルに、リカバリした表領域の古いインカンションのレコードが含まれません。そのため、この表領域に関連する現行の制御ファイルを使用したリカバリでは、時刻  $t$  より前に作成したバックアップは使用できません。ただし、時刻  $t$  より前の時点からバックアップ制御ファイルをリストアできる場合は、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。

## TSPITR の計画および準備

この項では、20-4 ページの「[TSPITR の前提条件および結果](#)」を参照済であると想定しています。TSPITR の準備では、次の手順を実行する必要があります。

- [TSPITR での正しい目標時点の選択](#)
- [リカバリ・セットの決定](#)
- [TSPITR の実行後に消失するオブジェクトの確認および保存](#)

## TSPITR での正しい目標時点の選択

TSPITR での正しい目標時点または SCN を選択することは、非常に重要です。20-4 ページの「[TSPITR の前提条件および結果](#)」で説明されているように、TSPITR の実行後に表領域をオンラインにすると、それより前の時点のバックアップは使用できなくなります。つまり、最初に誤った目標時点を選択すると、リカバリ・カタログを使用していないかぎり、TSPITR を再試行できません。

リカバリ・カタログを使用している場合は、表領域の履歴情報がカタログに格納されているため、TSPITR 操作を異なる目標時点まで繰り返し実行できます。ただし、Recovery Manager で制御ファイルのみが使用される場合、表領域の履歴は存在しません。この場合、Recovery Manager は現行の表領域のセットのみを認識します。TSPITR が実行された表領域の作成時刻は、オンラインになった時点になります。

リカバリ・カタログを使用しない状況を想定します。表領域に対して TSPITR を実行し、金曜日の午後 5 時に表領域をオンラインにします。金曜日の午後 5 時より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ません。この表領域に対して、金曜日の午後 5 時より前の目標時点を指定して TSPITR を再実行することはできません。また、現行の制御ファイルを使用して、データベースを金曜日の午後 5 時より前の時点までリカバリすることもできません。リストアした制御ファイルを使用して、データベース全体の Point-in-Time リカバリを実行する必要があります。

TSPITR の目標時点を指定するためにデータの過去の状態を調べるには、Oracle Flashback Query、Oracle Flashback Transaction Query および Oracle Flashback Version Query を使用して、データベースに不要な変更が発生した時点を検索します。

**参照：** フラッシュバック問合せ、フラッシュバック・トランザクション問合せおよびフラッシュバック・バージョン問合せの詳細は、『Oracle Database アドバンスト・アプリケーション開発者ガイド』を参照してください。

## リカバリ・セットの決定

最初、リカバリ・セットには、リカバリする表領域のデータファイルが含まれています。ただし、必要な表領域内のオブジェクトが他の表領域内のオブジェクトと関係（制約など）を持つ場合、TSPITR を実行する前にこの関係を処理する必要があります。このような関係に対応するには、次のいずれかの方法を選択します。

- 関連するオブジェクトを含む表領域をリカバリ・セットに追加する。
- 関係を削除する。
- TSPITR の実行中、関係を一時解消する。

## プライマリ・データベースの依存性の確認および解決

TS\_PITR\_CHECK ビューでは、リカバリ・セットの境界にまたがるオブジェクト間の関係を確認できます。このビューの問合せ時に行が戻された場合は、問題を調査および解決する必要があります。TS\_PITR\_CHECK ビューでリカバリ・セットに含まれていない表領域に対して行が戻されない場合のみ、TSPITR を続行します。TSPITR の完了後に一時解消または削除した関係を再作成できるように、この手順で実行するすべての処理を記録します。

例 20-1 の問合せでは、TS\_PITR\_CHECK ビューの使用方法を示します。たとえば、tools および users で構成される最初のリカバリ・セットを使用し、次のように TS\_PITR\_CHECK に対して SELECT 文を実行します。

### 例 20-1 表領域のサブセットに対する TS\_PITR\_CHECK の問合せ

```
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')
);
```

データベース内のすべての表領域（リカバリ・セットの表領域のみではなく）に対して、TSPITR を実行できるかどうかを完全に確認するには、例 20-2 の問合せを実行します。

### 例 20-2 すべての表領域に対する TS\_PITR\_CHECK の問合せ

```
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 ビューの列の列数および列幅のため、問合せの実行時に列を次のように調整する必要がある場合があります。

```
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 に、p1 および p2 の 2 つのパーティションが含まれ、それらがそれぞれ表領域 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、その索引に id1 および id2 の 2 つのパーティション（それぞれ表領域 id1 および id2 に存在する）が存在するとします。この場合、例 20-1 の問合せを実行すると、例 20-3 に示す出力が表示されます。

### 例 20-3 TS\_PITR\_CHECK の問合せの出力

```

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

```

例 20-3 は、SYSTEM.tp が表領域 id1 の ip1 および表領域 id2 の ip2 の 2 つのパーティションから構成されるパーティション索引 tpind を持っていることを示しています。TSPITR を実行するには、tpind を削除するか、または id1 および id2 をリカバリ・セットに含める必要があります。

**参照：** TS\_PITR\_CHECK ビューの詳細は、『Oracle Database リファレンス』を参照してください。

## TSPITR の実行後に消失するオブジェクトの確認および保存

表領域に対して Recovery Manager の TSPITR を実行すると、リカバリの目標時点より後に作成されたオブジェクトは消失します。このようなオブジェクトを確認した場合は、TSPITR の実行前にデータ・ポンプ・エクスポート・ユーティリティを使用してそれらをエクスポートし、後でデータ・ポンプ・インポートを使用して再インポートすることで、それらのオブジェクトを保存できます。

TSPITR を実行すると消失するオブジェクトを確認するには、プライマリ・データベースで TS\_PITR\_OBJECTS\_TO\_BE\_DROPPED ビューを問い合わせます。表 20-1 に、このビューの内容を示します。

表 20-1 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 を実行した結果、消失するオブジェクトの名前 |
| CREATION_TIME   | オブジェクトの作成タイムスタンプ             |
| TABLESPACE_NAME | オブジェクトを含む表領域の名前              |

CREATION\_TIME が TSPITR の目標時点より後であるオブジェクトのビューをフィルタ処理します。たとえば、users および tools で構成されるリカバリ・セットを使用し、リカバリの目標時点が 2007 年 11 月 2 日午前 7:03:11 の場合は、例 20-4 の文を発行します。

#### 例 20-4 TS\_PITR\_OBJECTS\_TO\_BE\_DROPPED の問合せ

```
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('02-NOV-06:07:03:11','YY-MON-DD:HH24:MI:SS')
ORDER BY TABLESPACE_NAME, CREATION_TIME;
```

TO\_CHAR および TO\_DATE ファンクションを使用すると、国によって異なる日付書式を使用した場合の問題を回避できます。ユーザーは、ユーザー自身の作業では現地の日付書式を使用できます。

**参照：** TS\_PITR\_OBJECTS\_TO\_BE\_DROPPED ビューの詳細は、『Oracle Database リファレンス』を参照してください。

## 完全に自動化された Recovery Manager の TSPITR の実行

完全に自動化された TSPITR を実行すると、プロセス全体が Recovery Manager によって管理されます。

Recovery Manager の TSPITR は、ターゲット・データベースでの構成にできるかぎり基づきます。TSPITR の実行中、リカバリ・セットのデータファイルはターゲット・データベースの現行の場所に書き込まれます。バックアップからファイルをリストアする場合は、ターゲット・データベースで有効なチャネル構成と同じ構成が補助インスタンスで使用されます。ただし、補助セットのデータファイルおよび補助インスタンスの他のファイルは、補助の格納場所に格納されます。

Recovery Manager で補助セットのデータファイルおよび補助インスタンスの他のファイルを格納する補助の格納場所を設定するには、AUXILIARY\_DESTINATION パラメータを使用します。補助の格納場所は、補助セットのデータファイルを格納するための領域が十分にあるディスク上の場所にする必要があります。他の方法を使用して一部またはすべての補助セットのデータファイル名を変更した場合でも、AUXILIARY\_DESTINATION パラメータを指定すると、名前が指定されていない補助セットのデータファイルにデフォルトの場所が提供されます。補助セットの一部のデータファイルに名前を指定しなかった場合でも、TSPITR は正常に実行されます。

#### 完全に自動化された Recovery Manager の TSPITR を実行する手順

1. 20-4 ページの「TSPITR の前提条件および結果」の情報を確認します。
2. 20-5 ページの「TSPITR の計画および準備」のタスクを実行します。
3. ターゲット・データベースで Recovery Manager セッションを開始し、必要に応じてリカバリ・カタログに接続します。

---

**注意：** 自動化された TSPITR で Recovery Manager クライアントを起動する場合は、補助インスタンスに接続しないでください。Recovery Manager は、RECOVER TABLESPACE を実行するときに補助インスタンスに接続していると、20-18 ページの「独自の補助インスタンスを使用した Recovery Manager の TSPITR の実行」で説明されているように、ユーザー独自の補助インスタンスを管理しようとしていると判断し、接続している補助インスタンスを TSPITR に使用します。

---



4. TSPITR に必要なすべてのチャンネルを、ターゲット・インスタンスで構成します。  
補助インスタンスでは、TSPITR の実行時に、ターゲット・インスタンスと同じチャンネル構成が使用されます。
5. UNTIL 句および AUXILIARY DESTINATION パラメータを指定して、RECOVER TABLESPACE コマンドを実行します。

例 20-5 では、users および tools 表領域をログ順序番号 1300 まで戻し、補助インスタンスのファイルを /disk1/auxdest ディレクトリに格納します。

#### 例 20-5 2 つの表領域に対する TSPITR の実行

```
RECOVER TABLESPACE users, tools
UNTIL LOGSEQ 1300 THREAD 1
AUXILIARY DESTINATION '/disk1/auxdest';
```

この次の手順は、RECOVER コマンドの結果によって異なります。

- TSPITR の実行中にエラーが発生しなかった場合は、手順 6 に進みます。

表領域は、Recovery Manager によってオフラインにされ、バックアップからリストアされて補助インスタンスで目標時点までリカバリされた後、ターゲット・データベースに再インポートされます。表領域はオフラインのままです。補助セットのすべてのデータファイルおよび補助インスタンスの他のファイルは、補助の格納場所から削除されます。

- TSPITR の実行中にエラーが発生した場合は、20-23 ページの「[Recovery Manager の TSPITR のトラブルシューティング](#)」に進みます。

補助セットのデータファイルおよび補助インスタンスの他のファイルは、補助の格納場所に残され、トラブルシューティングに使用できます。リカバリ・セット・ファイルの状態は、エラーのタイプによって決まります。問題の解決後、TSPITR を再試行できます。

---

**注意：** 補助の格納場所にあるファイルは Oracle Managed Files です。これらのファイルは、ターゲット・データベースではカタログに追加されないため、2 回目の TSPITR では使用されません。ただし、補助セットに対して AUXILIARY DESTINATION のかわりに SET NEWNAME を指定し、オンライン・ログがリセットされる前に障害が発生した場合、Recovery Manager は既存のデータファイルを再利用することができます。

---

6. TSPITR が正常に完了した場合は、リカバリした表領域をオンラインにする前に、その表領域をバックアップします。

たとえば、次のコマンドを入力します。

```
BACKUP TABLESPACE users, tools;
```

表領域に対して TSPITR を実行した後、TSPITR が完了し、表領域がオンラインになるまで、その表領域のバックアップは使用できません。バックアップを作成せずにリカバリした表領域を使用すると、これらの表領域の使用可能なバックアップが存在しない状態でデータベースを実行することになります。

7. 表領域をオンラインに戻します。

たとえば、次のコマンドを入力します。

```
RMAN> SQL "ALTER TABLESPACE users, tools ONLINE";
```

これで、リカバリした表領域を使用できます。

## Recovery Manager 管理の補助インスタンスを使用した Recovery Manager のカスタマイズ TSPITR の実行

Recovery Manager の TSPITR では、多くの操作で 20-8 ページの「[完全に自動化された Recovery Manager の TSPITR の実行](#)」の手順に従う必要がありますが、次のタスクはカスタマイズできます。

- ファイル名を Oracle Managed Files 形式に変更するタスク。  
このタスクについては、20-10 ページの「[TSPITR での Oracle Managed Files の名前の変更](#)」を参照してください。
- TSPITR の実行後に、リカバリした表領域を構成するデータファイルが元の場所に格納されないように、リカバリ・セットのデータファイルを名前変更または再配置するタスク。  
これは、表領域が格納されていたディスクを使用できない場合などに行う必要があります。  
このタスクについては、20-10 ページの「[SET NEWNAME を使用した TSPITR のリカバリ・セットのデータファイル名の変更](#)」を参照してください。
- 補助セットの一部またはすべてのデータファイルの格納場所を、補助の格納場所以外の場所に指定するタスク。このオプションは、すべての補助セット・ファイルを格納するための十分な領域がある場所がディスク上に存在しない場合に選択します。  
このタスクについては、20-11 ページの「[TSPITR の補助セットのデータファイル名の指定](#)」を参照してください。
- TSPITR の実行中にデータファイルをリストアする必要がないように、データファイルのイメージ・コピー・バックアップを事前に設定するタスク。  
このタスクについては、20-14 ページの「[イメージ・コピーを使用した Recovery Manager の TSPITR の高速化](#)」を参照してください。
- Recovery Manager 管理の補助インスタンスの初期化パラメータをカスタマイズするタスク。  
このタスクについては、20-14 ページの「[TSPITR における自動補助インスタンスの初期化パラメータのカスタマイズ](#)」を参照してください。

### TSPITR での Oracle Managed Files の名前の変更

データファイルまたはオンライン REDO ログ・ファイルが Oracle Managed Files (OMF) 形式であるときに TSPITR をカスタマイズする場合は、次の方法でファイル名を変更できます。名前の変更方法を推奨順に示します。

1. 補助の格納場所を使用します (20-8 ページの「[完全に自動化された Recovery Manager の TSPITR の実行](#)」を参照)。
2. 補助インスタンスに Oracle Managed Files 初期化パラメータ `DB_CREATE_FILE_DEST`、`DB_RECOVERY_FILE_DEST`、および `DB_CREATE_ONLINE_LOG_DEST_n` を 1 つ以上使用して格納場所を指定します。 `DB_FILE_NAME_CONVERT` または `LOG_FILE_NAME_CONVERT` 初期化パラメータは設定しないでください。
3. 補助インスタンスに `DB_FILE_NAME_CONVERT` および `LOG_FILE_NAME_CONVERT` 初期化パラメータを設定します。ASM で Oracle Managed Files を使用すると、Recovery Manager は、ASM ディスク・グループ名のみを変換するパターンを使用して、変換されたディスク・グループの有効な OMF ファイル名を生成します。

### SET NEWNAME を使用した TSPITR のリカバリ・セットのデータファイル名の変更

リカバリ・セットのデータファイルは、元の場所にリストアおよびリカバリしない方がよい場合があります。SET NEWNAME コマンドを使用すると、新しい格納場所を指定できます。CONFIGURE AUXNAME を使用してリカバリ・セットのデータファイルの名前を変更することもできますが、その影響は異なります。CONFIGURE AUXNAME の詳細は、20-14 ページの「[イメージ・コピーを使用した Recovery Manager の TSPITR の高速化](#)」を参照してください。



リカバリ・セットの新しいファイル名を指定するには、RUN ブロックを作成し、そのブロック内で SET NEWNAME コマンドを使用します。複数のファイル名が相互に競合しない名前、または現行のデータファイル名と競合しない名前を割り当ててください。例 20-6 に、基本的な方法を示します。

#### 例 20-6 リカバリ・セットのファイル名の変更

```
RUN
{
.
.
.
SET NEWNAME FOR DATAFILE 'ORACLE_HOME/oradata/trgt/users01.dbf'
  TO '/newfs/users01.dbf';
...other setup commands...
RECOVER TABLESPACE users, tools UNTIL SEQUENCE 1300 THREAD 1;
}
```

TSPITR の実行中に、指定した各データファイルが新しい場所にリストアされ、その場所でリカバリされます。また、制御ファイルが更新され、制御ファイル内の古いデータファイルが新しくリカバリしたデータファイルに置き換えられます。Recovery Manager は、新しく指定した場所で検出されたデータファイルの既存のイメージ・コピー・バックアップを上書きします。

Recovery Manager は、リカバリを実際に実行するまで、SET NEWNAME で設定された名前とターゲット・データベースの現行のデータファイル名との競合を検出しません。Recovery Manager が競合を検出すると、TSPITR は失敗し、Recovery Manager はエラーをレポートします。有効なデータファイルは上書きされません。

## TSPITR の補助セットのデータファイル名の指定

通常は元の場所に格納されるリカバリ・セットのデータファイルとは異なり、補助セットのデータファイルは、ターゲット・データベース内の対応する元のファイルを上書きできません。元の場所とは異なる補助セット・ファイルの格納場所を指定しない場合、TSPITR は失敗します。Recovery Manager が元のデータベース内の対応するファイルの上書きを試行し、そのファイルが使用中であることを検出した場合、障害が発生します。

補助セットのデータファイルの格納場所には、TSPITR の補助の格納場所を指定する方法が最も簡単です。ただし、Recovery Manager では、補助セットのデータファイルの場所を制御するための代替方法がサポートされています。表 20-2 に、それらの方法を優先順位に従って示します。

表 20-2 ファイル名を指定する場合の優先順位

| 順位 | 方法   | 参照先   |
|----|--|---|
| 1  | SET NEWNAME                                      | 「SET NEWNAME を使用した補助セットのデータファイル名の指定」(20-12 ページ)                     |
| 2  | CONFIGURE AUXNAME                                | 「補助セットのイメージ・コピーでの SET NEWNAME および CONFIGURE AUXNAME の使用」(20-15 ページ) |
| 3  | DB_FILE_NAME_CONVERT                             | 「DB_FILE_NAME_CONVERT を使用した補助セットのデータファイル名の指定」(20-12 ページ)            |
| 4  | RECOVER TABLESPACE に対する AUXILIARY DESTINATION 引数 |   |

2 つの設定が同時に適用された場合、優先順位の高い設定が優先順位の低い設定より優先されます。たとえば、補助セットの一部のデータファイルの補助名が CONFIGURE AUXNAME で構成されているときにターゲット・データベースで RECOVER TABLESPACE... AUXILIARY DESTINATION を実行した場合などです。

前述のいずれかの方法を使用して特定のファイルの格納場所を指定する場合でも、RECOVER TABLESPACE に AUXILIARY DESTINATION 引数を指定することをお勧めします。これによって、補助セットのデータファイル名を変更し忘れた場合でも、TSPITR が正常に実行されます。名前を変更していないファイルは、補助の格納場所に格納されます。

---

**注意：**『Oracle Database バックアップおよびリカバリ・リファレンス』で説明するように、SHOW AUXNAME コマンドを実行することによって、現在の CONFIGURE AUXNAME 設定を表示できます。

---

## SET NEWNAME を使用した補助セットのデータファイル名の指定

補助セットのデータファイルに新しい名前を指定するには、RECOVER TABLESPACE コマンドを RUN ブロックに含め、SET NEWNAME コマンドを RUN ブロック内で使用してファイル名を変更します。例 20-7 に、基本的な方法を示します。

### 例 20-7 補助セットのファイル名の変更

```

RUN
{
  SET NEWNAME FOR DATAFILE '?/oradata/prod/system01.dbf'
  TO '/disk1/auxdest/system01.dbf';
  SET NEWNAME FOR DATAFILE '?/oradata/prod/sysaux01.dbf'
  TO '/disk1/auxdest/sysaux01.dbf';
  SET NEWNAME FOR DATAFILE '?/oradata/prod/undotbs01.dbf'
  TO '/disk1/auxdest/undotbs01.dbf';
  RECOVER TABLESPACE users, tools
  UNTIL LOGSEQ 1300 THREAD 1
  AUXILIARY DESTINATION '/disk1/auxdest';
}

```

この結果は、RECOVER TABLESPACE の実行時に /disk1/auxdest/system01.dbf が存在するかどうかによって異なります。?/oradata/system01.dbf が指定した場所に存在し、TSPITR の UNTIL の時点より前の SCN で作成されている場合、動作は 20-15 ページの「補助セットのイメージ・コピーでの SET NEWNAME および CONFIGURE AUXNAME の使用」で説明されているとおりになります。それ以外の場合、Recovery Manager は、デフォルトの場所ではなく NEWNAME で指定した場所に補助セットのデータファイルをリストアします。補助セットのデータファイルが格納される場所のみを制御する場合は、TSPITR の実行前に SET NEWNAME で指定した場所にファイルが格納されていないことを確認します。

## DB\_FILE\_NAME\_CONVERT を使用した補助セットのデータファイル名の指定

補助セットのすべてのデータファイルには補助の格納場所を使用せず、すべてのデータファイルには個々に名前も指定しない場合を想定します。この場合、補助インスタンスで使用される初期化パラメータ・ファイルに、DB\_FILE\_NAME\_CONVERT 初期化パラメータを含めることができます。この方法は、次の場合にのみ使用できます。

- 20-16 ページの「TSPITR における自動補助インスタンスの初期化パラメータのカスタマイズ」で説明するように、自動管理補助インスタンスに対して独自の初期化パラメータ・ファイルを作成します。
- 20-18 ページの「独自の補助インスタンスを使用した Recovery Manager の TSPITR の実行」で説明するように、独自の補助インスタンスを作成します。

補助インスタンス内の DB\_FILE\_NAME\_CONVERT 初期化パラメータによって、ターゲット・データベース・インスタンス内の対応するファイルの元の名前から補助インスタンスのファイルの名前を導出する方法が指定されます。このパラメータは、文字列のペアのリストで構成されます。ペアの 1 つ目の文字列を含むファイル名の場合、補助インスタンスの一致するファイル名は、ペアの 2 つ目の文字列を元のファイル名に置き換えることによって生成されます。

たとえば、ターゲット・インスタンス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
- SYSAUX 表領域の ?/oradata/trgt/sysaux01.dbf
- undotbs 表領域の ?/oradata/trgt/undotbs01.dbf

補助インスタンス内の対応するファイルを /bigtmp 内に配置する場合は、次の行を補助インスタンスのパラメータ・ファイルに追加します。

```
DB_FILE_NAME_CONVERT=('?/oradata/trgt', '/bigtmp')
```

対応する補助インスタンス・ファイルの新しいファイル名は、  
/bigtmp/trgt/system01.dbf、/bigtmp/trgt/sysaux01.dbf および  
/bigtmp/trgt/undotbs01.dbf です。

最も重要な点は、補助インスタンスのパラメータ・ファイルに DB\_FILE\_NAME\_CONVERT を含める**必要がある**ことです。補助インスタンスを手動で作成した場合は、その補助インスタンスのパラメータ・ファイルに DB\_FILE\_NAME\_CONVERT を追加します。

SET NEWNAME または CONFIGURE AUXNAME を使用して、補助セットの個々のデータファイル名を変更することもできます。また、DB\_FILE\_NAME\_CONVERT で指定したパターンと一致しないファイルの名前は変更されません。RECOVER TABLESPACE の AUXILIARY DESTINATION パラメータを使用して、補助セットのすべてのデータファイルが格納場所に転送されていることを確認することをお勧めします。使用した名前の変更方法によって補助インスタンスの新しいファイル名が提供されない場合、TSPITR は失敗します。

**TSPITR での DB\_FILE\_NAME\_CONVERT を使用した OMF データファイル名の変更** Oracle Managed Files (OMF) は、ASM ストレージまたは非 ASM ストレージで使用できます。

DB\_FILE\_NAME\_CONVERT 初期化パラメータが設定されている場合は、OMF のストレージが ASM であるかどうかによって、TSPITR で実行される名前の変換方法が異なります。ASM に格納されている OMF ファイルの場合は、データベースによってディスク・グループ名のみが +DISK1 から +DISK2 などように変換されます。ASM に格納されていない OMF ファイルの場合は、データベースによって DB\_FILE\_NAME\_CONVERT 格納場所に格納され、部分文字列が置き換えられます。ターゲット・インスタンスの OMF ファイル名の部分文字列を置き換えることによって、補助インスタンスに有効な非 ASM の OMF ファイル名を生成することはできません。そのため、DB\_FILE\_NAME\_CONVERT を使用して、この状況で新しい名前の生成を制御することはできません。

この問題を回避するために、OMF ファイル (ASM に格納されているファイルを含む) の名前を生成するためにサポートされているその他のオプションのいずれかを使用します。

- 補助の格納場所を使用します (20-8 ページの「完全に自動化された Recovery Manager の TSPITR の実行」を参照)。
- 補助インスタンスのパラメータ・ファイルで DB\_CREATE\_FILE\_DEST 初期化パラメータを使用して、SET NEWNAME または CONFIGURE AUXNAME では新しい名前が指定されないすべての補助インスタンス・ファイルの場所を指定します。
- ASM ファイルに対して、SET NEWNAME を使用して、個々のファイルを補助インスタンスからアクセスできる固有のディスク・グループにリダイレクトできます (また、データベースではディスク・グループ内でファイル名を生成できます)。たとえば、次のように入力します。

```
RUN
{
  SET NEWNAME FOR DATAFILE 1 TO "+DISK2";
  SET NEWNAME FOR DATAFILE 2 TO "+DISK3";
  RECOVER TABLESPACE users, tools
    UNTIL LOGSEQ 1300 THREAD 1
    AUXILIARY DESTINATION '/disk1/auxdest';
}
```

**TSPITR の実行中の一時ファイルの名前の変更** 一時ファイルは、データベースの補助セットの一部とみなされます。補助インスタンスのインスタンス化時に、Recovery Manager は、ターゲット・データベースの一時表領域を再作成し、補助データファイル名の通常の規則に従って名前を生成します。

SET NEWNAME FOR TEMPFILE、DB\_FILE\_NAME\_CONVERT または AUXILIARY DESTINATION コマンドを使用すると、一時ファイルの名前を変更できます。補助インスタンスのオープン時に、Recovery Manager は、名前の変更規則に従って一時ファイルを再作成します。補助インスタンスを削除すると、残りの補助インスタンス・ファイルとともに一時ファイルが削除されます。

## イメージ・コピーを使用した Recovery Manager の TSPITR の高速化

リカバリ・セットおよび補助セットのデータファイルの既存のイメージ・コピーを Recovery Manager が使用するようリダイレクトすることによって、TSPITR のパフォーマンスを向上させることができます。この場合、Recovery Manager はバックアップからデータファイルをリストアする必要がありません。次の方法を使用して、データファイルのイメージ・コピーの存在を Recovery Manager に認識させることができます。

- リカバリ・セットのデータファイルまたは補助セットのデータファイルを指定して CONFIGURE AUXNAME コマンドを使用します。
- 補助セットのデータファイルのイメージ・コピーを指定して SET NEWNAME コマンドを使用します。

通常、指定した場所で適切なイメージ・コピーが使用可能な場合は、そのイメージ・コピーが Recovery Manager によってターゲット・インスタンスの Recovery Manager リポジトリのカタログから削除され、補助インスタンスの制御ファイルのカタログに追加されます。このイメージ・コピーを使用して、補助インスタンスで Point-in-Time リカバリが実行されます。詳細は、使用するコマンド、およびファイルが補助セット・ファイルかリカバリ・セット・ファイルかによって異なります。

### リカバリ・セットのイメージ・コピーでの CONFIGURE AUXNAME の使用

TSPITR の実行中、Recovery Manager は、AUXNAME で指定した場所でデータファイルを検索します。また、目標時点までリカバリ可能な古いデータファイル・チェックポイント SCN を持つデータファイルのイメージ・コピー・バックアップが存在するかどうかを確認します。使用可能なイメージ・コピーが検出されると、Recovery Manager は、そのイメージ・コピーを TSPITR で使用します。検出されない場合、Recovery Manager は、データファイルをリストアし、元の場所でリカバリします。AUXNAME で指定した場所のファイルは変更または削除されません。例 20-8 に、この方法を示します。

#### 例 20-8 CONFIGURE AUXNAME の使用

```
CONFIGURE AUXNAME FOR DATAFILE 'ORACLE_HOME/oradata/trgt/users01.dbf'
  TO '/newfs/users1.dbf';
...other RMAN commands, if any...
RECOVER TABLESPACE users, tools UNTIL SEQUENCE 1300 THREAD 1;
```

CONFIGURE AUXNAME は、主にリストア時間を削減して TSPITR を高速化するために使用されます。TSPITR を実行する可能性がある表領域が存在する場合は、影響を受けるデータファイルのイメージ・コピー・セットをバックアップ・ルーチンで保持し、そのデータファイルのイメージ・コピーを TSPITR を実行する可能性がある最も早い目標時点まで定期的に更新できます。予測される実行手順のモデルは、次のとおりです。

1. この計画の作成時に、ファイルに対して AUXNAME を 1 回構成します。
2. BACKUP AS COPY DATAFILE n FORMAT auxname を定期的に行って、更新されたイメージ・コピーを保持します。パフォーマンスを向上させるために、増分更新バックアップ方法を使用して、データファイルの全体バックアップを実行せずにイメージ・コピーを最新に保ちます。

3. TSPITR を実行する必要がある場合は、イメージ・コピーの最新の更新より後の目標時点を指定します。

イメージ・コピーが必要な表領域が事前にわからない場合があることに注意してください。20-6 ページの「リカバリ・セットの決定」で説明されているように、リカバリする表領域と他の表領域の間に関係が存在すると、最後のリカバリ・セットに表領域を追加する必要がある場合があります。また、補助セット内に別の表領域が残る場合もあります。リカバリ・セットに含める可能性がある各データファイルに対して AUXNAME を構成し、すべてのデータファイルのイメージ・コピーを頻繁に更新する必要があります。

すべての表領域がリカバリ・セットに含まれることを正しく予測できない場合があります。また、オーバーヘッドを考慮して、リカバリ・セットに含まれる可能性があるすべての表領域のコピーを保持しないでおく場合もあります。いずれの場合も、データファイルのサブセットのみを準備できます。TSPITR のプロセスは同じです。Recovery Manager は、元の場所にイメージ・コピーがないリカバリ・セットのデータファイルがリカバリされる必要があるため、プロセスには時間がかかります。

ネーミング・メソッドの優先順位は、CONFIGURE AUXNAME を使用してリカバリ・セット・ファイルの名前を変更する場合にも適用されることに注意してください。TSPITR コマンドの一部としてのリカバリ・セット・ファイルに対する SET NEWNAME は、同じファイルに対する CONFIGURE AUXNAME コマンドより優先されます。このインスタンスの動作は、20-10 ページの「SET NEWNAME を使用した TSPITR のリカバリ・セットのデータファイル名の変更」での説明のとおりになります。リカバリ・セット・ファイルで使用される SET NEWNAME では、イメージ・コピーは参照されません。

### 補助セットのイメージ・コピーでの SET NEWNAME および CONFIGURE AUXNAME の使用

リカバリ・セットのデータファイルと同様に、CONFIGURE AUXNAME を使用すると、補助セットのデータファイルのイメージ・コピーの永続的な代替格納場所を設定できます。一方、SET NEWNAME を使用すると、RUN コマンド実行中の代替格納場所を設定できます。ただし、Recovery Manager は、リカバリ・セットのデータファイルとは異なる方法で補助セットのデータファイルの値を処理します。つまり、補助セットのファイルでは、常に AUXNAME または NEWNAME が使用されますが、リカバリ・セットのファイルでは、有効なコピーが存在する場合にのみ AUXNAME または NEWNAME が使用されます。

SET NEWNAME または CONFIGURE AUXNAME を使用して、補助セットのデータファイルの新しい格納場所を指定するとします。また、TSPITR で使用可能な SCN を持つイメージ・コピーがその場所に存在するとします。この場合、Recovery Manager はイメージ・コピーを使用します。ただし、使用可能なイメージ・コピーがその場所に存在しない場合、Recovery Manager は、使用可能なコピーをバックアップからリストアします。(イメージ・コピーは存在するが、SCN が TSPITR の目標時点より後の場合は、リストアしたファイルによってデータファイルが上書きされます。)

すべての補助セットのファイルと同様に、TSPITR が正常に実行されると、ファイルは削除されます。TSPITR が失敗すると、トラブルシューティングで使用するために、ファイルは削除されません。そのファイルが、TSPITR の実行前に作成されたイメージ・コピーであるか、TSPITR の実行中に Recovery Manager によってバックアップからリストアされたかに関係なく、このように動作します。

### CONFIGURE AUXNAME およびイメージ・コピーを使用した TSPITR の実行の例

TSPITR で使用するデータベース全体のイメージ・コピーを保存するための十分なディスク領域があるとします。TSPITR を実行する必要がある場合の準備として、次の手順を実行します。

- 次の形式のコマンドを使用して、データベース内の各データファイルの AUXNAME を構成します。

```
CONFIGURE AUXNAME FOR DATAFILE n TO auxname_n;
```



- 次の形式のコマンドを使用して、毎週日曜日にデータベースのイメージ・コピーを作成します。

```
BACKUP AS COPY DATAFILE n FORMAT auxname_n
```

すべてのイメージ・コピーがディスクの同じ場所に格納され、元のデータファイルに類似した名前が付けられている場合は、すべてのデータファイルのバックアップが個々に実行されないようにできます。かわりに、BACKUP コマンドの FORMAT または DB\_FILE\_NAME\_CONVERT オプションを指定して、BACKUP AS COPY DATABASE を使用できます。たとえば、構成した補助名が場所 maindisk を auxdisk に変換したものである場合は、次のコマンドを使用できます。

```
BACKUP AS COPY
DATABASE
DB_FILE_NAME_CONVERT (maindisk, auxdisk);
```

---

**注意：** 通常、Oracle 管理ファイルの名前は簡単な置換を使用して変換できないため、DB\_FILE\_NAME\_CONVERT を使用して、OMF に格納されたイメージ・コピーの名前を生成する方法は使用できません。

---

これで、バックアップからリストアせずに TSPITR の準備が完了します。たとえば、2007 年 11 月 15 日 19:00:00 に誤ったバッチ・ジョブが開始され、これによって表領域 parts の表が不正に更新された場合、次のコマンドを使用して、表領域 parts で TSPITR を実行できます。

```
RECOVER TABLESPACE parts UNTIL TIME 'November 15 2007, 19:00:00';
```

AUXNAME の場所を構成すると、TSPITR の目標時点より前の SCN 以降のデータファイルのコピーを参照するため、補助セットおよびリカバリ・セットのデータファイルは、バックアップからリストアされません。かわりに、データファイルのコピーはリカバリに直接使用され、リストアのオーバーヘッドが解消されます。

TSPITR の完了時に、表領域 parts のデータファイルが元のデータファイルの場所には格納されていないことに注意してください。かわりに、それらのデータファイルは AUXNAME の場所に格納されます。補助セットに AUXNAME の場所のみを使用する場合は、リカバリ・セットのファイルに CONFIGURE AUXNAME ... CLEAR を実行してから、TSPITR を開始します。この場合、Recovery Manager はデータファイルをリストアする必要があります。

## TSPITR における自動補助インスタンスの初期化パラメータのカスタマイズ

自動補助インスタンスでは、オペレーティング・システム依存のファイルで初期化パラメータが検索されます。TSPITR の実行時、Recovery Manager では、常に、自動補助インスタンスのデフォルト・パラメータが検索されます。ファイルが検出されない場合でも、Recovery Manager でエラーは発生しません。かわりに、Recovery Manager は、自動補助インスタンスに対して表 20-3 の基本初期化パラメータを定義します。

表 20-3 デフォルトの初期化パラメータ

| 初期化パラメータ            | 値   |
|---------------------|---|
| DB_NAME             | ソース・データベースの DB_NAME と同じです。  |
| COMPATIBLE          | ターゲット・データベースの互換性設定と同じです。  |
| DB_UNIQUE_NAME      | DB_NAME の値に基づいて生成された一意の値。   |
| DB_BLOCK_SIZE       | ターゲット・データベースの DB_BLOCK_SIZE と同じです。  |
| DB_CREATE_FILE_DEST | 補助の格納場所 (AUXILIARY DESTINATION 引数が設定されている場合のみ)。Recovery Manager は、Oracle 管理の制御ファイルおよびオンライン・ログをこの場所に作成します。 |

表 20-3 デフォルトの初期化パラメータ (続き)

| 初期化パラメータ         | 値  |
|------------------|--|
| CONTROL_FILES    | 補助の格納場所で生成されたファイル名 (AUXILIARY DESTINATION 引数が設定されている場合のみ)。Recovery Manager は、制御ファイルをこの場所に作成します。<br><br>デフォルトでは、Recovery Manager は、補助インスタンスに対して 1 つの制御ファイルをオペレーティング・システム固有の場所に作成します。UNIX の場合、デフォルトの場所は ?/dbs/cntrl_@.dbf になります。ここで、? は ORACLE_HOME、@ は ORACLE_SID を表します。自動補助インスタンスの場合、ORACLE_SID は Recovery Manager によってランダムに生成されます。<br><br><b>参照:</b> CONTROL_FILES 初期化パラメータの使用の詳細は、『Oracle Database リファレンス』を参照してください。 |
| LARGE_POOL_SIZE  | 1M   |
| SHARED_POOL_SIZE | 110M   |
| PROCESSES        | 50   |

ほとんどの場合、特に RECOVER TABLESPACE に AUXILIARY DESTINATION 引数を指定している場合は、パラメータ・ファイルを変更する必要はありません。表 20-3 の初期化パラメータのいずれかを不適切な値で上書きすると、補助インスタンスで問題が発生し、TSPITR が失敗する場合があります。ただし、これらの基本パラメータ以外に、その他のパラメータも必要に応じて追加できます。たとえば、DB\_FILE\_NAME\_CONVERT を使用して補助セットのデータファイルの名前を指定できます。

自動補助インスタンスのパラメータを指定するもう 1 つの方法として、ファイルに初期化パラメータを指定する方法があります。この場合、TSPITR を実行する前に SET AUXILIARY INSTANCE PARAMETER FILE コマンドを使用して、このファイルの場所を指定できます。SET AUXILIARY INSTANCE PARAMETER を使用して指定したパスは、ターゲット・インスタンスまたは補助インスタンスではなく、Recovery Manager クライアントが実行されるシステム上のパスであることに注意してください。

### TSPITR における補助インスタンスの制御ファイルの格納場所の指定

初期化パラメータ・ファイルを使用すると、補助インスタンスの制御ファイルに対して独自の格納場所を指定できます。CONTROL\_FILES 初期化パラメータを設定して、制御ファイルの格納場所を指定します。

制御ファイルの格納場所を明示的に指定しない場合、TSPITR 実行時に AUXILIARY DESTINATION パラメータを使用すると、Recovery Manager は、そのファイルを補助格納場所に格納します。AUXILIARY DESTINATION を使用しない場合、補助インスタンスの制御ファイルは、オペレーティング・システム固有の場所に格納されます。

格納場所にかかわらず、補助インスタンスの制御ファイルは、TSPITR 操作が正常に実行された後に削除されます。制御ファイルはサイズが比較的小さいため、補助制御ファイルの作成時に問題が発生することはほとんどありません。ただし、制御ファイルを作成するための十分な領域がない場合、TSPITR は失敗します。

### TSPITR における補助インスタンスのオンライン・ログの格納場所の指定

補助インスタンスのパラメータ・ファイルで LOG\_FILE\_NAME\_CONVERT 初期化パラメータを指定すると、このパラメータによってオンライン REDO ログの格納場所が決定されます。このパラメータを指定しない場合、Recovery Manager によって補助の格納場所が使用され、補助インスタンスが管理されているときは、補助格納場所にオンライン REDO ログが作成されます。

LOG\_FILE\_NAME\_CONVERT または AUXILIARY DESTINATION を使用してオンライン REDO ログの格納場所を指定しない場合は、TSPITR でオンライン REDO ログが作成されません。初期化パラメータ・ファイルに DB\_CREATE\_FILE\_DEST または LOG\_FILE\_CREATE\_DEST が指定されている場合でも、TSPITR の実行時に、補助インスタンスのオンライン REDO ログの作成は制御されません。

# 独自の補助インスタンスを使用した Recovery Manager の TSPITR の実行

通常、Recovery Manager の TSPITR の実行時に使用する補助インスタンスの作成および削除は、Recovery Manager によって管理する必要があります。ただし、ユーザーは、独自の補助インスタンスを作成および使用することもできます。

ユーザー独自のインスタンスを作成する理由の 1 つとして、TSPITR で使用されるチャンネルを制御できることがあげられます。自動補助インスタンスでは、ターゲット・データベースの構成済チャンネルが、補助インスタンスで構成されてリストア中に使用されるチャンネルの基本チャンネルとして使用されます。ターゲット・データベースの設定を変更するために、CONFIGURE を使用せずに異なるチャンネルを設定する必要がある場合があります。この場合、独自の補助インスタンスを使用できます。

## Recovery Manager の TSPITR に使用する独自の補助インスタンスの準備

補助インスタンスとしての使用に適した Oracle インスタンスを作成するには、次のすべての手順を実行する必要があります。

- [手順 1: 補助インスタンス用の Oracle パスワード・ファイルの作成](#)
- [手順 2: 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](#)
- [手順 3: Oracle Net から補助インスタンスへの接続性の確認](#)

### 手順 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・ファイルを作成およびメンテナンスする方法については、『Oracle Database 管理者ガイド』を参照してください。

### 手順 2: 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

テキスト・エディタを使用して、ターゲット・データベース・ホストで補助インスタンスの初期化パラメータ・ファイルを作成します。たとえば、パラメータ・ファイルが /tmp/initAux.ora に格納されていると想定します。表 20-4 で説明するように、パラメータを設定します。

**注意：** TSPITR の場合、ターゲット・データベースおよび補助データベースは同一ホスト上にある必要があります。

表 20-4 補助インスタンスの初期化パラメータ

| パラメータ                     | 必須かどうか | 値   |
|---------------------------|--------|---|
| DB_NAME                   | 必須     | ターゲット・データベースと同じ名前。  |
| DB_UNIQUE_NAME            | 必須     | 同じ Oracle ホーム内のいずれのデータベースとも異なる値。簡略化するために、_dbname を指定します。たとえば、ターゲット・データベース名が trgt の場合は、_trgt を指定します。 |
| REMOTE_LOGIN_PASSWORDFILE | 必須     | EXCLUSIVE (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)。それ以外の場合は、NONE に設定します。                                    |
| COMPATIBLE                | 必須     | ターゲット・データベースのパラメータと同じ値。   |
| DB_BLOCK_SIZE             | 必須     | 補助インスタンスと同じ値 (この初期化パラメータがターゲット・データベースで設定されている場合)。   |



表 20-4 補助インスタンスの初期化パラメータ (続き)

| パラメータ                 | 必須かどうか | 値   |
|-----------------------|--------|---|
| LOG_FILE_NAME_CONVERT | 必須     | <p>補助データベースのオンライン REDO ログのファイル名を生成するパターン。ターゲット・データベースのオンライン REDO ログ名に基づきます。V\$LOGFILE.MEMBER を問い合わせるターゲット・インスタンスのオンライン 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たファイル名の書式と一致することを確認します。</p> <p>補助インスタンスのオンライン REDO ログの名前は、このパラメータでのみ指定できます。このパラメータを指定しない場合は、オンライン・ログを作成できないため、補助インスタンスのオープン時に TSPITR が失敗します。</p> <p><b>注意:</b> 末尾にスラッシュ (\ または /) を指定するパターンがサポートされていないプラットフォームもあります。</p> <p><b>参照:</b> OMF ファイル名を使用した LOG_FILE_NAME_CONVERT に対して設定可能な値の制限については、20-17 ページの「<a href="#">TSPITR における補助インスタンスのオンライン・ログの格納場所の指定</a>」を参照してください。</p> |
| DB_FILE_NAME_CONVERT  | 任意     | <p>補助データベースのデータファイルのファイル名を変換するパターン。このパラメータを使用すると、SET NEWNAME または CONFIGURE AUXNAME で名前を付けなかったファイルのファイル名を生成できます。V\$DATAFILE.NAME を問い合わせる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たファイル名の書式と一致することを確認します。また、このパラメータを RECOVER コマンド自体に指定することもできます。</p> <p><b>注意:</b> 末尾にスラッシュ (\ または /) を指定するパターンがサポートされていないプラットフォームもあります。</p> <p><b>参照:</b> 20-12 ページの「<a href="#">DB_FILE_NAME_CONVERT を使用した補助セットのデータファイル名の指定</a>」を参照してください。</p>  |
| CONTROL_FILES         | 任意     | <p>ターゲット・インスタンス (または他の既存のファイル) の制御ファイル名と競合しないファイル名。</p>   |

Oracle Net を介して SYSDBA として接続するためのパラメータを含めて、その他のパラメータを必要に応じて設定します。

次に、TSPITR の補助インスタンスで設定可能な初期化パラメータの例を示します。

```
DB_NAME=trgt
DB_UNIQUE_NAME=_trgt
CONTROL_FILES=/tmp/control01.ct1
DB_FILE_NAME_CONVERT=('/oracle/oradata/trgt/', '/tmp/')
LOG_FILE_NAME_CONVERT=('/oracle/oradata/trgt/redo', '/tmp/redo')
REMOTE_LOGIN_PASSWORDFILE=exclusive
COMPATIBLE =11.0.0
DB_BLOCK_SIZE=8192
```

**注意:** これらの初期化パラメータを設定した後、ターゲット・データベースの本番ファイルの初期化設定を上書きしていないことを確認してください。

**参照:** Oracle Net の詳細は、『Oracle Database Net Services 管理者ガイド』を参照してください。

### 手順 3: Oracle Net から補助インスタンスへの接続性の確認

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 を使用して補助インスタンスに SYSDBA で接続できることを確認します。

## 独自の補助インスタンスを使用した TSPITR に使用する Recovery Manager コマンドの準備

独自の補助インスタンスを実行すると、TSPITR に必要な一連のコマンドが長くなる場合があります。この状況は、バックアップからのリストアに複雑なチャンネル構成を割り当てたり、DB\_FILE\_NAME\_CONVERT を使用せずにファイル名を指定した場合に発生する可能性があります。

TSPITR に使用する一連のコマンドは、Recovery Manager のコマンド・ファイルに格納できません。コマンド・ファイルにエラーがないかを慎重に確認します。このコマンド・ファイルを Recovery Manager に読み込むには、@ コマンド（または Recovery Manager の起動時に CMDFILE コマンドライン引数）を使用します。

次の例では、/tmp/tspitr.rman というコマンド・ファイルを実行します。

```
@/tmp/tspitr.rman;
```

結果は、前述の例と同じになります。

**参照:** 「Recovery Manager でのコマンド・ファイルの使用」 (4-4 ページ)

### 独自の補助インスタンスを使用した TSPITR に使用するチャンネルの計画

独自の補助インスタンスを実行すると、デフォルトでターゲット・インスタンスの自動チャンネル構成が使用されます。ただし、独自のチャンネル構成を割り当てる場合は、RUN ブロックに ALLOCATE AUXILIARY CHANNEL コマンドおよび TSPITR 用の RECOVER TABLESPACE コマンドを含めます。必要に応じてこれらのコマンドの使用を計画し、TSPITR に使用する一連のコマンドに追加します。

**参照:** TSPITR スクリプトにチャンネル割当てを含める方法については、20-21 ページの「独自の補助インスタンスを使用した TSPITR の実行例」を参照してください。

### 独自の補助インスタンスのデータファイル名の計画 : SET NEWNAME

SET NEWNAME コマンドを使用して、補助セット・ファイルの既存のイメージ・コピーを参照して TSPITR のパフォーマンスを向上させるか、または TSPITR の実行後にリカバリ・セット・ファイルに新しい名前を割り当てることができます。必要に応じてこれらのコマンドの使用を計画し、TSPITR に使用する一連のコマンドに追加します。

## 独自の補助インスタンスを使用した TSPITR の実行

準備および TSPITR コマンドの計画が完了すると、TSPITR を実行できます。次の手順を実行する必要があります。

- [手順 1: NOMOUNT モードでの補助インスタンスの起動](#)
- [手順 2: ターゲット・インスタンスおよび補助インスタンスへの Recovery Manager クライアントの接続](#)
- [手順 3: RECOVER TABLESPACE コマンドの実行](#)

## 手順 1: NOMOUNT モードでの補助インスタンスの起動

Recovery Manager の TSPITR を開始する前に、SQL\*Plus を起動して SYSOPER 権限で補助インスタンスに接続します。

補助インスタンスを NOMOUNT モードで起動し、必要に応じてパラメータ・ファイルを指定します。たとえば、次の SQL\*Plus コマンドを入力します。

```
SQL> STARTUP NOMOUNT PFILE='/tmp/initAux.ora'
```

PFILE を指定する場合は、PFILE のパスに、SQL\*Plus を実行するホスト上のクライアント側のパスを指定することに注意してください。

補助インスタンスに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。制御ファイルを作成したり、TSPITR 用の補助インスタンスをマウントまたはオープンしないでください。

## 手順 2: ターゲット・インスタンスおよび補助インスタンスへの Recovery Manager クライアントの接続

Recovery Manager を起動し、ターゲット・データベースおよび手動で作成した補助インスタンスに接続します。

## 手順 3: RECOVER TABLESPACE コマンドの実行

最も簡単な方法は、Recovery Manager プロンプトで RECOVER TABLESPACE ... UNTIL コマンドを実行する方法です。

```
RECOVER TABLESPACE ts1, ts2... UNTIL TIME 'time';
```

ALLOCATE AUXILIARY CHANNEL コマンドまたは SET NEWNAME コマンドを使用する場合は、RUN コマンドで RECOVER TABLESPACE コマンドの前にこれらのコマンドを指定します。

次に、この方法の例を示します。

```
RUN
{
  ALLOCATE AUXILIARY CHANNEL c1 DEVICE TYPE DISK;
  ALLOCATE AUXILIARY CHANNEL c2 DEVICE TYPE sbt;
  # and so on...
  RECOVER TABLESPACE ts1, ts2 UNTIL TIME 'time';
}
```

## 独自の補助インスタンスを使用した TSPITR の実行例

この項では、RECOVER TABLESPACE... UNTIL 操作の実行例を示します。この実行例では、Recovery Manager の TSPITR の次の機能について説明します。

- 独自の補助インスタンスの管理
- ディスクおよび SBT デバイスからのバックアップのリストアに対するチャンネルの構成
- SET NEWNAME を使用した補助セットのデータファイルのリカバリ可能なイメージ・コピーの使用
- SET NEWNAME を使用したリカバリ・セットのデータファイルの新しい名前の指定

次の手順を実行するとします。

1. 20-18 ページの「[Recovery Manager の TSPITR に使用する独自の補助インスタンスの準備](#)」で説明するように、補助インスタンスを準備します。補助インスタンスのパスワードをパスワード・ファイルに指定し、補助インスタンス・パラメータ・ファイル /bigtmp/init\_tspitr\_prod.ora を次のように設定します。

```
DB_NAME=PROD
DB_UNIQUE_NAME=tspitr_PROD
CONTROL_FILES=/bigtmp/tspitr_cntrl.dbf'
DB_FILE_NAME_CONVERT=('?/oradata/prod', '/bigtmp')
```

```
LOG_FILE_NAME_CONVERT=('?' /oradata/prod', '/bigtmp')
COMPATIBLE=11.0.0
BLOCK_SIZE=8192
REMOTE_LOGIN_PASSWORD=exclusive
```

2. 補助インスタンスのサービス名 `pitprod` を作成し、接続性を確認します。
3. `SQL*Plus` を使用して、`SYSOPER` 権限で補助インスタンスに接続します。インスタンスを `NOMOUNT` モードで起動します。

```
SQL> STARTUP NOMOUNT PFILE=/bigtmp/init_tspitr_prod.ora
```

4. `Recovery Manager` を起動し、ターゲット・データベース・インスタンスおよび補助データベース・インスタンスに接続します。
5. 次のコマンドを `RUN` ブロック内で入力し、`TSPITR` を設定および実行します。

```
RUN
{
# Specify NEWNAMES for recovery set datafiles
SET NEWNAME FOR DATAFILE '?' /oradata/prod/clients01.dbf'
    TO '?' /oradata/prod/clients01_rec.dbf';
SET NEWNAME FOR DATAFILE '?' /oradata/prod/clients02.dbf'
    TO '?' /oradata/prod/clients02_rec.dbf';
SET NEWNAME FOR DATAFILE '?' /oradata/prod/clients03.dbf'
    TO '?' /oradata/prod/clients03_rec.dbf';
SET NEWNAME FOR DATAFILE '?' /oradata/prod/clients04.dbf'
    TO '?' /oradata/prod/clients04_rec.dbf';

# Specified newnames for some of the auxiliary set
# datafiles that have a valid image copy to avoid restores:
SET NEWNAME FOR DATAFILE '?' /oradata/prod/system01.dbf'
    TO '/backups/prod/system01_monday_noon.dbf';
SET NEWNAME FOR DATAFILE '?' /oradata/prod/system02.dbf'
    TO '/backups/prod/system02_monday_noon.dbf';
SET NEWNAME FOR DATAFILE '?' /oradata/prod/sysaux01.dbf'
    TO '/backups/prod/sysaux01_monday_noon.dbf';
SET NEWNAME FOR DATAFILE '?' /oradata/prod/undo01.dbf'
    TO '/backups/prod/undo01_monday_noon.dbf';

# Specified the types of channels to use
ALLOCATE AUXILIARY CHANNEL c1 DEVICE TYPE DISK;
ALLOCATE AUXILIARY CHANNEL t1 DEVICE TYPE sbt;

# Recovered the clients tablespace to 24 hours ago:
RECOVER TABLESPACE clients UNTIL TIME 'sysdate-1';
}
```

コマンド・ファイルにこの一連のコマンドを格納し、コマンド・ファイルを実行することを検討してください。

`TSPITR` 操作が正常に実行されると、結果は次のようになります。

- リカバリ・セットのデータファイルが、`SET NEWNAME` で指定した名前および `TSPITR` の目標時点の内容で、ターゲット・データベースの制御ファイルに登録されます。
- `Recovery Manager` によって、補助インスタンスの制御ファイル、オンライン `REDO` ログ、補助セットのデータファイルなどの補助ファイルが削除されます。
- 補助インスタンスが停止します。

`TSPITR` 操作が失敗すると、トラブルシューティングのために補助ファイルがディスクに残されます。`Recovery Manager` によって補助インスタンスが作成された場合、その補助インスタンスは停止します。それ以外の場合は、`TSPITR` 操作が失敗したときの状態のまま残されます。

## Recovery Manager の TSPITR のトラブルシューティング

プロセスの完了前に、次のような様々な問題で TSPITR が失敗する可能性があります。

- ターゲット・データベース内の既存のファイル、SET NEWNAME または CONFIGURE AUXNAME コマンドで割り当てられたファイル名と、DB\_FILE\_NAME\_CONVERT パラメータの影響で生成されたファイル名の間で、名前の競合が発生する場合があります。
- リカバリしたオブジェクトに関するメタデータが Recovery Manager によって補助インスタンスからエクスポートされると、ソートのために一時表領域の領域が使用されます。一時表領域内にソート操作のための十分な領域がない場合は、ソート用に使用可能な領域を増やす必要があります。

### ファイル名の競合のトラブルシューティング

SET NEWNAME、CONFIGURE AUXNAME および DB\_FILE\_NAME\_CONVERT を使用したため、補助セットまたはリカバリ・セット内の複数のファイルに同じ名前が指定されたとします。この場合、Recovery Manager は、TSPITR の実行中にエラーをレポートします。この問題を解決するには、これらのパラメータに異なる値を使用して、重複する名前を排除します。

### UNDO セグメントを含む表領域の識別のトラブルシューティング

TSPITR の実行時、Recovery Manager には、TSPITR の目標時点においてどの表領域に UNDO セグメントが含まれていたかの情報が必要です。リカバリ・カタログが使用されている場合、通常、この情報はリカバリ・カタログから取得できます。

リカバリ・カタログが存在していないか、またはリカバリ・カタログで情報が検出されない場合を想定します。この場合、Recovery Manager は、目標時点で UNDO セグメントが含まれていた表領域のセットが、現在 UNDO セグメントが含まれている表領域のセットと同じであるとみなします。この仮定が正しくない場合、TSPITR は失敗してエラーが表示されます。この場合は、UNDO TABLESPACE 句を使用して、目標時点での UNDO セグメントを表領域のリストに含めます。

### TSPITR の失敗後の手動による補助インスタンスの再起動のトラブルシューティング

独自の補助インスタンスを管理している場合に、TSPITR で障害が発生したとします。TSPITR を再試行する前に、補助インスタンスを停止し、TSPITR を中断させた問題を解決する必要があります。その後、補助インスタンスを NOMOUNT モードで起動してから TSPITR を再試行します。



# 第 VI 部

---

## チューニングおよびトラブルシューティング

第 VI 部では、Recovery Manager 操作のチューニングおよびトラブルシューティングの方法について説明します。第 VI 部に含まれる章は次のとおりです。

- [第 21 章「Recovery Manager のパフォーマンスのチューニング」](#)
- [第 22 章「Recovery Manager 操作のトラブルシューティング」](#)





---

## Recovery Manager のパフォーマンスのチューニング

この章の内容は、次のとおりです。

- Recovery Manager のパフォーマンスのチューニングの目的
- Recovery Manager のパフォーマンスのチューニングの基本的な概念
- Recovery Manager のパフォーマンスの問題を診断するための V\$ ビューの使用方法
- Recovery Manager のバックアップのパフォーマンスのチューニング

## Recovery Manager のパフォーマンスのチューニングの目的

Recovery Manager のバックアップ・ジョブまたはリストア・ジョブは、別々のフェーズまたはコンポーネントに分けることができます。Recovery Manager ジョブでのこれらのフェーズのうち、最も低速なものを**ボトルネック**とといいます。Recovery Manager のチューニングの目的は、ジョブのボトルネックを特定し、Recovery Manager コマンド、初期化パラメータまたは物理メディアの調整を利用して、パフォーマンスを向上することです。

## Recovery Manager のパフォーマンスのチューニングの基本的な概念

Recovery Manager のパフォーマンスをチューニングするには、Recovery Manager がバックアップをどのように作成するかについて、詳細に理解しておく必要があります。3-4 ページの「[Recovery Manager チャンネル](#)」で説明されているように、バックアップの作業は、1 つ以上のチャンネルで実行されます。**チャンネル**は、ストレージ・デバイスへのバイト・ストリームを表しています。

バイト・ストリームは、メモリー内の入力バッファから CPU 経由で出力バッファに渡され、さらにそこからストレージ・デバイスに渡されます。2 つのテープ・デバイスにバックアップするように指定するには、2 つのテープ・チャンネルを割り当て、各バイト・ストリームが別々のデバイスに渡されるようにします。

各チャンネルの作業は、チャンネルのタイプがディスクまたは SBT のいずれであるかにかかわらず、次のフェーズに細分化されます。

### 1. 読取りフェーズ

ディスクのブロックが入力バッファに読み取られます。

### 2. コピー・フェーズ

入力バッファから出力バッファにブロックがコピーされ、ブロックに対する追加処理が実行されます。

### 3. 書込みフェーズ

出力バッファのブロックがストレージ・メディアに書き込まれます。書込みフェーズは、バックアップ・メディアのタイプに応じて、次の相互排他的な形式のいずれかになります。

- [SBT への書込みフェーズ](#)
- [ディスクへの書込みフェーズ](#)

[図 21-1](#) に、3 つのディスクに格納されているデータをバックアップする 2 つのチャンネルを示します。各チャンネルはデータを入力バッファに読み取り、入力バッファから出力バッファへのコピー時にデータを処理し、出力バッファからのデータをディスクに書き込みます。

図 21-1 ディスクへのマルチチャネル・バックアップのフェーズ

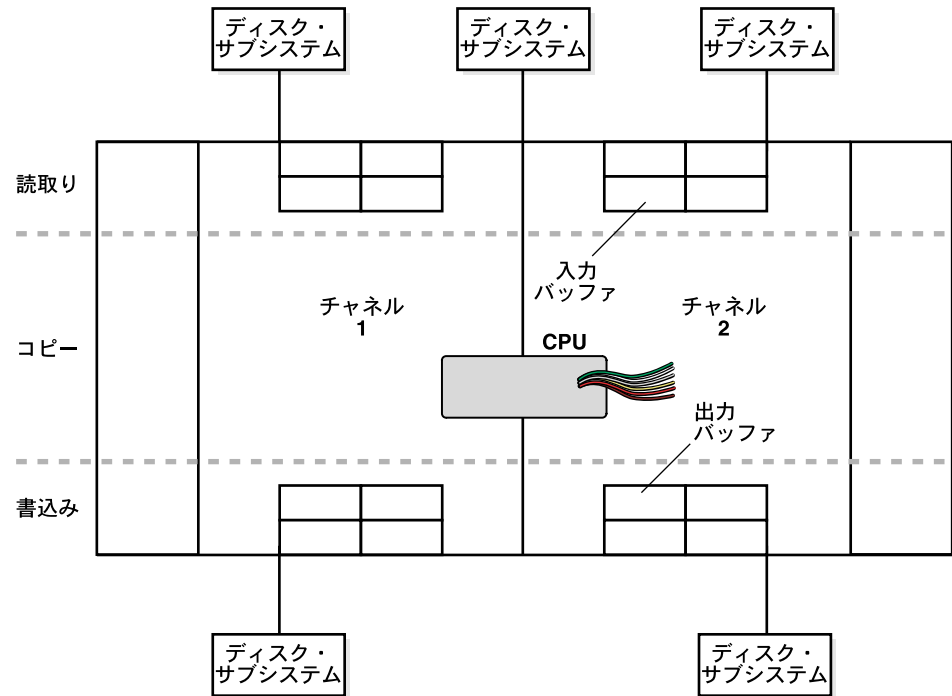
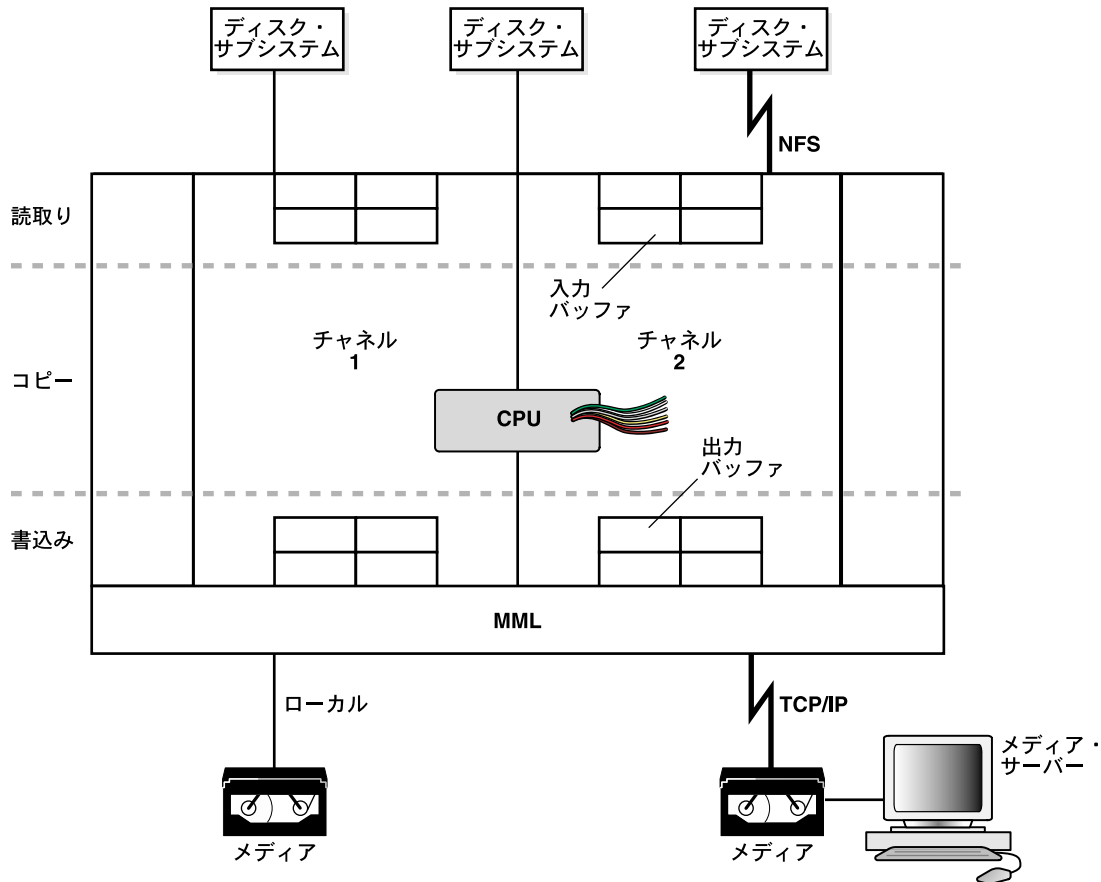


図 21-2 は、同様に、3つのディスクに格納されているデータをバックアップする2つのチャンネルを示していますが、ディスクの1つがネットワーク上にリモートでマウントされています。各チャンネルはデータを入力バッファに読み取り、入力バッファから出力バッファへのコピー時にデータを処理し、出力バッファからのデータをテープに書き込みます。チャンネル1はローカル接続されたテープ・ドライブにデータを書き込み、チャンネル2はネットワークを介してリモート・メディア・サーバーにデータを送信します。

図 21-2 テープへのマルチチャネル・バックアップのフェーズ



データをリストアする場合、これらの手順が逆順で実行され、読取りと書込みの操作が逆になります。次の項では、Recovery Manager のチューニングの概念について、バックアップを例に説明します。

## 読取りフェーズ

この項では、Recovery Manager チャンネルがディスクのデータを読み取る場合に、パフォーマンスに影響する次の要因について説明します。

- 入力ディスク・バッファの割当て
- 同期ディスク I/O と非同期ディスク I/O
- ディスク I/O スレーブ
- チャンネルの RATE パラメータ

### 入力ディスク・バッファの割当て

バックアップ中、Recovery Manager チャンネルは、入力ファイルのブロックを I/O ディスク・バッファに読み取ります。ディスク・サブシステムのデータベース・ファイルは、**自動ストレージ管理**、他のボリューム・マネージャまたはファイル・システムで管理できます。バックアップのチューニングに関する考慮事項は、データベース・ファイルを ASM で管理するかどうかによって異なります。

入力バッファの割当ては、ファイルをどのように多重化するかによって異なります。バックアップの**多重化**は、バックアップ内の多数のファイルを複数のソースから同時に読み取り、1つのバックアップ・ピースに書き込む Recovery Manager の機能です。同時に読み取られ、同じ**バックアップ・ピース**に書き込まれる入力ファイルの数を表す**多重化レベル**を決定するアルゴリズムの詳細は、7-7 ページの「**多重バックアップ・セット**」を参照してください。この章を読む前に、これらの項を参照してください。

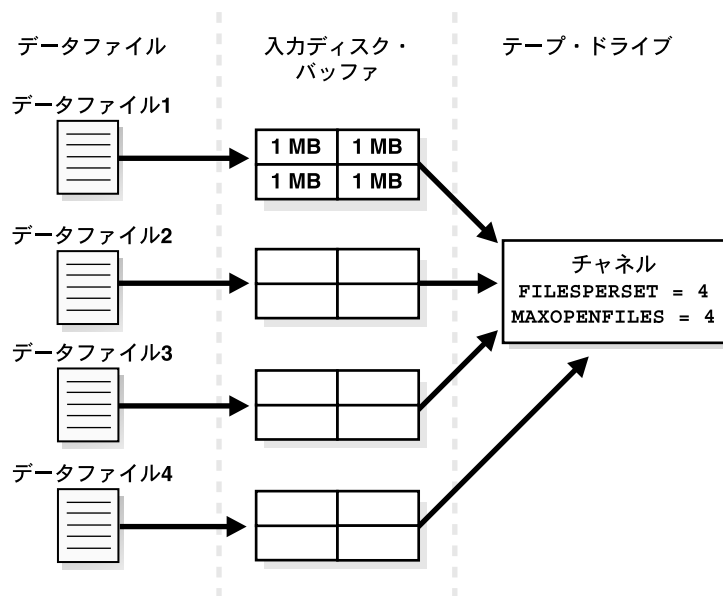
Recovery Manager チャンネルは、ディスクのファイルをバックアップするときに、次の表に示す規則を使用して、作成する入力ディスクのバッファのサイズを決定します。

表 21-1 データファイルの読取りバッファのサイズを決定するアルゴリズム

| 多重化レベル       | 入力ディスクのバッファのサイズ   |
|--------------|---|
| 4 以下         | すべての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 のバッファが 16 個割り当てられます。              |
| 4 より大きく 8 以下 | すべての入力ファイル用の合計バッファ・サイズが 16MB 未満になるように、512KB のディスク・バッファがいくつか割り当てられます。      |
| 8 より大きい      | 各入力ファイル用の合計バッファ・サイズが 512KB になるように、各ファイルに 128KB のディスク・バッファが 4 つずつ割り当てられます。 |

図 21-3 に、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する例を示します。MAXOPENFILES は 4 に設定され、FILESPESET は 4 に設定されています。したがって、多重化のレベルは 4 です。このため、各データファイル用の合計バッファ・サイズは 4MB になります。すべてのバッファを組み合わせたサイズは 16MB になります。

図 21-3 ディスク・バッファの割当て



BACKUP\_DISK\_IO\_SLAVESが0でない場合はSGA  
または  
BACKUP\_DISK\_IO\_SLAVESが0の場合はPGA

ASM に格納されているファイルをチャンネルがバックアップする場合、入力ディスクのバッファの数は、ASM ディスク・グループの物理ディスクの数と等しくなります。たとえば、16 個の物理ディスクを含む ASM ディスク・グループにデータファイルが格納されている場合、チャンネルはデータファイルのバックアップ用に 16 個の入力バッファを割り当てます。

チャンネルがディスクからバックアップをリストアする場合は、4 つのバッファが割り当てられます。バッファのサイズは、オペレーティング・システムによって異なります。

## 同期ディスク I/O と非同期ディスク I/O

チャンネルによるディスクに対する読取りまたは書込みでは、I/O は**同期 I/O** と**非同期 I/O** のいずれかです。ディスク I/O が同期の場合、サーバー・プロセスは一度に 1 つのタスクのみを実行できます。ディスク I/O が非同期の場合、サーバー・プロセスは 1 つの I/O を開始し、その I/O が完了するまで待機している間に他の作業を実行できます。また、1 つ目の I/O の完了の待機に入る前に、複数の I/O 操作を開始することもできます。

ASM **ディスク・グループ**から読み取るときには、可能な場合は非同期ディスク I/O を使用してください。また、ボリューム・マネージャで管理されている **RAW デバイス**からチャンネルが読み取る場合も、非同期ディスク I/O が適切に動作します。一部のオペレーティング・システムでは、固有の非同期ディスク I/O をサポートしています。データベースは、使用可能な場合はその機能を利用します。

## ディスク I/O スレーブ

固有の非同期 I/O をサポートしていないオペレーティング・システムの場合、データベースは、特別な I/O スレーブ・プロセスを使用してその機能をシミュレートできます。このようなプロセスは、別のプロセスのかわりに I/O を実行する専用のプロセスです。

ディスク I/O スレーブを制御するには、動的でない `DBWR_IO_SLAVES` 初期化パラメータを設定します。このパラメータでは、データベース・ライター・プロセス (DBWR) で使用される I/O サーバー・プロセスの数を指定します。デフォルトでは、この値は 0 (ゼロ) であり、I/O サーバー・プロセスは使用されません。`DBWR_IO_SLAVES` が 0 (ゼロ) 以外の値である場合に、非同期 I/O が無効であると、4 つのバックアップ・ディスク I/O スレーブが割り当てられます。

データベースが I/O スレーブ用の共有バッファを取得する際、次のことが行われます。

- `LARGE_POOL_SIZE` が設定されていて、かつ `DBWR_IO_SLAVES` パラメータの値が 0 (ゼロ) 以外に設定されている場合、データベースでは、ラージ・プールからのメモリの取得が試行されます。この値が十分大きくない場合、アラート・ログにエラーが記録されません。また、共有プールからのバッファの取得は試行されず、非同期 I/O は使用されません。
- `LARGE_POOL_SIZE` が設定されていない場合、または 0 (ゼロ) に設定されている場合、データベースは、共有プールからのメモリの取得を試行します。
- データベースが十分なメモリを取得できない場合、PGA から I/O バッファ・メモリを取得し、このバックアップに同期 I/O を使用するというメッセージを `alert.log` ファイルに書き込みます。

ラージ・プールのメモリは、共有サーバー、パラレル問合せ、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、Recovery Manager がメモリ取得のために他のサブシステムと競合することを防止できます。

共有プールからの連続メモリの割当ての要求は、通常は小さいサイズ (5KB 未満) の要求です。ただし、大きいサイズの連続メモリの割当てが要求されると、割当てが失敗するか、または要求された量の連続メモリーを解放するための大量のクリーンアップが必要となる場合があります。このメモリ要求は、共有プールでは満たせない場合がありますが、ラージ・プールでは満たすことができます。ラージ・プールには最低使用頻度 (LRU) リストが存在しないため、ラージ・プールからの古いメモリーの削除は試行されません。

## チャネルの RATE パラメータ

ALLOCATE コマンドまたは CONFIGURE CHANNEL コマンドでは、RATE パラメータで、チャネルで読み取られる速度（バイト / 秒）を指定します。このパラメータを使用すると、Recovery Manager が過度のディスク帯域幅を使用してオンライン・パフォーマンスが低下することを防止するために、読取りバイト数の上限を設定できます。基本的に、RATE は、バックアップの制限として機能します。たとえば、RATE=1500K と設定し、各ディスク・ドライブの速度が 3MB/ 秒である場合、チャネルは、ディスク帯域幅の一部をオンライン・システム用に確保します。

## コピー・フェーズ

このフェーズでは、入力バッファから出力バッファにブロックがコピーされ、追加処理が実行されます。たとえば、データがディスクから読み取られてテープにバックアップされる場合、データはディスク・バッファから出力テープ・バッファにコピーされます。

コピー・フェーズには、次の処理が含まれています。

- 検証
- 圧縮
- 暗号化

ブロックの**検証**を実行すると、ブロックが破損していないかどうかを確認されます。検証については、[第 15 章「データベース・ファイルおよびバックアップの検証」](#)を参照してください。通常、この処理では CPU に負荷はかかりません。

**バイナリ圧縮**を実行すると、Recovery Manager で、バックアップ・セット内のデータに圧縮アルゴリズムが適用されます。バイナリ圧縮は、CPU に負荷がかかる場合があります。Recovery Manager でバックアップに使用する圧縮アルゴリズムを選択できます。デフォルトでは、Recovery Manager は、圧縮率に優れている BZIP2 を使用します。ZLIB 圧縮では、COMPATIBLE を 11.0.0 以上に設定する必要があり、高速ですが圧縮率は他のアルゴリズムより悪くなります。バイナリ圧縮については、8-6 ページの「[圧縮バックアップの作成](#)」を参照してください。

**バックアップの暗号化**を実行すると、V\$RMAN\_ENCRYPTION\_ALGORITHMS に表示されているアルゴリズムのいずれかを使用して、バックアップ・セットが暗号化されます。Recovery Manager には、透過モード、パスワード保護モードおよびデュアル・モードの 3 つの暗号化モードがある。バックアップの暗号化については、9-11 ページの「[Recovery Manager バックアップの暗号化](#)」を参照してください。バックアップの暗号化は、CPU に負荷がかかる場合があります。

## SBT への書込みフェーズ

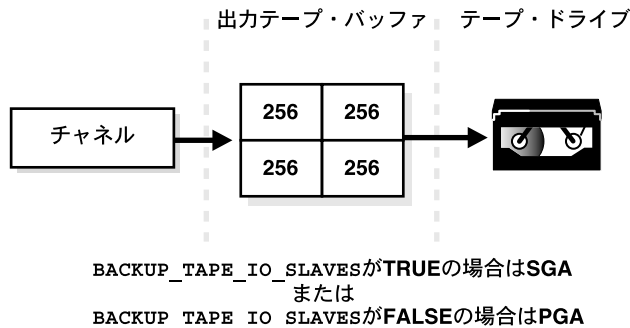
SBT へのバックアップ時、Recovery Manager は、[メディア・マネージャ](#)にバイト・ストリームを渡し、そのストリームに一意的な名前を関連付けます。ストリームの格納方法および格納場所の詳細は、メディア・マネージャによって管理されます。したがって、テープへのバックアップでは、Recovery Manager とメディア・マネージャの両方が相互に作用します。

### SBT への書込みフェーズにおける Recovery Manager の構成要素

SBT への書込みフェーズに影響する Recovery Manager 固有の要因は、ディスクの読取りに影響する要因と類似しています。いずれの場合も、バッファの割当て、スレーブ・プロセスおよび同期 I/O または非同期 I/O によってパフォーマンスが左右されます。

**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、またはそこからリストアする場合、デフォルトでは、テープ書込み用（データをリストアする場合は読取り用）の各チャネルに 4 つのバッファが割り当てられます。テープ I/O バッファのサイズは、プラットフォームによって異なります。この値は、ALLOCATE CHANNEL コマンドまたは CONFIGURE CHANNEL コマンドの PARMS パラメータおよび BLKSIZE パラメータで変更できます。

図 21-4 テープ・バッファの割当て



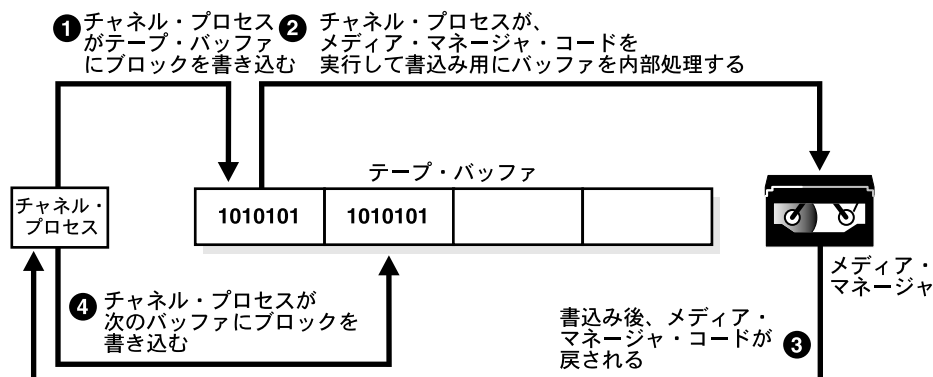
**テープ I/O スレーブ** Recovery Manager は、I/O スレーブが使用されているかどうかに応じて、SGA または PGA にテープ・バッファを割り当てます。初期化パラメータ `BACKUP_TAPE_IO_SLAVES=true` を設定した場合、テープ・バッファは SGA から割り当てられます。テープ・デバイスにアクセスできるのは、一度に 1 つのプロセスだけです。そのため、テープ・デバイス数に対応する必要な数のスレーブが起動されます。 `LARGE_POOL_SIZE` 初期化パラメータも設定されている場合は、ラージ・プールからバッファが割り当てられます。 `BACKUP_TAPE_IO_SLAVES=false` を設定した場合、バッファは PGA から割り当てられます。

I/O スレーブを使用する場合、 `LARGE_POOL_SIZE` 初期化パラメータを設定し、SGA メモリーをサイズの大きいメモリー割当ての保持専用に確保しておきます。このパラメータによって、SGA メモリーのために Recovery Manager の I/O バッファとライブラリ・キャッシュが競合することを防止できます。テープ I/O の I/O スレーブが要求されたものの、そのために十分な領域が SGA に存在しない場合、スレーブは使用されず、アラート・ログにメッセージが書き込まれます。

`BACKUP_TAPE_IO_SLAVES` には、スレーブ・プロセスの数ではなく、スレーブ・プロセスを使用するかどうかを指定することに注意してください。テープ・デバイスにアクセスできるのは、一度に 1 つのプロセスだけであり、Recovery Manager では、テープ・デバイス数に対応する必要な数のスレーブが使用されます。

**同期 I/O と非同期 I/O** SBT チャンネルによるデータの読取りまたはテープへの書込みでは、I/O は常に同期です。テープ I/O では、(手動または自動で) 割当て済の各チャンネルは 1 つのサーバー・プロセスに対応しています。ここでは、このプロセスをチャンネル・プロセスと呼びます。次の図に、テープへのバックアップでの同期 I/O を示します。

図 21-5 同期テープ I/O



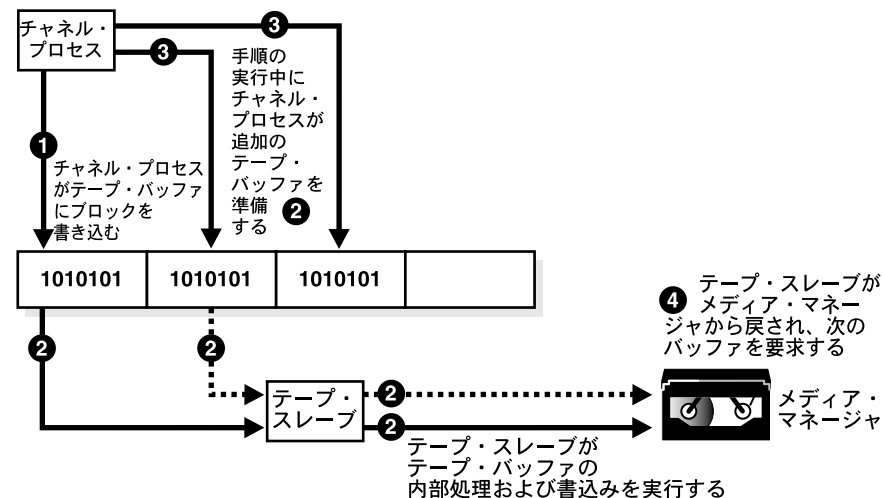


次の手順が実行されます。

1. チャンネル・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. チャンネル・プロセスが、テープ・バッファを処理するメディア・マネージャ・コードを実行して、メディア・マネージャが追加の処理および格納を行えるように内部処理を行います。
3. メディア・マネージャ・コードが、書き込みが完了したことを示すメッセージをサーバー・プロセスに戻します。
4. チャンネル・プロセスが新しいタスクを開始できる状態になります。

次の図に、テープへのバックアップでの非同期 I/O を示します。テープへの非同期 I/O は、テープ・スレーブを使用してシミュレートされます。この場合、各割当て済チャンネルは 1 つのサーバー・プロセスに対応しています。この項の説明では、サーバー・プロセスをチャンネル・プロセスと呼びます。各チャンネル・プロセスに対して 1 つ（コピーが複数の場合は複数）のテープ・スレーブが起動されます。

図 21-6 非同期テープ I/O



次の手順が実行されます。

1. チャンネル・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. チャンネル・プロセスが、テープ・バッファを処理するためのメッセージをテープ・スレーブ・プロセスに送信します。テープ・スレーブ・プロセスが、テープ・バッファを処理するメディア・マネージャ・コードを実行して、メディア・マネージャが処理を行えるように内部処理を行います。
3. テープ・スレーブ・プロセスが書き込みを行っている間、チャンネル・プロセスはデータファイルからデータを読み取り、追加の出力バッファを準備できます。
4. テープ・スレーブ・チャンネルは、メディア・マネージャ・コードから戻されると、新しいテープ・バッファを要求します。通常、その時点で新しいバッファが使用可能になっています。このため、チャンネル・プロセスの待機時間が減少し、バックアップ時間が短縮されます。

## SBT への書き込みフェーズにおけるメディア・マネージャの構成要素

テープへのバックアップの速度に影響する要因を次に示します。

- ネットワークのスループット
- 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

**ネットワークのスループット** リモートのテープ・デバイスの場合、メディア・マネージャは、ネットワーク経由でデータを転送する必要があります。たとえば、Oracle Secure Backup の管理ドメインには、ネットワークで接続された複数のクライアント・ホスト、メディア・サーバーおよびテープ・デバイスが含まれていることがあります。あるホストにデータベースがあり、別のホストに出力用のテープ・ドライブが接続されている場合、Oracle Secure Backup がネットワーク経由のデータ転送を管理します。ネットワークのスループットが、バックアップのパフォーマンスの上限になります。

**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速度です。この速度は、バックアップ・レートの上限を表します。バックアップのパフォーマンスの上限は、すべてのテープ・デバイスの転送レートの集計となります。バックアップ操作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、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、バックアップのパフォーマンスに重大な影響を及ぼします。テープの圧縮レベルが高い場合、持続的なバックアップの転送レートは高くなります。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/秒である場合、バックアップ速度は 12MB/秒になります。この場合、Recovery Manager は、12MB/秒より高速なスループットでディスクを読み取ることができる必要があります。そうでない場合はこのディスクはバックアップのボトルネックになります。

---

**注意：** メディア・マネージャによるテープの圧縮機能と Recovery Manager によるバイナリ圧縮機能の両方を使用しないでください。メディア・マネージャによる圧縮が十分である場合は、こちらを使用することをお勧めします。Recovery Manager でのデータ圧縮による CPU オーバーヘッドを許容できる場合は、Recovery Manager の圧縮バックアップ・セットを使用すると、圧縮していないバックアップ・セットをネットワークを介してメディア・マネージャに移動するときに使用される帯域幅を削減することができます。

---

**テープ・ストリーム** 書き込み操作中のテープ・ストリームは、テープへのバックアップ・パフォーマンスに重大な影響を及ぼします。現行のほぼすべての商用テープ・ドライブは、固定速度のストリーム・テープ・ドライブです。そのようなドライブはデータの書き込み速度を変更できないため、テープへ書き込むデータがなくなると、テープを減速して停止する必要があります。通常、ドライブのバッファが空になっても、テープの移動が速すぎて書き込み終了位置を越えてしまいます。そのためドライブで書き込みを継続するには、書き込みが終了した位置までテープを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、バックアップのパフォーマンスに影響する場合があります。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の量です。通常、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・サイズは、Recovery Manager または Oracle Database サーバーではなく、メディア管理ソフトウェアが制御します。詳細は、ご使用の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。

## ディスクへの書き込みフェーズ

ディスクへの書き込みフェーズに影響する主な要因は、バッファ・サイズです。バックアップの出力がディスクに存在する場合、各チャンネルは、1MB ずつの出力バッファを 4 つ割り当てます。ディスク・チャンネルが、ディスク・サブシステムにブロックを書き込みます。ファイルをリストアするときの読取りフェーズは、ファイルをバックアップするときの書き込みフェーズとほぼ同じです。ただし、ブロックの動く向きは逆です。

ディスクから非同期の読取りが行われる場合は、ディスクへの書き込みも非同期になります。ディスクへの書き込み時、読取り時と同様にディスク I/O スレーブを利用できます。

Recovery Manager が、複数のディスクにストライプ化されたディスクベースの出力先にファイルをバックアップする場合は、複数のチャンネルを割り当てることができます。チャンネルの数は、出力先のストライプ化されたディスクの数に制限されます。ASM は、複数のディスクにストライプ化された出力先の一例です。

## Recovery Manager のパフォーマンスの問題を診断するための V\$ ビューの使用法

通常、チューニング・プロセスを開始する場合は、v\$ ビューを使用して、Recovery Manager のバックアップ操作およびリストア操作のどこで問題が発生しているかを特定します。

### V\$SESSION\_LONGOPS による Recovery Manager ジョブの進捗状況の監視

V\$SESSION\_LONGOPS ビューを問い合わせると、バックアップ・ジョブおよびリストア・ジョブの進捗状況を監視できます。Recovery Manager は、V\$SESSION\_LONGOPS で詳細行と集計行の 2 つのタイプの行を使用します。

詳細行には、1 つのジョブ手順によって処理されているファイルの説明が表示され、集計行には、Recovery Manager コマンドのすべてのジョブ手順によって処理されたファイルの説明が表示されます。ジョブ手順とは、1 つのバックアップ・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、バックアップ手順中、バッファに対する読取りまたは書き込みが行われるたびに更新されるため、更新の粒度は小さくなります。集計行は、各ジョブ手順の完了時に更新されるため、更新の粒度は大きくなります。

表 21-2 に、Recovery Manager に最も関連する V\$SESSION\_LONGOPS の列を示します。通常、各バックアップ・セットの進捗状況を確認するには、集計行ではなく詳細行を表示します。

表 21-2 Recovery Manager に関連する V\$SESSION\_LONGOPS 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更新されます。  |
| OPNAME  | 行の説明テキスト。詳細行の例は、RMAN: datafile copy、RMAN: full datafile backup、RMAN: full datafile restore などです。<br><b>注意:</b> 集計行は、RMAN: aggregate input および RMAN: aggregate output のみです。 |
| CONTEXT | バックアップ出力行では、この値は 2 です。それ以外のすべての行では、値は 1 です (ただし、この列を更新しないプロキシ・コピーは除く)。   |

表 21-2 Recovery Manager に関連する V\$SESSION\_LONGOPS の列 (続き)

| 列         | 詳細行の説明   |
|-----------|--|
| SOFAR     | この列の内容は、行に表示される操作のタイプによって次のように異なります。<br><ul style="list-style-type: none"> <li>■ イメージ・コピーの場合、読み取られたブロック数。</li> <li>■ バックアップ入力行の場合、バックアップされるファイルから読み取られたブロック数。</li> <li>■ バックアップ出力行の場合、バックアップ・ピースに書き込まれたブロック数。</li> <li>■ リストアの場合、この1つのジョブ手順でリストアされているファイルに処理されたブロック数。</li> <li>■ プロキシ・コピーの場合、コピーされたファイル数。</li> </ul>   |
| TOTALWORK | この列の内容は、行に表示される操作のタイプによって次のように異なります。<br><ul style="list-style-type: none"> <li>■ イメージ・コピーの場合、ファイルのブロックの合計数。</li> <li>■ バックアップ入力行の場合、このジョブ手順で処理されるすべてのファイルから読み取られるブロックの合計数。</li> <li>■ バックアップ出力行の場合、Recovery Manager はバックアップ・ピースに書き込むブロック数を認識しないため、値は0になります。</li> <li>■ リストアの場合、このジョブ手順でリストアされるすべてのファイルのブロックの合計数。</li> <li>■ プロキシ・コピーの場合、このジョブ手順でコピーされるファイルの合計数。</li> </ul> |

バックアップ・ジョブまたはリストア・ジョブを実行する各サーバー・セッションは、1つのジョブ手順に必要な処理の合計と比較して、進捗状況をレポートします。たとえば、2つのチャネルを使用してデータベースをリストアし、各チャネルが2つのバックアップ・セットをリストアする場合 (合計4セットをリストアする場合)、各サーバー・セッションは、1つのバックアップ・セットにおける進捗状況をレポートします。あるセットのリストアが完了すると、Recovery Manager は、リストアする次のセットでの進捗状況のレポートを開始します。

### Recovery Manager ジョブの進捗状況の監視手順

1. Recovery Manager ジョブを開始する前に、次の SQL 文を含むスクリプト・ファイル (この例では longops) を作成します。

```
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;
```

2. Recovery Manager を起動し、ターゲット・データベースおよびリカバリ・カタログ (使用している場合) に接続します。
3. Recovery Manager ジョブを起動します。たとえば、次のように入力します。

```
RMAN> RESTORE DATABASE;
```

4. Recovery Manager ジョブの実行中に、SQL\*Plus を起動し、ターゲット・データベースに接続して longops スクリプトを実行し、Recovery Manager ジョブの進捗状況を確認します。Recovery Manager ジョブの実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
```

5. 2分以上の間隔で longops スクリプトを実行しているときに、%\_COMPLETE 列の値が増加しない場合は、Recovery Manager で問題が発生しています。詳細は、22-8 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。

長時間の作業の実行を頻繁に監視する場合、SQL\*Plus を実行するホスト・オペレーティング・システムでシェル・スクリプトまたはバッチ・ファイルを作成して、この問合せを繰り返し実行できます。

## V\$BACKUP\_SYNC\_IO および V\$BACKUP\_ASYNC\_IO によるボトルネックの特定

バックアップまたはリストアのボトルネックの原因を特定し、バックアップ・ジョブの詳細な進捗を確認するには、V\$BACKUP\_SYNC\_IO ビューおよび V\$BACKUP\_ASYNC\_IO ビューを使用できます。

バックアップを実行中のプロセス（一部のプラットフォームではスレッド）に I/O が同期している場合は、

V\$BACKUP\_SYNC\_IO に行が表示されます。I/O が非同期の場合は、V\$BACKUP\_ASYNC\_IO に行が表示されます。非同期 I/O は、I/O プロセスによって、または基礎となるオペレーティング・システムがサポートしている場合に実行されます。

データベース・インスタンスが停止するまで、バックアップ・ジョブまたはリストア・ジョブの結果はメモリーに残ります。このため、ジョブの完了後にビューを問い合わせることができます。

### I/O が同期している場合に、テープがストリーム化されているかどうかを確認する手順

1. SQL\*Plus を起動し、ターゲット・データベースに接続します。
2. V\$BACKUP\_SYNC\_IO ビューまたは V\$BACKUP\_ASYNC\_IO ビューの EFFECTIVE\_BYTES\_PER\_SECOND 列を問い合わせます。

EFFECTIVE\_BYTES\_PER\_SECOND がハードウェアのロー・キャパシティ未満の場合、そのテープはストリーム化されていません。EFFECTIVE\_BYTES\_PER\_SECOND がハードウェアのロー・キャパシティより大きい場合、そのテープはストリーム化されている場合とされていない場合があります。これは、圧縮によって、EFFECTIVE\_BYTES\_PER\_SECOND が実際の I/O 速度より高速になる場合があるためです。

**参照：** これらのビューの詳細は、『Oracle Database リファレンス』を参照してください。

### 同期 I/O のボトルネックの特定

同期 I/O では、すべての同期 I/O 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唯一の方法は、I/O レート (バイト / 秒) をデバイスの最大スループット・レートと比較することです。I/O レートが、デバイスで指定されているレートより低い場合、バックアップ・プロセスおよびリストア・プロセスのこの側面をチューニングすることを検討します。

#### 同期 I/O のレートを確認する手順

1. SQL\*Plus を起動し、ターゲット・データベースに接続します。
2. V\$BACKUP\_SYNC\_IO ビューの DISCRETE\_BYTES\_PER\_SECOND 列を問い合せて、I/O レートを表示します。

V\$BACKUP\_SYNC\_IO にデータが表示されている場合は、非同期 I/O を有効にしているか、またはディスク I/O スレーブを使用していないことが問題となっています。

### 非同期 I/O のボトルネックの特定

**ロング・ウェイト**は、バックアップ・プロセスまたはリストア・プロセスが、I/O が完了するまで待機するようにオペレーティング・システムに通知した回数です。**ショート・ウェイト**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**は、I/O が使用可能な状態になっており、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。

#### 非同期 I/O のレートを確認する手順

1. SQL\*Plus を起動し、ターゲット・データベースに接続します。
2. V\$BACKUP\_SYNC\_IO ビューの LONG\_WAITS 列および IO\_COUNT 列を問い合せて、I/O レートを表示します。

ボトルネックを特定する最も簡単な方法は、LONG\_WAITS を IO\_COUNT で割った比率が最も大きいデータファイルを見つける方法です。たとえば、次の問合せを使用できます。

```
SELECT   LONG_WAITS/IO_COUNT, FILENAME
FROM     V$BACKUP_ASYNC_IO
WHERE    LONG_WAITS/IO_COUNT > 0
ORDER BY LONG_WAITS/IO_COUNT DESC;
```

---

**注意：** 同期 I/O の使用時に BACKUP\_DISK\_IO\_SLAVES を設定した場合、I/O は V\$BACKUP\_ASYNC\_IO に表示されます。

---

**参照：** V\$BACKUP\_SYNC\_IO ビューおよび V\$BACKUP\_ASYNC\_IO ビューの詳細は、『Oracle Database リファレンス』を参照してください。

## Recovery Manager のバックアップのパフォーマンスのチューニング

バックアップのパフォーマンスは、多くの要因に影響を受けます。多くの場合、低速なバックアップを改善するには、試行錯誤が伴います。この項では、バックアップで最高のパフォーマンスを得るための手順を順に示します。

この項の内容は、次のとおりです。

- [手順 1: チャネル設定からの RATE パラメータの削除](#)
- [手順 2: 同期ディスク I/O を使用する場合の DBWR\\_IO\\_SLAVES の設定](#)
- [手順 3: 共有メモリーの割当てに失敗する場合の LARGE\\_POOL\\_SIZE の設定](#)
- [手順 4: 読取り、書き込みおよびコピーのフェーズのチューニング](#)

### 手順 1: チャネル設定からの RATE パラメータの削除

21-7 ページの「[チャネルの RATE パラメータ](#)」で説明されているように、チャネルの RATE パラメータは、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・スループットを（増加ではなく）削減するためのパラメータです。バックアップがテープにストリーム配信されていない場合、RATE パラメータを設定していないことを確認してください。

#### RATE パラメータを削除する手順

1. バックアップ・スクリプトを確認します。
2. 次のいずれかの処理を実行します。
  - バックアップが RUN コマンド内に指定されていて、ALLOCATE コマンドに RATE パラメータが指定されている場合は、このパラメータを削除します。残りの手順はスキップします。
  - バックアップが RUN コマンド内に指定されていない場合は、Recovery Manager を起動し、ターゲット・データベースに接続して次の手順に進みます。
3. SHOW ALL コマンドを実行して、現在構成されている設定を表示します。
4. CONFIGURE CHANNEL コマンドに RATE パラメータが設定されている場合は削除します。

### 手順 2: 同期ディスク I/O を使用する場合の DBWR\_IO\_SLAVES の設定

21-6 ページの「[同期ディスク I/O と非同期ディスク I/O](#)」に示されているように、一部のオペレーティング・システムでは、固有の非同期 I/O をサポートしています。ご使用のディスクが非同期 I/O をサポートしない場合にのみ、DBWR\_IO\_SLAVES を設定します。

DBWR\_IO\_SLAVES に 0（ゼロ）以外の値を設定すると、バックアップおよびリストアに固定数のディスク I/O スレーブが使用され、非同期 I/O がシミュレートされます。

#### ディスク I/O スレーブを有効にする手順

1. SQL\*Plus を起動し、ターゲット・データベースに接続します。
2. データベースを停止します。
3. DBWR\_IO\_SLAVES 初期化パラメータを 0（ゼロ）以外の値に設定します。
 

DBWR\_IO\_SLAVES を設定すると、データベース・ライター・プロセスがスレーブを使用します。したがって、PROCESSES 初期化パラメータの値を増やす必要がある場合があります。
4. データベースを再起動します。
5. Recovery Manager バックアップを再開します。

### 手順 3: 共有メモリの割当てに失敗する場合の LARGE\_POOL\_SIZE の設定

LARGE\_POOL\_SIZE 初期化パラメータは、メモリー不足のために I/O スレーブを起動しないというエラーがアラート・ログに書き込まれた場合に設定します。メッセージの例を次に示します。

```
ksfqxcrc: failure to allocate shared memory means sync I/O will be used whenever
async I/O to file not supported natively
```

ラージ・プールは、Recovery Manager およびその他の用途に使用されるため、すべてのユーザーに対応できる合計サイズにする必要があります。このことは特に、DBWR\_IO\_SLAVES が設定されており、DBWR プロセスにバッファが必要な場合に該当します。

#### ラージ・プールのサイズを設定する手順

1. SQL\*Plus を起動し、ターゲット・データベースに接続します。
2. 必要に応じて、V\$SGASTAT.POOL を問い合せて、特定のオブジェクト用のメモリーが存在するプール（共有プールまたはラージ・プール）を確認します。
3. ターゲット・データベースに LARGE\_POOL\_SIZE 初期化パラメータを設定します。

ALTER SYSTEM SET 文を実行して、パラメータを動的に設定します。  
LARGE\_POOL\_SIZE を設定するための計算式を次に示します。

$$\text{LARGE\_POOL\_SIZE} = \text{number\_of\_allocated\_channels} * (16 \text{ MB} + (4 * \text{size\_of\_tape\_buffer}))$$

4. Recovery Manager バックアップを再開します。

**参照：** ラージ・プールの詳細は『Oracle Database 概要』、初期化パラメータの詳細は『Oracle Database リファレンス』を参照してください。

### 手順 4: 読取り、書込みおよびコピーのフェーズのチューニング

次のいくつかのタスクを実行して、バックアップのパフォーマンスに影響を及ぼすボトルネックを特定および解消できます。

#### バックアップの検証を使用した読取りと書込みのボトルネックの識別

任意のバックアップ・ジョブで出力デバイスまたは入力ディスク I/O のどちらがボトルネックになっているかを確実に確認する方法の 1 つは、あるバックアップ・タスクの実行時間と、同じタスクの BACKUP VALIDATE の実行時間を比較することです。バックアップの BACKUP VALIDATE は、実際のバックアップと同じディスク読取りを実行しますが、出力デバイスに対する I/O は実行しません。

#### バックアップ時間と検証時間を比較する手順

1. NLS 環境の日付書式変数が、時刻を表示するように設定されていることを確認します。たとえば、NLS 変数を次のように設定します。

```
setenv NLS_LANG AMERICAN_AMERICA.WE8DEC;
setenv NLS_DATE_FORMAT "MM/DD/YYYY HH24:MI:SS"
```
2. BACKUP コマンドでなく BACKUP VALIDATE コマンドを使用するようにバックアップ・スクリプトを編集します。
3. バックアップ・スクリプトを実行します。
4. Recovery Manager 出力を調べ、Starting backup at と Finished backup at に示された時間の差を計算します。
5. BACKUP VALIDATE コマンドでなく BACKUP コマンドを使用するようにバックアップ・スクリプトを編集します。
6. バックアップ・スクリプトを実行します。



7. Recovery Manager 出力を調べ、Starting backup at と Finished backup at に示された時間の差を計算します。
8. 検証と実際のバックアップでバックアップ時間を比較します。

テープに対する BACKUP VALIDATE の時間が、テープへの実際のバックアップの時間とほとんど変わらない場合は、ディスクからの読取りがボトルネックになっている可能性があります。詳細は、21-17 ページの「[読取りフェーズのチューニング](#)」を参照してください。

テープに対する BACKUP VALIDATE の時間が、テープへの実際のバックアップの時間より大幅に短い場合は、出力デバイスへの書込みがボトルネックになっている可能性があります。詳細は、21-18 ページの「[コピーおよび書込みのフェーズのチューニング](#)」を参照してください。

## 読取りフェーズのチューニング

Recovery Manager では、出力デバイスの占有を継続するのに十分な速度で出力デバイスにデータ・ブロックを送信できないことがあります。たとえば、Recovery Manager は、[増分バックアップ](#)中、同じ計画の一環として以前にデータファイルをバックアップした後に変更されたブロックのみをバックアップします。[ブロック・チェンジ・トラッキング](#)を有効にしない場合、Recovery Manager は、変更されたブロックを検出するためにデータファイル全体をスキャンし、変更されたブロックを検出しながら出力バッファを一杯にする必要があります。変更されたブロックがほとんどなく、SBT バックアップを行っている場合、Recovery Manager は、テープ・ドライブへのストリームを継続するのに十分な速度で出力バッファを一杯にできないことがあります。

同時に読み取られ、Recovery Manager の同じバックアップ・ピースに書き込まれる入力ファイルの数を表す[多重化レベル](#)を調整すると、バックアップのパフォーマンスを向上させることができます。多重化のレベルは、チャンネル上の MAXOPENFILES 設定および各バックアップ・セット内の入力ファイル数の最小値です。次の表に、多重化のレベルを調整するときの推奨事項を示します。

表 21-3 多重化のレベルの調整

| ASM | ストライプ化されたディスク | 推奨事項  |
|-----|---------------|---|
| なし  | あり            | 多重化のレベルを上げます。MAXOPENFILES または各バックアップ・セット内のファイル数のいずれかが最小値かを確認してから、この値を増やします。<br><br>このようにして、Recovery Manager がテープ・バッファを一杯にする速度を上げ、ストリームを継続するのに十分な速度でバッファがメディア・マネージャに送信される確率を高くします。 |
| なし  | なし            | チャンネル上の MAXOPENFILES 設定値を増やします。   |
| あり  | N/A           | チャンネル上の MAXOPENFILES パラメータを 1 または 2 に設定します。   |

### 参照：

- MAXOPENFILES および FILESPERSET の設定が多重化のレベルにどのような影響を与えるかについては、7-7 ページの「[多重バックアップ・セット](#)」を参照してください。
- 概要については、7-13 ページの「[増分バックアップ](#)」を参照してください。

## コピーおよび書込みのフェーズのチューニング

読取りフェーズのパフォーマンスが良好な場合、ボトルネックとなっている可能性が高いのは、コピー・フェーズまたは書込みフェーズです。特に、Recovery Manager が、ストリームをサポートするのに十分な速度でテープ・ドライブにデータ・ブロックを送信しているときにテープがストリーム化されていない場合は、SBT への書込みフェーズがボトルネックです。次のように、パフォーマンスを向上させます。

- バックアップが**全体バックアップ**の場合、増分バックアップの使用を検討します。

増分レベル 1 のバックアップでは、変更されたブロックのみがデータファイルからテープに書き込まれるため、テープへの書込みに関するボトルネックは、バックアップ計画全体にはあまり影響を及ぼしません。特に、バックアップするデータベースのノードにテープ・ドライブがローカルに接続されていない場合、増分バックアップはより高速になる場合があります。詳細は、8-15 ページの「[増分バックアップの作成および更新](#)」を参照してください。
- デフォルトの圧縮アルゴリズムである BZIP2 をバックアップで使用している場合は、圧縮アルゴリズムを BZIP2 から ZLIB に変更します。

ZLIB は、BZIP2 ほど CPU に負荷がかかりません。詳細は、6-7 ページの「[バックアップ圧縮アルゴリズムの構成](#)」を参照してください。
- データベースのホストが複数の CPU を使用し、バックアップではバイナリ圧縮が使用されている場合は、チャンネルの数を増やします。
- バックアップが暗号化されている場合は、暗号化アルゴリズムを AES128 に変更します。

AES128 アルゴリズムは、最も CPU に負荷がかからない処理です。詳細は、6-10 ページの「[バックアップ暗号化アルゴリズムの構成](#)」を参照してください。
- Recovery Manager がテープにバックアップしている場合は、次のような調整を試してください。
  - テープ I/O バッファのサイズを調整します。

サイズを設定するには、ALLOCATE CHANNEL コマンドまたは CONFIGURE CHANNEL コマンドの PARMs パラメータおよび BLKSIZE パラメータを使用します。テープ I/O バッファのサイズは、プラットフォームによって異なります。BLKSIZE 設定は、デフォルトより優先されます。
  - メディア管理ソフトウェアの設定を調整します。

テープ・ブロック・サイズを含め、メディア・マネージャの設定の数は、バックアップのパフォーマンスに影響する場合があります。
- Recovery Manager が ASM にファイルをバックアップしている場合は、チャンネルの数を増やします。

たとえば、Recovery Manager が、16 個の物理ディスクを含む 1 つのディスク・グループにデータベースをバックアップしている場合は、ディスク・チャンネルの数を 4 ~ 16 の範囲で割り当てるか構成します。

---

## Recovery Manager 操作の トラブルシューティング

この章では、Recovery Manager をトラブルシューティングする方法について説明します。  
この章では、次の項目について説明します。

- Recovery Manager のメッセージ出力の解釈
- Recovery Manager のトラブルシューティングでの V\$ ビューの使用
- Media Management API のテスト
- Recovery Manager コマンドの終了

## Recovery Manager のメッセージ出力の解釈

Recovery Manager は、問題のトラブルシューティングに役立つ詳細なエラー・メッセージを提供します。また、Oracle Database サーバーおよびサード・パーティのメディア・ベンダーは、独自の有効なデバッグ出力を生成します。この項では、発生する可能性がある様々なエラーの識別方法および解釈について説明します。

### メッセージ出力タイプの識別

障害が発生したか、ハングアップした Recovery Manager ジョブのトラブルシューティングに役立つ出力は、次の表に示すとおり、様々な場所に表示または格納されます。

表 22-1 メッセージ出力タイプ

| 出力タイプ                  | 作成元                   | 場所  | 説明  |
|------------------------|-----------------------|---|---|
| Recovery Manager メッセージ | Recovery Manager      | ジョブの詳細情報は、V\$RMAN_STATUS および RC_RMAN_STATUS に表示されます。現行のジョブの情報は、V\$RMAN_OUTPUT に表示されます。<br><br>Recovery Manager をコマンドラインから実行すると、出力を次の場所へ送ることができます。 <ul style="list-style-type: none"> <li>標準出力</li> <li>コマンドラインの LOG または SPOOL LOG コマンドで指定したログ・ファイル</li> <li>Recovery Manager 出力をリダイレクトすることで作成したファイル (UNIX の &gt; 演算子などによる)</li> </ul> | Recovery Manager ジョブに関連するアクション、および Recovery Manager、データベース・サーバー、メディア・ベンダーによって生成されたエラー・メッセージが含まれています。Recovery Manager のエラー・メッセージには、RMAN-xxxx という接頭辞が付いています。通常のアクションの説明には接頭辞は付きません。<br><br>次の PL/SQL を実行すると、V\$RMAN_STATUS からすべてのエントリを削除できることに注意してください。<br><br>SYS.DBMS_BACKUP_RESTORE.resetCfileSection(28);<br><br>このファンクションでは、すべてのジョブ関連エントリが削除されます。新しいバックアップ・ジョブが V\$RMAN_BACKUP_JOB_DETAILS に表示されるまで、行は表示されません。 |
| alert_SID.log          | Oracle Database       | 自動診断リポジトリ・ホームの alert サブディレクトリ   | エラー、初期化パラメータ設定および管理操作の時系列の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ます。  |
| Oracle トレース・ファイル       | Oracle Database       | ADR ホームの trace サブディレクトリ   | Oracle サーバー・プロセスによって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-600 または ORA-3113 エラー・メッセージが発生したとき、Recovery Manager がチャネルを割り当てられないとき、およびデータベースがメディア管理ライブラリのロードに失敗したときに作成されます。   |
| sbtio.log              | サード・パーティのメディア管理ソフトウェア | ADR ホームの trace サブディレクトリ   | メディア管理ソフトウェアによって生成されたベンダー固有の情報が含まれています。このログには、Oracle サーバーまたは Recovery Manager のエラーは含まれていません。  |
| 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 | sbtio.log 以外のすべてのメディア・マネージャのログのファイル名は、メディア管理ソフトウェアによって決定されます。   | 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・スタックの識別

Recovery Manager は、エラーの発生時にそれらのエラーを通知します。回復不可能なエラーの場合、Recovery Manager は別のチャネルへのフェイルオーバーを実行して特定のジョブ手順を完了することができず、すべてのジョブ・セットの完了後にエラーの概要レポートを出力します。この機能は、遅延エラー・レポートともいいます。

22-7 ページの「[Recovery Manager のリターン・コードの識別](#)」で説明するように、Recovery Manager にエラーが発生したかどうかを確認する方法の1つは、リターン・コードを調べることです。2つ目の方法は、Recovery Manager の出力内で RMAN-00569 文字列を検索することです。RMAN-00569 は、エラー・スタック・バナーのメッセージ番号です。すべての Recovery Manager エラーの前に、このエラー・メッセージが表示されます。出力内に RMAN-00569 メッセージが表示されない場合、エラーはありません。次に、構文エラーの例を示します。

#### 例 22-1 Recovery Manager 構文エラー

```
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-00558: error encountered while parsing input commands
RMAN-01005: syntax error: found ")": expecting one of: "archivelog, backup, backupset,
controlfilecopy, current, database, datafile, datafilecopy, (, plus, ;, tablespace"
RMAN-01007: at line 1 column 18 file: standard input
```

## エラー・コードの識別

通常、Recovery Manager メッセージ・スタックには次のタイプのエラー・コードが含まれています。

- 接頭辞が RMAN- のエラー
- 接頭辞が ORA- のエラー
- Additional information: の後に表示されるエラー

**参照:** RMAN および ORA のエラー・コードの詳細は、『Oracle Database 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 22-2 に、一般的な Recovery Manager エラー・メッセージのエラー範囲を示します。すべてのメッセージの詳細は、『Oracle Database エラー・メッセージ』を参照してください。

表 22-2 Recovery Manager のエラー・メッセージの範囲

| エラー範囲       | 原因                                   |
|-------------|--------------------------------------|
| 0550-0999   | コマンドライン・インタプリタ                       |
| 1000-1999   | キーワード・アナライザ                          |
| 2000-2999   | 構文アナライザ                              |
| 3000-3999   | 主レイヤー                                |
| 4000-4999   | サービス・レイヤー                            |
| 5000-5499   | RESTORE コマンドまたは 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 間のフェーズ間エラー |
| 12000-12999 | リカバリ・カタログ・パッケージ                      |

## ORA-19511: メディア・マネージャ・エラー

メディア・マネージャ・エラーが発生した場合、ORA-19511 が表示され、説明を含むエラー・メッセージがメディア・マネージャから Recovery Manager に戻されます。Recovery Manager は、メディア・マネージャから戻されたエラーを表示します。たとえば、次のようなエラーが表示されます。

```
ORA-19511: Error received from media manager layer, error text:
  sbtpvt_open_input: file .* does not exist or cannot be accessed, ermo = 2
```

メディア・マネージャからのメッセージには、根本的な問題を修正するために十分な情報が含まれています。十分でない場合、ご使用のメディア・マネージャのドキュメントを参照するか、またはメディア管理ベンダーのサポート担当者に詳細を問い合せてください。ORA-19511 エラーは、Oracle Database ではなくメディア・マネージャによって生成されます。データベースは、メディア・マネージャからのメッセージを渡すのみです。原因を解決できるのは、メディア管理ベンダーのみです。

**SBT 1.1** 対応のメディア管理レイヤーを使用している場合、その他のエラー・メッセージ・テキストが表示される場合があります。SBT 1.1 対応のメディア管理レイヤーからの出力は、次のものと類似しています。

```
ORA-19507: failed to retrieve sequential file, handle="c-140148591-20031014-06",
parms=""
ORA-27007: failed to open file
Additional information: 7000
Additional information: 2
ORA-19511: Error received from media manager layer, error text:
  SBT error = 7000, ermo = 0, sbtopen: backup file not found
```

「Additional information」には、SBT 1.1 に固有なエラー・コードが表示されます。表示される値は、表 22-3 に示すメディア・マネージャ・メッセージ番号およびエラー・テキストに対応しています。Recovery Manager は、「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。」を再度表示し、メディア・マネージャから戻されたエラー・コードに関連する一般的なエラー・メッセージおよび SBT 1.1 エラー番号を表示します。

参考として、SBT 1.1 エラー・メッセージを示します。表 22-3 に、メディア・マネージャ・メッセージ番号および各番号に対応するエラー・テキストを示します。エラー・コード内の O/S は、オペレーティング・システムを意味します。アスタリスクが付いているエラーは内部エラーであり、通常の操作中に一般的に表示されるものではありません。

**表 22-3 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    | メッセージ                                |
|---------|-------|--------------------------------------|
| sbtopen | 7000  | バックアップ・ファイルが見つかりません。(読取りの場合のみ戻されます。) |
|         | 7001  |                                      |
|         | 7002* | バックアップ・ファイルは存在します。(書込みの場合のみ戻されます。)   |
|         | 7003  | 不正なモードが指定されました。                      |
|         | 7004  | 指定されたブロック・サイズが無効です。                  |
|         | 7005  | デバイスがありません。                          |
|         | 7006  | 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。      |
|         | 7007  | ボリュームが見つかりません。                       |
|         | 7008  | ボリュームは使用中です。                         |
|         | 7009  | I/O エラーです。                           |
|         | 7010  | Media Manager と接続できません。              |
|         | 7011  | 許可されません。                             |
|         | 7012* | システム・エラー - 例 : malloc, fork のエラー     |
|         |       | 引数が無効です。                             |

表 22-3 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    | メッセージ                         |
|-----------|-------|-------------------------------|
| sbtclose  | 7020* | ファイル・ハンドルが無効か、ファイルをオープンできません。 |
|           | 7021* | フラグが無効です。                     |
|           | 7022  | I/O エラーです。                    |
|           | 7023  | O/S エラー。                      |
|           | 7024* | 引数が無効です。                      |
|           | 7025  | Media Manager と接続できません。       |
| sbtwrite  | 7040* | ファイル・ハンドルが無効か、ファイルをオープンできません。 |
|           | 7041  | ボリュームの終わりに達しました。              |
|           | 7042  | I/O エラーです。                    |
|           | 7043  | O/S エラー。                      |
|           | 7044* | 引数が無効です。                      |
| sbtread   | 7060* | ファイル・ハンドルが無効か、ファイルをオープンできません。 |
|           | 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 エラーです。                    |
|           | 7064  | O/S エラー。                      |
|           | 7065* | 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 エラーです。                    |
|           | 7083  | Media Manager と接続できません。       |
|           | 7084  | 許可されません。                      |
|           | 7085  | O/S エラー。                      |
|           | 7086* | 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 エラーです。                    |
|           | 7092  | Media Manager と接続できません。       |
|           | 7093  | 許可されません。                      |
|           | 7094  | O/S エラー。                      |
|           | 7095* | 引数が無効です。                      |
| sbtinit   | 7110* | 引数が無効です。                      |
|           | 7111  | O/S エラー。                      |

## Recovery Manager エラー・スタックの解釈

Sometimes Recovery Manager エラー・スタック内の役立つメッセージを特定することが困難な場合があります。次のヒントおよび推奨事項を参考にしてください。

- メッセージは下から上に読みます。これは、Recovery Manager は下から順にメッセージを発行するためです。スタックの最後から1つか2つのエラーが、通常は最も重要です。
- SBT 1.1 メディア管理レイヤーを使用しており、「Additional Information:」およびエラー・コード (番号) を含む SBT 1.1 形式のエラー・メッセージが表示された場合、その後続く ORA-19511 メッセージの、メディア・マネージャによって Recovery Manager に戻されたエラー・メッセージのテキストを確認します。これらのメッセージには、メディア管理レイヤーでの実際の障害が示されます。

- エラー・バナーの直後にある RMAN-03002 メッセージまたは RMAN-03009 メッセージを確認します (RMAN-03009 は RMAN-03002 と同じですが、RMAN-03009 にはチャンネル ID が含まれます)。これらのメッセージには、正常に実行されなかったコマンドが示されます。構文エラーの場合、RMAN-00558 が生成されます。
- 表 22-2 に示すエラー範囲を参照して基本的なエラー・タイプを判断し、最も重要なメッセージについて、『Oracle Database エラー・メッセージ』を参照して確認します。

## Recovery Manager エラーの解釈の例

users 表領域のバックアップを試行し、次のメッセージが戻されたと想定します。

```
Starting backup at 29-AUG-02
using channel ORA_DISK_1
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of backup command at 08/29/2002 15:14:03
RMAN-20202: tablespace not found in the recovery catalog
RMAN-06019: could not translate tablespace name "USESR"
```

RMAN-03002 エラーは、BACKUP コマンドが正常に実行されなかったことを示しています。スタックの最後の 2 つのメッセージを参照すると、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 user という名前の表領域が見つからないことがわかります。

## サーバー・エラーの解釈の例

表領域のリカバリを試行し、次のエラーが戻されたと想定します。

```
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: failure of recover command at 08/29/2007 15:18:43
RMAN-11003: failure during parse/execution of SQL statement: alter database recover if
needed tablespace USERS
ORA-00283: recovery session canceled due to errors
ORA-01124: cannot recover data file 8 - file is in use or recovery
ORA-01110: data file 8: '/oracle/oradata/trgt/users01.dbf'
```

前述の推奨事項に従って、スタックの下から読み始めます。ORA-01110 メッセージは、users01.dbf データファイルのリカバリで問題が発生したことを示しています。下から 2 つ目のメッセージは、そのデータファイルが使用中かリカバリ中であるため、リカバリ不可能であることを示しています。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ています。このデータファイルはリカバリ中ではないため、問題はデータファイルがオンラインであることであり、このファイルをオフラインにしてバックアップをリストアする必要があると判断できます。

## SBT 2.0 のメディア管理エラーの解釈の例

テープ・ドライブを使用したバックアップ・ジョブ中に、次の出力が戻されたと想定します。

```
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
ORA-19624: operation failed, retry possible
ORA-19507: failed to retrieve sequential file, handle="/tmp/mydir", parms=""
ORA-27029: skgfrtrv: sbtrestore returned error
```



```
ORA-19511: Error received from media manager layer, error text:
  sbtpvt_open_input:file /tmp/mydir does not exist or cannot be accessed, errno=2
```

ORA-19511 エラーの後に表示されるエラー・テキストはメディア・マネージャによって生成されたものであり、障害の根源を示しています。このエラーを解釈する方法は、メディア・マネージャのドキュメントを参照してください。

## SBT 1.1 のメディア管理エラーの解釈の例

テープ・ドライブを使用したバックアップ・ジョブ中に、次の出力が戻されると想定します。

```

RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03009: failure of backup command on c1 channel at 09/04/2007 13:18:19
ORA-19506: failed to create sequential file, name="07d36ecp_1_1", parms=""
ORA-27007: failed to open file
SVR4 Error: 2: No such file or directory
Additional information: 7005
Additional information: 1
ORA-19511: Error received from media manager layer, error text:
  SBT error = 7005, errno = 2, sbtopen: system error

```

SBT 1.1 メディア・マネージャによって戻される最も重要な情報は、「Additional information」行にあるエラー・コードです。

Additional information: 7005

表 22-3 「メディア・マネージャのエラー・メッセージの範囲」を参照すると、7005 エラーが、メディア管理デバイスがビジー状態であることを示していることがわかります。メディア管理ソフトウェアが使用中か、またはそのソフトウェアに問題が発生しているため、そのソフトウェアを使用してデバイスに書き込むことができません。

---

**注意：** `sbtio.log` には、Oracle Database ではなくメディア管理ソフトウェアによって生成された情報が含まれています。そのため、それらのエラー・コードおよびメッセージを解釈するには、ご使用のメディア・ベンダーのドキュメントを参照する必要があります。`sbtio.log` に情報が記述されない場合、別の場所にエラー・メッセージが記述されているかどうか、またはメディア・マネージャ・エラーを `sbtio.log` に表示するために必要な手順があるかどうかを、ご使用のメディア・マネージャのサポートに問い合せてください。

---

## 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確認する方法の 1 つは、リターン・コードまたは終了ステータスを調べることです。Recovery Manager クライアントは、そのクライアントが起動されたシェルに、エラーが発生しなかった場合は 0 (ゼロ) を返し、エラーが発生した場合は 0 (ゼロ) 以外のエラー値を返します。

このリターン・コードへのアクセス方法は、Recovery Manager クライアントを起動した環境によって異なります。たとえば、UNIX および C シェルを実行している場合、Recovery Manager が完了すると、`$status` というシェル変数にリターン・コードが配置されます。終了ステータスを戻す方法は、Recovery Manager クライアントではなくホスト・オペレーティング・システムの詳細によって異なります。

## Recovery Manager のトラブルシューティングでの V\$ ビューの使用

LIST、REPORT および SHOW を使用しても、Recovery Manager アクティビティに必要なすべての情報が表示されない場合は、多数の V\$ ビューで詳細を参照できます。

バックアップおよびリカバリ・ジョブを実行するサーバー・セッションで現在実行されている動作を確認すると有効な場合があります。表 22-4 に示すビューは、Recovery Manager ジョブに関する情報を取得する場合に有効です。

**表 22-4** トラブルシューティングで有効な V\$ ビュー

| 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\$SESSION      | 現在アクティブなセッションを識別します。このビューを使用して、Recovery Manager が割り当てたチャンネルに対応するデータベース・サーバー・セッションを判断します。 |
| V\$SESSION_WAIT | セッションが待機中のイベントまたはリソースのリストを表示します。   |

前述のビューを使用して、次のタスクを実行できます。

- Recovery Manager とメディア・マネージャの相互作用の監視
- サーバー・セッションと Recovery Manager チャンネルの関連付け

### 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・ビューのイベント名を使用して、Media Management API に対する Recovery Manager コールを監視できます。イベント名は、SBT ファンクションと 1 対 1 で対応します。次に例を示します。

```
Backup: sbtinit
Backup: ssbtopen
Backup: ssbtread
Backup: ssbtwrite
Backup: ssbtbackup
```

```
.
```

SBT イベントの完全なリストを取得するには、次の問合せを使用します。

```
SELECT NAME
FROM V$EVENT_NAME
WHERE NAME LIKE '%sbt%';
```

サーバーは、Media Management API でいずれかのファンクションをコールする前に、V\$SESSION\_WAIT に行を追加して、STATE 列に文字列 WAITING を含めます。V\$SESSION\_WAIT.SECONDS\_IN\_WAIT 列には、サーバーが、このコールが戻されるのを待機している秒数が表示されます。SBT ファンクションがメディア・マネージャから戻されると、この行は削除されます。

SBT イベント名に対応する V\$SESSION\_WAIT の行には、問題は表示されません。これは、サーバーがこれらの行を実行時に更新するためです。これらの行は、コールが実行されると表示され、戻されると削除されます。ただし、SECONDS\_IN\_WAIT 列の値が高い場合、メディア・マネージャがハングアップ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,
       sw.STATE, CLIENT_INFO
FROM   V$SESSION_WAIT sw, V$SESSION s, V$PROCESS p
WHERE  sw.EVENT LIKE 's%bt%'
AND    s.SID=sw.SID
AND    s.PADDR=p.ADDR;

```

SQL 出力を調べて、待機中の SBT 機能を確認します。たとえば、次の出力には、Recovery Manager が sbtbackup ファンクションが戻されるのを 10 分間待機していることが示されています。

| SPID | EVENT             | SEC_WAIT | STATE   | CLIENT_INFO                 |
|------|-------------------|----------|---------|-----------------------------|
| 8642 | Backup: sbtbackup | 600      | WAITING | rman channel=ORA_SBT_TAPE_1 |

**注意:** V\$SESSION\_WAIT ビューにはデータベース・イベントのみが表示され、メディア・マネージャ・イベントは表示されません。

**参照:** V\$SESSION\_WAIT の詳細は、『Oracle Database リファレンス』を参照してください。

## サーバー・セッションと Recovery Manager チャンネルの関連付け

どのサーバー・セッションがどの Recovery Manager チャンネルに対応しているかを確認するには、V\$SESSION および V\$PROCESS を問い合わせます。V\$PROCESS の SPID 列に、オペレーティング・システムのプロセスまたはスレッドの ID 番号が示されます。たとえば、UNIX では SPID 列にプロセス ID が表示され、Windows 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、複数の Recovery Manager セッションが同時にアクティブになっているかどうかに応じて、2 つの基本的な方法があります。

### アクティブな Recovery Manager セッションが 1 つの場合のサーバー・セッションとチャンネルの一致

アクティブな Recovery Manager セッションが 1 つのみの場合、Recovery Manager チャンネルに対応するサーバー・セッション ID を確認する最も簡単な方法は、Recovery Manager ジョブの実行中、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%';

```

出力例を次に示します。

| SID | SPID | CLIENT_INFO                 |
|-----|------|-----------------------------|
| 14  | 8374 | rman channel=ORA_SBT_TAPE_1 |

システム生成のデフォルト ID ではなく、Recovery Manager の SET COMMAND ID コマンドを使用して ID を設定する場合、'rman%' ではなく、CLIENT\_INFO 列の値を検索します。

## Recovery Manager セッションが複数の場合のサーバー・セッションとチャネルの一致

アクティブな Recovery Manager セッションが複数の場合、V\$SESSION.CLIENT\_INFO 列に、各セッションのチャネルに対して同じ情報が表示される場合があります。たとえば、次のように入力します。

```
SID SPID          CLIENT_INFO
-----
14 8374          rman channel=ORA_SBT_TAPE_1
9 8642          rman channel=ORA_SBT_TAPE_1
```

この場合、次の方法で SID 値に対応するチャネルを確認します。

**Recovery Manager 出力からのチャネル ID の取得** この方法では、まず Recovery Manager 出力から sid 値を取得して、その値を SQL 問合せで使用する必要があります。

### バックアップ中にプロセスをチャネルに関連付ける手順

1. いずれかのアクティブなセッションで、Recovery Manager ジョブを通常どおりに実行し、出力を確認してチャネルの sid を取得します。たとえば、次の出力が表示される場合があります。

```
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 ジョブの実行中に、SQL\*Plus セッションを開始して、V\$SESSION ビューと V\$PROCESS ビューを結合して問い合わせます。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確認します。

```
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関連付け** 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。これによって、この文字列の V\$SESSION.CLIENT\_INFO を問い合わせることができます。

### バックアップ中にプロセスをチャネルに関連付ける手順

1. 各セッションで、チャネルの割当て後、COMMAND ID を別々の値に設定して、目的の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 disk;
  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 ジョブの実行中に、SQL\*Plus セッションを開始して、`V$SESSION` ビューと `V$PROCESS` ビューを結合して問い合わせます。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` コマンドを実行した場合、`CLIENT_INFO` 列は次の形式で表示されます。

```
id=command_id,rman channel=channel_id
```

出力例を次に示します。

| 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` を含む行に、バックアップを実行中のチャンネルが表示されます。残りの行には、ターゲット・データベースへの接続が表示されます。

**参照：** `SET COMMAND ID` 構文については、『Oracle Database バックアップおよびリカバリ・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、『Oracle Database リファレンス』を参照してください。

## Media Management API のテスト

一部のプラットフォームでは、Oracle によって `sbttest` という診断ツールが提供されます。このユーティリティは、Oracle Database サーバーと同様にメディア・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 ユーティリティの入手

UNIX 上では、`sbttest` ユーティリティは、通常 `$ORACLE_HOME/bin` に存在します。なんらかの理由でプラットフォームにこのユーティリティが存在しない場合、Oracle サポート・サービスからこのプログラムの C バージョンを入手してください。このバージョンの `sbttest` 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` を使用する際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、再リンクが必要な場合があります。

## sbtttest ユーティリティのオンライン・ドキュメントの取得

sbtttest のオンライン・ドキュメントを入手するには、コマンドラインで次のコマンドを発行します。

```
% sbtttest
```

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: sbtttest 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]>
      <-libname sbt_library_name>
```

各引数の意味も表示されます。例として、2つのオプション・パラメータの説明を次に示します。

Optional parameters:

- dbname specifies the database name which will be used by SBT to identify the backup file. The default is "sbtldb"
- 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

sbtttest を使用すると、メディア・マネージャの簡単なテストを実行できます。

sbtttest によって 0 (ゼロ) が戻される場合、テストがエラーなしで実行されています。これは、メディア・マネージャが正しくインストールされ、データ・ストリームを受け入れ、要求に応じて同じデータを戻すことができることを意味します。sbtttest によって 0 以外の値が戻される場合、メディア・マネージャがインストールされていないか、正しく構成されていません。

### sbtttest の使用手順

1. コマンドラインで sbtttest と入力し、プログラムがインストール済でシステム・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、オンライン・ドキュメントが表示されます。

2. オンライン・ドキュメントに表示された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、some\_file.f テスト・ファイルを作成して sbtio.log に出力を生成するには、次のコマンドを入力します。

```
% sbtttest some_file.f -trace sbtio.log
```

既存のデータファイルのバックアップをテストすることもできます。たとえば、prod データベースの tbs\_33.f データファイルをテストするには、次のコマンドを入力します。

```
% sbttest tbs_33.f -dbname prod
```

- 出力を確認します。プログラムの実行中にエラーが発生した場合、障害を説明するメッセージが出力されます。たとえば、データベースがライブラリを検出できない場合、次の出力が戻されます。

```
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: sbttest: fatal: libobk.so: open failed: No such file or directory
```

sbttest を実行可能でも、Recovery Manager バックアップを実行でき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。
- データベース・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か、または必要に応じてメディア管理ライブラリを動的にロードできない。この場合、sbttest は機能しますが、メディア・マネージャに対する Recovery Manager によるバックアップは失敗します。
- sbttest プログラムはシェルからすべての環境変数を渡しているが、Recovery Manager は渡していない。

## Recovery Manager コマンドの終了

実行中の Recovery Manager コマンドは、次の複数の方法で終了できます。

- 推奨する方法は、Recovery Manager インタフェースで [CTRL] キーを押しながら [C] キー（または、ご使用のシステムのアテンション・キーの組合せ）を押すことです。この操作では、テープのマウントを待機している場合など、割当て済チャンネルがメディア管理コードでハンガアップしていないかぎり、そのチャンネルも終了します。
- SQL の ALTER SYSTEM KILL SESSION 文を実行すると、Recovery Manager チャンネルに対応するサーバー・セッションを強制終了できます。
- オペレーティング・システム上の Recovery Manager チャンネルに対応するサーバー・セッションを終了します。

## ALTER SYSTEM KILL SESSION によるセッションの終了

Recovery Manager チャンネル用の Oracle セッション ID は、Recovery Manager ログ内の、次のような書式のメッセージに示されています。

```
channel chl: sid=15 devtype=SBT_TAPE
```

割当て済チャンネルごとに、sid および devtype が表示されます。Oracle の sid はオペレーティング・システム・プロセス ID とは異なることに注意してください。セッションを強制終了するには、SQL の ALTER SYSTEM KILL SESSION 文を使用します。

ALTER SYSTEM KILL SESSION には、シリアル番号および Recovery Manager メッセージに出力された sid の 2 つの引数を指定できます。いずれの引数も、V\$SESSION を問い合わせ取得できます。たとえば、次の文を実行します。ここで、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;
```

問合せによって取得した `sid_in_rman_output` およびシリアル番号を代入して次の文を実行します。

```
ALTER SYSTEM KILL SESSION 'sid_in_rman_output,serial#';
```

メディア・マネージャ・コードでセッションがハングアップしている場合は、この文を実行してもハングアップ状態は解消しません。

## オペレーティング・システム・レベルでのセッションの終了

サーバー・セッションに関連付けられたプロセスの検出および強制終了の方法は、オペレーティング・システムによって異なります。サーバー・セッションがどのプロセスにも関連付けられていないプラットフォームもあります。詳細は、ご使用のオペレーティング・システム固有のドキュメントを参照してください。

## メディア・マネージャでハングアップした Recovery Manager セッションの終了

メディア・マネージャでハングアップした Recovery Manager ジョブを強制終了する必要がある場合があります。チャンネル接続がメディア・マネージャでハングアップした場合に Recovery Manager を終了する最も適切な方法は、メディア・マネージャ内のセッションを強制終了することです。この操作で問題が解決しない場合、UNIX などの一部のプラットフォームでは、接続を行っている Oracle プロセスを強制終了できる場合があります。(Oracle プロセスを強制終了すると、メディア・マネージャに問題が発生する可能性があることに注意してください。詳細は、ご使用のメディア・マネージャのドキュメントを参照してください。)

### Recovery Manager セッションの構成要素

Recovery Manager セッションの特性は、オペレーティング・システムに応じて異なります。UNIX では、Recovery Manager セッションには次のプロセスが関連付けられています。

- Recovery Manager クライアント・プロセス自体。
- ターゲット・データベースへの初期接続である **デフォルト・チャンネル**。
- 各割当て済チャンネルに 1 つ対応付けられた、ターゲット・データベースへの **ターゲット接続**。
- リカバリ・カタログを使用する場合、リカバリ・カタログ・データベースへの **カタログ接続**。
- DUPLICATE 操作または TSPITR 操作中は、補助インスタンスへの **補助接続**。
- 様々な割当て済チャンネルに対する Recovery Manager コマンドの実行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 は 1 つのポーリング接続を確立します。ALLOCATE CHANNEL コマンドまたは CONFIGURE CHANNEL コマンドに異なる接続文字列を使用すると、Recovery Manager は追加のポーリング接続を確立します。ポーリング接続は、ALLOCATE CHANNEL コマンドまたは CONFIGURE CHANNEL コマンドに使用した各接続文字列に 1 つ存在します。

### ジョブのハングアップ中のプロセス動作

Recovery Manager がハングアップする理由は、通常、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・リソースが使用可能になるまで待機するためです。カタログ接続およびデフォルト・チャンネルがハングアップしているように見えるのは、Recovery Manager からの指示を待機しているためです。ポーリング接続は、Recovery Manager プロセスの制御下で RPC をポーリングする間は無限にループしているように見えます。



Recovery Manager プロセス自体を強制終了すると、カタログ接続、補助接続、デフォルト・チャンネルおよびポーリング接続が切断されます。メディア・マネージャ・コードでハングアップしていないターゲット接続および補助接続も切断されます。メディア管理レイヤーで実行しているターゲット接続か補助接続のいずれかは切断されません。これらのプロセスを終了するには、オペレーティング・システム・レベルで手動で強制終了する必要があります。

すべてのメディア・マネージャが Oracle プロセスの終了を検出できるわけではありません。終了を検出しないメディア・マネージャは、リソースをビジー状態のままにしたり、処理を継続する場合があります。詳細は、ご使用のメディア・マネージャのドキュメントを参照してください。

カタログ接続を切断しても、Recovery Manager プロセスは終了されません。これは、Recovery Manager は、バックアップまたはリストアの実行中にはカタログ操作を実行しないためです。デフォルト・チャンネルおよびポーリング接続を削除すると、Recovery Manager プロセスはチャンネルの1つが終了したことを検出し、終了処理を実行します。この場合、前述のとおり、ハングアップしたチャンネルへの接続はアクティブのままです。

## Recovery Manager セッションの終了の基本手順

メディア・マネージャ・コードでハングアップしたチャンネルが強制終了されると、Recovery Manager プロセスはその終了を検出し、終了処理を実行します。その際、メディア管理レイヤーで実行可能なターゲット接続を除く、すべての接続を削除します。この場合にも、メディア・マネージャ・リソースに関する警告が当てはまります。

### メディア・マネージャでハングアップした Oracle プロセスの終了手順

- 22-8 ページの「[Recovery Manager のトラブルシューティングでの V\\$ ビューの使用](#)」で説明するように、V\$SESSION および V\$SESSION\_WAIT を問い合わせます。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, s.EVENT, s.SECONDS_IN_WAIT AS SEC_WAIT,
       sw.STATE, s.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、待機中の SBT 機能を確認します。たとえば、次のような出力が表示されます。

```
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
```

- ご使用のプラットフォームに適切なオペレーティング・システム・レベルのツールを使用して、ハングアップしたセッションを強制終了します。たとえば、Linux では kill -9 コマンドを実行します。

```
% kill -9 8642 8374
```

一部のプラットフォームには、orakill というコマンドライン・ユーティリティが含まれています。このユーティリティを使用すると、特定のスレッドを強制終了できます。コマンド・プロンプトから、次のコマンドを実行します。ここで、sid はターゲットに対するデータベース・インスタンスで、thread\_id は手順 1 の問合せで取得した SPID 値です。

```
orakill sid thread_id
```

3. メディア・マネージャがプロセスも消去したことを確認します。消去されていないプロセスが存在する場合、次のバックアップ処理またはリストア処理で、以前のハングアップが原因で再度ハングアップする可能性があります。一部のメディア・マネージャでは、この唯一の解決策は、メディア・マネージャを停止して再起動することです。メディア・マネージャのドキュメントに必要な情報が記載されていない場合、メディア・マネージャのテクニカル・サポートに問い合せてください。

**参照：** 関連するコマンドの詳細は、ご使用のオペレーティング・システム固有のドキュメントを参照してください。

# 第 VII 部

---

## Recovery Manager を使用したデータの送信

第 VII 部では、データベースおよび表領域の転送および移行に Recovery Manager を使用する方法について説明します。第 VII 部に含まれる章は次のとおりです。

- 第 23 章「データベースの複製」
- 第 24 章「トランスポータブル表領域セットの作成」
- 第 25 章「プラットフォーム間でのデータ転送」
- 第 26 章「ASM でのデータの移行の実行」



---

## データベースの複製

この章では、DUPLICATE コマンドを使用して、複製データベースを作成する方法を説明します。この章では、次の項目について説明します。

- [Recovery Manager データベースの複製の概要](#)
- [バックアップおよびアーカイブ・ログから複製インスタンスへのアクセス可能化](#)
- [複製ファイルの名前の指定方法の選択](#)
- [補助インスタンスの準備](#)
- [Recovery Manager の起動および複製前の構成](#)
- [データベースの複製](#)
- [代替方法を使用した複製ファイルの名前の指定](#)
- [Recovery Manager による複製の例](#)

## Recovery Manager データベースの複製の概要

この項では、データベースを複製する場合の基本的な概念およびタスクについて説明します。

### データベースの複製の目的

データベースの複製では、**複製データベース**の作成を目的としています。複製データベースとは、**ソース・データベース**にあるすべてのデータまたはデータのサブセットが含まれている別のデータベースのことです。複製データベースは様々な目的で有効であり、ほとんどの場合、テストに使用されます。複製データベースでは、次のタスクを実行できます。

- バックアップ手順およびリカバリ手順のテスト
- Oracle Database の新しいリリースへの更新のテスト
- データベースのパフォーマンスにアプリケーションが与える影響のテスト
- レポートの生成
- 本番データベースから誤って削除した表などのデータのエクスポート、およびそのデータの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 上に複製し、host1 上の本番データベースを通常どおりに実行したまま、host2 上の複製データベースを使用してこのデータベースのリストアおよびリカバリを練習できます。

DUPLICATE を使用してではなくオペレーティング・システムのユーティリティを使用してデータベースをコピーした場合、コピーしたデータベースの DBID は元のデータベースと同じになります。このコピーした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 (『Oracle Database ユーティリティ』を参照)。これに対して、DUPLICATE コマンドを実行すると、コピーされるデータベースに対して異なる DBID が自動的に割り当てられるため、このデータベースをコピー元のデータベースと同じリカバリ・カタログに登録できます。

複製データベースは**フィジカル・スタンバイ・データベース**とは異なる目的で使用されますが、いずれのデータベースも 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・データベースのアーカイブ・ログを使用して継続的に更新できます。プライマリ・データベースにアクセスできない場合は、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、それを新しいプライマリ・データベースにすることができます。一方、複製データベースはこのような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目的としていないため、スタンバイの様々なリカバリ・オプションおよびフェイルオーバー・オプションをサポートしていません。

**参照：** 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、『Oracle Data Guard 概要および管理』を参照してください。

### データベースの複製の基本的な概念

Recovery Manager の DUPLICATE コマンドを使用して、複製データベースを作成します。複製データベースはコピー先のデータベースであり、ソース・データベースはコピー元のデータベースです。複製データベースの DBID はソース・データベースの DBID とは異なっており、完全に独立して機能します。

**ソース・ホスト**はソース・データベースをホストするコンピュータであり、**転送先ホスト**は複製データベースをホストするコンピュータです。ソース・ホストと転送先ホストは、同じである場合も異なる場合もあります。ソース・データベース・インスタンスは、ソース・データベースに関連付けられた Oracle インスタンスです。複製データベースに関連付けられたインスタンスは、**補助インスタンス**と呼ばれます。

複製データベースには、ソース・データベースと同じ内容か、またはソース・データベース内の表領域のサブセットのみを含めることができます。たとえば、TABLESPACE オプションを使用すると、指定した表領域のみを複製できます。また、SKIP READONLY オプションを使用すると、複製データベースから読取り専用表領域を除外できます。

**アクティブなデータベースの複製**または**バックアップベースの複製**のいずれかの方法を使用できます。アクティブなデータベースの複製では、ネットワークを介して使用中のソース・データベースが複製データベース・インスタンスにコピーされますが、バックアップベースの複製では、既存の Recovery Manager バックアップおよびコピーが使用されます。

複製の主な作業は、**補助チャンネル**によって実行されます。このチャンネルは、転送先ホスト上の補助インスタンスのサーバー・セッションに対応しています。Recovery Manager は、複製操作のうち、次の処理を自動化します。

1. 複製データベースの制御ファイルの作成
2. 補助インスタンスの再起動および複製制御ファイルのマウント
3. 複製データファイルの作成および増分バックアップとアーカイブ REDO ログを使用した複製データファイルのリカバリ

ソース・データベースのオンライン REDO ログ・ファイル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 で不完全リカバリを実行する必要があります。Recovery Manager では、ソース・データベースによってアーカイブされた最新の REDO ログの時点まで、複製データベースをリカバリできます。

4. RESETLOGS オプションを使用した複製データベースのオープン

**参照：** 複製データベースにコピーされるファイルの詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』の DUPLICATE に関する項を参照してください。

## データベースの複製の基本手順

データベースを複製するには、多くの前提条件を満たしている必要があります。たとえば、ソース・データベースおよび複製データベースは、同じプラットフォーム上にある必要があります。詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』の DUPLICATE コマンド・エントリの前提条件に関する項を参照してください。

### データベースを複製する手順

1. バックアップベースの複製を使用する場合は、データベース・バックアップおよびアーカイブ REDO ログを転送先ホスト上の補助インスタンスで使用できるようにします。アクティブなデータベースの複製を使用する場合、この手順はスキップします。

このタスクについては、23-4 ページの「[バックアップおよびアーカイブ・ログから複製インスタンスへのアクセス可能化](#)」を参照してください。

2. 複製制御ファイル、データファイル、オンライン REDO ログおよび一時ファイルに名前を指定する方法を決定します。ソース・ホストと同じディレクトリ構造を使用する別のホストに複製を行う場合、およびソース・データベース・ファイルと同じ名前を複製ファイルに指定する場合は、次の手順にスキップします。

このタスクについては、23-6 ページの「[複製ファイルの名前の指定方法の選択](#)」を参照してください。

3. 転送先ホスト上の補助インスタンスで使用する初期化パラメータ・ファイルを作成してから、補助インスタンスを起動します。

このタスクについては、23-8 ページの「[補助インスタンスの準備](#)」を参照してください。

4. Recovery Manager を起動して、TARGET としてソース・データベースに、AUXILIARY として複製データベースのインスタンスに、必要に応じて CATALOG としてリカバリ・カタログ・データベースに接続します。必要に応じて、DUPLICATE コマンドで使用するチャンネルを構成します。

このタスクについては、23-11 ページの「[Recovery Manager の起動および複製前の構成](#)」を参照してください。

5. DUPLICATE コマンドを実行します。

このタスクについては、23-12 ページの「[データベースの複製](#)」を参照してください。

## バックアップおよびアーカイブ・ログから複製インスタンスへのアクセス可能化

アクティブなデータベースの複製を使用している場合は、この項をスキップし、23-6 ページの「複製ファイルの名前の指定方法の選択」に進みます。

Recovery Manager は、複製制御ファイルのメタデータを使用して、複製に必要なバックアップおよびアーカイブ・ログを検索します。補助チャネルが、複製データベースのリストアおよびリカバリに必要なすべてのデータファイル・バックアップおよびアーカイブ REDO ログにアクセスできることを確認する必要があります。アクセスできない場合、複製操作は失敗します。

データベース・バックアップは、BACKUP DATABASE で生成す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および増分バックアップを組み合わせで使用できます。たとえば、データファイル 1、2 および 3 の全体バックアップを作成したとします。また、データファイル 4、5、6 のレベル 0 およびレベル 1 のバックアップも作成したとします。この場合、DUPLICATE を実行すると、全体バックアップからファイル 1、2 および 3、レベル 0 のバックアップからファイル 4、5 および 6 をリストアできます。Recovery Manager は、増分レベル 1 の増分バックアップを 4、5 および 6 に適用し、アーカイブ REDO ログを 6 つすべてのデータファイルに適用できます。

バックアップベースの複製を使用している場合は、データベース・バックアップを補助インスタンスで使用できるようにする方法を決定する必要があります。また、複製データベースを目的の時点にリカバリする場合に必要なアーカイブ REDO ログ・ファイルは、複製データベースが作成されるホストから同じパスでアクセスできるようになっている必要があります。これらのログは、バックアップ（メディア・マネージャを使用する場合など）またはイメージ・コピー（または実際のアーカイブ REDO ログ・ファイル）のいずれかとして使用できます。

### SBT バックアップから複製インスタンスへのアクセス可能化

SBT バックアップを使用する場合は、バックアップが含まれているテープから転送先ホストにアクセスできるようにする必要があります。メディア管理ソフトウェアが転送先ホスト上にインストールされている場合は、リモート・ホストに接続されているドライブにテープを物理的に移動するか、またはネットワーク・アクセスが可能なテープ・サーバーを使用することによって実現できます。通常、メディア・マネージャによっては、リモート・メディア管理ソフトウェアにテープの存在について通知する必要があります。

### ディスク・バックアップから複製インスタンスへのアクセス可能化

ディスク・バックアップを使用する場合、ソース・ホストのファイル・システムと転送先ホストのファイル・システムが同じである場合が最も簡単です。たとえば、ソース・データベースのバックアップが /dsk1/bkp に格納されているとします。この場合、次のいずれかの方法で、ディスク・バックアップから転送先ホストにアクセスできるようになります。

- バックアップおよびコピーを、ソース・ホストからソース・ホストのパスと同様に設定した転送先ホストのパスに手動で移動します。たとえば、バックアップがソース・ホストの /dsk1/bkp にある場合は、それらのバックアップを転送先ホストの /dsk1/bkp に FTP 転送します。
- NFS または共有ディスクを使用して、転送先ホストで同じパスにアクセスできるようにします。たとえば、NFS を使用して転送先ホストに /dsk1/bkp をマウントし、/dsk1/bkp をマウント・ポイント名として使用します。

ソース・ホストで使用するディレクトリ名と同じディレクトリ名を転送先ホストで使用できない場合は複雑です。この場合にバックアップをアクセス可能にする方法は、共有ディスクを使用するかどうかによって異なります。次の項で説明します。



## 共有ディスクを使用しないディスク・バックアップのアクセス可能化

NFS または共有ディスクを使用できない場合、バックアップを格納するパスはソース・ホストおよび転送先ホストの両方に存在する必要があります。srchost および dsthost という 2 つのホストがあるとします。srchost 上のデータベースは、srcdb です。srcdb の Recovery Manager バックアップは、ホスト srchost 上の /dsk1/bkp 内に存在します。ディレクトリ /dsk1/bkp はすでに転送先ホストで使用されているため、バックアップは転送先ホスト上の /dsk2/dup に格納します。複製サイトの制御ファイルをダンプファイルの新しい場所で更新する必要があります。

バックアップをソース・ホストから転送先ホストに転送するには、次の手順を実行します。

1. バックアップを格納する転送先ホスト上にあるディレクトリと同じ名前のディレクトリをソース・ホスト上に新しく作成します。

たとえば、Recovery Manager バックアップを転送先ホストの /dsk2/dup に格納する場合、ソース・ホスト上に /dsk2/dup を作成します。

2. ソース・ホスト上で、前の手順で作成したディレクトリにバックアップをコピーして、バックアップをカタログに追加します。次のいずれかの方法を使用できます。
  - Recovery Manager を TARGET としてソース・データベースに接続し、8-28 ページの「[Recovery Manager バックアップのバックアップ](#)」で説明されている BACKUP コマンドを使用して、バックアップのバックアップを作成します。たとえば、BACKUP COPY OF DATABASE コマンドを使用して、ソース・ホストの /dsk1/bkp にあるバックアップをソース・ホストの /dsk2/dup にコピーします。この場合、Recovery Manager は、新しい場所のバックアップを自動的にカタログに追加します。
  - オペレーティング・システム・ユーティリティを使用して、ソース・ホストの /dsk1/bkp にあるバックアップを、ソース・ホストの /dsk2/dup にコピーします。後で、Recovery Manager を TARGET としてソース・データベースに接続し、CATALOG コマンドを使用して、手動で転送したバックアップの場所で、ソース・データベースの制御ファイルを更新します。
3. ソース・ホスト上の新しいディレクトリから転送先ホスト上の同じ名前のディレクトリに、バックアップを手動で転送します。

たとえば、FTP を使用して、ソース・ホストの /dsk2/dup にあるバックアップを、転送先ホストの /dsk2/dup に転送します。

複製の一部として、Recovery Manager は、バックアップを表示する複製制御ファイルを /dsk2/dup パスに作成します。その後、補助チャネルでは、転送先ホスト上の /dsk2/dup でバックアップを検索してリストアできます。

## 共有ディスクを使用したディスク・バックアップのアクセス可能化

srchost および dsthost という 2 つのホストがあるとします。srchost 上のデータベースは、srcdb です。srcdb のバックアップは、ホスト srchost 上の /dsk1/bkp 内に存在します。ディレクトリ /dsk1/bkp は転送先ホストで使用されていますが、ディレクトリ /dsk2/dup は、いずれのホストでも使用されていません。

バックアップをソース・ホストから転送先ホストに転送するには、次の手順を実行します。

1. ソース・ホストまたは転送先ホストのいずれかに、バックアップを格納するディレクトリを新しく作成します。

たとえば、転送先ホスト上にバックアップ・ディレクトリ /dsk2/dup を作成します。

2. 前の手順で作成したディレクトリを、もう一方のホストにマウントします。ディレクトリとマウント・ポイントの名前が同一であることを確認します。

たとえば、転送先ホストに /dsk2/dup を作成した場合は、NFS を使用してこのディレクトリを /dsk2/dup としてソース・ホストにマウントします。

3. バックアップを転送先ホスト上の新しい場所で使用できるようにします。次のいずれかの方法を使用できます。
  - Recovery Manager を TARGET としてソース・データベースに接続し、8-28 ページの「[Recovery Manager バックアップのバックアップ](#)」で説明されている BACKUP コマンドを使用して、バックアップのバックアップを作成します。たとえば、BACKUP COPY OF DATABASE コマンドを使用して、ソース・ホストの /dsk1/bkp にあるバックアップをソース・ホストの /dsk2/dup にコピーします。この場合、Recovery Manager は、新しい場所のバックアップを自動的にカタログに追加します。
  - オペレーティング・システム・ユーティリティを使用して、バックアップを新しい場所に転送します。たとえば、ソース・ホストの /dsk1/bkp から転送先ホストの /dsk2/dup にバックアップを FTP 転送するか、cp コマンドを使用して、ソース・ホストの /dsk1/bkp からソース・ホストの /dsk2/dup にバックアップをコピーします。後で、Recovery Manager を TARGET としてソース・データベースに接続し、CATALOG コマンドを使用して、手動で転送したバックアップの場所で、ソース・データベースの制御ファイルを更新します。

複製の一部として、Recovery Manager は、バックアップを表示する複製制御ファイルを /dsk2/dup パスに作成します。その後、補助チャンネルでは、転送先ホスト上の /dsk2/dup でバックアップを検索してリストアできます。

**参照：** BACKUP コマンドについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## 複製ファイルの名前の指定方法の選択

Recovery Manager は、データベースの複製時に複製データベース・ファイルの名前を生成します。転送先ホストでソース・ホストと同じディレクトリ構造が使用されている場合は、複製データベース・ファイルに、ソース・データベース・ファイルで使用されている名前と同じ名前を使用できます。この場合、複製ファイルの名前を変更する必要はありませんが、DUPLICATE コマンドで NOFILENAMECHECK オプションを指定する必要があります。

ホストで異なるディレクトリ構造が使用されている場合、または同じ構造が使用されているが、複製ファイルに異なる名前を使用する必要がある場合は、複製データベース・ファイルの名前の生成方法を決定する必要があります。具体的には、制御ファイル、データファイル、オンライン REDO ログ・ファイルおよび一時ファイルに名前を指定する方法を決定する必要があります。

Recovery Manager では、複製ファイルの名前を生成する次の方法がサポートされています。

- DUPLICATE コマンドに SPFILE 句を指定すると、DB\_FILE\_NAME\_CONVERT を除いて、ファイル名に関連するすべての必要なパラメータを設定できます。

最も簡単な方法であるため、この方法を使用することをお勧めします。DUPLICATE ... SPFILE を実行すると、Recovery Manager は、サーバー・パラメータ・ファイルをバックアップからリストアするか、またはアクティブなデータベースからコピーします。Recovery Manager は、SPFILE\_PARAMETER\_VALUE\_CONVERT および SPFILE SET の値に（この順序で）基づいて、コピーされたサーバー・パラメータ・ファイルの初期化パラメータ値を更新します。その後、Recovery Manager は、サーバー・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。
- 23-17 ページの「[代替方法を使用した複製ファイルの名前の指定](#)」で説明されているいずれかの代替方法を使用します。

テキスト・エディタを使用して複製インスタンス上の初期化パラメータ・ファイルでパラメータを設定するか、DUPLICATE コマンドに LOGFILE および DB\_FILE\_NAME\_CONVERT 句を指定するか、または SET および CONFIGURE コマンドを発行します。これらのオプションは、希望する結果となるように任意に組み合わせることができます。

この項では、SPFILE の方法についてのみ説明します。次のリストに、複製ファイルの名前を指定する場合に使用可能な DUPLICATE オプションを示します。DUPLICATE コマンドの構文およびセマンティクスの詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

- SPFILE ... PARAMETER\_VALUE\_CONVERT 'string\_pattern'

DB\_FILE\_NAME\_CONVERT および LOG\_FILE\_NAME\_CONVERT パラメータを除いて、パス名を指定する値を持つ初期化パラメータの変換文字列を指定します。

PARAMETER\_VALUE\_CONVERT の主な目的は、初期化パラメータの集合を設定して、1 つずつ明示的に設定しなくてもすむようにすることです。

---

**注意：** PARAMETER\_VALUE\_CONVERT を実行すると、パス名が含まれている文字列値のみでなく、すべての文字列値を更新できます。これらの値では、大 / 小文字が区別されます。

---

- SPFILE ... SET 'string\_pattern'

指定した初期化パラメータを指定した値に設定します。SET を使用すると、オンライン REDO ログに対して LOG\_FILE\_NAME\_CONVERT パラメータを設定できます。

- DB\_FILE\_NAME\_CONVERT 'string\_pattern'

データファイルおよび一時ファイルのファイル名を作成する規則を指定します。

DUPLICATE コマンドに指定された DB\_FILE\_NAME\_CONVERT によって、初期化パラメータ DB\_FILE\_NAME\_CONVERT (初期化パラメータ・ファイルに設定されている場合) が上書きされます。

- NOFILENAMECHECK

ソース・データベース・データファイルとオンライン REDO ログ・ファイルの名前が、複製されたファイルと同じ名前かどうかを Recovery Manager が確認しなくなります。このオプションは、ソース・データベースのホストと同じディスク構成、ディレクトリ構成およびファイル名を持つ別のホスト上で、複製データベースを作成する場合に必要となります。ソース・データベースと同じホスト上でデータベースを複製する場合は、NOFILENAMECHECK が設定されていないことを確認してください。

例 23-1 に、SPFILE 句を使用して複製ファイルに名前を指定する DUPLICATE コマンドを示します。PARAMETER\_VALUE\_CONVERT オプションによって、ファイル名を指定するすべての初期化パラメータの /disk2 が /disk1 に置き換えられます (ただし、DB\_FILE\_NAME\_CONVERT および LOG\_FILE\_NAME\_CONVERT を除く)。SET LOG\_FILE\_NAME\_CONVERT 句によって、複製データベースのオンライン REDO ログのファイル名の /disk2 が /disk1 に置き換えられます。DB\_FILE\_NAME\_CONVERT オプションによって、複製データファイルおよび一時ファイルの名前の /disk1 が /disk2 に置き換えられます。

### 例 23-1 SPFILE 句を使用した複製ファイルの名前の指定

```
DUPLICATE TARGET DATABASE TO dup1
FROM ACTIVE DATABASE
DB_FILE_NAME_CONVERT '/disk1','/disk2'
SPFILE
PARAMETER_VALUE_CONVERT '/disk1', '/disk2'
SET LOG_FILE_NAME_CONVERT '/disk1','/disk2'
SET SGA_MAX_SIZE '200M'
SET SGA_TARGET '125M';
```

**参照：** DB\_FILE\_NAME\_CONVERT 初期化パラメータの詳細は、『Oracle Database リファレンス』を参照してください。

## 補助インスタンスの準備

DUPLICATE DATABASE コマンドで使用する補助インスタンスを準備するには、次のタスクを実行します。

- 手順 1: 補助インスタンス用の Oracle パスワード・ファイルの作成
- 手順 2: Oracle Net から補助インスタンスへの接続の確立
- 手順 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
- 手順 4: SQL\*Plus を使用した補助インスタンスの起動

### 手順 1: 補助インスタンス用の Oracle パスワード・ファイルの作成

次のいずれかの条件に該当する場合にのみ、補助インスタンスでパスワード・ファイルが必要となります。

- 転送先ホスト以外のホスト上で Recovery Manager クライアントを使用します。
- アクティブなデータベースから複製を行います。

---

**注意：** パスワード・ファイルは、バックアップベースの複製では必要ではありません。ソース・データベースと同じホストに複製する場合は、補助接続にオペレーティング・システム認証を使用できます。

---

FROM ACTIVE DATABASE オプションを使用すると、ソース・データベース・インスタンス (Recovery Manager が TARGET として接続されるデータベース・インスタンス) は、補助データベース・インスタンスに直接接続されます。この接続では、同じ SYSDBA パスワードが含まれているパスワード・ファイルが必要です。パスワード・ファイルを手動で作成し、ソース・データベースと同じ SYSDBA パスワードを使用するようにします。補助インスタンスを起動し、ソース・データベースをその補助インスタンスに接続できるように、1つのパスワードが含まれているパスワード・ファイルを作成することができます。

また、DUPLICATE コマンドに PASSWORD FILE オプションを指定することもできます。この場合、Recovery Manager は、ソース・データベース・パスワード・ファイルを転送先ホストにコピーし、既存のすべての補助インスタンスのパスワード・ファイルを上書きします。この方法は、複製データベースで使用できるようにするパスワードがソース・データベースのパスワード・ファイルに複数含まれている場合に有効です。

アクティブなデータベースの複製でスタンバイ・データベースを作成する場合、Recovery Manager は、パスワード・ファイルを常にスタンバイ・ホストにコピーします。PASSWORD FILE オプションを指定する必要はありません。また、補助インスタンス用の既存のパスワード・ファイルを上書きします。

#### パスワード・ファイルを作成する手順

- パスワード・ファイルを作成するには、『Oracle Database 管理者ガイド』の手順を実行します。パスワード・ファイルに使用できるファイル名の形式は、オペレーティング・システムに固有です。

**参照：** 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、『Oracle Data Guard 概要および管理』を参照してください。

## 手順 2: Oracle Net から補助インスタンスへの接続の確立

次のいずれかの条件を満たしている場合、補助インスタンスは、Oracle Net を介して使用できる必要があります。

- 転送先ホスト以外のホスト上で Recovery Manager クライアントを使用します。
- アクティブなデータベースから複製を行います。

アクティブなデータベースから複製する場合は、まず、ネット・サービス名を使用して、SYSDBA として補助インスタンスに接続する必要があります。このネット・サービス名は、ソース・データベース・インスタンス上でも使用可能である必要があります。ソース・データベース・インスタンス (Recovery Manager が TARGET として接続されるインスタンス) は、このネット・サービス名で補助データベース・インスタンスに直接接続されます。

## 手順 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テキストベースの初期化パラメータ・ファイルを作成します。

複製ファイルの名前の指定に、SPFILE の方法 (23-6 ページの「複製ファイルの名前の指定方法の選択」を参照) を使用する場合、必要なパラメータは、任意の値に指定できる DB\_NAME のみとなります。DB\_FILE\_NAME\_CONVERT、LOG\_FILE\_NAME\_CONVERT などのパラメータは、初期化パラメータ・ファイルで設定する必要はありません。これらのパラメータおよび他のすべてのパラメータは、DUPLICATE コマンド自体で設定できるためです。

SPFILE の方法を使用していない場合は、初期化パラメータを初期化パラメータ・ファイルで設定する必要があります。表 23-1 に、これらの初期化パラメータを示します。

表 23-1 補助インスタンスの初期化パラメータ

| 初期化パラメータ              | 値   | ステータス                                |
|-----------------------|---|--------------------------------------|
| DB_NAME               | DUPLICATE コマンドで使用する名前と同じ名前。<br>複製データベースの DB_NAME 設定は、その Oracle ホームのデータベース間で一意である必要があります。  | 必須                                   |
| CONTROL_FILES         | 制御ファイルの場所。<br><b>参照:</b> 23-17 ページの「複製制御ファイルの名前の指定」を参照してください。   | 必須 (Oracle 管理ファイル用のパラメータを設定していない場合)  |
| DB_BLOCK_SIZE         | 複製データベースのブロック・サイズ。<br>このブロック・サイズは、ソース・データベースのブロック・サイズと一致している必要があります。ソース・データベースのパラメータ・ファイルに DB_BLOCK_SIZE 初期化パラメータの値が含まれている場合、補助インスタンスに同じ値を指定する必要があります。ただし、ソース・データベースの初期化パラメータ・ファイルに DB_BLOCK_SIZE が指定されていない場合は、補助インスタンスに DB_BLOCK_SIZE は指定しないでください。<br><b>参照:</b> 23-17 ページの「代替方法を使用した複製ファイルの名前の指定」を参照してください。 | 必須 (ソース・データベースでこの初期化パラメータが設定されている場合) |
| DB_FILE_NAME_CONVERT  | データファイルおよび一時ファイルの名前を変換する文字列のペア。DUPLICATE コマンド自体に DB_FILE_NAME_CONVERT を指定することもできます。<br><b>参照:</b> 23-18 ページの「複製データファイルの名前の指定」および 23-19 ページの「複製一時ファイルの名前の指定」を参照してください。   | オプション                                |
| LOG_FILE_NAME_CONVERT | オンライン REDO ログ・ファイルの名前を指定する文字列のペア。<br><b>参照:</b> 23-17 ページの「複製オンライン REDO ログ・ファイルのネーミング」を参照してください。  | オプション                                |

表 23-1 補助インスタンスの初期化パラメータ（続き）

| 初期化パラメータ                    | 値  | ステータス |
|-----------------------------|--|-------|
| DB_CREATE_FILE_DEST         | Oracle 管理データファイルの場所。<br><b>参照</b> ：23-17 ページの「複製データファイルの名前の指定」および 23-17 ページの「複製一時ファイルの名前の指定」を参照してください。 | オプション |
| DB_CREATE_ONLINE_LOG_DEST_# | Oracle 管理オンライン REDO ログ・ファイルの場所。<br><b>参照</b> ：23-17 ページの「複製オンライン REDO ログ・ファイルのネーミング」を参照してください。         | オプション |
| DB_RECOVERY_FILE_DEST       | フラッシュ・リカバリ領域の場所。<br><b>参照</b> ：23-17 ページの「代替方法を使用した複製ファイルの名前の指定」を参照してください。                             | オプション |

必要に応じて、Oracle Net を介して SYSDBA として接続するためのパラメータも含めて、その他の初期化パラメータを設定します。

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を作成する場合は、パス名を指定するすべての初期化パラメータ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。

例 23-2 に、サンプル初期化パラメータ・ファイルの設定例を示します。

#### 例 23-2 補助インスタンス用のサンプル初期化パラメータ・ファイル

```
DB_NAME=dupdb
CONTROL_FILES=(/dup/oracle/oradata/prod/control01.ct1,
               /dup/oracle/oradata/prod/control02.ct1)
DB_FILE_NAME_CONVERT=(/oracle/oradata/prod/,/dup/oracle/oradata/prod/)
LOG_FILE_NAME_CONVERT=(/oracle/oradata/prod/redo,/dup/oracle/oradata/prod/redo)
```

## 手順 4: SQL\*Plus を使用した補助インスタンスの起動

Recovery Manager の複製を開始する前に、SQL\*Plus を使用して SYSOPER 権限で補助データベース・インスタンスに接続します。インスタンスを NOMOUNT モードで起動し、必要に応じて PFILE パラメータを指定します。

```
SQL> STARTUP NOMOUNT
```

---

**注意：** 補助インスタンスに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、複製処理中に補助インスタンスを停止して再起動します。このため、補助インスタンス用のサーバー側の初期化パラメータ・ファイルを転送先ホスト上のデフォルトの場所に含めることをお勧めします。

DUPLICATE コマンドの SPFILE 句を使用すると、Recovery Manager は、ソース・データベースのサーバー・パラメータ・ファイルを転送先ホスト上の補助インスタンスのデフォルトの場所にコピーまたはリストアします。SPFILE 句を使用しない場合は、サーバー・パラメータ・ファイルを転送先ホストにコピーするか、または DUPLICATE コマンドの PFILE パラメータが含まれているテキストベースの初期化パラメータ・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テキストベースの初期化パラメータ・ファイルは、複製の実行に使用される Recovery Manager クライアントと同じホスト上に存在する必要があります。



## Recovery Manager の起動および複製前の構成

DUPLICATE DATABASE コマンドを実行する前に、次のタスクを実行します。

- 手順 1: Recovery Manager の起動およびデータベース・インスタンスへの接続
- 手順 2: ソース・データベースのマウントまたはオープン
- 手順 3: 複製で使用するための Recovery Manager チャンネルの構成

### 手順 1: Recovery Manager の起動およびデータベース・インスタンスへの接続

Recovery Manager を起動して、TARGET としてソース・データベースに、AUXILIARY として複製データベース・インスタンスに、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。すべてのデータベース・インスタンスに接続可能であるかぎり、どのホスト上でも Recovery Manager クライアントを起動できます。補助インスタンスにテキストベースの初期化パラメータ・ファイルが必要な場合は、Recovery Manager クライアント・アプリケーションが実行されるホストと同じホスト上にそのファイルが存在する必要があります。

#### Recovery Manager を起動し、ターゲット・インスタンスおよび補助インスタンスに接続する手順

1. Recovery Manager クライアントを起動します。

たとえば、オペレーティング・システム・プロンプトで次のコマンドを入力します。

```
% rman
```

2. Recovery Manager プロンプトで、各データベース・インスタンスに対して CONNECT コマンドを実行します。

次の例では、3 つのすべてのデータベース・インスタンスにネット・サービス名を使用して接続します。

```

RMAN> CONNECT TARGET SYS@prod      # source database

target database Password: password
connected to target database: PROD (DBID=39525561)

RMAN> CONNECT AUXILIARY SYS@dupdb   # duplicate database instance

auxiliary database Password: password
connected to auxiliary database: DUPDB (not mounted)

RMAN> CONNECT CATALOG rman@catdb    # recovery catalog database

recovery catalog database Password: password
connected to recovery catalog database

```

### 手順 2: ソース・データベースのマウントまたはオープン

Recovery Manager による複製を開始する前に、ソース・データベースがマウントまたはオープンされていない場合は、マウントまたはオープンします。ソース・データベースがオープンしている場合は、アーカイブを有効にする必要があります。ソース・データベースがオープンしていない場合、およびソース・データベースがスタンバイ・データベースでない場合は、一貫性のある状態で停止する必要があります。

## 手順 3: 複製で使用するための Recovery Manager チャンネルの構成

バックアップベースの複製に必要な場合は、補助データベース・インスタンス上で使用するチャンネルを構成します。バックアップをリストアするのは補助インスタンスのチャンネルであり、ソース・データベース・インスタンスではないことに注意してください。

Recovery Manager は、AUXILIARY オプションが指定されていなくても、ソース・データベースに設定された同じチャンネル構成を複製に使用できます。ただし、補助チャンネルに特別なパラメータが必要な場合（別のメディア・マネージャを指す場合など）は、CONFIGURE コマンドの AUXILIARY オプションを指定して自動チャンネルを構成します。

バックアップベースの複製では、チャンネルのチャンネル・タイプ（DISK または sbt）は、ソース・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にある場合は、割り当てるチャンネルの数を増やすと、複製の速度も速くなります。テープ・バックアップの場合は、チャンネルの数を可能なデバイスの数に制限します。

アクティブなデータベースの複製では、ソース・データベースのチャンネル構成を変更したり、AUXILIARY チャンネルを構成する必要はありません。ただし、Recovery Manager がネットワークを介して平行にファイルをコピーできるように、ソース・データベースのディスク・チャンネルの平行化設定を増加させる必要がある場合があります。

**参照：** CONFIGURE コマンドについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## データベースの複製

この項では、複製データベースを作成する最も基本的な手順について説明します。この手順は、データベースおよびホストの構成方法によって異なります。この項の内容は、次のとおりです。

- 同じディレクトリ構造を持つリモート・ホストで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・ホストでのデータベースの複製
- ローカル・ホストでの複製データベースの作成
- Oracle Managed Files または自動ストレージ管理を使用したデータベースの複製

**参照：** DUPLICATE コマンドで指定できるオプションおよび句の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## 同じディレクトリ構造を持つリモート・ホストでのデータベースの複製

最も簡単な方法は、アクティブなデータベースの複製を使用してデータベースを異なるホストに複製し、同じディレクトリ構造を使用する方法です。ソース・データベースでサーバー・パラメータ・ファイルが使用されている（またはバックアップが使用可能である）場合は、転送先ホストに一時初期化パラメータ・ファイルを作成し、DB\_NAME パラメータのみを設定できます。複製データベース・ファイル名に追加のパラメータを設定したり、バックアップを複製ホストに転送する必要はありません。

### 同じディレクトリ構造を持つリモート・ホストにデータベースを複製する手順

1. 23-8 ページの「補助インスタンスの準備」の手順を実行します。

この例では、初期化パラメータ・ファイルに、任意の値に設定された DB\_NAME のみが含まれています。

2. 23-11 ページの「Recovery Manager の起動および複製前の構成」の手順を実行します。
3. DUPLICATE コマンドを実行します。



例 23-3 に、アクティブな複製に DUPLICATE を使用する方法を示します。この例では、ソース・データベース・ファイルの名前が複製データベース・ファイルの名前と同じであるため、NOFILENAMECHECK オプションが必要です。

### 例 23-3 同じディレクトリ構造を持つホストへの複製（アクティブ）

```
DUPLICATE TARGET DATABASE
  TO dupdb
  FROM ACTIVE DATABASE
  SPFILE
  NOFILENAMECHECK;
```

Recovery Manager は、サーバー・パラメータ・ファイルを転送先ホストに自動的にコピーし、サーバー・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し、必要なすべてのデータベース・ファイルおよびアーカイブ REDO ログをネットワークを介して転送先ホストにコピーし、データベースをリカバリします。最後に、Recovery Manager は、RESETLOGS オプションを指定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。

ソース・データベースの 1 週間前のデータを表示するために、複製データベースをその時点までリカバリするとします。例 23-4 では、バックアップベースの複製を使用して、ソース・データベースの 1 週間前の状態の複製を作成します。UNTIL 句は、アクティブなデータベースの複製ではサポートされていません。

### 例 23-4 過去の特定の時点までのデータベースの複製（バックアップ・ベース）

```
DUPLICATE TARGET DATABASE
  TO dupdb
  PASSWORD FILE
  SPFILE
  NOFILENAMECHECK
  UNTIL TIME 'SYSDATE-7';
```

例 23-4 のコマンドでは、Recovery Manager がソース・データベースからパスワード・ファイルを複製する必要があることを示す PASSWORD FILE オプションが指定されています。

## 異なるディレクトリ構造を持つリモート・ホストでのデータベースの複製

異なるディレクトリ構造を持つホストに複製データベースを作成する場合は、複製データベースのデータファイルにファイル名を生成する方法を使用する必要があります。最も簡単な方法は、アクティブなデータベースの複製を使用し、SPFILE 句を使用してファイルの名前を指定する方法です（23-6 ページの「複製ファイルの名前の指定方法の選択」を参照）。1 ソース・データベースでサーバー・パラメータ・ファイルが使用されている（またはバックアップが使用可能である）場合は、転送先ホストに一時初期化パラメータ・ファイルを作成し、DB\_NAME パラメータのみを設定できます。

ソース・データベースのデータファイルが /oracle/oradata/prod/ にあり、それらのデータファイルを /scratch/oracle/oradata/dupdb/ に複製するとします。すべてのソース・データベースのオンライン REDO ログが /oracle/oradata/prod/redo/ にあり、それらのオンライン REDO ログを /scratch/oracle/oradata/dupdb/redo/ に複製するとします。

### 異なるディレクトリ構造を持つリモート・ホストにデータベースを複製する手順

1. 23-8 ページの「補助インスタンスの準備」の手順を実行します。

この例では、DB\_NAME のみがパラメータとして指定されている、複製インスタンス用の一時初期化パラメータ・ファイルを作成します。このパラメータは、任意の値に設定できません。

2. 23-11 ページの「**Recovery Manager の起動および複製前の構成**」の手順を実行します。

この例では、ソース・データベースはオープンされており、自動チャンネルが構成されていると想定しています。次のように入力して、**Recovery Manager** を起動し、ソース・データベース・インスタンスおよび補助データベース・インスタンスに接続します。

```

RMAN> CONNECT TARGET SYS@prod

target database Password: password
connected to target database: PROD1 (DBID=39525561)

CONNECT AUXILIARY SYS@dupdb

auxiliary database Password: password
connected to auxiliary database: DUPDB (not mounted)

```

3. **DUPLICATE** コマンドを実行します。

例 23-5 では、**DB\_FILE\_NAME\_CONVERT** パラメータと組み合わせて **SPFILE** 句を使用して、複製データベース・ファイルの名前を指定します。**PARAMETER\_FILE\_CONVERT** パラメータは **LOG\_FILE\_NAME\_CONVERT** または **DB\_FILE\_NAME\_CONVERT** に影響を及ぼしません。このため、これらの 2 つのパラメータは個別に設定する必要があります。

### 例 23-5 異なるディレクトリ構造を持つホストへの複製 (アクティブ)

```

DUPLICATE TARGET DATABASE
  TO dupdb
  FROM ACTIVE DATABASE
  DB_FILE_NAME_CONVERT '/oracle/oradata/prod/', '/scratch/oracle/oradata/dupdb/'
  SPFILE
  PARAMETER_VALUE_CONVERT '/oracle/oradata/prod/',
    '/scratch/oracle/oradata/dupdb/'
  SET SGA_MAX_SIZE '300M'
  SET SGA_TARGET '250M'
  SET LOG_FILE_NAME_CONVERT '/oracle/oradata/prod/redo/',
    '/scratch/oracle/oradata/dupdb/redo/';

```

**Recovery Manager** は、サーバー・パラメータ・ファイルを転送先ホストに自動的にコピーし、**PARAMETER\_VALUE\_CONVERT** および **SET** で指定されている値に基づいて、転送先ホスト上でサーバー・パラメータ・ファイルを更新します。その後、**Recovery Manager** は、サーバー・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し、必要なすべてのデータベース・ファイルおよびアーカイブ **REDO** ログをネットワークを介して転送先ホストにコピーし、データベースをリカバリします。最後に、**Recovery Manager** は、**RESETLOGS** オプションを指定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。

## ローカル・ホストでの複製データベースの作成

ソース・データベースと同じホスト上に複製データベースを作成する場合は、23-13 ページの「異なるディレクトリ構造を持つリモート・ホストでのデータベースの複製」に示す、異なるディレクトリ構造を持つリモート・ホストでの複製と同じ手順を実行します。

ソース・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、ソース・データベースとは異なるデータベース名を使用し、別のホスト上でファイル名を変換する場合と同じ方法で、ファイル名を変換する必要があります。

---

**注意：** プライマリ・データベースと同じ **Oracle** ホームへ複製する場合、**NOFILENAMECHECK** オプションは使用しないでください。このオプションを使用すると、**DUPLICATE** コマンドによって、ソース・データベースのデータファイルが上書きされる場合があります。

---

## Oracle Managed Files または自動ストレージ管理を使用したデータベースの複製

この項では、複製データベースの一部またはすべてのファイルが Oracle 管理ファイルであるか、またはそれらのファイルで自動ストレージ管理 (ASM) が使用されている場合に、複製データベースを作成する際の要件について説明します。

- Oracle 管理ファイルの初期化パラメータの設定
- ASM の初期化パラメータの設定

**参照:** ASM の概要は、『Oracle Database ストレージ管理者ガイド』を参照してください。

### Oracle 管理ファイルの初期化パラメータの設定

Oracle 管理ファイルを使用する複製データベースを作成する場合は、補助インスタンスで初期化パラメータを設定する必要があります。DUPLICATE の SPFILE 句を使用してファイルの名前を指定する場合は、SPFILE 句で初期化パラメータを設定できます。次の表に、関連する初期化パラメータおよび推奨設定を示します。

表 23-2 Oracle 管理ファイルの初期化パラメータ

| 初期化パラメータ                    | 用途   | 推奨事項  |
|-----------------------------|--|---|
| DB_CREATE_FILE_DEST         | Oracle 管理データファイルのデフォルトの場所を指定します。この場所は、どの DB_CREATE_ONLINE_LOG_DEST 初期化パラメータも指定されていない場合、Oracle 管理の制御ファイルおよびオンライン・ログのデフォルトの場所でもあります。 | このパラメータを Oracle 管理ファイルの場所に設定します。他の場所が指定されていないデータベース・ファイルが DUPLICATE によって DB_CREATE_FILE_DEST で作成されます。23-23 ページの「 <a href="#">SET NEWNAME を使用した Oracle 管理ファイルの名前の指定</a> 」で説明するように、SET NEWNAME を使用して特定のファイルのデフォルト値を上書きできます。 |
| DB_CREATE_ONLINE_LOG_DEST_n | Oracle 管理の制御ファイルおよびオンライン REDO ログのデフォルトの場所を指定します。複数のパラメータが設定されている場合は、それぞれの場所に制御ファイルおよびオンライン REDO ログが 1 つずつ作成されます。                   | 複数の場所の制御ファイルおよび REDO ログを多重化する場合のみ、これらのパラメータを設定します (_1、_2 など)。   |
| DB_RECOVERY_FILE_DEST       | フラッシュ・リカバリ領域のデフォルトの場所を指定します。フラッシュ・リカバリ領域には、現行の制御ファイルおよびオンライン REDO ログの多重コピーが含まれています。  | リカバリ領域に制御ファイルおよびオンライン REDO ログ・ファイルの多重コピーが必要な場合は、このパラメータを設定します。  |
| CONTROL_FILES               | 1 つ以上の制御ファイル名をカンマで区切って指定します。   | Oracle Managed Files 形式でデータベース制御ファイルを複製する場合は、このパラメータを設定しないでください。制御ファイルに OMF 形式を使用する場合は、複製データベースでサーバー・パラメータ・ファイルを使用することをお勧めします。  |
| DB_FILE_NAME_CONVERT        | プライマリ・データベース上の新しいデータファイルのファイル名を複製データベース上のファイル名に変換します。  | このパラメータを設定しないでください。このパラメータを省略することによって、データベースは複製データファイルに Oracle 管理ファイルの有効な名前を生成できます。   |

表 23-2 Oracle 管理ファイルの初期化パラメータ（続き）

| 初期化パラメータ              | 用途  | 推奨事項  |
|-----------------------|---|---|
| LOG_FILE_NAME_CONVERT | プライマリ・データベース上の新しいログ・ファイルのファイル名をスタンバイ・データベース上のログ・ファイルのファイル名に変換します。 | このパラメータを設定しないでください。このパラメータを省略することによって、データベースは Oracle 管理オンライン REDO ログ・ファイルの有効な名前を生成できます。<br><br>複製データベースのオンライン REDO ログ・ファイルを Oracle 管理ストレージに格納するには、DB_CREATE_FILE_DEST、DB_RECOVERY_FILE_DEST または DB_CREATE_ONLINE_LOG_DEST_n 初期化パラメータを使用して、オンライン・ログの Oracle 管理の場所を指定します。 |

### ASM の初期化パラメータの設定

ASM の場所に複製データベースを作成する手順は、23-15 ページの「Oracle 管理ファイルの初期化パラメータの設定」に示す手順と同様です。補助インスタンスで、ASM ディスク・グループへの複製時にファイルが作成される場所を制御する初期化パラメータを設定する必要がある点が異なります。たとえば、DB\_CREATE\_FILE\_DEST、DB\_CREATE\_ONLINE\_DEST\_n および CONTROL\_FILES を設定します。

**ファイル・システムから ASM へのデータベースの複製の例** この例では、アクティブなデータベースの複製を使用します。ソース・データベースでサーバー・パラメータ・ファイルが使用されている（またはバックアップが使用可能である）場合は、転送先ホストに一時初期化パラメータ・ファイルを作成し、DB\_NAME パラメータのみを設定できます。

ソース・データベース prod は host1 上にあり、データファイルは非 ASM ファイル・システムに保存されていると想定しています。prod の制御ファイルは、/oracle/oradata/prod/ に存在します。ソース・データベースを複製し、リモート・ホスト host2 上に dupdb データベースを作成します。ASM ディスク・グループ +DISK1 に複製データベース・ファイルを保存します。

Recovery Manager をターゲット・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・データベースに接続してから、例 23-6 に示す Recovery Manager スクリプトを実行してデータベースを複製します。

#### 例 23-6 ファイル・システムから ASM への複製（アクティブ）

```
DUPLICATE TARGET DATABASE
  TO dupdb
  FROM ACTIVE DATABASE
  SPFILE
  PARAMETER_VALUE_CONVERT '/oracle/oradata/prod/', '+DISK1'
  SET DB_CREATE_FILE_DEST +DISK1;
```

DUPLICATE コマンドが完了すると、データファイル、オンライン REDO ログおよび制御ファイルとともに複製データベースが ASM ディスク・グループ +DISK1 に作成されます。

**ASM から ASM へのデータベースの複製の例** この例では、アクティブなデータベースの複製を使用します。ソース・データベースでサーバー・パラメータ・ファイルが使用されている（またはバックアップが使用可能である）場合は、転送先ホストに一時初期化パラメータ・ファイルを作成し、DB\_NAME パラメータのみを設定できます。

ソース・データベース prod は host1 上にあり、データファイルは ASM ディスク・グループ +DISK1 に保存されていると想定しています。ターゲット・データベースを複製し、リモート・ホスト host2 上に dupdb データベースを作成します。ASM に dupdb のデータファイルを格納します。具体的には、ディスク・グループ +DISK2 にデータファイルおよび制御ファイルを保存します。

DUPLICATE コマンドに PARAMETER\_VALUE\_CONVERT を設定し、すべてのディレクトリの場所を +DISK1 から +DISK2 に変換します。+DISK2 の新しいファイル名は、ASM によって生成され、ディスク・グループ +DISK1 の元のファイル名とは一致しません。

ターゲット・データベース、複製データベースおよびカタログ・データベースに接続してから、例 23-7 に示す Recovery Manager スクリプトを実行してデータベースを複製します。

### 例 23-7 ASM から ASM への複製（アクティブ）

```
DUPLICATE TARGET DATABASE
  TO dupdb
  FROM ACTIVE DATABASE
  SPFILE PARAMETER_VALUE_CONVERT '+DISK1','+DISK2'
  SET DB_FILE_NAME_CONVERT '+DISK1','+DISK2'
  SET LOG_FILE_NAME_CONVERT '+DISK1','+DISK2';
```

DUPLICATE コマンドが完了すると、データファイル、オンライン REDO ログおよび制御ファイルとともに複製データベースが ASM ディスク・グループ +DISK2 に作成されます。

## 代替方法を使用した複製ファイルの名前の指定

この項では、複製ファイルの名前の指定に SPFILE 句の方法を使用しない場合、または他の名前指定方法で SPFILE 句の方法を補完する場合を想定しています。

次の項で、複製データベースでファイルの名前を指定する方法について説明します。

- [複製制御ファイルの名前の指定](#)
- [複製オンライン REDO ログ・ファイルのネーミング](#)
- [複製データファイルの名前の指定](#)
- [複製一時ファイルの名前の指定](#)

### 複製制御ファイルの名前の指定

複製データベースの制御ファイルの名前を指定する場合は、補助初期化パラメータ・ファイルで初期化パラメータを設定する必要があります。CONTROL\_FILES 初期化パラメータでファイル名を指定するか、または他のパラメータを使用して Oracle 管理ファイルの場所を指定することができます。

複製データベースの制御ファイルの名前を決定する場合の優先順位の規則は、SQL 文 CREATE CONTROLFILE で使用される規則と同じです。複製データベースの制御ファイルの名前を選択するときは、補助データベースの初期化パラメータ・ファイルにパラメータを正しく設定します。そうしない場合、ソース・データベースの制御ファイルが上書きされます。

**参照：**『Oracle Database SQL 言語リファレンス』の CREATE CONTROLFILE エントリのセマンティクスに関する項を参照してください。

### 複製オンライン REDO ログ・ファイルのネーミング

Recovery Manager では、複製データベースのオンライン REDO ログ・ファイルの新しい名前が必要です。DUPLICATE コマンドで明示的に名前を指定するか、または Recovery Manager によって名前を生成できます。

1. DUPLICATE コマンドの LOGFILE 句を指定します。

このオプションを指定すると、指定されたとおりに複製データベースにオンライン REDO ログが作成されます。

2. LOG\_FILE\_NAME\_CONVERT 初期化パラメータを設定してください。

このパラメータを設定すると、log\_\* から duplog\_\* などのように、ソース・データベース・ファイルの名前で文字列を置き換えることによって、複製ログ・ファイルの名前が作成されます。変換する組合せは複数指定できます。

Oracle 管理ファイルで LOG\_FILE\_NAME\_CONVERT を使用するの詳細は、23-15 ページの「Oracle 管理ファイルの初期化パラメータの設定」を参照してください。

Recovery Manager は、オンライン REDO ログの作成時に REUSE パラメータを使用します。指定された場所にすでにオンライン REDO ログ・ファイルが存在し、適切な大きさの場合は、そのオンライン REDO ログ・ファイルが複製のために再利用されます。

3. Oracle Managed Files 初期化パラメータ DB\_CREATE\_FILE\_DEST、DB\_CREATE\_ONLINE\_DEST\_n または DB\_RECOVERY\_FILE\_DEST のいずれかを設定します。

このパラメータを設定すると、ソース・データベース・ファイルの名前で文字列を置き換えることによって、複製ログ・ファイルの名前が作成されます。これらのパラメータの優先順位の規則は、SQL 文 ALTER DATABASE ADD LOGFILE で使用される規則と同じです。

4. 前述のいずれの手順も実行しません。

複製ファイル名がソース・データベース・ファイル名と同じになります。この方法を使用する場合は、NOFILENAMECHECK オプションを指定する必要があります。また、複製のオンライン REDO ログが複製元と競合しないように、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。

優先順位の高い規則は、優先順位の低い規則より優先されます。たとえば、LOGFILE 句および 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。

---

---

**注意：**

- ソース・データベースと複製データベースが同じホストに存在する場合、ソース・データベースが現在使用しているオンライン REDO ログの名前は使用しないでください。
  - 複製データベースが異なるホストに存在し、NOFILENAMECHECK を指定していない場合は、複製先ホストでデータベースによって現在使用されているオンライン REDO ログ・ファイルの名前は使用しないでください。
- 
- 

## 複製データファイルの名前の指定

いくつかの方法を使用して、複製データベースのデータファイルに新しい名前を指定できます。次に、それらの方法を優先順位の高い順に示します。

1. SET NEWNAME コマンドと DUPLICATE コマンドの両方が含まれている RUN ブロック内で、Recovery Manager のコマンド SET NEWNAME FOR DATAFILE を使用します。
2. Recovery Manager のコマンド CONFIGURE AUXNAME を使用して、既存のデータファイルに新しい名前を指定します。CONFIGURE AUXNAME コマンドは、DUPLICATE コマンドの前に実行します。
3. DUPLICATE ... DB\_FILE\_NAME\_CONVERT を実行し、SET NEWNAME または CONFIGURE AUXNAME によって名前が変更されないデータファイルの名前を変換する規則を指定します。

---

---

**注意：** DUPLICATE コマンドの DB\_FILE\_NAME\_CONVERT 句を使用して、ソース・データベース・インスタンスでは Oracle Managed Files (OMF) である複製インスタンスのファイルの新しい名前を制御することはできません。この制限の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

---

---

4. DB\_FILE\_NAME\_CONVERT 初期化パラメータを設定してください。

---

---

**注意：** DB\_FILE\_NAME\_CONVERT 初期化パラメータを使用して、ソース・データベース・インスタンスでは Oracle Managed Files (OMF) である複製インスタンスのファイルの新しい名前を制御することはできません。DUPLICATE コマンドの DB\_FILE\_NAME\_CONVERT パラメータと同じセマンティクスおよび制限に従います。詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

---

---

5. DB\_CREATE\_FILE\_DEST 初期化パラメータを設定して、指定した場所で Oracle Managed Files のデータファイルを作成します。

前述のいずれの手順も実行しない場合、複製データベースではソース・データベースの元のデータファイルの場所が再利用されます。

## 複製一時ファイルの名前の指定

Recovery Manager は、データベース複製プロセスの一部として一時表領域のデータファイルを再作成します。複製データベースの一時ファイルの場所を指定する方法は複数あります。次に、それらの方法を優先順位の高い順に示します。

1. SET NEWNAME コマンドと DUPLICATE コマンドの両方が含まれている RUN ブロック内で、SET NEWNAME FOR TEMPFILE コマンドを使用します。
2. DUPLICATE コマンドに DB\_FILE\_NAME\_CONVERT 句を指定して、SET NEWNAME によって名前が変更されない一時ファイルを変換する規則を指定します。

---

---

**注意：** DB\_FILE\_NAME\_CONVERT 句を使用して、ソース・データベース・インスタンスでは Oracle Managed Files (OMF) である複製インスタンスのファイルの新しい名前を制御することはできません。この制限の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

---

---

3. DB\_FILE\_NAME\_CONVERT 初期化パラメータを設定してください。

---

---

**注意：** DB\_FILE\_NAME\_CONVERT 初期化パラメータは、DUPLICATE コマンドの DB\_FILE\_NAME\_CONVERT パラメータと同じセマンティクスおよび制限に従います。詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

---

---

4. DB\_CREATE\_FILE\_DEST 初期化パラメータを設定して、Oracle Managed Files の一時ファイルを作成します。



## Recovery Manager による複製の例

この項では、Recovery Manager の DUPLICATE の代表的な使用例を示します。

- ソース・データベース表領域のサブセットの複製
- DUPLICATE を使用したアーカイブ・バックアップのリストア
- SET NEWNAME を使用した複製ファイルの名前の指定
- CONFIGURE AUXNAME を使用した複製ファイルの名前の指定

### ソース・データベース表領域のサブセットの複製

常に、データベースのすべての表領域を複製する必要があるわけではありません。たとえば、ソース・データベースから表領域のサブセットのみを必要とする、複製のレポートを生成する場合があります。DUPLICATE DATABASE コマンドには、読取り専用の表領域または現在 OFFLINE NORMAL である表領域をスキップできるオプションがあります。また、TABLESPACE 句を使用して、複製データベースに含める表領域を指定することもできます。この項の内容は、次のとおりです。

- 読取り専用表領域の除外
- OFFLINE NORMAL 表領域の除外
- 指定した表領域の追加および除外

#### 読取り専用表領域の除外

DUPLICATE コマンドに 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。例 23-8 は、例 23-3 を少し変更した例です。読取り専用表領域が除外される点が異なります。

##### 例 23-8 読取り専用表領域の除外

```
DUPLICATE TARGET DATABASE
  TO dupdb
  FROM ACTIVE DATABASE
  SKIP READONLY
  NOFILENAMECHECK;
```

#### OFFLINE NORMAL 表領域の除外

DUPLICATE 操作の前に、表領域が OFFLINE NORMAL オプションによってオフラインにされた場合、Recovery Manager はそれらの表領域のデータファイルの複製およびその複製データベースに対する DROP TABLESPACE の発行を行いません。したがって、特別な構文を指定してこれらの表領域を除外する必要はありません。複製後、すべてのデータファイルおよび表領域はオンラインになります。

---

**注意：** IMMEDIATE オプションを指定して表領域をオフラインにすると、この表領域のリカバリが必要になるため、Recovery Manager はこの表領域を複製します。バックアップベースの複製方法を使用する場合は、オンライン表領域のときと同様に、Recovery Manager にこれらの表領域の有効なバックアップが必要です。

---

#### 指定した表領域の追加および除外

SKIP TABLESPACE パラメータを指定すると、複製データベースから表領域を除外できます。SYSTEM 表領域、SYSAUX 表領域、UNDO 表領域およびロールバック・セグメントを含む表領域は除外できないことに注意してください。TABLESPACE パラメータを使用すると、指定したデータベースに含める必要がある表領域を指定することもできます。複製データベースから除外する必要がある表領域を指定する SKIP TABLESPACE とは異なり、このオプションを使用すると、含める必要がある表領域が指定された後、残りの表領域はスキップされます。



例 23-9 は、例 23-3 を少し変更した例です。tools 表領域が除外される点が異なります。

#### 例 23-9 指定した表領域の除外

```
DUPLICATE TARGET DATABASE
  TO dupdb
  FROM ACTIVE DATABASE
  SKIP TABLESPACE tools
  NOFILENAMECHECK;
```

例 23-10 は例 23-3 を少し変更した例です。users 表領域が含まれ、SYSTEM 表領域、SYSAUX 表領域および UNDO セグメントまたはロールバック・セグメントが含まれている表領域以外の他のすべての表領域が除外される点が異なります。

#### 例 23-10 指定した表領域の追加

```
DUPLICATE TARGET DATABASE
  TO dupdb
  FROM ACTIVE DATABASE
  TABLESPACE users
  NOFILENAMECHECK;
```

## DUPLICATE を使用したアーカイブ・バックアップのリストア

8-24 ページの「長期格納用のデータベース・バックアップの作成」の説明に従って、**アーカイブ・バックアップ**を行うことができます。アーカイブ・バックアップは、データベースのリストアおよびリカバリに必要なすべてのファイルが含まれているという点で包括的なバックアップです。テストのためにアーカイブ・バックアップをリストアする場合は、一時インスタンスを作成し、DUPLICATE コマンドを使用する方法をお勧めします。この方法を使用すると、本番システムに影響が及ぼされません。

Recovery Manager を起動し、ターゲット・データベースおよびカタログ・データベースに接続するとします。例 23-11 では、タグ TESTDB が含まれているアーカイブ・バックアップを一時ディスクに作成します。

#### 例 23-11 一時的なアーカイブ・バックアップの作成

```
BACKUP DATABASE
  FORMAT '/disk1/oraclebck/%U'
  TAG TESTDB
  KEEP UNTIL 'SYSDATE+1'
  RESTORE POINT TESTDB06;
```

例 23-11 では、リカバリ・カタログおよびバックアップ制御ファイルの両方に存在し、バックアップをリカバリする時点を示すラベルとなるリストア・ポイントを作成します。アーカイブ・ログは、オンライン・バックアップの場合にのみバックアップされることに注意してください。アーカイブ・ログは、オフライン・バックアップでは必要ないためバックアップされません。

例 23-11 で作成されたバックアップをリストアする手順は、23-12 ページの「データベースの複製」の手順と同様です。唯一の追加要件として、DUPLICATE コマンドで、アーカイブ・バックアップを使用して作成されたリストア・ポイントを指定する必要があります。

#### アーカイブ・バックアップをリストアする手順

1. Recovery Manager を、TARGET としてソース・データベースに、AUXILIARY として複製データベース・インスタンスに、およびリカバリ・カタログに接続します。
2. LIST RESTORE POINT を実行し、使用可能なリストア・ポイントを表示します（手順は、10-10 ページの「リストア・ポイントの表示」を参照）。

```
LIST RESTORE POINT;
```

3. データベースを複製する場合の DUPLICATE コマンド自体の発行の前までのすべての手順に従います。
4. DUPLICATE コマンドを実行し、リストアするアーカイブ・バックアップのリストア・ポイントを指定します。

次の例では、補助インスタンスを作成し、テスト・ホスト上で Recovery Manager を実行していると想定しています。UNTIL RESTORE POINT 句に TESTDB06 が指定されています。これは、例 23-11 で作成されたリストア・ポイントです。

```
DUPLICATE TARGET DATABASE
  TO mytest
  UNTIL RESTORE POINT TESTDB06
  DB_FILE_NAME_CONVERT '/prod/oracledb/', '/test/oracledb'
  PFILE 'test/oracledb/init.ora';
```

前述の DUPLICATE コマンドによって、データベース全体がリストアされ、データベースの名前が mytest に変更されます。DUPLICATE コマンドでは、元の制御ファイルはリストアされませんが、新しい制御ファイルが作成されます。したがって、リストア・ポイントが制御ファイル内に存在しているときに、Recovery Manager がカタログまたはソース・データベースに接続されている場合は、DUPLICATE ... UNTIL RESTORE POINT のみを指定できます。

## SET NEWNAME を使用した複製ファイルの名前の指定

この例では、バックアップベースの複製を使用して複製データベースを作成します。ソース・データベースでサーバー・パラメータ・ファイルが使用されないため、SPFILE の方法を使用して複製データファイルの名前を指定することはできません。複製データファイルがいくつかのディレクトリに分散されるため、SET NEWNAME コマンドを使用してファイル名を指定します。

ソース・データベース prod は host1 上にあり、複数のディレクトリに分散した 8 つのデータファイルが含まれているとします。ソース・データベースを複製し、リモート・ホスト host2 上に dupdb データベースを作成します。dupdb データベースでは、表領域 tools が除外され、その他のすべての表領域が保持されます。

この例では、host1 と host2 のディレクトリ構造が異なります。host2 のデータファイルを /oradata1 から /oradata7 サブディレクトリに格納します。ソース・データベースには 8 つのデータファイルが存在しますが、tools 表領域は除外するため、ターゲット・データファイルに指定する必要があるのは 7 つの場所のみです。転送先ホスト上のディレクトリ /duplogs に 2 つのオンライン REDO ログ・グループを作成し、各グループに 200KB のメンバーを 2 つ含めます。host1 および host2 は、NFS などの方法を使用して相互のファイル・システムにマウントできないとします。

ソース・データベースのすべてのデータファイルおよびアーカイブ REDO ログのディスク・コピーまたはバックアップ・セットがディスク上に存在しており、オペレーティング・システム・ユーティリティを使用して、それらを host2 に手動でコピー済です。これらのバックアップおよびコピーは、host2 上の、host1 上と同じ場所に存在します。

オペレーティング・システム・ユーティリティを使用して、初期化パラメータ・ファイルを host1 から host2 の適切な場所にコピー済です。末尾に \_DEST が付いていて、パス名を指定するすべての初期化パラメータを再設定済である。RUN コマンド自体でデータファイルおよびオンライン・ログの名前を指定するため、DB\_FILE\_NAME\_CONVERT および LOG\_FILE\_NAME\_CONVERT は設定しません。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の初期化パラメータ・ファイルを使用します。そのため、DUPLICATE コマンドに PFILE パラメータを指定する必要はありません。

例 23-12 に、複製データベースを作成するスクリプトを示します。SET NEWNAME は RUN 内でのみ実行できるため、RUN コマンドが必要であることを注意してください。データファイル 7 は複製データベースから除外する tools 表領域にあるため、このデータファイルには新しい名前を設定しません。

**例 23-12 SET NEWNAME を使用した複製**

```

RUN
{
SET NEWNAME FOR DATAFILE 1 TO '/oradata1/system01.dbf';
SET NEWNAME FOR DATAFILE 2 TO '/oradata2/undotbs01.dbf';
SET NEWNAME FOR DATAFILE 3 TO '/oradata3/cwmlite01.dbf';
SET NEWNAME FOR DATAFILE 4 TO '/oradata4/drsys01';
SET NEWNAME FOR DATAFILE 5 TO '/oradata5/example01.dbf';
SET NEWNAME FOR DATAFILE 6 TO '/oradata6/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 '/oradata7/users01.dbf';
DUPLICATE TARGET DATABASE TO dupdb
  SKIP TABLESPACE tools
  LOGFILE
    GROUP 1 ('/duplogs/redo01a.log',
             '/duplogs/redo01b.log') SIZE 4M REUSE,
    GROUP 2 ('/duplogs/redo02a.log',
             '/duplogs/redo02b.log') SIZE 4M REUSE;
}

```

**SET NEWNAME を使用した Oracle 管理ファイルの名前の指定**

残りのデータベース・ファイルの場所とは別に、Oracle 管理ファイルの複製先に特定のデータファイルまたは一時ファイルを格納する場合は、次の手順を実行します。

- 補助インスタンスの DB\_CREATE\_FILE\_DEST 初期化パラメータを任意の場所に設定します。
- RUN ブロックに DUPLICATE コマンドを含めて、SET NEWNAME FOR DATAFILE ... TO NEW および SET NEWNAME FOR TEMPFILE ... TO NEW を使用します。

指定したデータファイルまたは一時ファイルは、DB\_CREATE\_FILE\_DEST で指定された場所に Oracle 管理ファイルの名前で作成されます。

例 23-13 に示すように、SET NEWNAME を使用して、個々のデータファイルまたは一時ファイルを特定の ASM ディスク・グループに格納することもできます。

**例 23-13 SET NEWNAME を使用した ASM ディスク・グループでのファイルの作成**

```

RUN
{
SET NEWNAME FOR DATAFILE 1 TO "+dgroup1";
SET NEWNAME FOR DATAFILE 2 TO "+dgroup2";
.
.
.
DUPLICATE TARGET DATABASE
  TO dupdb
  FROM ACTIVE DATABASE
  SPFILE SET DB_CREATE_FILE_DEST +dgroup3;
}

```

**参照：** SET NEWNAME の使用方法の詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## CONFIGURE AUXNAME を使用した複製ファイルの名前の指定

この項では、23-22 ページの「SET NEWNAME を使用した複製ファイルの名前の指定」で説明されている環境と同じ環境を想定しています。例 23-12 は例 23-14 を少し変更した例です。CONFIGURE AUXNAME を使用して新しいデータファイル名を指定します。これらの新しいファイル名は制御ファイルに記録され、将来のすべての複製で使用されます。

また、例 23-14 では、自動チャネルおよび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データベースを複製し、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。この場合、SET NEWNAME を使用していないため、RUN コマンドは不要です。

### 例 23-14 CONFIGURE AUXNAME を使用したデータベースのファイル名の生成

```
CONFIGURE AUXNAME FOR DATAFILE 1 TO '/oradata1/system01.dbf';
CONFIGURE AUXNAME FOR DATAFILE 2 TO '/oradata2/undotbs01.dbf';
CONFIGURE AUXNAME FOR DATAFILE 3 TO '/oradata3/cwmlite01.dbf';
CONFIGURE AUXNAME FOR DATAFILE 4 TO '/oradata4/drsys01';
CONFIGURE AUXNAME FOR DATAFILE 5 TO '/oradata5/example01.dbf';
CONFIGURE AUXNAME FOR DATAFILE 6 TO '/oradata6/indx01.dbf';
```

```
DUPLICATE TARGET DATABASE
  TO dupdb
  SKIP TABLESPACE tools
  LOGFILE
    GROUP 1 ('/duplogs/redo01a.log',
             '/duplogs/redo01b.log') SIZE 4M REUSE,
    GROUP 2 ('/duplogs/redo02a.log',
             '/duplogs/redo02b.log') SIZE 4M REUSE;
```

Recovery Manager は、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、RESETLOGS オプションを指定して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。

複製の完了後、データファイルが以降の操作によって上書きされないように、複製データベース内のデータファイル用に設定した補助名を消去します。たとえば、次のコマンドを入力します。

```
CONFIGURE AUXNAME FOR DATAFILE 1 CLEAR;
CONFIGURE AUXNAME FOR DATAFILE 2 CLEAR;
CONFIGURE AUXNAME FOR DATAFILE 3 CLEAR;
CONFIGURE AUXNAME FOR DATAFILE 4 CLEAR;
CONFIGURE AUXNAME FOR DATAFILE 5 CLEAR;
CONFIGURE AUXNAME FOR DATAFILE 6 CLEAR;
```

かわりに、複製元のデータベースと複製データベースを定期的に同期化する必要がある場合があります。この場合は、DUPLICATE コマンドを再度実行して、基本的に複製データベースを再作成します。この方法では、複製データベースのデータファイルの完全コピーを作成する必要があります。複製データベースとソース・データベースを同期化する場合は、常に例 23-15 に示すスクリプトを実行します。たとえば、このスクリプトは、毎日または毎週実行できます。

### 例 23-15 データベースの複製

```
DUPLICATE TARGET DATABASE TO dupdb
  SKIP TABLESPACE tools
  LOGFILE
    GROUP 1 ('/duplogs/redo01a.log',
             '/duplogs/redo01b.log') SIZE 4M REUSE,
    GROUP 2 ('/duplogs/redo02a.log',
             '/duplogs/redo02b.log') SIZE 4M REUSE;
```

---

## トランスポートブル表領域セットの作成

この章では、Recovery Manager を使用し、バックアップをリストアしてトランスポートブル表領域セットを作成する方法について説明します。ここでは、『Oracle Database 管理者ガイド』で説明する [トランスポートブル表領域](#) での手順に習熟していることを前提としています。この章で説明する手順は、トランスポートブル表領域セットの生成に使用する代替方法です。

この章では、次の項目について説明します。

- [トランスポートブル表領域セットの作成の概要](#)
- [補助インスタンスの初期化パラメータのカスタマイズ](#)
- [トランスポートブル表領域セットの作成](#)
- [トランスポートブル表領域セットの作成のトラブルシューティング](#)
- [トランスポートブル表領域セットの使用例](#)

## トランスポートブル表領域セットの作成の概要

この項では、Recovery Manager のバックアップからトランスポートブル表領域セットを作成する場合の基本的な概念およびタスクについて説明します。

### トランスポートブル表領域セットの作成の目的

**トランスポートブル表領域セット**には、表領域セットのデータファイル、および表領域セットの構造メタデータを含むエクスポート・ファイルが含まれています。エクスポート・ファイルは、データ・ポンプ・エクスポートによって生成されます。

トランスポートブル表領域セットを使用する例としては、表領域リポジトリの作成があげられます。たとえば、四半期レポートに使用する複数の表領域を含むデータベースがある場合、これらの表領域のトランスポートブル・セットを表領域リポジトリのストレージに作成できます。その後、表領域のバージョンをリポジトリから要求して別のデータベースに組み込み、レポートの生成に使用することができます。

トランスポートブル表領域を使用するもう 1 つの例としては、Oracle Streams 環境での使用があげられます。Oracle Streams を使用して、転送先データベースとソース・データベースを継続して同期化するための準備をする場合には、Oracle Streams のインスタンス化を実行する必要があります。Oracle Streams を実際に使用して後続の更新をソース・データベースから転送先データベースに移動する前に、2 つのデータベースが同期化されたことがわかっている時点の SCN まで転送先データベースを戻す必要があります。バックアップからのトランスポートブル表領域セットは、Oracle Streams のインスタンス化の一部として作成できます。

Recovery Manager の TRANSPORT TABLESPACE コマンドの主なメリットは、転送する表領域のアクティブなデータファイルにアクセスする必要がないことです。これに対して、『Oracle Database 管理者ガイド』で説明されているトランスポートブル表領域の方法では、転送時に、転送する表領域を読取り専用でオープンする必要があります。したがって、バックアップから転送すると、転送中、転送する表領域を書込み用にオープンしたままにできるため、特に大規模な表領域の場合にデータベースの可用性が向上します。また、現行のデータベース・アクティビティによっては、表領域を読取り専用モードにする操作に時間が長くなる場合もあります。

Recovery Manager の TRANSPORT TABLESPACE コマンドを使用すると、リカバリ期間内に目標時点、SCN またはリストア・ポイントを指定し、表領域データを指定した時点の状態で転送することもできます (24-9 ページの「[指定した時点または SCN でのトランスポートブル表領域セットの作成](#)」を参照)。たとえば、[バックアップの保存方針](#)で 1 週間のリカバリ期間を保証しており、月の最終日のデータベースの内容に基づいてトランスポートブル表領域を作成する場合、Recovery Manager は、翌月の第 1 週目の任意の時点にそのタスクを実行できます。

---

---

#### 参照：

- TRANSPORT TABLESPACE コマンドの詳細は、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。
  - Recovery Manager および表領域リポジトリの詳細は、『Oracle Streams レプリケーション管理者ガイド』を参照してください。
  - Recovery Manager および Oracle Streams のインスタンス化の詳細は、『Oracle Streams レプリケーション管理者ガイド』を参照してください。
- 
-

## トランスポートブル表領域セットの基本的な概念

Recovery Manager をソース・データベースに TARGET として接続してから、TRANSPORT TABLESPACE コマンドを実行して、トランスポートブル表領域セットを作成します。ソース・データベースには、転送される表領域が含まれています。

Recovery Manager では、TRANSPORT TABLESPACE 操作の目標時点までリカバリ可能な、必要となるすべての表領域のバックアップおよびアーカイブ REDO ログ・ファイルが使用可能になっている必要があります。図 24-1 に、トランスポートブル表領域の基本プロセスを示します。

図 24-1 バックアップからの Recovery Manager トランスポートブル表領域：アーキテクチャ

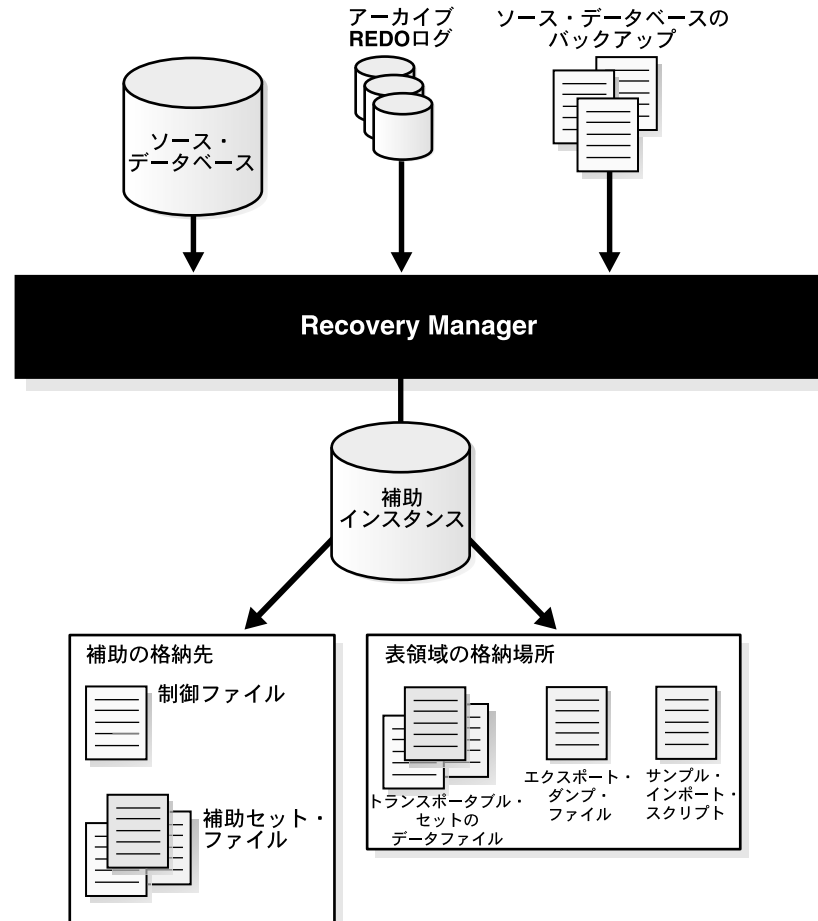


図 24-1 に示されているプロセスは、次のフェーズで実行されます。

1. Recovery Manager によって、**補助インスタンス**が起動されます。

Recovery Manager によって、表領域のリストアおよびリカバリを実行するために、ソース・データベースと同じホスト上に補助インスタンスが作成されます。また、補助インスタンスの初期化パラメータ・ファイルが自動的に作成され、インスタンスが NOMOUNT モードで起動されます。

2. Recovery Manager によって、補助インスタンスの制御ファイルとして機能するソース・データベースの制御ファイルのバックアップがリストアされ、その制御ファイルがマウントされます。
3. Recovery Manager によって、ソース・データベースのバックアップから、**補助セット**およびトランスポートブル・セットのデータファイルがリストアされます。



**補助セット**には、表領域の転送に必要なデータファイルおよびその他のファイルが含まれていますが、それらのファイル自体はトランスポータブル表領域セットの一部ではありません。通常、補助セットには、SYSTEM 表領域、SYSAUX 表領域、一時ファイル、およびロールバック・セグメントまたは UNDO セグメントを含むデータファイルが含まれています。補助インスタンスには、それ自体の制御ファイル、パラメータ・ファイル、オンライン・ログなどの他のファイルが関連付けられますが、これらのファイルは補助セットの一部ではありません。

Recovery Manager によって、選択した**補助の格納場所**に補助データファイルが格納されます。補助の格納場所は、転送実行中に、補助インスタンスのパラメータ・ファイル、データファイル（トランスポータブル・セットのデータファイル以外）、制御ファイル、オンライン・ログなどの補助セット・ファイルを Recovery Manager によって格納できるディスク上の場所です。転送が正常に実行されると、これらのファイルは Recovery Manager によって削除されます。

Recovery Manager によって、**表領域の格納場所**にトランスポータブル・セット・ファイルが格納されます。表領域の格納場所は、表領域の転送コマンドの完了時に、デフォルトでデータファイルのコピーおよび他の出力が格納されるディスク上の場所です。

4. Recovery Manager は、補助インスタンスで**データベースの Point-in-Time リカバリ**を実行します。

このリカバリによって、補助セットおよびトランスポータブル・セットのデータファイルの内容が、TRANSPORT DATABASE コマンドに対して指定された目標時点の状態に更新されます。目標時点が指定されていない場合、Recovery Manager では、すべての使用可能な REDO ログを使用してリカバリが行われます。必要に応じて、バックアップからアーカイブ REDO ログが補助の格納場所（またはその他の場所）にリストアされ、適用後に削除されます。

5. Recovery Manager によって、補助データベースが RESETLOGS オプションを使用してオープンされます。

これにより、データファイルに、表領域の転送操作の目的の SCN 時点の表領域内容が反映されます。

6. Recovery Manager によって、補助インスタンスのトランスポータブル・セットの表領域が読取り専用モードに設定されます。また、データ・ポンプ・エクスポートがトランスポータブル表領域モードで起動され、トランスポータブル・セットの**エクスポート・ダンプ・ファイル**が作成されます。

デフォルトでは、ダンプ・ファイルは表領域の格納場所に格納されます。ダンプ・ファイルの場所を指定するには、24-10 ページの「**データ・ポンプ・ファイルの場所の指定**」を参照してください。

また、Recovery Manager によって、転送した表領域をターゲット・データベースに組み込む場合に使用するサンプル・データ・ポンプ・インポート・スクリプトも生成されます。このスクリプトの内容は、表領域の格納場所の `impscript.sql` という名前のファイルに書き込まれます。このスクリプト用のコマンドも、Recovery Manager のコマンド出力に含まれます。

7. 前述の手順が正常に実行されると、Recovery Manager によって、補助インスタンスが停止され、TRANSPORT TABLESPACE の実行中に作成されたすべてのファイル（トランスポータブル・セット・ファイル、データ・ポンプ・エクスポート・ファイルおよびサンプル・インポート・スクリプトを除く）が削除されます。



## トランスポータブル表領域セットの作成の基本手順

トランスポータブル表領域セットを作成する前に、多くの前提条件を満たす必要があります。これらの前提条件については、『Oracle Database バックアップおよびリカバリ・リファレンス』の TRANSPORT TABLESPACE エントリを参照してください。

基本手順は次のとおりです。

1. Recovery Manager クライアントを起動し、ソース・データベースおよびリカバリ・カタログ（使用する場合）に接続します。
2. 必要に応じて、補助インスタンスのパラメータ・ファイルで追加のパラメータを設定します。  
このタスクについては、24-5 ページの「補助インスタンスの初期化パラメータのカスタマイズ」を参照してください。
3. TRANSPORT TABLESPACE コマンドを実行します。  
この方法については、24-7 ページの「トランスポータブル表領域セットの作成」を参照してください。この方法については、24-9 ページの「トランスポータブル表領域セットの使用例」を参照してください。
4. TRANSPORT TABLESPACE コマンドが失敗した場合は、問題のトラブルシューティングを行い、コマンドが正常に実行されるまで再試行します。  
この方法については、24-8 ページの「トランスポータブル表領域セットの作成のトラブルシューティング」を参照してください。
5. 表領域の転送手順（『Oracle Database 管理者ガイド』を参照）に戻ります。

## 補助インスタンスの初期化パラメータのカスタマイズ

Recovery Manager は、補助インスタンスの作成時に、初期化パラメータ・ファイルを作成します。デフォルト値は、ほぼすべての TRANSPORT TABLESPACE の例（特に TRANSPORT TABLESPACE に AUXILIARY DESTINATION を指定する場合）で正常に動作します。

また、Recovery Manager は、追加の初期化パラメータの値を含む補助インスタンスのパラメータ・ファイルも使用できます。これらの値によって、デフォルトの初期化パラメータ・ファイルで定義されているパラメータの値が上書きされます。次の理由から、補助インスタンスのパラメータ・ファイルを使用する場合があります。

- STREAMS\_POOL\_SIZE および SHARED\_POOL\_SIZE を増加するため（データ・ポンプ・エクスポートが必要な場合）。
- 補助インスタンスのデータファイルの場所を管理するため（24-12 ページの「初期化パラメータを使用した補助ファイル名の指定」を参照）。たとえば、すべてのファイルの場所を個別に指定せずに、補助インスタンスのすべてのデータファイルが同一の場所に格納されないようにする場合などです。
- LOG\_FILE\_NAME\_CONVERT を使用してオンライン REDO ログの名前を指定するため（24-12 ページの「初期化パラメータを使用した補助ファイル名の指定」を参照）。

補助インスタンスのパラメータ・ファイルは、補助インスタンスの完全な初期化パラメータ・ファイルとはみなされません。指定したパラメータは、補助インスタンスのデフォルト・パラメータに追加されるか、これらのパラメータを上書きします。上書きしない初期化ファイルに、パラメータを指定する必要はありません。

## 補助インスタンスの初期化パラメータの設定

Recovery Manager は、自動補助インスタンスに対して表 24-1 の基本初期化パラメータを定義します。

表 24-1 デフォルトの初期化パラメータ

| 初期化パラメータ            | 値   |
|---------------------|---|
| DB_NAME             | ソース・データベースの DB_NAME と同じです。  |
| COMPATIBLE          | ソース・データベースの互換性設定と同じです。  |
| DB_UNIQUE_NAME      | DB_NAME の値に基づいて生成された一意の値。   |
| DB_BLOCK_SIZE       | ソース・データベースの DB_BLOCK_SIZE と同じです。  |
| DB_FILES            | ソース・データベースの DB_FILES と同じです。   |
| SHARED_POOL_SIZE    | 110MB。データ・ポンプ・エクスポートに必要な最小領域サイズ。<br>共有プールのデフォルトのサイズが、データ・ポンプ・エクスポートの正常な実行に十分でない場合は、より大きい値を指定します。<br><b>参照:</b> SHARED_POOL_SIZE 初期化パラメータの使用法の詳細は、『Oracle Database リファレンス』を参照してください。   |
| LARGE_POOL_SIZE     | 1MB   |
| DB_CREATE_FILE_DEST | 補助の格納場所 (TRANSPORT TABLESPACE に AUXILIARY DESTINATION 引数が設定されている場合のみ)。Recovery Manager は、Oracle 管理の制御ファイルおよびオンライン・ログをこの場所に作成します。  |
| CONTROL_FILES       | 補助の格納場所で生成されたファイル名 (TRANSPORT TABLESPACE に AUXILIARY DESTINATION 引数が設定されている場合のみ)。Recovery Manager は、制御ファイルをこの場所に作成します。<br><br>デフォルトでは、Recovery Manager は、補助インスタンスに対して 1 つの制御ファイルをオペレーティング・システム固有の場所に作成します。Linux および UNIX の場合、デフォルトの場所は <code>?/dbs/cntrl_@.dbf</code> になります。ここで、? は ORACLE_HOME、@ は ORACLE_SID を表します。自動補助インスタンスの場合、ORACLE_SID は Recovery Manager によってランダムに生成されます。<br><b>参照:</b> CONTROL_FILES 初期化パラメータの使用法の詳細は、『Oracle Database リファレンス』を参照してください。 |

表 24-1 に示されている、補助インスタンスのパラメータ・ファイル内の基本の初期化パラメータを不適切な値で書き出すと、TRANSPORT TABLESPACE が失敗する場合があります。問題が発生する場合は、初期化パラメータをデフォルト値に戻してください。

**参照:** ファイルの名前を指定するために DB\_FILE\_NAME\_CONVERT および LOG\_FILE\_NAME\_CONVERT を使用する方法については、24-12 ページの「初期化パラメータを使用した補助ファイル名の指定」を参照してください。

## 補助インスタンスのパラメータ・ファイルの場所の設定

デフォルトでは、Recovery Manager は、Recovery Manager クライアントが実行されているホスト上のオペレーティング・システム依存の場所で、補助の初期化パラメータ・ファイルを検索します。この場所は、補助インスタンスが実行されているホストではない可能性があることに注意してください。UNIX の場合、この場所は `*/rdbms/admin/params_auxint.ora` です。ここで、クエスチョン・マーク (?) は、Recovery Manager が実行されるホスト上の `ORACLE_HOME` を表します。デフォルトの場所でファイルが検出されない場合でも、Recovery Manager はエラーを生成しません。

補助インスタンスのデフォルトの初期化パラメータを使用する場合は、`TRANSPORT TABLESPACE` を実行する前に、補助インスタンスのパラメータ・ファイルが存在するかどうかを確認してください。

Recovery Manager で `TRANSPORT TABLESPACE` の前に `RUN` ブロック内の `SET AUXILIARY INSTANCE PARAMETER FILE` コマンドを使用すると、補助インスタンスのパラメータ・ファイルに別の場所を指定することができます。補助インスタンスのパラメータ・ファイルのデフォルトの場所と同様に、`SET AUXILIARY INSTANCE PARAMETER FILE` を使用して指定したパスは、クライアント側のパスです。

Recovery Manager クライアントが実行されているホスト上に、`/tmp/auxinstparams.ora` というファイルを作成するとします。このファイルには、次の初期化パラメータが含まれています。

```
SHARED_POOL_SIZE=150M;
```

例 24-1 に示すように、`TRANSPORT TABLESPACE` を指定して初期化パラメータ・ファイルを使用できます。Recovery Manager が補助インスタンスを作成する際に、`/tmp/auxinstparams.ora` の `SHARED_POOL_SIZE` パラメータによって、`SHARED_POOL_SIZE` で使用されるデフォルト値が上書きされます。

### 例 24-1 補助インスタンスのパラメータ・ファイルの指定

```
RUN
{
  SET AUXILIARY INSTANCE PARAMETER FILE TO '/tmp/auxinstparams.ora';
  TRANSPORT TABLESPACE tbs_2
    TABLESPACE DESTINATION '/disk1/transportdest'
    AUXILIARY DESTINATION '/disk1/auxdest';
}
```

## トランスポートブル表領域セットの作成

この項では、最大限に自動化された最も基本的な場合の `TRANSPORT TABLESPACE` の使用について説明します。基本例を少し変更した例については、24-9 ページの「[トランスポートブル表領域セットの使用例](#)」を参照してください。

『Oracle Database バックアップおよびリカバリ・リファレンス』の `TRANSPORT TABLESPACE` エントリに関する項で説明されている前提条件を満たしていると想定しています。また、『Oracle Database 管理者ガイド』で説明されている前提条件も満たしていると想定しています。

- ソース・プラットフォームと転送先プラットフォーム間で、表領域の転送がサポートされていることを確認します。
- トランスポートブル・セットに含める自己完結型の表領域セットを識別します。

### トランスポートブル表領域セットを作成する手順

1. Recovery Manager クライアントを起動し、ソース・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。

2. Recovery Manager で、TRANSPORT TABLESPACE コマンドを実行します。

最も基本的な場合では、AUXILIARY DESTINATION 句を指定します。これはオプションですが、指定することをお勧めします。Recovery Manager は、ほとんどの場合で機能するデフォルト値を使用します。補助の格納場所を指定しない場合は、補助インスタンスのすべてのファイルに対して場所が指定されていることを確認します。補助ファイルに名前を指定する方法については、24-11 ページの「[補助ファイルの場所の指定](#)」で説明されている規則を参照してください。

**例 24-2** では、表領域 tbs\_2 および tbs\_3 を含むトランスポータブル表領域セットを作成します。

#### 例 24-2 トランスポータブル表領域セットの作成

```
TRANSPORT TABLESPACE tbs_2, tbs_3
  TABLESPACE DESTINATION '/disk1/transportdest'
  AUXILIARY DESTINATION '/disk1/auxdest';
```

コマンドが正常に実行されると、結果は次のようになります。

- トランスポータブル・セットのデータファイルが、元の名前で /disk1/transportdest という場所に格納されます。TRANSPORT TABLESPACE では、トランスポータブル表領域セットのデータファイルは、転送先データベースのエンディアン形式に自動的に変換されません。必要に応じて、トランスポータブル・セットを作成した後に Recovery Manager の CONVERT コマンドを使用して、データファイルを転送先データベースのエンディアン形式に変換します。
  - トランスポータブル・セットのデータ・ポンプ・エクスポート・ダンプ・ファイルの名前は dmpfile.dmp、エクスポート・ログの名前は explog.log、サンプル・インポート・スクリプトの名前は impscript.sql になります。  
  
すべてのファイルは、/disk1/transportdest 内に作成されます。表領域の格納場所にエクスポート・ダンプ・ファイルの名前と同じ名前のファイルがすでに存在する場合は、TRANSPORT TABLESPACE でデータ・ポンプ・エクスポートをコールすると失敗します。前の TRANSPORT TABLESPACE 操作を繰り返す場合には、以前の出力ファイル（エクスポート・ダンプ・ファイルを含む）を削除してください。
  - 補助セット・ファイルが /disk1/auxdest から削除されます。
3. 必要に応じて、サンプル・インポート・スクリプトを編集します。  
  
サンプル・インポート・スクリプトでは、表領域を転送先データベースにインポートするために使用するファイルは、TRANSPORT TABLESPACE で作成された場所と同じ場所に格納されていると想定されます。組込み前にファイルがディスク内の新しい場所に移動している場合、スクリプトを使用して転送した表領域を組み込む前に、ファイルの新しい場所を使用してサンプル・スクリプトを更新する必要があります。
  4. 表領域の転送手順（『Oracle Database 管理者ガイド』を参照）に戻ります。

## トランスポータブル表領域セットの作成のトラブルシューティング

Recovery Manager の TRANSPORT TABLESPACE コマンドが失敗すると、トラブルシューティングのために、失敗した補助インスタンスのファイルがその補助インスタンスの転送先にそのまま残ります。

共有プールのサイズに関連するエラーが表示された場合は、補助インスタンスのパラメータ・ファイルを作成し、STREAMS\_POOL\_SIZE および SHARED\_POOL\_SIZE に大きい値を指定します。この方法については、24-6 ページの「[補助インスタンスの初期化パラメータの設定](#)」を参照してください。

SET NEWNAME、CONFIGURE AUXNAME および DB\_FILE\_NAME\_CONVERT を使用したため、補助セット内またはトランスポートブル表領域セット内の複数のファイルに同じ名前が付けられた場合は、TRANSPORT TABLESPACE コマンドの実行中にエラーがレポートされます。この問題を解決するには、これらのパラメータに異なる値を使用して、重複するファイル名が作成されないようにします。ネーミング方法については、24-11 ページの「[補助ファイルの場所の指定](#)」を参照してください。

## トランスポートブル表領域セットの使用例

この項の内容は、次のとおりです。

- [指定した時点または SCN でのトランスポートブル表領域セットの作成](#)
- [データ・ポンプ・ファイルの場所の指定](#)
- [補助ファイルの場所の指定](#)

### 指定した時点または SCN でのトランスポートブル表領域セットの作成

TRANSPORT TABLESPACE コマンドを使用すると、目標時点または SCN を指定できます。表領域の転送操作中、Recovery Manager は、目標時点より前のバックアップを使用して補助インスタンスで表領域をリストアし、補助データベースで Point-in-Time リカバリを実行して指定の目標時点にリカバリします。この Point-in-Time リカバリに必要なバックアップおよびアーカイブ REDO ログが使用可能である必要があります。

[例 24-3](#) に示すように、(現行のインカネーションまたはその祖先内の) SCN を使用して目標時点を指定できます。

#### 例 24-3 終了 SCN の指定

```
TRANSPORT TABLESPACE tbs_2
  TABLESPACE DESTINATION '/disk1/transportdest'
  AUXILIARY DESTINATION '/disk1/auxdest'
  UNTIL SCN 11379;
```

[例 24-4](#) に示すように、リストア・ポイントを指定することもできます。

#### 例 24-4 終了リストア・ポイントの指定

```
TRANSPORT TABLESPACE tbs_2
  TABLESPACE DESTINATION '/disk1/transportdest'
  AUXILIARY DESTINATION '/disk1/auxdest'
  UNTIL RESTORE POINT 'before_upgrade';
```

[例 24-5](#) に示すように、終了時刻を指定することもできます。

#### 例 24-5 終了時刻の指定

```
TRANSPORT TABLESPACE tbs_2
  TABLESPACE DESTINATION '/disk1/transportdest'
  AUXILIARY DESTINATION '/disk1/auxdest'
  UNTIL TIME 'SYSDATE-1';
```

## データ・ポンプ・ファイルの場所の指定

トランスポートابل・セットのデータ・ポンプ・エクスポート・ダンプ・ファイル、ターゲット・データベースで使用するサンプル・インポート・スクリプト、データ・ポンプ・エクスポートで生成されるログ・ファイル、およびこれらのファイルの書き込み先となるディレクトリの名前は変更できます。

デフォルトでは、これらのファイルは、表領域の格納場所に格納され、次のような名前になります。

- データ・ポンプ・エクスポート・ダンプ・ファイルは、`dmpfile.dmp` という名前になります。
- エクスポート・ログ・ファイルは、`explog.log` という名前になります。
- サンプル・インポート・スクリプトは、`impscript.sql` という名前になります。

`DATAPUMP DIRECTORY` 句を使用してデータベース・ディレクトリ・オブジェクトの名前で渡すことによって、ダンプ・ファイルとエクスポート・ログ・ファイルを異なるディレクトリに格納できます。`DATAPUMP DIRECTORY` で使用するデータベース・ディレクトリ・オブジェクトは、実際のファイル・システム・ディレクトリのディレクトリ・パスではないことに注意してください。渡される値は、データ・ポンプ・エクスポートの `DIRECTORY` コマンドライン引数に対応しています。データ・ポンプ・エクスポートでのディレクトリ・オブジェクトの使用方法の詳細は、『Oracle Database ユーティリティ』を参照してください。

これらのファイルの名前は、`TRANSPORT TABLESPACE` の `DUMP FILE`、`EXPORT LOG`、および `IMPORT SCRIPT` 句を使用して変更できます。ファイル名には、ディレクトリ名を使用した完全なファイル・パスを含めることができません。`DUMP FILE` または `EXPORT LOG` というファイル名によってファイル・パスを指定した場合、`TRANSPORT TABLESPACE` でエクスポート・ダンプ・ファイルを生成しようとする失敗します。`DATAPUMP DIRECTORY` 句を使用して、データ・ポンプ・エクスポートの出力場所を識別するデータベース・ディレクトリ・オブジェクトを指定します。

次の例では、`DATAPUMP DIRECTORY`、`DUMP FILE`、`EXPORT LOG` および `IMPORT SCRIPT` というファイル名を指定した `TRANSPORT TABLESPACE` を使用します。次のように入力して、データ・ポンプ・エクスポートで使用するためにデータベース・ディレクトリ・オブジェクトを作成するとします。

```
CREATE OR REPLACE DIRECTORY mypumpdir as '/datapumpdest';
```

例 24-6 に、ファイルの出力場所を指定するオプションの引数を使用した `TRANSPORT TABLESPACE` コマンドを示します。

### 例 24-6 ファイルの出力場所の指定

```
TRANSPORT TABLESPACE tbs_2
  TABLESPACE DESTINATION '/transportdest'
  AUXILIARY DESTINATION '/auxdest'
  DATAPUMP DIRECTORY mypumpdir
  DUMP FILE 'mydumpfile.dmp'
  IMPORT SCRIPT 'myimportscript.sql'
  EXPORT LOG 'myexportlog.log';
```

正常に実行されると、Recovery Manager は、補助の格納場所をクリーンアップし、`DATAPUMP DIRECTORY (/datapumpdest/mydumpfile.dmp` および `/datapumpdest/myexportlog.log)` によって参照されるディレクトリにデータ・ポンプ・エクスポート・ダンプ・ファイルおよびエクスポート・ログを作成し、トランスポートابل・セットのデータファイルを `/transportdest` に格納します。



## 補助ファイルの場所の指定

いくつかの規則が、転送実行中に作成される補助インスタンス・ファイルの場所に影響します。最も簡単な方法としては、AUXILIARY DESTINATION を使用して、Recovery Manager で自動的にすべてのファイルの場所を管理する方法があげられます。補助インスタンスの一部またはすべてのファイルを再配置するために、ファイルの場所を指定する次のオプションを優先順位の高い順に示します。

### 1. SET NEWNAME

24-11 ページの「[補助データファイルに対する SET NEWNAME の使用](#)」の説明に従ってこのコマンドを使用して、データファイルの名前を指定することができます。

### 2. CONFIGURE AUXNAME

24-11 ページの「[補助データファイルに対する CONFIGURE AUXNAME の使用](#)」の説明に従ってこのコマンドを使用して、データファイルの名前を指定することができます。

### 3. AUXILIARY DESTINATION

24-12 ページの「[AUXILIARY DESTINATION を使用した補助ファイルの場所の指定](#)」の説明に従ってこのコマンドを使用して、補助ファイルの場所を指定することができます。

### 4. LOG\_FILE\_NAME\_CONVERT および DB\_FILE\_NAME\_CONVERT (初期化パラメータ・ファイル内)

24-12 ページの「[初期化パラメータを使用した補助ファイル名の指定](#)」の説明に従ってこのコマンドを使用して、補助ファイルの場所を指定することができます。

複数の規則を使用する場合は、ファイルに適用されるリスト内の最初の規則によってファイル名が決定されます。

## 補助データファイルに対する SET NEWNAME の使用

RUN ブロック内で SET NEWNAME コマンドを使用すると、TRANSPORT TABLESPACE で使用するファイル名を指定できます。例 24-7 に示す SET NEWNAME コマンドを使用すると、これらの補助インスタンスのデータファイルは /disk1/auxdest ではなく、指定した場所にリストアされます。

### 例 24-7 SET NEWNAME を使用した補助データファイル名の指定

```
RUN
{
  SET NEWNAME FOR DATAFILE '/oracle/dbs/tbs_12.f'
  TO '/bigdrive/auxdest/tbs_12.f';
  SET NEWNAME FOR DATAFILE '/oracle/dbs/tbs_11.f'
  TO '/bigdrive/auxdest/tbs_11.f';
  TRANSPORT TABLESPACE tbs_2
  TABLESPACE DESTINATION '/disk1/transportdest'
  AUXILIARY DESTINATION '/disk1/auxdest';
}
```

SET NEWNAME は、1 回のみ操作に最適です。特定の表領域セット用に、定期的にバックアップからトランスポートブル表領域を作成する予定がある場合、補助インスタンスのデータファイルの場所に永続的な設定を行うために、SET NEWNAME のかわりに CONFIGURE AUXNAME を使用することを検討してください。

## 補助データファイルに対する CONFIGURE AUXNAME の使用

CONFIGURE AUXNAME コマンドを使用して、トランスポートブル表領域セットまたは補助セットのデータファイル用の永続的な場所を指定できます。Recovery Manager は、リカバリの前に、CONFIGURE AUXNAME コマンドが使用されている各データファイルを指定された場所にリストアします。Recovery Manager は、操作が失敗しないかぎり、操作の完了時に補助セットのデータファイルを削除します。

CONFIGURE AUXNAME と TRANSPORT ... AUXILIARY DESTINATION の間の関係を表す例を示します。表領域 tbs\_11 を転送するとします。表領域 tbs\_12 は、補助セットの一部であり、データファイル tbs\_12.f を含んでいます。次の手順を実行します。

1. CONFIGURE AUXNAME 文を使用して、補助セットのデータファイル /oracle/dbs/tbs\_12.f に対してデフォルト以外の永続的な場所を設定します。

たとえば、次のコマンドを入力します。

```
CONFIGURE AUXNAME FOR '/oracle/dbs/tbs_12.f'  
TO '/disk1/auxdest/tbs_12.f';
```

2. AUXILIARY DESTINATION パラメータを指定して TRANSPORT TABLESPACE コマンドを実行します。

たとえば、次のコマンドを入力します。

```
TRANSPORT TABLESPACE tbs_11  
AUXILIARY DESTINATION '/myauxdest';
```

前述の例で、Recovery Manager は、データファイル /oracle/dbs/tbs\_12.f の補助セットのコピーを、AUXILIARY DESTINATION で指定された場所ではなく /disk1/auxdest/tbs\_12.f に格納します。CONFIGURE AUXNAME 設定の優先順位が、AUXILIARY DESTINATION の優先順位より高いためです。

---

**注意：**『Oracle Database バックアップおよびリカバリ・リファレンス』で説明するように、SHOW AUXNAME コマンドを実行して、現行の CONFIGURE AUXNAME 設定を表示できます。

---

## AUXILIARY DESTINATION を使用した補助ファイルの場所の指定

AUXILIARY DESTINATION 引数を TRANSPORT TABLESPACE とともに使用する場合、SET NEWNAME または CONFIGURE AUXNAME コマンドを使用して別の場所に移動されていない補助セット・ファイルは、TRANSPORT TABLESPACE 操作の間に補助の格納場所に格納されます。

AUXILIARY DESTINATION を使用しない場合は、LOG\_FILE\_NAME\_CONVERT を使用して、補助インスタンスのオンライン REDO ログ・ファイルの場所を指定する必要があります。SET NEWNAME または CONFIGURE AUXNAME のいずれを使用しても、補助インスタンスのオンライン REDO ログの場所に影響を与えることはできません。このため、AUXILIARY DESTINATION または LOG\_FILE\_NAME\_CONVERT を使用しない場合、Recovery Manager にはオンライン REDO ログを作成する場所に関する情報がありません。

## 初期化パラメータを使用した補助ファイル名の指定

補助インスタンスのパラメータ・ファイルで LOG\_FILE\_NAME\_CONVERT および DB\_FILE\_NAME\_CONVERT 初期化パラメータを使用すると、補助インスタンスのオンライン REDO ログおよびその他のデータベース・ファイルの名前を指定できます。TRANSPORT TABLESPACE コマンドに AUXILIARY DESTINATION 句を指定しない場合は、これらのパラメータによって、CONFIGURE AUXNAME または SET NEWNAME コマンドが実行されなかったすべてのファイルの場所が決定されます。

元のファイルが Oracle Managed Files (OMF) ファイルに格納される場合、LOG\_FILE\_NAME\_CONVERT または DB\_FILE\_NAME\_CONVERT を使用して、補助インスタンスのファイルに新しい OMF ファイル名を生成することはできません。データベースによって、各 OMF 転送先で一意のファイル名が生成されるように管理されます。

AUXILIARY DESTINATION 句を使用して、オンライン REDO ログ・ファイルの場所を制御する必要があります。AUXILIARY DESTINATION 句、SET NEWNAME コマンド、CONFIGURE AUXNAME コマンドまたは DB\_CREATE\_FILE\_DEST 初期化パラメータを使用して、OMF データファイルの場所を指定する必要があります。



**参照:** LOG\_FILE\_NAME\_CONVERT および DB\_FILE\_NAME\_CONVERT 初期化パラメータの詳細は、『Oracle Database リファレンス』を参照してください。



---

## プラットフォーム間でのデータ転送

Recovery Manager を使用すると、異なるエンディアン形式のプラットフォーム間で表領域を転送できます。また、2つのプラットフォームのエンディアン形式が同じであるかぎり、Recovery Manager を使用してデータベース全体を別のプラットフォームに転送することもできます。

この章の内容は、次のとおりです。

- [クロス・プラットフォーム・データ転送の概要](#)
- [ソース・ホストでのクロス・プラットフォーム表領域変換の実行](#)
- [ソース・ホストでのクロス・プラットフォーム・データファイル変換の実行](#)
- [クロス・プラットフォーム・データベース変換の前のデータベースの確認](#)
- [データベースを転送する場合のソース・ホストでのデータファイルの変換](#)
- [データベースを転送する場合の転送先ホストでのデータファイルの変換](#)

## クロス・プラットフォーム・データ転送の概要

この項では、プラットフォーム間で表領域およびデータベースを転送する場合の基本的な概念およびタスクについて説明します。

### クロス・プラットフォーム・データ転送の目的

あるプラットフォームで実行されているデータベースの表領域を別のプラットフォームで実行されているデータベースに転送できます。クロス・プラットフォーム・トランスポータブル表領域の通常の使用方法を次に示します。

- 表領域を変換することでプラットフォームに関係なく既存のデータベースに表領域を統合できるカスタマへ、構造化データをトランスポータブル表領域として公開する。
- 大規模なデータ・ウェアハウス・サーバーから小規模なコンピュータ（Linux ベースのワークステーションやサーバーなど）上のデータ・マートへデータを移動する。
- すべてのホストのエンディアン形式が同一である異機種間クラスタ間で、読取り専用表領域を共有する。

トランスポータブル表領域、その使用方法、およびトランスポータブル表領域を作成および使用する場合の様々な方法の詳細は、『Oracle Database 管理者ガイド』を参照してください。

また、Recovery Manager を使用して、プラットフォーム間でデータベース全体を転送することもできます。たとえば、ビジネス要件によって、別のプラットフォームを使用する、より低コストなサーバー上でデータベースを実行する必要があるとします。この場合、データベースを最初から再作成し、インポート・ユーティリティまたはトランスポータブル表領域を使用してデータを再移入するのではなく、Recovery Manager を使用してデータベース全体を転送できます。

転送先ホストまたはソース・ホスト上のデータベースを変換できます。転送先ホストで変換を行う理由は次のとおりです。

- 変換プロセスによるソース・ホストのパフォーマンスのオーバーヘッドを回避する場合。
- 1つのソース・システムから複数の異なるプラットフォーム上の複数の受信者にデータベースを分散する場合。
- 新しいプラットフォームへの移行パスを評価する場合

### クロス・プラットフォーム・データ転送の基本的な概念

ソース・プラットフォームと転送先プラットフォームが異なり、そのエンディアン形式も異なる場合は、トランスポータブル表領域の操作で Recovery Manager の CONVERT コマンドを使用する必要があります。エンディアン形式が同じプラットフォーム間での転送では、CONVERT を使用する必要はありません。オペレーティング・システム・ユーティリティを使用して、ソースから転送先にファイルをコピーできます。

#### 表領域およびデータファイルの変換

ソース・ホストでは Recovery Manager の CONVERT TABLESPACE コマンドを使用して表領域の変換を実行できますが、転送先ホストでは実行できません。CONVERT TABLESPACE コマンドでは、データファイルのインプレース変換は実行されません。かわりに、このコマンドでは、転送先プラットフォームで使用するために適切な形式の出力ファイルが生成されます。ソース・データベース内のデータファイルの内容は変更されません。

転送先ホストでは CONVERT DATAFILE コマンドを使用してファイルを変換できますが、ソース・ホストでは変換できません。データ・ポンプ・エクスポート・ユーティリティでは、**エクスポート・ダンプ・ファイル**が生成されます。このファイルは、転送先ホストに手動でコピーされたデータファイルとともに、転送先データベースにインポートできます。データファイルが転送先データベースに転送されるまで、それらのデータファイルはデータベース内の表領域名に関連付けられません。この場合、Recovery Manager は、表領域名を一連のデータファイルに変換できません。このため、CONVERT DATAFILE を使用して、ファイル名でデータファイルを指定する必要があります。

---

**注意：** CONVERT TABLESPACE または CONVERT DATAFILE の使用は、クロス・プラットフォーム・トランスポート表領域を使用する場合の 1 つの手順にすぎません。この章の手順を実行する前に、『Oracle Database 管理者ガイド』のトランスポート表領域に関する項を参照してください。

---

## データベースの変換

データベース全体を異なるプラットフォームに変換するには、両方のプラットフォームで同じエンディアン形式が使用されている必要があります。Recovery Manager の CONVERT DATABASE コマンドを使用すると、ソース・プラットフォームから転送先プラットフォームへのデータベース全体の移動を自動化できます。転送されたデータベースには、ソース・データベースと同じデータおよび（いくつかの例外を除いて）ソース・データベースと同じ設定が含まれています。

転送先プラットフォームに自動的に転送されるファイルを次に示します。

- 永続表領域に属するデータファイル

プラットフォーム間での表領域の転送とは異なり、データベース全体を転送する場合は、転送先プラットフォームとの互換性が維持されるように、特定の形式のブロック（UNDO セグメント内のブロックなど）を再フォーマットする必要があります。ソース・プラットフォームと転送先プラットフォームのエンディアン形式が同じである場合でも、トランスポート・データベースのデータファイルは変換する必要があります。表領域を転送する場合のように、あるプラットフォームから別のプラットフォームにデータファイルを単にコピーすることはできません。

- 初期化パラメータ・ファイルまたはサーバー・パラメータ・ファイル

データベースでテキストベースの初期化パラメータ・ファイルが使用されている場合、Recovery Manager はそのファイルを転送します。データベースでサーバー・パラメータ・ファイルが使用されている場合、Recovery Manager は、そのサーバー・パラメータ・ファイルに基づいて初期化パラメータ・ファイルを生成し、転送します。次に、初期化パラメータ・ファイルの設定に基づいて、転送先で新しいサーバー・パラメータ・ファイルを作成します。

ほとんどの場合、初期化パラメータ・ファイルの一部のパラメータを新しいデータベース用に手動で更新する必要があります。たとえば、DB\_NAME、および転送先ホスト上のファイルの場所を示す CONTROL\_FILES などのパラメータを変更する場合があります。

ソース・プラットフォームまたは転送先プラットフォームのいずれかのデータファイルの形式を変換できます。CONVERT DATABASE コマンド自体では、データファイルの形式は変換されません。かわりに、変換を実行するために手動で実行できるスクリプトが生成されます。CONVERT SCRIPT パラメータを指定すると、**変換スクリプト**が作成されます。このスクリプトを転送先ホストで手動で実行すると、データファイルのコピーをバッチ・モードで変換できます。TRANSPORT SCRIPT パラメータを指定すると、**転送スクリプト**が生成されます。このスクリプトには、転送先プラットフォーム上に新しいデータベースを作成する SQL 文が含まれています。

## ソース・ホストでのクロス・プラットフォーム表領域変換の実行

CONVERT の前提条件のリストについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。この項の手順を実行する前に、これらのすべての前提条件を満たしてください。

わかりやすくするため、Sun Solaris ホスト上で実行されているソース・データベース prod\_source から、表領域 finance および hr を転送する必要があると想定します。Linux PC 上で実行されている転送先データベース prod\_dest に、これらの表領域を転送することを計画します。ソース・ホスト上の一時ディレクトリ /tmp/transport\_linux/ に、変換済データファイルを格納することを計画しています。

**ソース・ホストでクロス・プラットフォーム表領域変換を実行する手順**

1. SQL\*Plus を起動し、管理者権限でソース・データベース prod\_source に接続します。
2. V\$TRANSPORTABLE\_PLATFORM で転送先プラットフォームの名前を問い合わせます。

データベースには、クロス・プラットフォームのデータ転送をサポートする各プラットフォームの固有な内部名のリストが存在します。CONVERT コマンドのパラメータとして、ソース・プラットフォームまたは転送先プラットフォームの正確な名前が必要な場合があります。V\$TRANSPORTABLE\_PLATFORM を問い合わせ、プラットフォームの名前を取得します。次の例では、Linux プラットフォームの名前を問い合わせます。

```
SELECT PLATFORM_ID, PLATFORM_NAME, ENDIAN_FORMAT
FROM   V$TRANSPORTABLE_PLATFORM
WHERE  UPPER(PLATFORM_NAME) LIKE '%LINUX%';
```

PC 上の Linux の PLATFORM\_NAME は、Linux IA (32-bit) です。

3. 転送する表領域を読み取り専用モードに設定します。たとえば、次のように入力します。

```
ALTER TABLESPACE finance READ ONLY;
ALTER TABLESPACE hr READ ONLY;
```

4. 出力ファイルの名前を指定する方法を選択します。

CONVERT に FORMAT または DB\_FILE\_NAME\_CONVERT 引数を指定して、出力ファイルの名前を制御します。次に、優先順位の高い順に規則を示します。

- a. CONVERT ...DB\_FILE\_NAME\_CONVERT 句のパターンに一致するファイルは、そのパターンに基づいて名前が指定されます。
- b. FORMAT 句を指定すると、CONVERT ...DB\_FILE\_NAME\_CONVERT 句のパターンに基づいて名前が指定されていないすべてのファイルに対して、FORMAT 句に基づいて名前が指定されます。

---

**注意：** ソース・ファイルおよび転送先ファイルが Oracle Managed Files の名前である場合、CONVERT ... DB\_FILE\_NAME\_CONVERT を使用して CONVERT の出力ファイル名を生成することはできません。

---

5. Recovery Manager を起動し、TARGET として（転送先データベースではなく）ソース・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
RMAN> CONNECT TARGET SYS@prod_source
```

6. CONVERT TABLESPACE コマンドを実行して、データファイルを転送先ホストのエンディアン形式に変換します。

次の例で FORMAT 引数は、変換されたデータ・ファイルの名前および場所を制御しています。

```
RMAN> CONVERT TABLESPACE finance,hr
2> TO PLATFORM 'Linux IA (32-bit)'
3> FORMAT '/tmp/transport_linux/%U';
```

結果として、/tmp/transport\_linux/ ディレクトリに、Linux IA (32 ビット) プラットフォームに適切なエンディアン形式のデータを含む一連の変換済データファイルが格納されます。

**参照：** CONVERT コマンドの完全なセマンティクスについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

7. 表領域を転送する場合の一般的な手順の残りの手順を実行します。
  - a. Oracle Data Pump Export ユーティリティを使用して、ソース・ホスト上にエクスポート・ダンプ・ファイルを作成します。
  - b. 変換済データファイルおよびエクスポート・ダンプ・ファイルをソース・ホストから転送先ホストの目的のディレクトリに移動します。
  - c. インポート・ユーティリティを使用して、表領域を新しいデータベースに組み込みます。
  - d. 必要に応じて、転送された表領域を読み取り / 書き込みモードに設定します。

## ソース・ホストでのクロス・プラットフォーム・データファイル変換の実行

CONVERT の前提条件のリストについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。この項の手順を実行する前に、これらの前提条件を満たしてください。

### 転送先ホストでのクロス・プラットフォーム・データファイル変換

データファイルの変換では、出力ファイルの名前を指定する方法を選択する必要があります。CONVERT に FORMAT 引数または DB\_FILE\_NAME\_CONVERT 引数を指定して、出力ファイルの名前を指定する方法を制御します。次に、優先順位の高い順に規則を示します。

1. CONVERT ...DB\_FILE\_NAME\_CONVERT 句のパターンに一致するファイルは、そのパターンに基づいて名前が指定されます。
2. FORMAT 句を指定すると、CONVERT ...DB\_FILE\_NAME\_CONVERT 句のパターンに基づいて名前が指定されていないすべてのファイルに対して、FORMAT 句に基づいて名前が指定されます。

---

**注意：** ソース・ファイルおよび転送先ファイルの両方が Oracle Managed Files である場合、CONVERT ... DB\_FILE\_NAME\_CONVERT を使用して、CONVERT の出力ファイル名を生成することはできません。

---

ソース・プラットフォームおよび転送先プラットフォームが異なる場合は、FROM PLATFORM パラメータを指定する必要があります。V\$TRANSPORTABLE\_PLATFORM を問い合せて、プラットフォームの名前を確認します。エラーを回避するには、FROM\_PLATFORM の値が、変換するデータファイルの形式と一致する必要があります。FROM\_PLATFORM を指定しない場合、このパラメータはデフォルトで転送先プラットフォームの値になります。

### CONVERT DATAFILE を使用したデータファイル形式の変換

この項では、CONVERT DATAFILE の使用方法について説明します。この項では、表領域 finance (データファイル fin/fin01.dbf および fin/fin02.dbf) および hr (データファイル hr/hr01.dbf および hr/hr02.dbf) を、prod\_source というソース・データベースから転送すると想定しています。このデータベースは、Sun Solaris ホスト上で実行されています。Linux PC 上で実行されている prod\_dest という転送先データベースに、これらの表領域を転送することを計画します。転送先ホスト上で変換を実行することを計画しています。

データファイルが転送先データベースに組み込まれる場合は、これらのデータファイルが /orahome/dbs に格納され、現行のディレクトリ構造が保持されるように計画します。つまり、hr 表領域のデータファイルは /orahome/dbs/hr サブディレクトリに、finance 表領域のデータファイルは /orahome/dbs/fin ディレクトリに格納されます。

**転送先ホストでクロス・プラットフォーム・データファイル変換を実行する手順**

1. SQL\*Plus を起動し、管理者権限でソース・データベース prod\_source に接続します。
2. V\$TRANSPORTABLE\_PLATFORM でソース・プラットフォームの名前を問い合わせます。

データベースには、クロス・プラットフォームのデータ転送をサポートする各プラットフォームの固有な内部名のリストが存在します。CONVERT コマンドのパラメータとして、ソース・プラットフォームまたは転送先プラットフォームの正確な名前が必要な場合があります。たとえば、次のように入力すると、接続したデータベースのプラットフォーム名を取得できます。

```
SELECT PLATFORM_NAME
FROM   V$TRANSPORTABLE_PLATFORM
WHERE  PLATFORM_ID =
      ( SELECT PLATFORM_ID
        FROM   V$DATABASE );
```

この例では、ソース・ホストの PLATFORM\_NAME は Solaris [tm] OE (64-bit) であると想定しています。

3. ソース・データベースから転送する表領域を識別し、それらの表領域を読取り専用モードに設定します。

たとえば、finance および hr を読取り専用モードに設定するには、次の SQL 文を入力します。

```
ALTER TABLESPACE finance READ ONLY;
ALTER TABLESPACE hr READ ONLY;
```

4. ソース・ホスト上で、データ・ポンプ・エクスポートを使用してエクスポート・ダンプ・ファイルを作成します。

この例では、ダンプ・ファイルは expdat.dmp という名前になります。

5. エクスポート・ダンプ・ファイルおよび転送するデータファイルを転送先ホストで使用可能にします。

NFS を使用して、ダンプ・ファイルおよび現行のデータベース・ファイル（コピーは対象外）をアクセス可能にします。また、オペレーティング・システム・ユーティリティを使用して、これらのファイルを転送先ホストにコピーすることもできます。

この例では、転送先ホストの /tmp/transport\_solaris/ ディレクトリにファイルを格納します。ファイルの元の場所からのサブディレクトリ構造を保持します。つまり、データファイルは次の場所に格納されます。

- /tmp/transport\_solaris/fin/fin01.dbf
- /tmp/transport\_solaris/fin/fin02.dbf
- /tmp/transport\_solaris/hr/hr01.dbf
- /tmp/transport\_solaris/hr/hr02.dbf

6. Recovery Manager を起動し、TARGET として（ソース・データベースではなく）転送先データベースに接続します。たとえば、次のコマンドを入力します。

```
% rman
RMAN> CONNECT TARGET SYS@prod_dest
```

7. CONVERT DATAFILE コマンドを実行して、データファイルを転送先ホストのエンディアン形式に変換します。

次の例では、DB\_FILE\_NAME\_CONVERT を使用して、変換済データファイルの名前および場所を制御します。また、FROM PLATFORM 句も指定します。

```
RMAN> CONVERT DATAFILE
2>  '/tmp/transport_solaris/fin/fin01.dbf',
3>  '/tmp/transport_solaris/fin/fin02.dbf',
4>  '/tmp/transport_solaris/hr/hr01.dbf',
```



```

5> '/tmp/transport_solaris/hr/hr02.dbf'
6> DB_FILE_NAME_CONVERT
7> '/tmp/transport_solaris/fin','/orahome/dbs/fin',
8> '/tmp/transport_solaris/hr','/orahome/dbs/hr'
9> FROM PLATFORM 'Solaris[tm] OE (64-bit)

```

結果として、/orahome/dbs/ ディレクトリに一連の変換済データファイルが格納されます。これらのファイルの名前は次のとおりです。

- /orahome/dbs/fin/fin01.dbf
- /orahome/dbs/fin/fin02.dbf
- /orahome/dbs/hr/hr01.dbf
- /orahome/dbs/hr/hr02.dbf

8. 表領域を転送する場合の一般的な手順の残りの手順を実行します。

- a. インポート・ユーティリティを使用して、表領域を新しいデータベースに組み込みます。
- b. 必要に応じて、転送された表領域を読取り専用モードに設定します。

**参照：** CONVERT コマンドの構文およびセマンティクスについては、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

## クロス・プラットフォーム・データベース変換の前のデータベースの確認

25-2 ページの「クロス・プラットフォーム・データ転送の基本的な概念」で説明されているように、Recovery Manager の CONVERT DATABASE コマンドを使用すると、あるプラットフォームから別のプラットフォームへのデータベース全体のコピーを自動化できます。

データベースを変換する前に、CONVERT DATABASE の前提条件のリストについて『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。たとえば、ソース・プラットフォームと転送先プラットフォームでは同じエンディアン形式を使用する必要があります。この項の手順を試行する前に、これらのすべての前提条件を満たしてください。

クロス・プラットフォーム・トランスポート・データベースの主な前提条件は、ソース・プラットフォームと転送先プラットフォームのエンディアン形式が同じである必要があるということです。たとえば、Microsoft Windows から Linux for x86（両方ともリトル・エンディアン）または HP-UX から AIX（両方ともビッグ・エンディアン）にデータベースを転送することはできますが、HP-UX から Linux for x86 にデータベース全体を転送することはできません。

---

**注意：** プラットフォームのエンディアン形式が異なるため、CONVERT DATABASE を使用できない場合は、転送先プラットフォーム上に新しいデータベースを手動で作成し、クロス・プラットフォーム・トランスポート表領域を使用して、ソース・データベースから必要な表領域を転送できます。

---

### データベースの変換を準備する手順

1. ソース・データベースで、SYSDBA として SQL\*Plus セッションを開始します。
2. データベースを読取り専用モードでオープンします。

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

3. SQL\*Plus でサーバー出力が有効になっていることを確認します。

たとえば、次の SQL\*Plus コマンドを入力します。

```
SET SERVEROUTPUT ON
```

4. DBMS\_TDB.CHECK\_DB ファンクションを実行します。

このチェックを実行すると、データベースの転送を妨げる条件がないことが確認されます。転送を妨げる条件には、不適切な互換性設定、インダウト・トランザクションやアクティブ・トランザクション、ソース・プラットフォームと転送先プラットフォーム間の互換性のないエンディアン形式などがあります。

CHECK\_DB を引数指定せずにコールすると、ソース・データベースに転送を妨げる条件があるかどうかを確認できます。このファンクションは、表 25-1 に示す引数を指定してコールすることもできます。

表 25-1 CHECK\_DB プロシージャのパラメータ

| パラメータ                | 説明   |
|----------------------|--|
| target_platform_name | V\$DB_TRANSPORTABLE_PLATFORM に表示される転送先プラットフォームの名前。<br><br>このパラメータは任意ですが、skip_option パラメータを使用する場合は必須です。省略すると、転送先プラットフォームがソース・プラットフォームと互換性があると想定され、プラットフォームの互換性に関連性のない条件のみがテストされます。  |
| skip_option          | データベースが転送可能かどうかをチェックする際にスキップするデータベースの部分（適切な場合）を指定します。指定可能な値（NUMBER 型）は次のとおりです。 <ul style="list-style-type: none"> <li>■ SKIP_NONE（または 0）：すべての表領域がチェックされます。</li> <li>■ SKIP_OFFLINE（または 2）：オフライン表領域内のデータファイルのチェックがスキップされます。</li> <li>■ SKIP_READONLY（または 3）：読取り専用表領域内のデータファイルのチェックがスキップされます。</li> </ul> |

例 25-1 では、32 ビット Windows にデータベースを転送するために、32 ビット Linux プラットフォーム上で CHECK\_DB を実行し、読取り専用表領域をスキップします。

例 25-1 DBMS\_TDB.CHECK\_DB の実行

```
DECLARE
  db_ready BOOLEAN;
BEGIN
  db_ready :=
    DBMS_TDB.CHECK_DB('Microsoft Windows IA (32-bit)',DBMS_TDB.SKIP_READONLY);
END;
/
```

PL/SQL procedure successfully completed.

警告が表示されない場合または DBMS\_TDB.CHECK\_DB によって TRUE が戻される場合は、この時点でデータベースを転送できます。手順 6 に進みます。

警告が表示された場合または DBMS\_TDB.CHECK\_DB によって FALSE が戻された場合は、この時点でデータベースを転送できません。次の手順に進みます。

5. 出力を調べてデータベースを転送できない理由を確認し、可能な場合は問題を解決してから、前の手順に戻ります。該当する条件については、DBMS\_TDB のドキュメントを参照してください。
6. DBMS\_TDB.CHECK\_EXTERNAL を実行して、すべての外部表、ディレクトリまたは BFILE を識別します。Recovery Manager は、これらのファイルの転送を自動化できません。したがって、ファイルを手動でコピーしてデータベースのディレクトリを再作成する必要があります。

例 25-2 に、DBMS\_TDB.CHECK\_EXTERNAL のコール方法を示します。

**例 25-2 DBMS\_TDB.CHECK\_EXTERNAL の実行**

```

DECLARE
    external BOOLEAN;
BEGIN
    /* value of external is ignored, but with SERVEROUTPUT set to ON
    * dbms_tdb.check_external displays report of external objects
    * on console */
    external := DBMS_TDB.CHECK_EXTERNAL;
END;
/

```

外部オブジェクトが存在しない場合は、このプロシージャを実行しても出力は行われません。ただし、外部オブジェクトが存在する場合は、次のような出力が表示されます。

The following external tables exist in the database:

SH.SALES\_TRANSACTIONS\_EXT

The following directories exist in the database:

SYS.DATA\_PUMP\_DIR, SYS.MEDIA\_DIR, SYS.DATA\_FILE\_DIR, SYS.LOG\_FILE\_DIR

The following BFILEs exist in the database:

PM.PRINT\_MEDIA

PL/SQL procedure successfully completed.

## データベースを転送する場合のソース・ホストでのデータファイルの変換

この項では、`CONVERT DATABASE` のすべての前提条件をすでに満たし、25-7 ページの「[クロス・プラットフォーム・データベース変換の前のデータベースの確認](#)」の手順を実行済であると想定しています。この手順の目的は、クロス・プラットフォーム・データベース転送の一部として、ソース・ホスト上のデータファイルの形式を変換することです。

Solaris 上で実行されるデータベースを Windows 上で実行されるデータベースに変換するとします。

### ソース・プラットフォーム上のデータベースを変換する手順

1. ソース・データベースを読取り専用モードでオープンします。

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

2. Recovery Manager を起動し、TARGET としてソース・データベースに接続します。たとえば、次のコマンドを入力します。

```
% rman
RMAN> CONNECT TARGET SYS@source_db
```

3. `CONVERT DATABASE` コマンドを実行します。

**例 25-3** に、`CONVERT DATABASE` コマンドを示します（出力例も示します）。`TRANSPORT SCRIPT` パラメータによって、生成される SQL スクリプトの場所を指定します。このスクリプトを使用して、新しいデータベースを作成できます。`TO PLATFORM` パラメータによって、転送先データベースのプラットフォームを指定します。

`DB_FILE_NAME_CONVERT` パラメータによって、生成されるデータファイルのネーミング方法を指定します。

### 例 25-3 ソース・ホスト上のデータベースの変換

```

RMAN> CONVERT DATABASE
2> NEW DATABASE 'newdb'
3> TRANSPORT SCRIPT '/tmp/convertdb/transportscript.sql'
4> TO PLATFORM 'Microsoft Windows IA (32-bit)'
5> DB_FILE_NAME_CONVERT '/disk1/oracle/dbs' '/tmp/convertdb';

```

```

Starting convert at 25-NOV-06
using channel ORA_DISK_1

External table SH.SALES_TRANSACTIONS_EXT found in the database

Directory SYS.DATA_PUMP_DIR found in the database
Directory SYS.MEDIA_DIR found in the database
Directory SYS.DATA_FILE_DIR found in the database
Directory SYS.LOG_FILE_DIR found in the database

BFIL FILE PM.PRINT_MEDIA found in the database

User SYS with SYSDBA and SYSOPER privilege found in password file
User OPER with SYSDBA privilege found in password file
channel ORA_DISK_1: starting datafile conversion
input datafile fno=00001 name=/disk1/oracle/dbs/tbs_01.f
converted datafile=/tmp/convertdb/tbs_01.f
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:15
channel ORA_DISK_1: starting datafile conversion
input datafile fno=00002 name=/disk1/oracle/dbs/tbs_ax1.f
converted datafile=/tmp/convertdb/tbs_ax1.f
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:03
.
.
.
channel ORA_DISK_1: starting datafile conversion
input datafile fno=00016 name=/disk1/oracle/dbs/tbs_52.f
converted datafile=/tmp/convertdb/tbs_52.f
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:01
Run SQL script /tmp/convertdb/transportscript.sql on the destination platform
to create database
Edit init.ora file init_00gb3vfv_1_0.ora. This PFILE will be used to
create the database on the destination platform
To recompile all PL/SQL modules, run utlirp.sql and utlirp.sql on
the destination platform
To change the internal database identifier, use DBNEWID Utility
Finished backup at 25-NOV-06

```

4. CONVERT DATABASE の完了後、ソース・データベースを読み取り / 書き込みで再度オープンできます。
5. CONVERT DATABASE で生成されたデータファイルを転送先ホスト上の目的の場所に移動します。

例 25-3 では、このコマンドによって、ソース・ホストの /tmp/convertdb/ ディレクトリにファイルが作成されます。転送先データベースのファイルが格納される転送先ホストのディレクトリに、これらのファイルを移動します。

6. データファイルのパスが転送先ホスト上でパスと異なる場合は、データファイルの新しい場所が参照されるように転送スクリプトを編集します。
7. 必要に応じて、初期化パラメータ・ファイルを編集して、転送先データベースの設定を変更します。

初期化パラメータ・ファイルの先頭のいくつかのエントリは、データベースを転送先プラットフォームに移動する際に編集する必要があります。たとえば、初期化パラメータ・ファイルは次のような内容になっている場合があります。

```

# Please change the values of the following parameters:
control_files          = "/tmp/convertdb/cf_D-NEWDBT_id-1778429277_00gb9u2s"
db_recovery_file_dest = "/tmp/convertdb/orcva"
db_recovery_file_dest_size= 10737418240
instance_name         = "NEWDBT"

```

```

service_names          = "NEWDBT.regress.rdbms.dev.us.oracle.com"
plsql_native_library_dir = "/tmp/convertdb/plsqlnld1"
db_name                = "NEWDBT"

```

- 必要に応じて、変換されたデータファイルの新しい名前を使用するように転送スクリプトを編集します。

例 25-3 では、転送スクリプトの名前は /tmp/convertdb/transportscript.sql です。データベースを実際に作成するには、転送先ホストでこのスクリプトを実行します。したがって、データファイルの正しい名前を使用してこのスクリプトを編集する必要があります。

- 転送先ホストで SQL\*Plus を起動し、オペレーティング・システム認証を使用して SYSDBA として転送先データベース・インスタンスに接続します。

たとえば、次のように接続します。

```
SQL> CONNECT / AS SYSDBA
```

オペレーティング・システム認証を使用しない場合は、まず、Oracle Net ファイルを構成して **パスワード・ファイル** を作成し、リスナーを起動する必要があります。その後、ネット・サービス名を使用してインスタンスに接続できます。

- SQL\*Plus で転送スクリプトを実行し、転送先ホスト上に新しいデータベースを作成します。

```
SQL> @transportscript
```

転送スクリプトが終了すると、新しいデータベースの作成が完了します。

## データベースを転送する場合の転送先ホストでのデータファイルの変換

この項では、CONVERT DATABASE のすべての前提条件をすでに満たし、25-7 ページの「[クロス・プラットフォーム・データベース変換の前のデータベースの確認](#)」の手順を実行済であると想定しています。この手順の目的は、クロス・プラットフォーム・データベース転送の一部として、転送先ホスト上のデータファイルの形式を変換することです。

データファイルの変換には、次のフェーズがあります。

- ソース・ホストでのデータファイル初期変換手順の実行
- 転送先ホストでのデータファイルの変換

### ソース・ホストでのデータファイル初期変換手順の実行

この手順では、ソース・ホスト上で CONVERT DATABASE コマンドを実行します。このコマンドによって、初期化パラメータ・ファイル、および転送先ホストで使用するために編集可能なスクリプトが生成されます。また、変換前のデータファイルをソース・ホストから転送先ホストにコピーします。

#### ソース・ホストでデータファイル初期変換を実行する手順

- データベースが読み取り専用モードでオープンしていることを確認します。
- Recovery Manager を起動し、TARGET としてソース・データベースに接続します。

たとえば、次のコマンドを入力します。

```

% rman
RMAN> CONNECT TARGET SYS@source_db

```

3. CONVERT DATABASE ON DESTINATION コマンドを実行します。

例 25-4 に、CONVERT DATABASE コマンドの例を示します（出力例も示します）。ON DESTINATION PLATFORM パラメータを使用して、データファイルが必要とされるすべての CONVERT コマンドがソース・プラットフォームではなく転送先プラットフォームで実行されるように指定します。FORMAT パラメータによって、生成されるファイルのネーミング方法を指定します。

#### 例 25-4 CONVERT DATABASE ON DESTINATION PLATFORM の実行

```

RMAN> CONVERT DATABASE
2> ON DESTINATION PLATFORM
3> CONVERT SCRIPT '/tmp/convertdb/convertscript-target'
4> TRANSPORT SCRIPT '/tmp/convertdb/transportscript-target'
5> NEW DATABASE 'newdbt'
6> FORMAT '/tmp/convertdb/%U';

Starting convert at 28-JAN-05
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=39 devtype=DISK

External table SH.SALES_TRANSACTIONS_EXT found in the database

Directory SYS.DATA_PUMP_DIR found in the database
Directory SYS.MEDIA_DIR found in the database
Directory SYS.DATA_FILE_DIR found in the database
Directory SYS.LOG_FILE_DIR found in the database

BFILE PM.PRINT_MEDIA found in the database

User SYS with SYSDBA and SYSOPER privilege found in password file
User OPER with SYSDBA privilege found in password file
channel ORA_DISK_1: starting to check datafiles
input datafile fno=00001 name=/disk1/oracle/dbs/tbs_01.f
channel ORA_DISK_1: datafile checking complete, elapsed time: 00:00:00
channel ORA_DISK_1: starting to check datafiles
input datafile fno=00002 name=/disk1/oracle/dbs/tbs_ax1.f
channel ORA_DISK_1: datafile checking complete, elapsed time: 00:00:00
channel ORA_DISK_1: starting to check datafiles
input datafile fno=00017 name=/disk1/oracle/dbs/tbs_03.f
channel ORA_DISK_1: datafile checking complete, elapsed time: 00:00:00
.
.
.
channel ORA_DISK_1: starting to check datafiles
input datafile fno=00015 name=/disk1/oracle/dbs/tbs_51.f
channel ORA_DISK_1: datafile checking complete, elapsed time: 00:00:00
channel ORA_DISK_1: starting to check datafiles
input datafile fno=00016 name=/disk1/oracle/dbs/tbs_52.f
channel ORA_DISK_1: datafile checking complete, elapsed time: 00:00:00
Run SQL script /tmp/convertdb/transportscript-target on the destination platform to
create database
Edit init.ora file /tmp/convertdb/init_00gb9u2s_1_0.ora. This PFILE will be used to
create the database on the destination platform
Run RMAN script /tmp/convertdb/convertscript-target on destination platform to convert
datafiles
To recompile all PL/SQL modules, run utlirp.sql and utlirp.sql on the destination
platform
To change the internal database identifier, use DBNEWID Utility
Finished backup at 28-JAN-05

```

例 25-4 のコマンドでは、転送スクリプト、新しいデータベースの初期化パラメータ・ファイル、および変換される各データファイル用の Recovery Manager の CONVERT DATAFILE を含む変換スクリプトが作成されます。

---

**注意：** CONVERT DATABASE ON DESTINATION PLATFORM では、変換済データファイルのコピーは生成されません。このコマンドでは、スクリプトのみが作成されます。

---

4. オペレーティング・システム・ユーティリティを使用して、転送先ホストの一時格納場所に次のファイルをコピーします。
  - 変換するデータファイル
  - 変換スクリプト
  - 転送スクリプト
  - 転送先データベースの初期化ファイル
5. ソース・データベースを読み取り / 書き込みモードにします。

## 転送先ホストでのデータファイルの変換

この項では、前の項で作成したスクリプトを使用して、転送先ホストのデータファイルを変換する方法について説明します。

前の項で作成した変換スクリプトでは、ソース・データベース・ファイルの元のデータファイル名を使用します。FORMAT パラメータによって、CONVERT DATABASE の FORMAT または DB\_FILE\_NAME\_CONVERT パラメータを使用して生成される名前を指定します。

同じパス名を使用して転送先ホストからソース・データベースのデータファイルにアクセス可能な場合、変換スクリプトは、ソース・データベースが読み取り専用であるかぎり、変更せずに転送先ホストで実行できます。たとえば、ソース・ホストおよび転送先ホストの両方で NFS を使用してソース・データファイルを含むディスクがマウントされており、両方のホストのマウント・ポイントが /fs1/dbs/ である場合、編集は必要ありません。

### 転送先ホスト上のデータファイルを変換する手順

1. 必要に応じて、変換スクリプトを編集します。

変換スクリプトでは、変換するデータファイルごとに 1 つの CONVERT DATAFILE コマンドが存在します。変換スクリプトで、変換前データファイルの現在の一時ファイル名および変換済データファイルの出力ファイル名を指定する必要があります。典型的な変換スクリプトは、次のような内容になります。

```

RUN
{
  CONVERT
  FROM PLATFORM 'Linux IA (32-bit)'
  PARALLELISM 10
  DATAFILE '/disk1/oracle/dbs/tbs_01.f'
  FORMAT
  '/tmp/convertedb/data_D-TV_I-1778429277_TS-SYSTEM_FNO-1_7qgb9u2s';

  DATAFILE '/disk1/oracle/dbs/tbs_ax1.f'
  FORMAT
  '/tmp/convertedb/data_D-TV_I-1778429277_TS-SYSAUX_FNO-2_7rgb9u2s';

  DATAFILE '/disk1/oracle/dbs/tbs_03.f'
  FORMAT
  '/tmp/convertedb/data_D-TV_I-1778429277_TS-SYSTEM_FNO-17_7sgb9u2s';

  DATAFILE '/disk1/oracle/dbs/tbs_51.f'
  FORMAT

```



```

'/tmp/convertdb/data_D-TV_I-1778429277_TS-TBS_5_FNO-15_8egb9u2u';

DATAFILE '/disk1/oracle/dbs/tbs_52.f'
FORMAT
'/tmp/convertdb/data_D-TV_I-1778429277_TS-TBS_5_FNO-16_8fgb9u2u';
}

```

変換スクリプト内の各 DATAFILE コマンドを編集し、各データファイルの一時格納場所を入力として指定します。また、各コマンドの FORMAT パラメータを編集し、転送済データベースのデータファイルの任意の最終格納場所を指定します。

- 必要に応じて、転送先ホスト上の初期化パラメータ・ファイルを編集し、転送先データベースの設定を変更します。

初期化パラメータ・ファイルの先頭のいくつかのエントリは、データベースを転送先プラットフォームに移動する前に編集する必要があります。たとえば、初期化パラメータ・ファイルは次のような内容になっている場合があります。

```

# Please change the values of the following parameters:
control_files          = "/tmp/convertdb/cf_D-NEWDBT_id-1778429277_00gb9u2s"
db_recovery_file_dest = "/tmp/convertdb/orcva"
db_recovery_file_dest_size= 10737418240
instance_name         = "NEWDBT"
service_names         = "NEWDBT.regress.rdbms.dev.us.oracle.com"
plsql_native_library_dir = "/tmp/convertdb/plsqlnld1"
db_name               = "NEWDBT"

```

- 転送先ホストで、SQL\*Plus を使用してデータベース・インスタンスを NOMOUNT モードで起動します。

前の手順でコピーした初期化パラメータ・ファイルを指定します。たとえば、次のコマンドを入力します。

```
SQL> STARTUP NOMOUNT PFILE='/tmp/init_convertdb_00i2gj63_1_0.ora'
```

- Recovery Manager を起動し、TARGET として（ソース・データベースではなく）転送先データベースに接続します。たとえば、次のコマンドを入力します。

```

% rman
RMAN> CONNECT TARGET SYS@prod_dest

```

- Recovery Manager プロンプトで、変換スクリプトを実行します。たとえば、次のコマンドを入力します。

```
RMAN> @/tmp/convertdb/convertscript-target
```

- データベース・インスタンスを停止します。

実行する必要がある転送スクリプトに STARTUP NOMOUNT コマンドがすでに含まれているため、この手順が必要となります。

- 必要に応じて、変換されたデータファイルの新しい名前を使用するように転送スクリプトを編集します。

例 25-3 では、転送スクリプトは /tmp/convertdb/transportscript.sql です。データベースを作成するには、転送先ホストでこのスクリプトを実行します。したがって、データファイルの正しい名前を使用してこのスクリプトを編集する必要があります。

- SQL\*Plus で転送スクリプトを実行します。

たとえば、次のように入力して転送先ホスト上に新しいデータベースを作成します。

```
SQL> @/tmp/convertdb/transportscript
```

転送スクリプトが終了すると、転送先データベースが作成されます。



---

## ASM でのデータの移行の実行

この章では、Recovery Manager を使用して、[自動ストレージ管理](#)ストレージへのデータの移行および ASM ストレージからのデータの移行を行う方法について説明します。この章の内容は、次のとおりです。

- [ASM でのデータの移行の概要](#)
- [ASM へのデータベースの移行の準備](#)
- [ASM へのデータベースの移行](#)
- [ASM から代替ストレージへのデータベースの移行](#)
- [ASM ディスク・グループ間でのデータファイルの移動](#)

## ASM でのデータの移行の概要

この項では、ASM にデータを移行する場合および ASM からデータを移行する場合の基本的な概念およびタスクについて説明します。

### ASM でのデータの移行の目的

ASM ストレージの代替には、ファイル・システム、RAW ディスクおよび SAN 構成があります。ASM には、これらの代替ストレージを上回る多くのメリット（パフォーマンスの最適化、冗長性保護、ロード・バランシングなど）があります。ASM によってディスクが管理されるため、サード・パーティの論理ボリューム・マネージャを入手する必要はありません。ASM には既製の共有ストレージが用意されているため、Oracle Real Application Clusters (Oracle RAC) データベースで ASM を使用すると有効です。

データベースで現在 ASM 以外のストレージ・システムが使用されている場合は、データベースのすべてまたは一部を ASM に移行すると、データベースの管理が簡単になります。また、フラッシュ・リカバリ領域も ASM に移行できます。

Linux の cp、Windows の COPY などのオペレーティング・システム固有のコマンドでは、ASM ストレージ内のファイルに対して書き込みまたは読取りを行うことはできません。Recovery Manager では ASM ファイルに対して読取りおよび書き込みを行うことができるため、Recovery Manager を使用して、ASM ストレージへのデータファイルのコピーおよび ASM ストレージからのデータファイルのコピーまたは ASM ディスク・グループ間でのデータファイルのコピーを実行できます。この方法は、バックアップをユーザー管理ディスクに格納する必要がある場合に有効です。

### ASM でのデータの移行の基本的な概念

Recovery Manager を主なバックアップ・ツールとして使用していない場合でも、Recovery Manager を使用して ASM にデータを移行できます。移行には、Recovery Manager データベース・バックアップが 1 つ必要です。

ASM と代替ストレージ・システムの両方にデータベース全体を保持するための十分なディスク領域がある場合は、データベースを ASM に直接移動できます。十分なストレージがない場合は、データベースをテープにバックアップし、古いディスク領域を使用する ASM ディスク・グループを作成して、テープから ASM にデータベースをリストアできます。

新しいリカバリ領域の場所を設定した後も、既存のバックアップは古いリカバリ領域に存在したままとなり、リカバリ領域のディスク割当て制限の合計の負担になります。領域が必要になると、バックアップが古いリカバリ領域から削除されます。これらのバックアップは、Recovery Manager で使用できます。古いバックアップは、ディスク領域が必要にならないかぎり、新しい ASM リカバリ領域に移動する必要はありません。古いリカバリ領域でファイルによって消費されている領域を解放するには、それらのファイルをテープにバックアップするか、または ASM リカバリ領域に移行します。

---

**注意：** 外部のアーカイブ REDO ログは、LogMiner セッションのロジカル・スタンバイ・データベースによって受信されるログです。外部のアーカイブ REDO ログは移行できません。

---

ASM から代替ストレージ・システムへのデータベースの移行は、代替ストレージ・システムから ASM への移行と類似しています。主な違いは、代替ストレージ・システムのファイルの場所を参照するように各手順を変更する点です。

**参照：** Enterprise Manager でデータベースを ASM に移行する方法については、『Oracle Database ストレージ管理者ガイド』を参照してください。

## ASM へのデータ移行の基本手順

データベース全体および**フラッシュ・リカバリ領域**を代替ストレージから ASM に移行する基本手順は、次のとおりです。

1. データベースおよびサーバー・パラメータ・ファイルをバックアップし、**Oracle Flashback Database** を無効にします。

この手順については、26-3 ページの「[ASM へのデータベースの移行の準備](#)」を参照してください。

2. ASM にファイルをリストアし、データベースをリカバリします。必要に応じて、フラッシュ・リカバリ領域を ASM に移行します。

この手順については、26-5 ページの「[ASM へのデータベースの移行](#)」を参照してください。

ファイルを代替ストレージから ASM に移行するには、26-8 ページの「[ASM から代替ストレージへのデータベースの移行](#)」を参照してください。

## ASM へのデータベースの移行の準備

この項では、移行のためにデータベースを準備する方法について説明します。この項では、次のことを想定しています。

- データベースを 2 つの ASM ディスク・グループ（データベース用の +DATA およびフラッシュ・リカバリ領域用の +FRA）に移行します。
- ASM ストレージに移行するデータベースは、mydb という名前です。

---

**注意：** フラッシュ・リカバリ領域を移行しない場合は、手順 10 をスキップします。

---

### ASM での移行のためにデータベースを準備する手順

1. データベース用の COMPATIBLE 設定が 11.0.0 よりも小さい場合は、読取り専用のトランスポート表領域を読取り / 書込みにします。

読取り専用トランスポート表領域は、Recovery Manager ではバックアップできないため、移行できません。

2. データベースが**フィジカル・スタンバイ・データベース**で、管理リカバリが開始されている場合は、管理リカバリを停止します。

たとえば、SYSDBA 権限を使用して SQL\*Plus をデータベースに接続し、次の文を実行して管理リカバリを停止します。

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
```

この端末ウィンドウを開いたままにします。

3. サーバー・パラメータ・ファイルまたは初期化パラメータ・ファイルを一時的な場所にコピーします。

次の例では、オペレーティング・システム・ユーティリティを使用してサーバー・パラメータ・ファイルをコピーします。

```
% cp spfileMYDB.ora orig_spfileMYDB.ora
```

4. 新しい端末ウィンドウで、Recovery Manager セッションを開始し、移行するデータベースに TARGET として接続します。必要に応じて、リカバリ・カタログに接続します。

5. ASM ディスク・グループにデータファイルをバックアップします。

次の例では、RUN コマンドを使用して**レベル0の増分バックアップ**を作成し、4つのチャネルを割り当ててバックアップの速度を上げます。この数は、必要に応じて増減させます。FORMAT 句では +DATA を指定します。これは、データベースの格納に使用される ASM ディスク・グループの名前です。

```
RUN
{
  ALLOCATE CHANNEL dev1 DEVICE TYPE DISK;
  ALLOCATE CHANNEL dev2 DEVICE TYPE DISK;
  ALLOCATE CHANNEL dev3 DEVICE TYPE DISK;
  ALLOCATE CHANNEL dev4 DEVICE TYPE DISK;
  BACKUP AS COPY
    INCREMENTAL LEVEL 0
    DATABASE
    FORMAT '+DATA'
    TAG 'ORA_ASM_MIGRATION';
}
```

6. **ブロック・チェンジ・トラッキング**がデータベースで有効になっている場合は、任意で、データベース・コピーを後でリカバリするのに使用できるレベル1の増分バックアップを作成します。

次の例では、前の手順で作成されたレベル0のバックアップの増分レベル1のコピーを作成します。

```
RUN
{
  ALLOCATE CHANNEL dev1 DEVICE TYPE DISK;
  ALLOCATE CHANNEL dev2 DEVICE TYPE DISK;
  ALLOCATE CHANNEL dev3 DEVICE TYPE DISK;
  ALLOCATE CHANNEL dev4 DEVICE TYPE DISK;
  BACKUP INCREMENTAL LEVEL 1
    FOR RECOVER OF COPY WITH TAG 'ORA_ASM_MIGRATION'
    DATABASE;
}
```

7. データベースが ARCHIVELOG モードであり、オープンしている場合は、オンライン・ログをアーカイブします。

次の例では、SQL コマンドを使用して、現行の REDO ログをアーカイブします。

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

8. 現在、データベース・インスタンスでサーバー・パラメータ・ファイルを使用している場合は、このファイルをバックアップします。

次の例では、サーバー・パラメータ・ファイルをバックアップします。

```
RMAN> BACKUP AS BACKUPSET SPFILE;
```

9. **ブロック・チェンジ・トラッキング**が有効になっている場合は無効にします。

次のコマンドを使用して、ブロック・チェンジ・トラッキングを無効にします。

```
RMAN> SQL "ALTER DATABASE DISABLE BLOCK CHANGE TRACKING";
```

10. フラッシュバック・データベースが有効になっている場合は無効にして、保証付きリストア・ポイントをすべて削除します。

---

---

**注意：** リカバリ領域を移行しない場合は、この手順をスキップします。

---

---

**フラッシュバック・ログ**は ASM に移行できないため、フラッシュバック・データベースを無効にする必要があります。次のコマンドを使用して、フラッシュバック・データベースを無効にします。

```
RMAN> SQL "ALTER DATABASE FLASHBACK OFF";
```

次のコマンドでは、Q106 という名前の保証付きリストア・ポイントが削除されます。

```
RMAN> SQL "DROP RESTORE POINT Q106";
```

11. 一貫性のある状態でデータベースを停止します。

次のコマンドは、データベースを停止します。

```
RMAN> SHUTDOWN IMMEDIATE;
```

## ASM へのデータベースの移行

次の手順の目的は、データベースの停止時間を最小限にすることです。移行するデータベースがプライマリまたはスタンバイかどうかによって、手順がわずかに異なることに注意してください。この手順では、26-3 ページの「[ASM へのデータベースの移行の準備](#)」と同じことを想定しています。リカバリ領域を ASM に移行しない場合は、手順の一部を変更する必要があることに注意してください。

---

**注意：** 次の手順では、SQL\*Plus と Recovery Manager とを切り替えます。そのため、各ユーティリティの端末ウィンドウは開いたままにしておきます。

---

### データベースを ASM に移行する手順

1. 26-3 ページの「[ASM へのデータベースの移行の準備](#)」の手順を実行します。

2. ASM ストレージでサーバー・パラメータ・ファイルをリストアまたは作成します。

データベースがサーバー・パラメータ・ファイルを使用しているかどうかによって、手順が異なります。

- データベースがサーバー・パラメータ・ファイルを使用している場合は、次のコマンドを使用して、ASM ディスク・グループにリストアします。ここで、*sid* はインスタンスの SID です。

```
RMAN> STARTUP MOUNT;
RMAN> RESTORE SPFILE TO '+DATA/spfilesid.ora';
RMAN> SHUTDOWN IMMEDIATE;
```

- データベースがサーバー・パラメータ・ファイルを使用していない場合は、このファイルを ASM で 1 つ作成します。次のように SQL\*Plus で CREATE SPFILE コマンドを実行します。ここで、*sid* はデータベースの SID です（コマンドは 2 行にわたります）。

```
SQL> CREATE SPFILE='+DATA/spfilesid.ora' FROM PFILE='?/dbs/initsid.ora';
```

その後、*\*/dbs* ディレクトリから *spfilesid.ora* および *initsid.ora* を削除し、次の内容の行を含む *initsid.ora* を新しく作成します。

```
SPFILE='+DATA/spfilesid.ora'
```

- Oracle Managed Files 初期化パラメータを ASM の場所に設定します。

---

**注意：** フラッシュ・リカバリ領域を移行しない場合は、DB\_RECOVERY\_FILE\_DEST および DB\_RECOVERY\_FILE\_DEST\_SIZE 初期化パラメータ設定を変更しないでください。ただし、オンライン REDO ログを移行する場合は、DB\_CREATE\_ONLINE\_LOG\_DEST\_n パラメータを ASM の場所に設定する必要があります。

---

DB\_CREATE\_FILE\_DEST と、オプションの DB\_CREATE\_ONLINE\_LOG\_DEST\_n 初期化パラメータを ASM ディスク・グループに設定します。データベースでリカバリ領域を使用する場合は、リカバリ領域の場所を ASM ディスク・グループに変更します。また、リカバリ領域のサイズも変更します。

次の例に示すように、SQL\*Plus でコマンドを実行します。この例では、フラッシュ・リカバリ領域のサイズが 100GB と想定され、フラッシュ・リカバリ領域に +FRA ディスク・グループが指定されています。

```
SQL> STARTUP FORCE NOMOUNT;
SQL> ALTER SYSTEM SET DB_CREATE_FILE_DEST='+DATA' SID='*';
SQL> ALTER SYSTEM SET DB_RECOVERY_FILE_DEST_SIZE=100G SID='*';
SQL> ALTER SYSTEM SET DB_RECOVERY_FILE_DEST='+FRA' SID='*';
```

- CONTROL\_FILES 初期化パラメータを ASM の場所に設定します。

フラッシュ・リカバリ領域を移行する場合は、SQL\*Plus で次のコマンドを入力してデータベース・インスタンスを再起動し、制御ファイルの場所をディスク・グループ +DATA および +FRA に設定します。

```
SQL> STARTUP FORCE NOMOUNT;
SQL> ALTER SYSTEM SET CONTROL_FILES='+DATA','+FRA' SCOPE=SPFILE SID='*';
```

フラッシュ・リカバリ領域を移行しない場合は、SQL\*Plus で次のコマンドを入力してデータベース・インスタンスを再起動し、制御ファイルの場所をディスク・グループ +DATA に設定します。

```
SQL> STARTUP FORCE NOMOUNT;
SQL> ALTER SYSTEM SET CONTROL_FILES='+DATA','+DATA' SCOPE=SPFILE SID='*';
```

- 制御ファイルを ASM に移行し、制御ファイルをマウントします。

Recovery Manager 端末に切り替えて、制御ファイルをリストアします。次の例では、original\_cf\_name は移行前の初期化パラメータ・ファイル内の制御ファイル名です。

```
RMAN> STARTUP FORCE NOMOUNT;
RMAN> RESTORE CONTROLFILE FROM 'original_cf_name';
RMAN> ALTER DATABASE MOUNT;
```

- データファイルを ASM に移行します。

Recovery Manager を使用して、26-3 ページの「ASM へのデータベースの移行の準備」の手順 5 で作成したデータベース・コピーに切り替えます。切替えによって、すべてのデータファイルの名前が ASM ディスク・グループ上のファイルに変更されます。その後、データベースをリカバリします。増分バックアップが行われた場合、Recovery Manager はリカバリ時にこれを適用します。たとえば、Recovery Manager プロンプトで次のコマンドを入力します。

```
SWITCH DATABASE TO COPY;
RUN
{
  ALLOCATE CHANNEL dev1 DEVICE TYPE DISK;
  ALLOCATE CHANNEL dev2 DEVICE TYPE DISK;
  ALLOCATE CHANNEL dev3 DEVICE TYPE DISK;
```

```

    ALLOCATE CHANNEL dev4 DEVICE TYPE DISK;
    RECOVER DATABASE;
}

```

7. データベースでブロック・チェンジ・トラッキングまたはフラッシュバック・データベースを使用する場合は、これらの機能を有効にします。

---

**注意：** リカバリ領域を移行しない場合は、フラッシュバック・データベースを無効にしていないため、有効にする必要はありません。

---

たとえば、SQL\*Plus で次の文を入力します。

```

SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING USING FILE '+DATA';
SQL> ALTER DATABASE FLASHBACK ON;

```

8. データベースを通常の操作モードに移行します。

通常の操作モードは、データベースがプライマリまたはスタンバイかどうかによって異なります。

- データベースがプライマリ・データベースの場合は、次のようにデータベースをオープンします。

```

SQL> ALTER DATABASE OPEN;

```

- スタンバイ・データベースの場合は、管理リカバリ・モードを次のように再開します。

```

SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE;

```

9. ASM で一時ファイルを削除し、再作成します。

SQL\*Plus を使用して、一時ファイルを再作成します。次の例では、元のストレージの一時ファイルの名前は *tempfile\_name* です。一時表領域の名前は、*temp\_tbs\_name* です。

```

SQL> ALTER DATABASE TEMPFILE 'tempfile_name' DROP;
SQL> ALTER TABLESPACE temp_tbs_name ADD TEMPFILE;

```

10. オンライン REDO ログ・ファイルを移行します。

これがプライマリ・データベースの場合は、ASM に新しいログ・グループ・メンバーを追加して古いメンバーを削除します。次の PL/SQL スクリプトを使用すると、オンライン REDO ログ・グループを ASM ディスク・グループに移行できます。この PL/SQL スクリプトでは、手順 3 で指定した Oracle Managed Files 初期化パラメータが設定されていることを想定しています。

### 例 26-1 オンライン REDO ログの移行

```

SET SERVEROUTPUT ON;
DECLARE
    CURSOR rlc IS
        SELECT GROUP# GRP, THREAD# THR, BYTES, 'NO' SRL
        FROM   V$LOG
        UNION
        SELECT GROUP# GRP, THREAD# THR, BYTES, 'YES' SRL
        FROM   V$STANDBY_LOG
        ORDER BY 1;
    stmt    VARCHAR2(2048);
BEGIN
    FOR rlcRec IN rlc LOOP
        IF (rlcRec.srl = 'YES') THEN
            stmt := 'ALTER DATABASE ADD STANDBY LOGFILE THREAD ' ||
                rlcRec.thr || ' SIZE ' || rlcRec.bytes;

```

```

EXECUTE IMMEDIATE stmt;
stmt := 'ALTER DATABASE DROP STANDBY LOGFILE GROUP ' || rlcRec.grp;
EXECUTE IMMEDIATE stmt;
ELSE
  stmt := 'ALTER DATABASE ADD LOGFILE THREAD ' ||
    rlcRec.thr || ' SIZE ' || rlcRec.bytes;
EXECUTE IMMEDIATE stmt;
BEGIN
  stmt := 'ALTER DATABASE DROP LOGFILE GROUP ' || rlcRec.grp;
  DEMS_OUTPUT.PUT_LINE(stmt);
  EXECUTE IMMEDIATE stmt;
EXCEPTION
  WHEN OTHERS THEN
    EXECUTE IMMEDIATE 'ALTER SYSTEM SWITCH LOGFILE';
    EXECUTE IMMEDIATE 'ALTER SYSTEM CHECKPOINT GLOBAL';
    EXECUTE IMMEDIATE stmt;
END;
END IF;
END LOOP;
END;
/

```

11. 必要に応じて、次のように古いフラッシュ・リカバリ領域のバックアップおよびコピーを ASM に移行します。

- a. リカバリ領域に外部アーカイブ・ログが存在する場合は、このログを ASM に移行することはできません。Recovery Manager プロンプトで次のコマンドを実行します。

```

RMAN> DELETE REMOTE ARCHIVELOG ALL;

```

- b. アーカイブ REDO ログ・ファイル、バックアップ・セットおよびデータファイルのコピーを ASM にバックアップします。たとえば、Recovery Manager プロンプトで次のコマンドを実行します。

```

RUN
{
  ALLOCATE CHANNEL dev1 DEVICE TYPE DISK;
  ALLOCATE CHANNEL dev2 DEVICE TYPE DISK;
  ALLOCATE CHANNEL dev3 DEVICE TYPE DISK;
  ALLOCATE CHANNEL dev4 DEVICE TYPE DISK;

  BACKUP AS COPY ARCHIVELOG ALL DELETE INPUT;
  BACKUP BACKUPSET ALL DELETE INPUT;
  BACKUP AS COPY DATAFILECOPY ALL DELETE INPUT;
}

```

## ASM から代替ストレージへのデータベースの移行

ASM から代替ストレージ・システムへのデータベースの移行は、基本的に ASM への移行の逆です。次のように、26-3 ページの「ASM へのデータベースの移行の準備」および 26-5 ページの「ASM へのデータベースの移行」の手順を変更します。

- 手順で Oracle Managed Files の場所を指定する場合は、代替ストレージの場所を使用するように手順を変更します。
- BACKUP コマンドの FORMAT 句で ASM の場所を指定する場合は、バックアップの書式を代替ストレージの場所に変更します。
- SQL 文で使用するファイル名が ASM の場所である場合は、これを代替ストレージの場所のファイル名に変更します。



## ASM ディスク・グループ間でのデータファイルの移動

ARCHIVELOG モードのデータベースで、アクティブなデータファイルを ASM ディスク・グループ間で移動する必要がある場合があります。この場合は、BACKUP AS COPY を使用してデータファイルを新しいディスク・グループにコピーし、SET NEWNAME および SWITCH コマンドを使用して制御ファイル内のデータファイルの名前を変更します。

この例では、ASM ディスク・グループ ASMSK2 および ASMSK1 を使用することを想定しています。データファイル +ASMSK2/rdbms/datafile/tbs\_5.256.565313879 を、ディスク・グループ ASMSK1 に移動します。

### ASM ディスク・グループ間でデータファイルを移動する手順

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. 次の問合せは、バックアップ・ピース名を示しています。

たとえば、Recovery Manager をターゲット・データベースに接続した後、次の REPORT コマンドを実行します。移動するファイルのデータファイル番号およびデータファイル名の両方を書き留めます。

```
REPORT SCHEMA;
```

3. 新しい ASM ディスク・グループにデータファイルをバックアップします。

たとえば、次の BACKUP AS COPY コマンドを発行して、ASMSK2 上のデータファイルを ASMSK1 にバックアップします。

```
BACKUP AS COPY
  DATAFILE "+ASMSK2/rdbms/datafile/tbs_5.256.565313879"
  FORMAT "+ASMSK1";
```

次の例に示すように、データファイル番号でデータファイルを指定することもできます。

```
BACKUP AS COPY
  DATAFILE 23
  FORMAT "+ASMSK1";
```

4. 新しいディスク・グループに移動するデータファイルの名前を検索してオフラインにします。

たとえば、Recovery Manager クライアントで次の SQL コマンドを実行します。データファイルの名前を 2 つの一重引用符で囲むことに注意してください。

```
SQL "ALTER DATABASE DATAFILE
  ' '+ASMSK2/rdbms/datafile/tbs_5.256.565313879 ' ' OFFLINE";
```

5. 制御ファイルが、新しく作成したデータファイルのコピーをポイントするように変更します。

たとえば、Recovery Manager クライアントで SWITCH ... TO COPY コマンドを次のように実行します。SWITCH の TO COPY オプションは、データファイルをデータファイルの最新コピーに切り替えます。データファイルは、名前または番号で指定できます。

```
SWITCH DATAFILE "+ASMSK2/rdbms/datafile/tbs_5.256.565313879" TO COPY;
```

このコマンドの出力には、データファイルの新しい名前が表示されます。

6. 名前を変更したデータファイルをリカバリします。

たとえば、Recovery Manager クライアントで RECOVER コマンドを次のように実行します。データファイルは、名前または番号で指定できます。

```
RECOVER DATAFILE "+ASMSK1/rdbms/datafile/tbs_5.256.603733209";
```

7. データファイルをオンラインにします。

たとえば、Recovery Manager クライアントで SQL コマンドを次のように実行します。

```
SQL "ALTER DATABASE DATAFILE  
    ' '+ASMDSK1/rdbms/datafile/tbs_5.256.603733209 ' ' ONLINE";
```

8. 元の ASM ディスク・グループからデータファイルのコピーを削除します。

この例では、+ASMDSK2/rdbms/datafile/tbs\_5.256.565313879 が ASMDSK2 内の元のデータファイルです。このデータファイルに対して SET NEWNAME および SWITCH コマンドを発行したため、元のファイルは現在 Recovery Manager にデータファイル・コピーとして記録されています。このファイルは、Recovery Manager クライアントで DELETE コマンドを次のように実行すると削除できます。

```
DELETE DATAFILECOPY "+ASMDSK2/rdbms/datafile/tbs_5.256.603733209";
```

# 第 VIII 部

---

## ユーザー管理のバックアップおよびリカバリの実行

第 VIII 部では、ユーザー管理のバックアップおよびリカバリ計画（Recovery Manager を必要としない計画）を使用してバックアップおよびリカバリを実行する方法について説明します。第 VIII 部に含まれる章は次のとおりです。

- [第 27 章「ユーザー管理データベース・バックアップの作成」](#)
- [第 28 章「ユーザー管理のデータベースのフラッシュバックおよびリカバリの実行」](#)
- [第 29 章「ユーザー管理のリカバリの実行：高度な例」](#)



---

## ユーザー管理データベース・バックアップの作成

この章では、ユーザー管理バックアップおよびリカバリ計画（Recovery Manager を必要としない計画）で Oracle Database をバックアップする方法について説明します。

この章の内容は、次のとおりです。

- V\$ ビューの問合せによるバックアップ情報の取得
- データベース全体のユーザー管理バックアップの作成
- 表領域およびデータファイルのユーザー管理バックアップの作成
- 制御ファイルのユーザー管理バックアップの作成
- アーカイブ REDO ログのユーザー管理バックアップの作成
- SUSPEND モードでのユーザー管理バックアップの作成
- RAW デバイスへのユーザー管理バックアップの作成
- Volume Shadow Copy Service (VSS) を使用したバックアップの作成
- ユーザー管理データファイル・バックアップの検証

## V\$ ビューの問合せによるバックアップ情報の取得

バックアップを作成する前に、データベース内のすべてのファイルを確認して、バックアップするファイルを決定する必要があります。V\$ ビューを使用すると、この情報を取得できます。

### バックアップ前のデータベース・ファイルの表示

データベースのデータファイルおよび制御ファイルを確認するには、V\$DATAFILE および V\$CONTROLFILE を使用します。次に示す手順は、これらのファイルの名前を手動で付けた場合も、Oracle Managed Files で付けた場合も、同様に使用できます。

---

**注意：** オンライン REDO ログ・ファイルはバックアップしないでください。

---

#### データファイルおよび制御ファイルを表示する手順

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

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

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

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

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

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

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

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

### オンライン表領域バックアップのデータファイルのステータスの確認

データファイルが現行のオンライン表領域のバックアップの一部であるかどうかを確認するには、V\$BACKUP ビューを問い合わせます。

このビューはユーザー管理のオンライン表領域バックアップでのみ有効です。Recovery Manager のバックアップおよびオフライン表領域のバックアップでは、表領域のデータファイルを**バックアップ・モード**に設定する必要がないためです。一部のユーザー管理バックアップ手順では、分裂ブロックの発生を防ぐため、表領域をバックアップ・モードにする必要があります。ただし、バックアップ・モードでは、データベースへの更新によって通常より多くの REDO が作成されます。

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

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

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

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

次の出力例は、tools および users 表領域が現在 ACTIVE ステータスであることを示しています。

| TB_NAME | DF# | DF_NAME                          | STATUS |
|---------|-----|----------------------------------|--------|
| TOOLS   | 7   | /oracle/oradata/trgt/tools01.dbf | ACTIVE |
| USERS   | 8   | /oracle/oradata/trgt/users01.dbf | ACTIVE |

STATUS 列に NOT ACTIVE が表示されている場合、ファイルが現在バックアップ・モードではない (ALTER TABLESPACE ... BEGIN BACKUP または ALTER DATABASE BEGIN BACKUP 文を実行していない) ことを意味します。ACTIVE が表示されている場合、ファイルが現在バックアップ・モードであることを意味します。

## データベース全体のユーザー管理バックアップの作成

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

データベースを ARCHIVELOG モードまたは NOARCHIVELOG モードのどちらかで実行していても、データベース全体のバックアップを作成できます。ただし、データベースを NOARCHIVELOG モードで実行する場合は、バックアップ前にデータベースを正しく停止して、一貫性のあるバックアップを作成する必要があります。

一貫性のあるデータベース全体のバックアップによって作成されたバックアップ・ファイルのセットでは、すべてのファイルで同じ SCN にチェックポイントが設定されているため、一貫性があります。一貫性のあるデータベースのバックアップは、リカバリを実行せずにリストアできます。データベースを ARCHIVELOG モードで実行している場合は、バックアップ・ファイルをリストアした後で、追加のリカバリ手順を実行してデータベースをより新しい時点までリカバリできます。また、データベースが ARCHIVELOG モードの場合には、一貫性のないデータベース全体のバックアップを作成することもできます。

制御ファイルは、データベースのリストアおよびリカバリに重要な役割を果たします。ARCHIVELOG モードで実行しているデータベースの場合には、ALTER DATABASE BACKUP CONTROLFILE TO '*filename*' 文を使用して、制御ファイルをバックアップしておくことをお勧めします。

**参照：** 制御ファイルのバックアップの詳細は、27-11 ページの「[制御ファイルのユーザー管理バックアップの作成](#)」を参照してください。

## 一貫性のあるデータベース全体のバックアップの作成

この項では、オペレーティング・システム・ユーティリティを使用してデータベースをバックアップする方法を説明します。

### 一貫性のあるデータベース全体のバックアップを作成する手順

1. データベースがオープンしている場合は、SQL\*Plus を使用して、NORMAL、IMMEDIATE または TRANSACTIONAL オプションを指定してデータベースを停止します。
2. オペレーティング・システム・ユーティリティを使用して、すべてのデータファイルと、初期化パラメータ・ファイルの CONTROL\_FILES パラメータで指定されたすべての制御ファイルのバックアップを作成します。また、初期化パラメータ・ファイルおよびその他の Oracle 製品の初期化ファイルもバックアップします。これらのファイルを検索するには、\*.ora を使用して、Oracle ホーム・ディレクトリから開始してすべてのサブディレクトリを再帰的に検索します。

たとえば、次に示すとおり、データファイル、制御ファイルおよびアーカイブ・ログを /disk2/backup にバックアップできます。

```
% cp $ORACLE_HOME/oradata/trgt/*.dbf /disk2/backup
% cp $ORACLE_HOME/oradata/trgt/arch/* /disk2/backup/arch
```

3. SQL\*Plus で STARTUP コマンドを使用して、データベースを再起動します。

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

## 表領域およびデータファイルのユーザー管理バックアップの作成

表領域およびデータファイルのユーザー管理バックアップを作成する方法は、ファイルがオフラインかオンラインかによって異なります。

### オフラインの表領域およびデータファイルのユーザー管理バックアップの作成

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

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

### オフライン表領域をバックアップする手順

1. 表領域のバックアップを開始する前に、DBA\_DATA\_FILES ビューを問い合わせ、表領域のデータファイルを確認します。たとえば、users 表領域をバックアップするとします。SQL\*Plus で次の文を入力します。

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

| TABLESPACE_NAME | FILE_NAME                        |
|-----------------|----------------------------------|
| USERS           | /oracle/oradata/trgt/users01.dbf |

この例の /oracle/oradata/trgt/users01.dbf は、users 表領域の中のデータファイルに対応する完全指定のファイル名です。



2. 可能であれば、通常優先度で表領域をオフラインにします。これによって、後で、リカバリを行わなくても表領域を確実にオンラインに戻せます。たとえば、次のように入力します。

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

3. オフライン・データファイルをバックアップします。たとえば、次のように入力します。

```
% cp /oracle/oradata/trgt/users01.dbf /d2/users01_'date "+%m_%d_%y"'.dbf
```

4. 表領域をオンラインにします。たとえば、次のように入力します。

```
ALTER TABLESPACE users ONLINE;
```

---

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

---

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

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

## オンラインの表領域およびデータファイルのユーザー管理バックアップの作成

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

---

**注意：** 一時表領域はバックアップしないでください。

---

### オンラインの読取り / 書込み表領域のユーザー管理バックアップの作成

表領域がオンラインで、データベースがオープンしているときに、データファイルのユーザー管理バックアップを作成するには、読取り / 書込み表領域をバックアップ・モードに設定する必要があります。ALTER TABLESPACE ... BEGIN BACKUP 文を使用すると、表領域をバックアップ・モードに設定できます。バックアップ・モードでは、変更されたデータ・ブロック全体が REDO ストリームにコピーされます。ユーザーが ALTER TABLESPACE ... END BACKUP または ALTER DATABASE END BACKUP 文を使用して表領域のバックアップ・モードを終了すると、**データファイル・チェックポイント SCN** が現在の**データベース・チェックポイント SCN** まで進みます。

この方法でバックアップされたデータファイルをリストアすると、リカバリが必要な場合は、REDO ログ・ファイルの適切なセットを適用するように求められます。REDO ログには、データファイルをリカバリし、データファイルを一貫性のある状態にするために必要なすべての変更が含まれています。

### オープン状態のデータベース中のオンラインの読取り / 書込み表領域をバックアップする手順

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

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

| TABLESPACE_NAME | FILE_NAME                        |
|-----------------|----------------------------------|
| USERS           | /oracle/oradata/trgt/users01.dbf |
| USERS           | /oracle/oradata/trgt/users02.dbf |

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

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

---

**注意：** BEGIN BACKUP を使用してオンライン表領域のバックアップの開始をマークしなかったか、またはオンライン表領域のコピーを開始する前にその文が完了していなかった場合は、生成されたデータファイルのコピーをその後のリカバリ操作で使用できません。このようなバックアップのリカバリは危険であり、エラーが発生してデータの一貫性がなくなる可能性があります。たとえば、リカバリ操作を試みると **ファジー・ファイル** の警告が発行され、データベースの一貫性が失われ、データベースをオープンできなくなる可能性があります。

---

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

```
% cp /oracle/oradata/trgt/users01.dbf /d2/users01_'date "+%m_%d_%y"' .dbf
% cp /oracle/oradata/trgt/users02.dbf /d2/users02_'date "+%m_%d_%y"' .dbf
```

- オンライン表領域のデータファイルをバックアップした後、END BACKUP オプションを指定した SQL 文 ALTER TABLESPACE を実行します。たとえば、次の文は、表領域 users のオンライン・バックアップを終了します。

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

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

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

---

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

---

## オンラインの読取り / 書込み表領域の複数のユーザー管理バックアップの作成

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

**オンライン表領域のパラレル・バックアップ** バックアップが必要な複数の表領域のデータファイルのコピーを、同時にバックアップ・モードで作成できます。ただし、すべての表領域を一度にオンライン・モードに設定すると、それらの表領域で多くの更新が実行されている場合は、大規模な REDO ログが生成される可能性があります。REDO には、変更された各データファイルの変更された各データ・ブロックのコピーを含める必要があるためです。次に示す手順を実行する前に、生成される REDO のサイズを検討してください。

### オンライン表領域をパラレルにバックアップする手順

1. 必要なすべての ALTER TABLESPACE 文を一度に発行して、オンライン表領域でバックアップの準備をします。たとえば、次のように表領域 users、tools および indx をバックアップ・モードにします。

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

すべての表領域をバックアップするには、次のコマンドを使用します。

```
SQL> ALTER DATABASE BEGIN BACKUP;
```

2. オンライン表領域のすべてのファイルをバックアップします。たとえば、Linux または UNIX ユーザーは、次のコマンドを実行して、\*.dbf という接尾辞を持つデータファイルをバックアップできます。

```
% cp $ORACLE_HOME/oradata/trgt/*.dbf /disk2/backup/
```

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

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

すべてのデータファイルを一度に処理する場合は、ALTER TABLESPACE のかわりに ALTER DATABASE コマンドを使用できます。

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

4. 表領域のバックアップのリカバリに必要な REDO を後でメディア・リカバリに使用できるように、オンライン REDO ログをアーカイブします。たとえば、次のように入力します。

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

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

### オンライン表領域をシリアルにバックアップする手順

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

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

この場合は、すべての表領域を同時にバックアップ・モードに設定する ALTER DATABASE BEGIN BACKUP は使用しません。オンライン・モードの表領域に関する不要な REDO ログ情報が生成されるためです。

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

```
% cp /oracle/oradata/trgt/users01.dbf /d2/users01_'date "+%m_%d_%y"' .dbf
```

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

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

4. 残りの各表領域で、この手順を繰り返します。

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

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

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

次の場合には、表領域のバックアップが失敗し、不完全になることがあります。

- バックアップが完了したが、ユーザーが ALTER TABLESPACE ... END BACKUP 文を実行していない場合。
- インスタンス障害または SHUTDOWN ABORT によってバックアップが中断された場合。

クラッシュ・リカバリが必要なときに、バックアップ・モードのデータファイルをオープンしようとした場合、リカバリ・コマンドが発行されるか、またはデータファイルのバックアップ・モードが終了するまでは、データベースはオープンされません。

たとえば、起動時に次のようなメッセージが表示される場合があります。

```
ORA-01113: file 12 needs media recovery
ORA-01110: data file 12: '/oracle/dbs/tbs_41.f'
```

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

高可用性が必要な場合および DBA がデータベースを監視していない場合は、ユーザーの介入を必要とする事態は回避する必要があります。このために、次の内容のクラッシュ・リカバリ・スクリプトを作成しておくことができます。

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

ALTER DATABASE END BACKUP を含む自動化されたクラッシュ・リカバリ・スクリプトは、次の場合に特に有効です。

- Oracle Real Application Clusters (Oracle RAC) 構成内のすべてのノードで障害が発生した場合。
- コールド・フェイルオーバー・クラスタ (最初のノードで障害が発生した場合に、2 番目のノードでデータベースをマウントしてリカバリする必要がある、RAC 構成ではないクラスタ) 内の 1 つのノードで障害が発生した場合。

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

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

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

### 複数の表領域でバックアップ・モードを同時に終了する手順

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

```
SQL> STARTUP MOUNT
```

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

```
SQL> SELECT * FROM V$BACKUP WHERE STATUS = 'ACTIVE';
FILE#          STATUS          CHANGE#    TIME
-----
          12 ACTIVE                20863 25-NOV-02
          13 ACTIVE                20863 25-NOV-02
          20 ACTIVE                20863 25-NOV-02
3 rows selected.
```

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

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

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

---

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

---

**SQL\*Plus の RECOVER コマンドを使用したバックアップ・モードの終了** オンライン・バックアップが失敗した場合の対処方法は、ALTER DATABASE END BACKUP 文を発行する以外に、SQL\*Plus の RECOVER コマンドを実行することもできます。他のユーザーがバックアップをリストアップしているかどうか分からない場合は、この方法が有効です。他のユーザーが実際にバックアップをリストアップしていた場合に、RECOVER コマンドによってバックアップが最新に更新されるためです。ALTER DATABASE END BACKUP または ALTER TABLESPACE ... END BACKUP 文は、ファイルが現行のものであることが確実な場合にのみ実行してください。

---

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

---

### RECOVER コマンドを使用して表領域のバックアップ・モードを終了する手順

- データベースをマウントします。たとえば、次のように入力します。

```
SQL> STARTUP MOUNT
```

- データベースを通常どおりにリカバリします。たとえば、次のように入力します。

```
SQL> RECOVER DATABASE
```

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

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

**参照：** データベースのリカバリについては、第 28 章「ユーザー管理のデータベースのフラッシュバックおよびリカバリの実行」を参照してください。

## 読取り専用表領域のユーザー管理バックアップの作成

オンラインの読取り専用表領域をバックアップする場合は、オンライン・データファイルをバックアップするのみで実行できます。データベースではデータファイルに対する変更が許可されないため、表領域をバックアップ・モードに設定する必要はありません。

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

**参照：** 表領域の転送方法については、『Oracle Database 管理者ガイド』を参照してください。

### オープン状態のデータベースのオンラインの読取り専用表領域をバックアップする手順

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

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

2. 読取り専用表領域のバックアップを開始する前に、DBA\_DATA\_FILES データ・ディクショナリ・ビューを問い合わせ、表領域のすべてのデータファイルを確認します。たとえば、history 表領域をバックアップする場合、次の問合せを実行します。

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

| TABLESPACE_NAME | FILE_NAME                          |
|-----------------|------------------------------------|
| HISTORY         | /oracle/oradata/trgt/history01.dbf |
| HISTORY         | /oracle/oradata/trgt/history02.dbf |

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

```
% cp $ORACLE_HOME/oradata/trgt/history*.dbf /disk2/backup/
```

---

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

---

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

```
% expdp DIRECTORY=dpump_dir1 DUMPFILE=hs.dmp TRANSPORT_TABLESPACES=history
LOGFILE=tts.log
```

**参照：** DBA\_DATA\_FILES ビューおよび DBA\_TABLESPACES ビューの詳細は、『Oracle Database リファレンス』を参照してください。

## 制御ファイルのユーザー管理バックアップの作成

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

### 制御ファイルのバイナリ・ファイルへのバックアップ

制御ファイルをバックアップする第一の手段は、SQL 文を使用してバイナリ・ファイルを生成する方法です。バイナリ・バックアップにはアーカイブ・ログ履歴、読取り専用およびオフライン表領域のオフライン範囲、バックアップ・セットとコピー (Recovery Manager を使用する場合) などの追加情報が含まれるため、トレース・ファイル・バックアップよりもバイナリ・バックアップを使用することをお勧めします。COMPATIBLE が 10.2 以上の場合、制御ファイルのバイナリ・バックアップには、一時ファイル・エントリが含まれます。

#### 構造変更の後に制御ファイルをバックアップする手順

1. データベースを変更します。たとえば、新規の表領域を作成します。

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

2. 出力バイナリ・ファイルのファイル名を指定して、データベースの制御ファイルをバックアップします。次の例は、/disk1/backup/cf.bak に制御ファイルをバックアップします。

```
ALTER DATABASE BACKUP CONTROLFILE TO '/disk1/backup/cf.bak' REUSE;
```

REUSE を指定すると、新規の制御ファイルで既存の制御ファイルを上書きできます。

### 制御ファイルのトレース・ファイルへのバックアップ

制御ファイルは、CREATE CONTROLFILE 文が含まれるテキスト・ファイルにバックアップできます。トレース・ファイルを編集し、トレース・ファイルの作成時点で現行のものであった制御ファイルに基づいて、新しい制御ファイルを作成するスクリプトを作成することができます。

SQL 文で RESETLOGS オプションも NORESETLOGS オプションも指定しなかった場合は、トレース・ファイルに RESETLOGS と NORESETLOGS の両方のオプション用の制御ファイルが含まれます。ALTER TABLESPACE ... ADD TEMPFILE 文を使用すると、出力に一時ファイル・エントリが含まれます。

NORMAL モードでオフラインにされた表領域または読取り専用表領域をリカバリしないように、CREATE CONTROLFILE 文を編集してそれらを除外してください。再作成された制御ファイルを使用してデータベースをオープンすると、データベースではこれらの省略されたファイルは MISSING としてマークされます。ALTER DATABASE RENAME FILE 文を実行すると、それらのファイルを元のファイル名に戻すことができます。

CREATE CONTROLFILE 文を含むトレース・ファイルは、DIAGNOSTIC\_DEST 初期化パラメータに指定されたサブディレクトリに格納されます。CREATE CONTROLFILE 文が書き込まれるトレース・ファイルの名前および場所については、データベースのアラート・ログで確認できます。アラート・ログの場所を確認する方法については、『Oracle Database 管理者ガイド』を参照してください。

#### 制御ファイルをトレース・ファイルにバックアップする手順

1. データベースをマウントまたはオープンします。
2. 次の SQL 文を実行します。

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

**参照：** CREATE CONTROLFILE 文に含まれる、読取り専用ファイル、NORMAL モードでオフライン化されたファイルおよび一時ファイルに関連する特殊な問題については、29-8 ページの「[再作成された制御ファイルを使用した読取り専用ファイルのリカバリ](#)」を参照してください。

## アーカイブ REDO ログのユーザー管理バックアップの作成

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

### アーカイブ REDO ログをバックアップする手順

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

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

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

```
% cp $ORACLE_HOME/oracle/trgt/arch/* /disk2/backup/arch
```

**参照：** データ・ディクショナリ・ビューの詳細は、『Oracle Database リファレンス』を参照してください。

## SUSPEND モードでのユーザー管理バックアップの作成

この項の内容は、次のとおりです。

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

### 一時停止 / 再開機能

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

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

ほとんどの場合は、ミラーの分割によるバックアップを作成するために SUSPEND/RESUME を使用する必要はありませんが、使用しているシステムで、ボリュームを分割する前にデータベース・キャッシュの[使用済バッファ](#)を排除しておく必要がある場合には、このコマンドが必要になります。RAID デバイスの中には、分割操作の実行中に書き込みを一時停止できるものがあります。使用しているシステムでこの機能を使用できるかどうかは、RAID のベンダーに確認してください。

ALTER SYSTEM SUSPEND 文は、データファイル・ヘッダー、データファイルおよび制御ファイルに対する I/O を停止して、データベースを一時停止します。データベースが一時停止すると、既存のすべての I/O 操作は完了できますが、データベースに対する新規の I/O アクセスの試行はキューされます。



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

## 一時停止中のデータベースのバックアップの作成

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

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

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

SUSPEND/RESUME 機能には次の制限があります。

- RAC 構成では、オリジナル・ノードの一時停止中に新規インスタンスを起動できません。
- ALTER SYSTEM SUSPEND 文または ALTER SYSTEM RESUME 文からはチェックポイントは開始されません。
- データベースの一時停止中は、IMMEDIATE、NORMAL または TRANSACTIONAL オプションを指定して SHUTDOWN コマンドを発行できません。
- すでに一時停止中のデータベースに対して SHUTDOWN ABORT を発行すると、データベースが再度アクティブになります。この操作で、メディア・リカバリまたはクラッシュ・リカバリのハンガアップを回避します。

### SUSPEND モードでミラーの分割によるバックアップを作成する手順

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

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

データベースのすべての表領域をバックアップする場合は、かわりに次のコマンドを使用できます。

```
ALTER DATABASE BEGIN BACKUP;
```

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

```
ALTER SYSTEM SUSPEND;
```

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

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

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

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

```
ALTER SYSTEM RESUME;
```
- V\$INSTANCE を問い合せて、データベースがアクティブであることを確認します。たとえば、次のように入力します。

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

```
DATABASE_STATUS
-----
ACTIVE
```
- 指定した表領域のバックアップ・モードを終了します。たとえば、表領域 `users` のバックアップ・モードを終了するには、次のように入力します。

```
ALTER TABLESPACE users END BACKUP;
```
- 通常のバックアップの場合と同様に、制御ファイルをコピーして、オンライン REDO ログをアーカイブします。

---

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

---

**参照：**

- 「[Recovery Manager を使用したミラーの分割によるバックアップの実行](#)」(9-9 ページ)
- SUSPEND/RESUME 機能の詳細は、『Oracle Database 管理者ガイド』を参照してください。
- ALTER SYSTEM SUSPEND 構文については、『Oracle Database SQL 言語リファレンス』を参照してください。

## RAW デバイスへのユーザー管理バックアップの作成

**RAW デバイス**とは、ファイル・システムを持たないディスクまたはパーティションです。つまり、RAW デバイスに保管できるファイルは1つのみです。RAW デバイスにファイルをバックアップする場合は、オペレーティング・システム固有の問題が発生します。後続の項では、UNIX、Linux および Windows でのこれらの問題の一部について説明します。

### Linux および UNIX での RAW デバイスへのバックアップ

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

dd を効率的に使用するには、ご使用のデータベースに基づいて適切なオプションを指定する必要があります。表 27-1 に、dd に使用するオプションに影響を及ぼすデータベースの詳細を示します。

表 27-1 dd の使用に重要なデータベースの詳細

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

表 27-1 の情報を使用して、表 27-2 に示す dd オプションを設定できます。

表 27-2 dd コマンドのオプション

| オプション | 説明  |
|-------|---|
| if    | 入力ファイル、つまり読み込むファイルの名前。  |
| of    | 出力ファイル、つまり書き出すファイルの名前。  |
| bs    | dd でデータのコピーに使用されるバッファ・サイズ。  |
| skip  | RAW オフセットが存在する場合に、入力 RAW デバイスでスキップする dd バッファの数。たとえば、64KB の RAW オフセットを持つ RAW デバイス上のファイルをバックアップする場合、dd バッファ・サイズが 8KB のときは、skip=8 と指定すると、64KB のオフセットからコピーを開始できます。  |
| seek  | RAW オフセットが存在する場合に、出力 RAW デバイスでスキップする dd バッファの数。たとえば、64KB の RAW オフセットを持つ RAW デバイス上にファイルをバックアップする場合、dd バッファ・サイズが 8KB のときは、skip=8 と指定すると、64KB のオフセットからコピーを開始できます。  |
| count | dd でコピーする、入力 RAW デバイス上のブロックの数。RAW デバイスからファイル・システムにコピーするときには、コピーする正確なブロック数を指定するのが最良です。そうでない場合、Oracle データファイルで使用されない、RAW ボリュームの最後の余分な領域がファイル・システムにコピーされることになります。<br><br>入力ファイルの合計サイズにブロック 0 (ゼロ) を必ず含めてください。たとえば、dd ブロック・サイズが 8KB のとき、30720KB のデータファイルをバックアップする場合は、count=3841 と設定できます。count にこの値を指定すると、実際には 30728KB がバックアップされます。追加の 8KB は Oracle ブロック 0 (ゼロ) 用です。 |

RAW デバイスはバックアップの入力または出力デバイスに使用できるため、4つのバックアップ・シナリオを想定できます。dd で選択できるオプションは、表 27-3 に示すように、どのシナリオを選択するかによって異なります。

表 27-3 dd バックアップのシナリオ

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

## Linux および UNIX での dd ユーティリティを使用したバックアップの例

この項で示す dd ユーティリティの使用法の例では、次のように想定しています。

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

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

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

次の例では、RAW デバイスからファイル・システムにバックアップします。

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

次の例では、ファイル・システムから RAW デバイスにバックアップします。

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

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

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

## Windows での RAW デバイスへのバックアップ

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

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

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

たとえば、次の名前は RAW ファイル名として使用できます。

```
%%.¥G:  
%%.¥PHYSICALDRIVE3
```

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

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

`OCOPY` のオンライン・マニュアルを表示するには、Windows プロンプトに `OCOPY` とのみ入力します。出力例を次に示します。

Usage of `OCOPY`:

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

次の表に、`OCOPY` の重要なオプションを示します。

表 27-4 `OCOPY` のオプション

| オプション          | 処理   |
|----------------|--|
| <code>b</code> | 入力ファイルを複数の出力ファイルに分割します。このオプションは、入力ファイルよりも小さいデバイスにバックアップする場合に有効です。                              |
| <code>r</code> | 複数の入力ファイルを組み合わせ、1つの出力ファイルに書き込みます。このオプションは、 <code>-b</code> オプションを指定して作成したバックアップをリストアする場合に有効です。 |

## `OCOPY` を使用したバックアップの例

この例では、次のように想定しています。

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

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

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

## `OCOPY` の `-b` および `-r` オプションの指定の例

この例では、次のように想定しています。

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

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

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

ドライブ `E:` が一杯になると、別のディスクを使用できます。この方法で、データファイル 7 のバックアップを複数のファイルに分割できます。

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

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

## Volume Shadow Copy Service (VSS) を使用したバックアップの作成

**Volume Shadow Copy Service (VSS)** は、アプリケーションでシャドウ・コピーと呼ばれる一貫性のあるスナップショットの作成を可能にする Windows API のセットです。**Oracle VSSライター**は、Windows システム上でのサービスとして実行され、VSS 対応アプリケーションと統合されます。これらのアプリケーションを使用すると、Oracle インスタンスによって管理されているデータベース・ファイルのスナップショットを作成できます。たとえば、Oracle Database を読取り / 書き込みでオープンしている場合は、その Oracle データベースのシャドウ・コピーを作成できます。

**参照：** VSS 対応アプリケーションでデータベースをバックアップおよびリカバリする方法については、Oracle Database のプラットフォーム・ガイドを参照してください。

## ユーザー管理データファイル・バックアップの検証

バックアップがリカバリに使用可能なものであることを確認するために、バックアップを定期的に検証する必要があります。

## データファイル・バックアップのリストアのテスト

データファイル・バックアップの有効性を確認する最適な方法は、別のホストにバックアップをリストアして、必要に応じてメディア・リカバリを実行し、データベースのオープンを試行する方法です。このオプションでは、リストア手順のために別のホストを使用する必要があります。

**参照：** SQL\*Plus でのファイルのリカバリ方法については、28-8 ページの「データベースの完全リカバリの実行」を参照してください。

## DBVERIFY ユーティリティの実行

DBVERIFY プログラムは、オフライン・データファイルに対して物理的なデータ構造の整合性チェックを実行する、外部のコマンドライン・ユーティリティです。DBVERIFY は主に、リストア前にデータファイルのユーザー管理バックアップが有効かどうかを確認する必要がある場合や、データ破損の問題が発生した場合の診断ツールとして使用します。

DBVERIFY の名前と位置はオペレーティング・システムごとに異なります。たとえば、Linux または UNIX 上でデータファイル users01.dbf の整合性チェックを実行する場合は、次のように dbv コマンドを実行します。

```
% dbv file=users01.dbf
```

dbv の出力例を次に示します。

```
DBVERIFY - Verification starting : FILE = users01.dbf
```

```
DBVERIFY - Verification complete
```

```
Total Pages Examined          : 250
Total Pages Processed (Data)   : 1
Total Pages Failing (Data)    : 0
Total Pages Processed (Index) : 0
Total Pages Failing (Index)   : 0
Total Pages Processed (Other) : 2
Total Pages Processed (Seg)   : 0
Total Pages Failing (Seg)     : 0
Total Pages Empty              : 247
Total Pages Marked Corrupt    : 0
Total Pages Influx            : 0
```

**参照：** DBVERIFY については、『Oracle Database ユーティリティ』を参照してください。





---

## ユーザー管理のデータベースの フラッシュバックおよびリカバリの実行

この章では、データベースをリストアおよびリカバリする方法、およびユーザー管理のバックアップおよびリカバリ計画（Recovery Manager を必要としない計画）を使用する場合に Oracle フラッシュバック機能を使用する方法について説明します。

この章の内容は、次のとおりです。

- [SQL\\*Plus](#) でのデータベースのフラッシュバックの実行
- ユーザー管理のメディア・リカバリの概要
- データベースの完全リカバリの実行
- データベースの不完全リカバリの実行
- NOARCHIVELOG モードでのデータベースのリカバリ
- メディア・リカバリのトラブルシューティング

## SQL\*Plus でのデータベースのフラッシュバックの実行

Oracle Flashback Database を使用すると、バックアップからファイルをリストアせずに、データベース全体を元の状態に戻すことができます。SQL\*Plus の FLASHBACK DATABASE コマンドの機能は、Recovery Manager の FLASHBACK DATABASE コマンドと同じです。このコマンドを実行すると、データベースが元の状態に戻されます。

フラッシュバック・データベースを使用する場合は、データベース用のフラッシュ・リカバリ領域を作成し、フラッシュバック・ログの収集を有効にする必要があります。フラッシュバック・データベース機能の動作、フラッシュバック・データベースの使用要件およびフラッシュバック・データベースに必要なフラッシュバック・ログの収集を有効にする方法の詳細は、[第 16 章「フラッシュバックおよびデータベースの Point-in-Time リカバリの実行」](#)を参照してください。Recovery Manager または SQL\*Plus のいずれを使用する場合でも、要件および準備は同じです。

SQL\*Plus でデータベースのフラッシュバックを実行する手順

1. ターゲット・データベースを問い合せて、フラッシュバックの SCN の範囲を確認します。次の SQL\*Plus 問合せを実行して、フラッシュバック・ウィンドウの最初と最後の SCN を示します。

```
SELECT CURRENT_SCN FROM V$DATABASE;

SELECT OLDEST_FLASHBACK_SCN, OLDEST_FLASHBACK_TIME
FROM V$FLASHBACK_DATABASE_LOG;
```

2. 必要に応じて他のフラッシュバック機能を使用し、データベースに対する不要な変更の SCN または時刻を確認します。
3. 管理者権限を使用して SQL\*Plus を起動して FLASHBACK DATABASE 文を実行し、データベースを以前の TIMESTAMP または SCN に戻します。たとえば、次のように入力します。

```
FLASHBACK DATABASE TO SCN 46963;
FLASHBACK DATABASE TO TIMESTAMP '2002-11-05 14:00:00';
FLASHBACK DATABASE
  TO TIMESTAMP to_timestamp('2002-11-11 16:00:00', 'YYYY-MM-DD HH24:MI:SS');
```

4. 操作が完了したら、データベースを読取り専用でオープンし、問合せを実行して必要なデータがリカバリされていることを確認します。

選択した目標時点が十分に前の時点でなかった場合は、別の FLASHBACK DATABASE 文を使用します。それ以外の場合は、RECOVER DATABASE を使用してデータベースを現在の時点に戻し、別の FLASHBACK DATABASE 文を実行します。

5. 結果が適切である場合は、RESETLOGS オプションを指定してデータベースをオープンします。

必要に応じて、データ・ポンプ・エクスポートを使用して消失したデータを保存し、RECOVER DATABASE でデータベースを現在の状態に戻して、消失したオブジェクトを再インポートすることもできます。

**参照：** フラッシュバック問合せ、フラッシュバック・トランザクション問合せなどの関連フラッシュバック機能の使用方法については、『Oracle Database アドバンスド・アプリケーション開発者ガイド』を参照してください。

## ユーザー管理のメディア・リカバリの概要

この項では、SQL\*Plus を使用したリカバリの概要について説明します。この項の内容は、次のとおりです。

- ユーザー管理のリストアおよびリカバリ
- RECOVER コマンドを使用した自動リカバリ
- アーカイブ・ログがデフォルトの場所にある場合のリカバリ
- アーカイブ・ログがデフォルト以外の場所にある場合のリカバリ
- リカバリの取消し
- パラレル・メディア・リカバリ

## ユーザー管理のリストアおよびリカバリ

通常、メディア障害またはユーザー・エラーによって 1 つ以上のデータファイルが破損したり削除された場合に、ファイルをリストアします。ユーザー管理のリストア操作では、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ファイルのバックアップをリストアします。

メディア障害がデータファイルに影響を与えた場合、リカバリの手順は次の条件によって異なります。

- データベースのアーカイブ・モード (ARCHIVELOG または NOARCHIVELOG)
- メディア障害のタイプ
- メディア障害によって影響されるファイル (データファイル、制御ファイル、アーカイブ REDO ログおよびサーバー・パラメータ・ファイルはすべてリストア操作の対象となります。)

永続的または一時的なメディア障害によって、NOARCHIVELOG モードで実行しているデータベースのデータファイルが影響を受けた場合、データベースは自動的に停止します。メディア障害が一時的な場合は、原因となっている問題を解決し、データベースを再起動します。通常、クラッシュ・リカバリによって、コミット済のすべてのトランザクションがオンライン REDO ログを使用してリカバリされます。メディア障害が永続的な場合は、28-18 ページの「[NOARCHIVELOG モードでのデータベースのリカバリ](#)」の説明に従ってデータベースをリカバリします。

表 28-1 に、ARCHIVELOG モードで実行されているデータベースでファイルを消失した場合に実行するメディア・リカバリについての説明を示します。

表 28-1 ユーザー管理のリストア操作

| 消失したもの   | 説明  |
|--|---|
| SYSTEM 表領域内のデータファイルまたはアクティブな UNDO セグメントを持つデータファイル                        | データベースは自動的に停止します。ハードウェアの問題が一時的な場合には、問題を解決し、データベースを再起動します。通常、クラッシュ・リカバリによって、消失したトランザクションがリカバリされます。ハードウェアの問題が永続的な場合は、28-9 ページの「 <a href="#">クローズしているデータベースのリカバリの実行</a> 」の説明に従って、バックアップからデータファイルをリストアし、データベースをリカバリします。       |
| SYSTEM 表領域内のものでないデータファイル、またはアクティブなロールバック・セグメントまたは UNDO セグメントを含まないデータファイル | 影響を受けたデータファイルがオフライン化されますが、データベースはオープンしたままです。データベースの影響を受けていない部分を使用可能なままにしておく必要がある場合には、データベースを停止しないでください。一時オプションを使用して、問題が発生したデータファイルを含む表領域をオフラインにした後、28-12 ページの「 <a href="#">オープンしているデータベースのリカバリの実行</a> 」の説明に従ってリカバリします。    |
| 現行の制御ファイルのすべてのコピー  | バックアップ制御ファイルをリストアし、RESETLOGS オプションを指定してデータベースをオープンする必要があります。<br><br>バックアップがない場合には、制御ファイルを再作成してください。可能であれば、ALTER DATABASE BACKUP CONTROLFILE TO TRACE の出力に含まれるスクリプトを使用します。制御ファイルの構造を現在のデータベース構造と一致させるために追加作業が必要になることがあります。 |
| 多重制御ファイルの 1 つのコピー  | 影響を受けていない多重制御ファイルの 1 つを破損または欠落した制御ファイルの場所にコピーして、データベースをオープンします。元の場所に制御ファイルをコピーできない場合は、初期化パラメータ・ファイルを編集して新しい場所を反映するか、または破損した制御ファイルを削除します。その後、データベースをオープンします。   |
| メディア・リカバリに必要な 1 つ以上のアーカイブ・ログ   | リカバリを実行するには、これらのアーカイブ・ログのバックアップをリストアする必要があります。デフォルトの場所またはデフォルト以外の場所にリストアできます。バックアップがない場合には、欠落している最初の REDO ログの前の SCN まで不完全リカバリを実行し、RESETLOGS でオープンする必要があります。   |
| サーバー・パラメータ・ファイル (SPFILE)   | サーバー・パラメータ・ファイルのバックアップがある場合には、これをリストアします。また、クライアント側の初期化パラメータ・ファイルのバックアップがある場合には、このファイルのバックアップをリストアし、インスタンスを起動した後、サーバー・パラメータ・ファイルを再作成できます。   |

**注意：** Oracle 管理ファイルのリストアおよびリカバリは、ユーザーが名前を付けたファイルのリストアおよびリカバリと同じです。

メディア・リカバリの実行には、SQL\*Plus の RECOVER 文を使用することをお勧めします。SQL 文の ALTER DATABASE RECOVER も使用できますが、ほとんどの場合、RECOVER 文を使用した方が簡単です。どのような種類のメディア・リカバリを開始する場合でも、次の制約に従う必要があります。

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

## RECOVER コマンドを使用した自動リカバリ

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

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

- RECOVER コマンドを発行する前に SET AUTORECOVERY ON を発行する。デフォルトである SET AUTORECOVERY OFF を実行した場合は、ファイル名を手動で入力するか、[Enter] キーを押して、提示されたファイル名を受け入れる必要があります。
- RECOVER コマンドのオプションとして AUTOMATIC キーワードを指定する。

いずれの場合にも、必要なファイルが正しい場所に正しい名前が存在しているかぎり RECOVER コマンドを発行したときに介入は必要ありません。データベースで REDO ログ・ファイルが正常に適用された場合は、次のメッセージが戻されます。

```
Log applied.
```

その後、順序内の次の REDO ログを求めるプロンプトが表示されます。最後に適用されたログが最後の必要なログである場合は、リカバリが終了します。

自動リカバリを使用した場合に使用されるファイル名は、LOG\_ARCHIVE\_FORMAT の値と LOG\_ARCHIVE\_DEST\_n の値を連結して導出されます。この n は、有効なすべてのローカル出力先の中で最大の値です。たとえば、データベース・インスタンスで初期化パラメータの次のような設定が有効になっているとします。

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

この場合、SQL\*Plus は自動的に /arc\_dest/loc2/arch\_%t\_%s\_%r.arc というファイル名を提示します（この %t はスレッド、%s は順序、%r は RESETLOG の ID です）。

## SET AUTORECOVERY を使用した自動リカバリ

データファイル・バックアップをリストアした後、SET AUTORECOVERY ON コマンドを実行すると、自動リカバリが有効になります。たとえば、SQL\*Plus に次のコマンドを入力すると、自動リカバリを実行し、データベースをオープンできます。

```
STARTUP MOUNT
SET AUTORECOVERY ON
RECOVER DATABASE
ALTER DATABASE OPEN;
```

---

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

---

## RECOVER コマンドの AUTOMATIC オプションを使用した自動リカバリ

SET AUTORECOVERY を使用して自動リカバリを有効にする他にも、RECOVER コマンドで AUTOMATIC キーワードを指定することもできます。たとえば、SQL\*Plus に次のコマンドを入力すると、自動リカバリを実行し、データベースをオープンできます。

```
STARTUP MOUNT
RECOVER AUTOMATIC DATABASE
ALTER DATABASE OPEN;
```

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

## アーカイブ・ログがデフォルトの場所にある場合のリカバリ

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

```
ORA-00279: change 53577 generated at 11/26/02 19:20:58 needed for thread 1
ORA-00289: suggestion : /oracle/oradata/trgt/arch/arcr_1_802.arc
ORA-00280: change 53577 for thread 1 is in sequence #802
Specify log: [<RET> for suggested | AUTO | FROM logsource | CANCEL ]
```

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

初期化パラメータ LOG\_ARCHIVE\_DEST\_n (この n は有効なすべてのローカル出力先の中で最大の値) および LOG\_ARCHIVE\_FORMAT の現在の値を連結し、制御ファイルのログ履歴情報を使用して、提示されるアーカイブ・ログのファイル名が構築されます。次のような設定が可能です。

```
LOG_ARCHIVE_DEST_1 = 'LOCATION = /oracle/oradata/trgt/arch/'
LOG_ARCHIVE_FORMAT = arcr_%t_%s.arc
```

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

```
NAME
-----
/oracle/oradata/trgt/arch/arcr_1_467.arc
/oracle/oradata/trgt/arch/arcr_1_468.arc
/oracle/oradata/trgt/arch/arcr_1_469.arc
```

このため、すべての必要なアーカイブ・ログ・ファイルが LOG\_ARCHIVE\_DEST\_1 という出力先にマウントされ、LOG\_ARCHIVE\_FORMAT の値が変更されていない場合は、ログ・ファイルが提示および適用され、メディア・リカバリが自動的に実行されます。

## アーカイブ・ログがデフォルト以外の場所にある場合のリカバリ

アーカイブ・ログがデフォルトの場所でない場合にメディア・リカバリを実行するには、追加の処理が必要です。次のいずれかの方法を選択します。

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

## アーカイブ・ログの場所の再設定

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

リカバリ前にアーカイブ・ログのデフォルトの場所を変更する手順

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

```
% cp /backup/arch/* /tmp/
```

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

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

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

```
STARTUP MOUNT
```

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

```
RECOVER DATABASE
```

## アーカイブ・ログの場所の変更

場合によっては、アーカイブの出力先パラメータの現在の設定を変更して、アーカイブ・ログ・ファイルの入力元を変更することもできます。

SET LOGSOURCE を使用して、デフォルト以外の場所のアーカイブ・ログをリカバリする手順

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

```
% cp $ORACLE_HOME/oradata/trgt/arch/* /tmp
```

2. SQL\*Plus 内でリカバリ操作のための代替場所を指定します。SET 文の LOGSOURCE パラメータまたは ALTER DATABASE 文の RECOVER ... FROM 句を使用します。たとえば、SQL\*Plus を起動して次のように実行します。

```
SET LOGSOURCE "/tmp"
```

3. オフライン表領域をリカバリします。たとえば、オフライン表領域 users をリカバリするには、次のコマンドを実行します。

```
RECOVER AUTOMATIC TABLESPACE users
```

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

```
RECOVER AUTOMATIC TABLESPACE users FROM "/tmp"
```

---

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

---

## リカバリの取消し

メディア・リカバリを開始した後で中断する必要がある場合は、REDO ログ・ファイルのプロンプトに対して CANCEL と入力します。また、個々のデータファイルのリカバリの実行時または自動リカバリの実行時に終了する必要がある場合は、オペレーティング・システムの割込みシグナルを使用します。リカバリが取り消された後、RECOVER コマンドを使用してリカバリを再開できます。リカバリは、取り消された位置から再開されます。

## パラレル・メディア・リカバリ

Oracle は、メディア・リカバリのロールフォワード・フェーズのパフォーマンスを向上するために、デフォルトで **パラレル・メディア・リカバリ** を使用します。パラレル・メディア・リカバリでは、ロールフォワード時に各データ・ブロックに別のプロセスを割り当てる分業体制を使用することで、処理の効率を高めています。使用されるプロセスの数は、CPU\_COUNT 初期化パラメータで指定されます。これは、デフォルトではシステム上の CPU の数と同じです。たとえば、CPU\_COUNT が 4 のシステム上でパラレル・リカバリを実行した場合、リカバリ対象のデータファイルが 1 つのみである場合には、4 つの起動されたプロセスがアーカイブ・ログからブロックを読み取り、REDO を適用します。

一般的に、メディア・リカバリはデータ・ブロックの読取りおよび書込みによって制限されます。パラレル・リカバリでは、パフォーマンスを向上させるために、システムで利用可能な I/O 帯域幅がすべて使用されます。システムの I/O ボトルネックが存在するか、非同期 I/O が十分にサポートされていない場合を除いて、パラレル・リカバリによってリカバリのパフォーマンスが向上する可能性があります。

パラレル・リカバリが実行されるデフォルトの動作を変更するには、NOPARALLEL オプションを指定して RECOVER を使用するか、または RECOVER PARALLEL 0 を使用します。RECOVERY\_PARALLELISM 初期化パラメータでは、インスタンス・リカバリまたはクラッシュ・リカバリのみが制御されます。メディア・リカバリは、RECOVERY\_PARALLELISM で使用される値の影響を受けません。

**参照：** SQL\*Plus の RECOVER ... PARALLEL および NOPARALLEL 文の詳細は、『SQL\*Plus ユーザーズ・ガイドおよびリファレンス』を参照してください。

## データベースの完全リカバリの実行

通常、メディア障害によって 1 つ以上のデータファイルにアクセスできなくなった場合に、データベースの完全リカバリを実行します。V\$RECOVER\_FILE ビューに、リカバリが必要なファイルが示されます。データベースの完全リカバリを実行する場合は、使用可能なすべての REDO を使用し、データベースを現行の SCN までリカバリします。

状況に応じて、データベース全体で一度にリカバリを実行することも、個々の表領域またはデータファイルでリカバリを実行することもできます。完全リカバリの場合は、RESETLOGS オプションを指定してデータベースをオープンする必要はないため、最初に一部のデータファイルのリカバリを行ってから、後で残りのデータファイルのリカバリを行うこともできます。

この項の手順では、次のことが想定されています。

- 現行の制御ファイルが使用可能です。制御ファイルをリストアまたは再作成する必要がある場合は、29-3 ページの「[現行の制御ファイルがすべて消失した場合のリカバリ](#)」および 29-6 ページの「[制御ファイルの再作成](#)」を参照してください。
- 必要なすべてのデータファイルのバックアップが存在しています。データファイルのバックアップが欠落している場合は、29-9 ページの「[バックアップが利用できない場合のデータファイルの再作成](#)」を参照してください。
- 必要なすべてのアーカイブ REDO ログが使用可能です。データベースを完全にリカバリするために必要な REDO が欠落している場合は、データベースの Point-in-Time リカバリを実行する必要があります。詳細は、28-14 ページの「[データベースの不完全リカバリの実行](#)」を参照してください。



この項では、完全メディア・リカバリ操作に必要な手順について説明します。この項の内容は、次のとおりです。

- [クローズしているデータベースのリカバリの実行](#)
- [オープンしているデータベースのリカバリの実行](#)

## クローズしているデータベースのリカバリの実行

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

破損または欠落しているデータファイルをリストアおよびリカバリする手順

1. データベースがオープンしている場合は、V\$RECOVER\_FILE を問い合わせ、リカバリする必要があるデータファイルおよびリカバリする必要がある理由を確認します。

Point-in-Time リカバリではなく完全リカバリの実行を計画している場合は、データベース全体ではなくリカバリが必要なデータファイルのみをリカバリすることができます。

Point-in-Time リカバリでは、TSPITR を実行しないかぎり、すべてのデータファイルをリストアおよびリカバリする必要があることに注意してください（[第 20 章「Recovery Manager の表領域の Point-in-Time リカバリ \(TSPITR\) の実行」](#)を参照）。また、フラッシュバック・データベースも使用できますが、すべてのデータファイルが影響を受けるため、データベース全体が過去の状態に戻ります。

V\$RECOVER\_FILE を問い合わせると、リカバリが必要なデータファイルを、そのステータス情報とエラー情報とともにデータファイル番号で表示できます。

```
SELECT FILE#, ERROR, ONLINE_STATUS, CHANGE#, TIME
FROM   V$RECOVER_FILE;
```

---

**注意：** V\$RECOVER\_FILE は、バックアップからリストアされた制御ファイル、またはデータファイルに影響を及ぼすメディア障害の発生後に再作成された制御ファイルには使用できません。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。

---

データファイル番号と V\$DATAFILE および V\$TABLESPACE ビューを使用した便利な結合を実行して、データファイルおよび表領域の名前を取得することもできます。次の SQL\*Plus コマンドを使用して、問合せの出力の書式を設定します。

```
COL DF# FORMAT 999
COL DF_NAME FORMAT A35
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#;
```

ERROR 列で、リカバリが必要な各ファイルに関する問題を識別します。

2. V\$ARCHIVED\_LOG および V\$RECOVERY\_LOG を問い合せて、必要なアーカイブ REDO ログ・ファイルを確認します。

V\$ARCHIVED\_LOG には、すべてのアーカイブ REDO ログのファイル名が表示されます。V\$RECOVERY\_LOG には、データベースでメディア・リカバリを実行する必要があるアーカイブ REDO ログのみが表示されます。V\$RECOVERY\_LOG ビューでは、LOG\_ARCHIVE\_FORMAT を使用して、考えられるファイルの名前も表示されます。

---

**注意：** V\$RECOVERY\_LOG は、データファイルのメディア・リカバリが必要な場合にのみ移入されます。このため、ユーザー・エラーなど、計画されたリカバリの場合にはこのビューは役に立ちません。

データファイルのリカバリが必要な場合に、データファイルのバックアップが存在しないときは、データベースにデータファイルが追加された後に生成されたすべての REDO が必要です。

---

3. デフォルトの場所ですべてのアーカイブ・ログが使用可能な場合は、次の手順にスキップします。

一部のアーカイブ・ログをリストアする必要があるときに、使用可能な領域が十分にある場合は、LOG\_ARCHIVE\_DEST\_1 で指定された場所に必要なアーカイブ REDO ログ・ファイルをリストアします。メディア・リカバリ中に正しいログが必要になると、自動的にログが検索されます。たとえば、Linux または UNIX では、次のコマンドを入力します。

```
% cp /disk2/arch/* $ORACLE_HOME/oradata/trgt/arch
```

使用可能な領域が十分にない場合は、必要なアーカイブ REDO ログ・ファイルの一部またはすべてを別の場所にリストアします。

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

```
SHUTDOWN IMMEDIATE
```

5. メディアを検査して問題の原因を確認します。

メディア障害の原因であるハードウェアの問題が一時的なもので、データが破損していない場合（ディスクやコントローラの停電など）は、メディア・リカバリは必要ありません。データベースを起動して、通常の操作を再開します。

問題を解決できない場合は、次の手順に進みます。

6. ファイルが永続的な損傷を受けた場合には、破損したファイルの最新のバックアップを識別します。メディア障害によって破損したデータファイルのみをリストアします。破損していないデータファイルまたはオンライン REDO ログ・ファイルはリストアしないでください。

たとえば、破損したファイルが ORACLE\_HOME/oradata/trgt/users01.dbf のみであり、このファイルの最新のバックアップが /backup/users01\_10\_24\_02.dbf であるとします。特定のデータファイルのバックアップがない場合には、リカバリするための空の置換ファイルを作成できます。

7. オペレーティング・システム・ユーティリティを使用して、デフォルトの場所または新しい場所にデータファイルをリストアします。たとえば、Linux または UNIX ユーザーが users01.dbf をデフォルトの場所にリストアする場合は、次のコマンドを入力できます。

```
% cp /backup/users01_10_24_06.dbf $ORACLE_HOME/oradata/trgt/users01.dbf
```

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

- ハードウェアの問題が解決され、デフォルトの場所にデータファイルをリストアできる場合は、デフォルトの場所にデータファイルをリストアし、メディア・リカバリを開始します。

- ハードウェアの問題が持続し、元の場所にデータファイルをリストアできない場合は、代替のストレージ・デバイスにデータファイルをリストアします。ALTER DATABASE RENAME FILE を使用して、これらのファイルの新しい場所を制御ファイルで指定します。必要に応じて、『Oracle Database 管理者ガイド』を参照してください。
  - データファイルを RAW ディスクまたは RAW パーティションにリストアする場合の方法は、ファイル・システム上のファイルにリストアする場合と基本的に同じです。ただし、RAW デバイス上のファイルのネーミング規則（オペレーティング・システムによって異なる）に注意して、RAW デバイスをサポートするオペレーティング・システム・ユーティリティを使用する必要があります。
8. 管理者権限を使用してデータベースに接続し、新しいインスタンスを起動して、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）。たとえば、次のように入力します。

```
STARTUP MOUNT
```

9. 1 つ以上の破損したデータファイルを代替の場所にリストアした場合は、データベースの制御ファイルを更新して、新しいデータファイル名を反映させます。たとえば、表領域 users のデータファイルのファイル名を変更するには、次のように入力します。

```
ALTER DATABASE RENAME FILE '?/oradata/trgt/users01.dbf' TO
'/disk2/users01.dbf';
```

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

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

11. リカバリが必要なすべてのデータファイルがオンラインにされていることを確認します。唯一の例外は、NORMAL モードでオフラインにされたオフライン表領域内のデータファイルまたは読取り専用表領域内のデータファイルです。たとえば、/oracle/dbs/tbs\_10.f という名前のデータファイルを確実にオンラインにするには、次のように入力します。

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

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

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

```
SQL> @onlineall
```

12. アーカイブ REDO ログを代替格納場所にリストアした場合は、SQL\*Plus で SET コマンドの LOGSOURCE パラメータを使用して、メディア・リカバリの前の場所を特定できます。たとえば、ログが /tmp 内にステージングされている場合は、次のコマンドを入力できます。

```
SET LOGSOURCE /tmp
```

あるいは、次の手順を実行して、RECOVER コマンドで FROM パラメータを使用します。たとえば、ログが /tmp 内にステージングされている場合は、次のコマンドを入力できます。

```
RECOVER AUTOMATIC FROM '/tmp' DATABASE
```

---

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

---

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

```
RECOVER AUTOMATIC DATABASE # whole database
RECOVER AUTOMATIC TABLESPACE users # specific tablespace
RECOVER AUTOMATIC DATAFILE '?/oradata/trgt/users01.dbf'; # specific datafile
```

アーカイブ REDO ログの適用を自動化しない場合は、ログが提示されるたびに、そのログを適用するかどうかを選択する必要があります。リカバリを自動化した場合は、ログは自動的に適用されます。必要なすべてのアーカイブおよびオンライン REDO ログが、リストアされたデータファイルに適用されるまで、リカバリは続行されます。メディア・リカバリが完了すると、次のように通知されます。

```
Media recovery complete.
```

完全メディア・リカバリにアーカイブ REDO ログ・ファイルが必要でない場合は、必要なすべてのオンライン REDO ログ・ファイルが適用され、リカバリが終了します。

14. リカバリ完了後に、データベースをオープンして使用します。

```
ALTER DATABASE OPEN;
```

**参照：** REDO ログ・ファイルの適用の詳細は、28-3 ページの「[ユーザー管理のメディア・リカバリの概要](#)」を参照してください。V\$ ビューについては、『Oracle Database リファレンス』を参照してください。

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

```
% rm /tmp/*.arc
```

**参照：** データ・ディクショナリ・ビューの詳細は、『Oracle Database リファレンス』を参照してください。メディア・リカバリ時のログの適用の概要は、28-3 ページの「[ユーザー管理のメディア・リカバリの概要](#)」を参照してください。

## オープンしているデータベースのリカバリの実行

データベースをオープンしているときにメディア障害が発生した場合は、破損していないデータファイルはオンラインのまま使用できる場合があります。データベース・ライターがファイルに書き込めない場合は、破損したファイルは自動的にオフラインにされますが、破損したファイルを含む表領域はオフラインにされません。破損したファイルを読み取れない問合せはエラーを戻しますが、問合せが失敗したためにデータファイルがオフラインにされることはありません。たとえば、SQL 問合せを実行して、次のような出力が表示される場合があります。

```
ERROR at line 1:
ORA-01116: error in opening database file 3
ORA-01110: data file 11: '/oracle/oradata/trgt/cwmlite02.dbf'
ORA-27041: unable to open file
SVR4 Error: 2: No such file or directory
Additional information: 3
```

データベースのオープン中は、この項の手順を使用して SYSTEM 表領域の完全メディア・リカバリを実行することはできません。メディア障害によって SYSTEM 表領域のデータファイルが破損した場合は、データベースは自動的に停止します。

オープン状態のデータベースでデータファイルをリストアする手順

1. 28-9 ページの「[クローズしているデータベースのリカバリの実行](#)」の手順 1～3 を実行します。

2. データベースがオープンしている場合は、破損したデータファイルを含むすべての表領域をオフラインにします。たとえば、破損したデータファイルが表領域 `users` および `tools` に含まれている場合は、次の SQL 文を実行します。

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

TEMPORARY を指定すると、Oracle Database によって、表領域内のすべてのオンライン・データファイルに対して **チェックポイント** が実行されることに注意してください。この文の発行時にオフラインであるファイルでは、表領域をオンラインに戻す前に、メディア・リカバリが必要な場合があります。IMMEDIATE を指定すると、表領域をオンラインに戻す前に、表領域に対してメディア・リカバリを実行する必要があります。

3. メディアを検査して問題の原因を確認します。

DBVERIFY ユーティリティを使用すると、オフライン・データファイルに対して整合性チェックを実行できます (27-18 ページの「**DBVERIFY ユーティリティの実行**」を参照)。

メディア障害の原因であるハードウェアの問題が一時的なもので、データが破損していない場合、メディア・リカバリは必要ありません。オフライン表領域をオンラインにし、通常の操作を再開することができます。問題を解決できない場合、または DBVERIFY によって破損ブロックがレポートされた場合は、次の手順に進みます。

4. ファイルが永続的に破損された場合は、オペレーティング・システム・コマンドを使用して、メディア障害によって破損されたデータファイルのみの最新のバックアップ・ファイルをリストアします。たとえば、データファイル `users01.dbf` をリストアするには、Linux または UNIX で `cp` コマンドを次のように実行します。

```
% cp /disk2/backup/users01.dbf $ORACLE_HOME/oradata/trgt/users01.dbf
```

ハードウェアの問題が解決され、データファイルを元の場所にリストアできる場合は、その場所にリストアします。それ以外の場合は、代替のストレージ・デバイスにデータファイルをリストアします。破損していないデータファイル、オンライン REDO ログまたは制御ファイルはリストアしないでください。

---

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

---

5. 1 つ以上の破損したデータファイルを代替の場所にリストアした場合は、データベースの制御ファイルを更新して、新しいデータファイル名を反映させます。たとえば、表領域 `users` のデータファイルのファイル名を変更するには、次のように入力します。

```
ALTER DATABASE RENAME FILE '/oradata/trgt/users01.dbf' TO
'/disk2/users01.dbf';
```

6. アーカイブ REDO ログを代替格納場所にリストアした場合は、SQL\*Plus で SET コマンドの LOGSOURCE パラメータを使用して、メディア・リカバリの前の場所を特定できます。たとえば、ログが `/tmp` 内にステージングされている場合は、次のコマンドを入力できます。

```
SET LOGSOURCE /tmp
```

あるいは、次の手順を実行して、RECOVER コマンドで FROM パラメータを使用します。たとえば、ログが `/tmp` 内にステージングされている場合は、次のコマンドを入力できます。

```
RECOVER AUTOMATIC FROM '/tmp' TABLESPACE users, tools;
```

---

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

---

7. 管理者権限を使用してデータベースに接続します。一度の手順で、1つ以上のオフライン表領域の破損したすべてのデータファイルについて、オフライン表領域のリカバリを開始します。たとえば、users および tools のリカバリを行うには、次のように入力します。

```
RECOVER AUTOMATIC TABLESPACE users, tools;
```

必要なアーカイブ REDO ログおよびオンライン REDO ログが適用され、リストアしたデータファイルが再構築されて、メディア・リカバリのロールフォワード・フェーズが開始されます。RECOVER AUTOMATIC または SET AUTORECOVERY ON によってファイルの適用が自動化されていないかぎり、必要な REDO ログ・ファイルごとにプロンプトが表示されます。

必要なすべてのアーカイブ・ログがデータファイルに適用されるまで、リカバリは続行されます。次に、オンライン REDO ログがリストアされたデータファイルに自動的に適用され、メディア・リカバリが完了します。完全メディア・リカバリにアーカイブ REDO ログが必要ない場合は、プロンプトは表示されません。かわりに、必要なすべてのオンライン REDO ログが適用され、メディア・リカバリが完了します。

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

```
ALTER TABLESPACE users ONLINE;
ALTER TABLESPACE tools ONLINE;
```

**参照：** データファイルの作成については、『Oracle Database 管理者ガイド』を参照してください。ALTER DATABASE RENAME FILE については、『Oracle Database SQL 言語リファレンス』を参照してください。

## データベースの不完全リカバリの実行

通常、次のような場合にデータベースの Point-in-Time リカバリ (DBPITR) を実行します。

- ユーザー・エラーまたは管理エラーの前の SCN までデータベースをリカバリする場合。
- データベースに破損ブロックが含まれている場合。
- 必要なすべてのアーカイブ REDO ログが使用可能ではなかったためデータベースの完全リカバリに失敗した場合。
- 本番データベース・バックアップからテスト・データベースまたはレポート・データベースを作成する場合。

データベースを ARCHIVELOG モードで実行しているときに、アーカイブ REDO ログ・ファイルの唯一のコピーが破損した場合には、破損したファイルが、データベースの現在の動作に影響を与えることはありません。表 28-2 に、REDO ログが書き込まれた場合、およびデータファイルをバックアップした場合に応じて発生する可能性がある状況を示します。

**表 28-2 アーカイブ REDO ログの消失**

| バックアップ対象   | 説明  |
|--|---|
| 一杯になったオンライン REDO ログ・グループ (現在アーカイブされている) が書き込まれた後のすべてのデータファイル | 完全メディア・リカバリに、一杯になったオンライン REDO ログ・グループのアーカイブされたバージョンは必要ありません。  |
| 一杯になったオンライン REDO ログ・グループが書き込まれる前の特定のデータファイル                  | 対応するデータファイルが永続的なメディア障害によって破損した場合は、破損したデータファイルの最新のバックアップを使用して、破損したアーカイブ REDO ログ・ファイルに至るまで、破損したデータファイルの表領域の Point-in-Time リカバリを実行します。 |

---

---

**注意：** アーカイブ REDO ログ・グループが破損したことがわかっている場合は、破損したアーカイブ REDO ログを必要としない、データベース全体のバックアップを作成するために、すぐにすべてのデータファイルをバックアップします。

---

---

DBPITR での方法は、特定の時刻または SCN を指定するか、あるいは CANCEL と入力することによって終了する点以外は、28-9 ページの「[クローズしているデータベースのリカバリの実行](#)」に示す方法と非常に類似しています。取消しベースのリカバリでは、アーカイブ REDO ログ・ファイルの推奨ファイル名がプロンプトに表示されます。ユーザーがファイル名のかわりに CANCEL を指定するか、すべての REDO がデータファイルに適用されると、リカバリは停止します。リカバリを終了するアーカイブ・ログをユーザーが制御する場合は、取消しベースのリカバリが最適です。

この項の手順では、次のことが想定されています。

- 現行の制御ファイルが使用可能です。制御ファイルをリストアまたは再作成する必要がある場合は、29-3 ページの「[現行の制御ファイルがすべて消失した場合のリカバリ](#)」を参照してください。
- 必要なすべてのデータファイルのバックアップが存在しています。データファイルのバックアップが欠落している場合は、29-9 ページの「[バックアップが利用できない場合のデータファイルの再作成](#)」を参照してください。

この項の内容は、次のとおりです。

- [取消しベースの不完全リカバリの実行](#)
- [時間ベースまたは変更ベースの不完全リカバリの実行](#)

## 取消しベースの不完全リカバリの実行

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

取消しベースのリカバリを実行する手順

1. 28-9 ページの「[クローズしているデータベースのリカバリの実行](#)」の手順 1～8 を実行します。
2. SQL\*Plus で次のコマンドを発行し、取消しベースのリカバリを開始します。

```
RECOVER DATABASE UNTIL CANCEL
```

---

---

**注意：** RECOVER コマンドの UNTIL 句を指定しなかった場合は、完全リカバリを実行するとみなされ、すべての REDO が適用されるまではデータベースをオープンできません。

---

---

リストアしたデータファイルを再構築するために必要な REDO ログ・ファイルが適用されます。データベースは、LOG\_ARCHIVE\_DEST\_1 から検索する名前を提示します。ユーザーは、そのログ・ファイルを適用するかどうかの決定を求められます。制御ファイルがバックアップの場合は、オンライン REDO ログの変更を適用する場合、オンライン REDO ログの名前を指定する必要があります。



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

CANCEL

リカバリが成功したかどうかを示されます。すべてのデータファイルが整合性のある SCN までリカバリされる前にリカバリを取り消し、その後データベースをオープンしようとする、さらにリカバリが必要な場合は、ORA-1113 エラーが表示されます。

V\$RECOVER\_FILE を問い合わせると、さらにリカバリが必要かどうか、または不完全リカバリの開始前にデータファイルのバックアップがリストアされていなかったかどうかを確認できます。

4. RESETLOGS オプションを指定してデータベースをオープンします。不完全リカバリまたはバックアップ制御ファイルを使用したリカバリの後は、ログをリセットする必要があります。たとえば、次のように入力します。

```
ALTER DATABASE OPEN RESETLOGS;
```

ログのリセットが不適切な状況で OPEN RESETLOGS を試行した場合、またはログをリセットする必要がある状況でログをリセットしなかった場合はエラーが戻され、データベースはオープンされません。問題を解決して再試行してください。

**参照：** ALTER DATABASE OPEN RESETLOGS が正常に実行されない原因については、28-19 ページの「ユーザー管理のメディア・リカバリの問題」を参照してください。

5. RESETLOGS オプションを使用してデータベースをオープンした後、アラート・ログを確認します。

---

---

**注意：** トレース・ファイルおよびアラート・ログの場所を特定する最も簡単な方法は、SQL 問合せ SELECT NAME, VALUE FROM V\$DIAG\_INFO を実行する方法です。

---

---

RESETLOGS オプションを指定してオープンした場合は、リカバリが完全か不完全かによって、戻されるメッセージが異なります。完全リカバリの場合は、アラート・ログに次のメッセージが表示されます。

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

不完全リカバリの場合は、アラート・ログに次のメッセージが表示されます。ここで、scn は、不完全リカバリの終了位置を表します。

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



また、アラート・ログも確認して、データ・ディクショナリと制御ファイルの間の矛盾がデータベースで検出されていないかを調べます。表 28-3 に、2 つの例を示します。

表 28-3 データ・ディクショナリと制御ファイルの間の矛盾

| 制御ファイルに示されているデータファイル | データ・ディクショナリに示されているデータファイル | 結果  |
|----------------------|---------------------------|---|
| あり                   | なし                        | 示されていないデータファイルへの参照が、制御ファイルから削除されます。アラート・ログのメッセージに、検出された内容が表示されます。   |
| なし                   | あり                        | 制御ファイルの MISSINGnnnnn の項 (nnnnn は 10 進数で示したファイル番号) に、プレースホルダ・エントリが作成されます。制御ファイルで MISSINGnnnnn には、オフラインで、メディア・リカバリが必要であるとのフラグが付けられます。MISSINGnnnnn に ALTER DATABASE RENAME FILE を使用して、データファイルをポイントするようにすると、MISSINGnnnnn に対応するデータファイルをアクセス可能にできます。このデータファイルのバックアップがない場合は、表領域を削除します。 |

## 時間ベースまたは変更ベースの不完全リカバリの実行

この項では、リカバリの終了位置に対して SCN または時刻を指定する方法について説明します。データベースが季節的な時間変更（たとえば夏時間）の影響を受ける場合には、REDO ログの中に 1 つの時間が 2 度出現し、2 番目の、つまり後の時間までリカバリを行う場合に、問題が発生します。時間の変更を処理するには、取消しベースまたは変更ベースのリカバリを実行します。

変更ベースまたは時間ベースのリカバリを実行する手順

- 28-9 ページの「クローズしているデータベースのリカバリの実行」の手順 1～8 を実行します。
- RECOVER DATABASE UNTIL 文を発行してリカバリを開始します。特定の SCN までリカバリする場合は、引用符を付けずに 10 進数を指定します。たとえば、SCN 10034 までリカバリするには、次のコマンドを発行します。

```
RECOVER DATABASE UNTIL CHANGE 10034;
```

特定の時刻までリカバリする場合は、一重引用符で区切った書式 ('YYYY-MM-DD:HH24:MI:SS') で時刻を指定します。次の文は、指定された時刻までデータベースのリカバリを行います。

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

- リストアされたデータファイルのリカバリに必要な REDO ログ・ファイルを適用します。正しい時刻になるとリカバリは自動的に停止し、リカバリが成功したかどうかを示すメッセージが戻されます。

---

**注意：** リカバリが自動化されていないかぎり、名前は LOG\_ARCHIVE\_DEST\_1 から提示され、各ログを適用するかどうかの決定を求められます。制御ファイルがバックアップの場合は、アーカイブ・ログを適用した後で、オンライン・ログの名前を指定する必要があります。

---

- 28-15 ページの「取消しベースの不完全リカバリの実行」の手順 4～5 を実行します。

## NOARCHIVELOG モードでのデータベースのリカバリ

メディア障害によって、NOARCHIVELOG モードのデータベースのデータファイルが破損した場合は、通常、一貫性のあるデータベース全体のバックアップをリストアすることによってのみリカバリできます。Oracle Data Pump Export で作成した論理バックアップを使用して通常の物理バックアップを補完している場合は、データベースのエクスポートされたバックアップを、再作成したデータベースまたは古いバックアップからリストアしたデータベースにインポートして、データベースのリストアを試行することもできます。

最新のデータベース全体のバックアップをリストアおよびリカバリする手順

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

```
SHUTDOWN IMMEDIATE
```

2. 可能であれば、メディアの問題を解決して、バックアップ・データベース・ファイルを元の場所にリストアできるようにします。

3. オペレーティング・システム・コマンドを使用して、最新のデータベース全体のバックアップをリストアします。破損したファイルのみでなく、データベース全体のバックアップの、すべてのデータファイルおよび制御ファイルをリストアします。ハードウェアの問題が解決できず、データベース・ファイルの一部またはすべてを代替の場所にリストアする必要がある場合は、データベース全体のバックアップを新しい場所にリストアします。次の例は、データベース全体のバックアップをデフォルトの場所にリストアします。

```
% cp /backup/*.dbf $ORACLE_HOME/oradata/trgt/
```

4. 必要に応じて、リストアされた初期化パラメータ・ファイルを編集して、制御ファイルの新しい場所を示します。たとえば、次のように入力します。

```
CONTROL_FILES = "/new_disk/oradata/trgt/control01.dbf"
```

5. リストアして編集したパラメータ・ファイルを使用してインスタンスを起動し、データベースをマウントします（ただし、オープンはしません）。たとえば、次のように入力します。

```
STARTUP MOUNT
```

6. リストアされたデータファイルのファイル名が異なる場合（同じノードまたは異なるノードの異なるファイル・システムまたはディレクトリにリストアする場合など）、制御ファイルを更新して新しいデータファイルの場所を反映させます。たとえば、データファイル 1 の名前を変更するには、次のように入力します。

```
ALTER DATABASE RENAME FILE '?/oradata/trgt/system01.dbf' TO  
'/new_disk/oradata/system01.dbf';
```

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

```
ALTER DATABASE RENAME FILE '?/oradata/trgt/redo01.log' TO  
'/new_disk/oradata/redo_01.log';  
ALTER DATABASE RENAME FILE '?/oradata/trgt/redo02.log' TO  
'/new_disk/oradata/redo_02.log';
```

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

```
RECOVER DATABASE UNTIL CANCEL  
CANCEL
```

9. RESETLOGS モードでデータベースをオープンします。このコマンドによって、オンライン REDO ログが消去され、ログ順序が 1 にリセットされます。

```
ALTER DATABASE OPEN RESETLOGS;
```

NOARCHIVELOG モードのデータベースのバックアップをリストアした後、ログをリセットすると、バックアップが作成されてから障害が発生するまでの間にデータベースに対して行われたすべての変更が廃棄されます。

**参照：** データファイルの名前の変更および再配置の詳細は、『Oracle Database 管理者ガイド』を参照してください。ALTER DATABASE RENAME FILE の詳細は、『Oracle Database SQL 言語リファレンス』を参照してください。

## メディア・リカバリのトラブルシューティング

この項では、ユーザー管理のメディア・リカバリ（Recovery Manager を使用しないで実行するメディア・リカバリ）のトラブルシューティングの方法について説明します。この項の内容は、次のとおりです。

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

## ユーザー管理のメディア・リカバリの問題

表 28-4 「メディア・リカバリの問題」に、メディア・リカバリ中に発生する可能性がある問題を示します。

表 28-4 メディア・リカバリの問題

| 問題   | 説明   |
|--|--|
| アーカイブ・ログの欠落または名前の誤り  | 制御ファイルに記録されたアーカイブ・ログが見つからないため、リカバリは停止します。  |
| データベースをオープンしようとする、ORA-1113 エラーが、データファイルのメディア・リカバリが必要であることを示す | このエラーの一般的な原因は次のとおりです。 <ul style="list-style-type: none"> <li>■ 不完全リカバリを実行していますが、必要なすべてのデータファイル・バックアップがリストアされていません。</li> <li>■ データファイルが一貫性のある SCN に達する前に、不完全リカバリが停止しました。</li> <li>■ オンライン・バックアップからデータファイルのリカバリを行っていますが、データファイルを一貫性のある状態にするための十分な REDO が適用されていません。</li> <li>■ バックアップ制御ファイルを使用してリカバリを行っていますが、必要なオンライン REDO ログの場所が指定されていません。</li> <li>■ ユーザーがデータベースをオープンしようとしたときに、データファイルに対してメディア・リカバリが実行されていました。</li> <li>■ リカバリの必要なデータファイルは、RECOVER DATABASE の実行前にオンラインにされていなかったため、リカバリが行われていません。</li> </ul> |

表 28-4 メディア・リカバリの問題（続き）

| 問題                                | 説明   |
|-----------------------------------|--|
| REDO レコードの問題                      | 2つの状況が考えられます。 <ul style="list-style-type: none"> <li>一貫性チェックが失敗したためにリカバリが停止しました。スタック・リカバリと呼ばれる問題です。スタック・リカバリは、基礎となるオペレーティング・システムまたはストレージ・システムで、データベースの通常稼働中に発行された書込みが消失した場合に発生することがあります。</li> <li>REDO を適用するときに内部エラーが通知されました。この問題は、Oracle Database の不具合によって発生することがあります。チェックサム検証が使用されていない場合は、REDO またはデータ・ブロックの破損によってエラーが発生することもあります。</li> </ul> |
| アーカイブ・ログの破損                       | 格納中またはストレージ・システム間でコピーされるときに、ログが破損した可能性があります。DB_BLOCK_CHECKSUM が有効になっている場合は、通常、チェックサム・エラーが通知されます。チェックサム・チェックが無効になっている場合、ログの破損が REDO の問題として表示されることがあります。   |
| 互換性のないパラレル REDO フォーマットによるアーカイブ・ログ | パラレル REDO 機能を有効にした場合、REDO ログが新規のフォーマットで生成されません。以前のリリースの Oracle では、パラレル REDO ログを適用できません。ただし、Oracle9i リリース 2 (9.2) より前のリリースでは、パラレル REDO フォーマットを検出し、エラー・メッセージ「External error 00303, 00000, "パラレル REDO は処理できません"」によって非一貫性の存在を確認できます。   |
| データ・ブロックの破損                       | データファイルのバックアップに破損したデータ・ブロックが含まれているか、リカバリ中またはバックアップにコピーされるときにデータ・ブロックが破損しました。DB_BLOCK_CHECKSUM が有効になっている場合、データベースは、通常の操作時にチェックサムを各ブロックで計算し、ブロックに保存してからディスクに書き込みます。データベースは、後でディスクからブロックを読み取る場合、チェックサムを再計算し、格納されている値と比較します。一致しない場合は、チェックサム・エラーを通知します。チェックサム・チェックが無効になっている場合は、問題が REDO の破損として表示されることがあります。   |
| その他の問題                            | リカバリ中には、メモリーの破損や、その他の一時的な問題が発生することがあります。   |

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

メディア・リカバリで、データベース・バックアップのリカバリ中に問題が発生した場合は、スタック・リカバリの問題または REDO 適用中の問題であるかどうかに関係なく、データベースは常に停止し、リカバリ中のデータファイルを一貫性のある状態、つまり、障害前の一貫性のある SCN のままにします。次のいずれかを実行できます。

- データベースを読み取り専用でオープンし、問題を調査します。
- RESETLOGS オプションを指定してデータベースをオープンします (RESETLOGS のオープンの要件が満たされている場合)。スタンバイ・データベースはメディア・リカバリの形態で更新されるため、RESETLOGS の制限は、[フィジカル・スタンバイ・データベース](#)をオープンする場合にも適用されます。

一般に、データベースを読み取り専用でオープンしたり、RESETLOGS オプションを指定してオープンする場合には、すべてのオンライン・データファイルを同じ SCN までリカバリする必要があります。この要件が満たされない場合は、ユーザーがオープンを試行すると、ORA-1113 またはその他のエラーが通知されることがあります。ORA-1113 の一般的な原因については、[表 28-4 「メディア・リカバリの問題」](#)を参照してください。

一般に、メディア・リカバリの問題には、次のように対処します。

1. 問題の原因を確認します。必要に応じて試行リカバリを実行します。
2. 問題が REDO ログの欠落に係るか、REDO ログ、メモリーまたはデータ・ブロックが破損している可能性がある場合は、表 28-5 に示す方法で解決を試みます。
3. 表 28-5 に示す方法を使用しても問題を解決できない場合は、次のいずれかを実行します。
  - データベース全体のバックアップのリカバリを行う場合は、RESETLOGS オプションを指定してデータベースをオープンします。シリアル・メディア・リカバリを実行した場合は、データベースには、破損が発生した SCN の前までのすべての変更が含まれています。この SCN 以降の変更は、データベースのリカバリされた部分に含まれていません。オンライン・バックアップをリストアした場合は、RESETLOGS によるオープンには、REDO ストリーム中のすべての ALTER ... END BACKUP 操作までリカバリが完了している場合にのみ成功します。
  - データ・ブロックを破損させることをメディア・リカバリに許可して、リカバリを続行します。メディア・リカバリが完了した後、Recovery Manager を使用して**ブロック・メディア・リカバリ**を実行してください。
  - それでも問題が解決しない場合は、Oracle サポート・サービスに連絡します。

**参照：** ブロック・メディア・リカバリについては、19-9 ページの「[障害リカバリの実行](#)」を参照してください。

## メディア・リカバリの問題の調査：フェーズ 1

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

最初に、問題の原因が、設定の誤り、REDO ログの破損、データ・ブロックの破損、メモリーの破損、またはその他の原因のどれかを確認します。データ・ブロックの**チェックサム**にエラーがある場合には、データ・ブロックが破損しています。REDO ログ・ブロックのチェックサムにエラーがある場合には、REDO ログが破損しています。

リカバリの問題の原因は、特定するのが困難な場合があります。問題の原因が完全にはわからない場合でも、この章に示す方法を使用すると、データベースのリカバリを迅速に実行できます。

メディア・リカバリの問題を調査する手順

1. alert.log を調べて、エラー・メッセージに問題の性質についての一般的な情報が記載されているかを確認します。たとえば、alert\_SID.log にチェックサムのエラーが示されているかどうかを確認します。また、メディア・リカバリを続行するためにデータ・ブロックの破損を許可する必要があることが示されているかどうかを確認します。
2. リカバリ中に Oracle プロセスによって生成されたトレース・ファイルを調べます。追加のエラー情報が含まれていることがあります。

## ブロックを破損させない修正の試行：フェーズ 2

メディア・リカバリの問題のタイプに応じて、いくつかの解決方法があります。表 28-5 に示す方法の 1 つ、あるいはその組合せを使用できます。これらの方法は比較的安全なものです。ほとんどの場合、データベースに対する害はありません。

表 28-5 メディア・リカバリの解決方法

| 発生した可能性がある問題                      | 説明  |
|-----------------------------------|---|
| アーカイブ REDO ログの欠落または名前の誤り          | ファイル名が正しく入力されているかを確認します。正しく入力されていた場合には、オペレーティング・システムからログが欠落しているかを確認します。欠落している場合に、バックアップがあれば、バックアップをリストアし、ログを適用します。バックアップがない場合には、可能であれば、欠落したログの時点まで不完全リカバリを実行します。  |
| ALTER DATABASE OPEN での ORA-1113   | 表 28-4 「メディア・リカバリの問題」で、このエラーの原因を検討します。リカバリが必要なすべての読取り / 書き込みデータファイルがオンラインにされていることを確認します。<br>リカバリにバックアップ制御ファイルを使用した場合には、制御ファイルとデータファイルの SCN に一貫性がないかぎり、データベースをオープンできません。必要な REDO が不在場合には、制御ファイルを再作成する必要があります。  |
| アーカイブ・ログの破損                       | ログ REDO ブロックのチェックサム検証が失敗したときには、ログが破損しています。リカバリ・セッション中、またはデータベースが REDO を生成したときに DB_BLOCK_CHECKSUM が有効になっていなかった場合には、リカバリの問題の原因としてログの破損が考えられます。ログが破損し、破損したログの代替コピーが使用できる場合には、このコピーを適用して、問題が修正されるかどうかを確認します。<br>DB_BLOCK_CHECKSUM 初期化パラメータは、REDO ログおよびデータ・ブロックに対してチェックサムを計算するかどうかを決定します。  |
| 互換性のないパラレル REDO フォーマットによるアーカイブ・ログ | Oracle9i リリース 2 (9.2) より前の Oracle リリースを実行している場合、パラレル REDO フォーマットで作成された REDO ログを適用するには、次の手順に従う必要があります。<br><ol style="list-style-type: none"> <li>1. データベースを Oracle9i リリース 2 (9.2) 以上にアップグレードします。</li> <li>2. メディア・リカバリを実行します。</li> <li>3. データベースを一貫した状態で停止し、バックアップを行います。</li> <li>4. データベースを元のリリースにダウングレードします。</li> </ol>                                      |
| メモリーの破損または一時的な問題                  | データベースを停止してリカバリを再開すると、問題を解決できる場合があります。2 回目の試行も失敗した場合は、データベースを一貫性のある状態のままにしておく必要があります。   |
| データ・ブロックの破損                       | ユーザー管理の方法を使用して、データファイルのリストアおよびリカバリを再度実行するか、または Recovery Manager の RECOVER ...BLOCK コマンドを使用して、個々のデータ・ブロックのリストアおよびリカバリを行います。この方法で問題が解決される場合もあります。<br>データ・ブロックのチェックサム検証が失敗した場合は、データ・ブロックが破損しています。DB_BLOCK_CHECKING が無効である場合は、データ・ブロックの破損の問題が、REDO の問題として表示されることがあります。メディア・リカバリを続行する必要がある場合は、ここでブロックを破損させてリカバリを続け、後で Recovery Manager を使用してブロック・メディア・リカバリを実行できます。 |

表 28-5 に示す方法で問題を解決できない場合は、データを消失せずに、簡単に問題を解決できる方法がない可能性があります。次のようなオプションがあります。

- RESETLOGS オプションを指定してデータベースをオープンします（データベース全体のリカバリのため）。

この解決方法では、REDO の問題が発生した時点以降のすべての変更が廃棄されますが、データベースの論理的な一貫性は保証されます。



- メディア・リカバリで1つ以上のデータ・ブロックを破損させることを許可して、続行します。

このオプションは、アラート・ログに、データ・ブロックを破損させることを許可すればリカバリを続行できることが示されている場合にのみ成功します。これはほとんどのリカバリの問題に当てはまります。データベースを迅速に起動し、すべての変更をリカバリする必要がある場合は、このオプションが最適です。このオプションを検討する場合は、28-23 ページの「[リカバリでブロックの破損を許可するかどうかの決定：フェーズ3](#)」に進みます。

**参照：** RECOVER ...BLOCK コマンドを使用してブロック・メディア・リカバリを実行する方法については、19-9 ページの「[障害リカバリの実行](#)」を参照してください。

## リカバリでブロックの破損を許可するかどうかの決定：フェーズ3

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

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

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

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

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

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

リカバリの問題が分離されているかどうかを確認するために、診断用の**試行リカバリ**を実行できます。試行リカバリでは、REDO ストリームをスキャンして問題が検索されますが、リカバリされたデータベースに対して実際の変更は行いません。試行リカバリでリカバリの問題が検出された場合、alert\_SID.log で報告されます。試行リカバリを起動するには、28-25 ページの「[RECOVER ... TEST 文の実行](#)」で説明するように、RECOVER ... TEST 文を使用します。

これらの調査を実行した後、表 28-6 のガイドラインに従って、リカバリでブロックの破損を許可するかどうかを決定します。

表 28-6 リカバリに破損を許可する場合のガイドライン

| 問題の状態    | ブロック               | 説明 ..  |
|----------|--------------------|--|
| 分離されていない | N/A                | 多くの場合、RESETLOGS オプションを指定してデータベースをオープンする必要があります。スタック・リカバリはオペレーティング・システムまたはストレージ・システムで書き込みが消失したために発生することがあるため、この対応はスタック・リカバリの問題にとって重要です。オペレーティング・システムまたはストレージ・システムで予期せぬエラーが発生した場合には、複数のブロックでスタック・リカバリの問題が発生する可能性があります。 |
| 分離されている  | SYSTEM 表領域内        | 最終的にデータベースをオープンできなくなる可能性があるため、ブロックは破損させないでください。ただし、SYSTEM 表領域の中のデータが重要でない場合もあります。SYSTEM ブロックを破損させ、すべての変更のリカバリを行う必要がある場合には、Oracle サポート・サービスに連絡してください。   |
| 分離されている  | 索引データ              | データベースのリカバリ後に索引を再作成できるため、索引ブロックを破損させることを検討してください。  |
| 分離されている  | ユーザー・データ           | データの重要性に基づいて決定します。データファイルのリカバリを続行し、ブロックを破損させると、ブロック内のデータが消失します。ただし、データファイルのリカバリが完了した後で、Recovery Manager を使用してブロック・メディア・リカバリを実行できます。RESETLOGS でオープンすると、データベースは一貫性のあるものになりますが、リカバリが停止された時点以降のすべての変更は失われます。             |
| 分離されている  | ロールバックまたは UNDO データ | すべてのトランザクションがコミットされる場合は、ロールバックまたは UNDO ブロックを破損させることを検討します。UNDO を生成したトランザクションをロールバックしない場合は、データベースに害はありません。ただし、これらのトランザクションをロールバックする場合は、UNDO ブロックを破損させると問題が発生する可能性があります。判断できない場合は、Oracle サポート・サービスに連絡してください。           |

**参照：** 試行リカバリの実行方法については、28-25 ページの「[試行リカバリの実行](#)」を参照してください。ブロックを破損させることを決定した場合は、28-24 ページの「[リカバリでのブロックの破損の許可：フェーズ 4](#)」を参照してください。

## リカバリでのブロックの破損の許可：フェーズ 4

リカバリの続行のために、ブロックの破損を許可することを決定した場合には、ALLOW *n* CORRUPTION 句を指定して RECOVER コマンドを実行します。この *n* は、破損させることを許可するブロックの数です。

### リカバリでブロックを破損させることを許可する手順

1. リカバリの通常の前提条件がすべて満たされていることを確認します。たとえば、データベースがオープンしている場合には、リカバリを試みる前に、表領域をオフライン化します。
2. RECOVER コマンドを実行します。たとえば、次のように入力します。

```
RECOVER DATABASE ALLOW 5 CORRUPTION
```



## 試行リカバリの実行

スタック・リカバリなどの問題が発生した場合、判断が困難な場合があります。ブロックが比較的重要でなく、問題が分離されている場合は、通常、ブロックを破損させます。ただし、問題が分離されていない場合は、RESETLOGS オプションを指定してデータベースをオープンする方がよい場合があります。

このため、Oracle Database では、**試行リカバリ**がサポートされています。試行リカバリは、通常のメディア・リカバリと同じ方法で REDO を適用しますが、ディスクに変更を書き込むのではなく、必ず変更をロールバックします。試行リカバリはメモリー内でのみ発生します。

**参照:** 「リカバリでのブロックの破損の許可: フェーズ 4」 (28-24 ページ)

### 試行リカバリの仕組み

デフォルトでは、試行リカバリでスタック・リカバリやその他の問題が検出されると、常に、メモリー内でデータ・ブロックに破損しているというマークが付けられます (このアクションによってリカバリを続行できる場合)。試行リカバリ中に生成されたエラーは、アラート・ファイルに書き込まれます。これらのエラーは、テスト実行のエラーであることが明示されます。

試行リカバリでは、通常のメディア・リカバリと同様に、アーカイブ・ログのファイル名を指定するプロンプトが表示され、ユーザーは、ログを適用するかどうかを判断するように求められます。試行リカバリは次の場合に終了します。

- 試行リカバリ用に使用できるメモリー内の最大バッファ数がすべて使用された場合
- リカバリ不能なエラー (データ・ブロックを破損させても解決できないエラー) が通知された場合
- ユーザーがリカバリ・セッションを取り消したか中断した場合
- REDO ストリーム内の次の REDO レコードによって制御ファイルが変更された場合
- 必要な REDO がすべて適用された場合

試行リカバリが終了すると、アラート・ファイル内のエラー・メッセージを除き、テスト実行のすべての影響がシステムから削除されます。試行リカバリ中にインスタンスの障害が発生した場合は、試行リカバリが変更をディスクに書き込むことはないため、試行リカバリのすべての影響がシステムから削除されます。

試行リカバリを使用することによって、通常のリカバリを続行した場合に発生する可能性のある問題を予測できます。進行中のメモリーの破損を原因とする問題の場合は、試行リカバリと通常のリカバリで発生するエラーが異なることがあります。

### RECOVER ... TEST 文の実行

TEST オプションはすべての RECOVER コマンドで使用できます。たとえば、SQL\*Plus を起動して次のコマンドのいずれかを発行できます。

```
RECOVER DATABASE TEST
RECOVER DATABASE USING BACKUP CONTROLFILE UNTIL CANCEL TEST
RECOVER TABLESPACE users TEST
RECOVER DATABASE UNTIL CANCEL TEST
```

デフォルトでは、試行リカバリを実行すると、メモリー内でブロックの破損が常に試行されます (このアクションによって試行リカバリを続行できる場合)。つまり、試行リカバリはデフォルトで、いくつでもデータ・ブロックを破損させることができます。RECOVER ... TEST 文で ALLOW n CORRUPTION 句を指定すると、試行リカバリの実行時にメモリー内で破損させることができるデータ・ブロックの数を制限できます。

試行リカバリのコマンドは、通常リカバリのコマンドを使用できるすべての場合に使用できます。ただし、試行リカバリを実行する必要があるのは、リカバリで問題が発生する場合のみです。



---

## ユーザー管理のリカバリの実行： 高度な例

この章では、複数の一般的なメディア障害の例について説明します。ユーザー管理のバックアップおよびリカバリ計画（Recovery Manager を必要としない計画）を使用した場合に、各障害からリカバリする方法について説明します。この章の内容は、次のとおりです。

- 現行の制御ファイルのサブセットが消失した場合の対応
- 現行の制御ファイルがすべて消失した場合のリカバリ
- 制御ファイルの再作成
- バックアップが利用できない場合のデータファイルの再作成
- NOLOGGING 表および索引のリカバリ
- トランスポータブル表領域のリカバリ
- オンライン REDO ログ・ファイルが消失した後のリカバリ
- フラッシュバック機能を使用しない、削除された表のリカバリ
- SQL\*Plus でのデータベースの削除

## 現行の制御ファイルのサブセットが消失した場合の対応

永続的なメディア障害によってデータベースの1つ以上の制御ファイルが破損したが、メディア障害によって消失していない現行の制御ファイルが1つ以上ある場合は、次の手順でデータベースをリカバリします。

### 多重制御ファイルのデフォルトの場所へのコピー

消失した制御ファイルを含むディスクおよびファイル・システムが影響を受けていない場合は、影響を受けていない制御ファイルのいずれかを、欠落した制御ファイルの場所にコピーできます。この場合は、CONTROL\_FILES 初期化パラメータを編集する必要はありません。

#### 多重制御ファイルをコピーして破損した制御ファイルを置き換える手順

1. インスタンスが実行されている場合は、停止します。

```
SQL> SHUTDOWN ABORT
```

2. メディア障害の原因であるハードウェアの問題を解決します。ハードウェアの問題をすぐに解決できない場合は、29-2 ページの「[多重制御ファイルのデフォルト以外の場所へのコピー](#)」の説明に従って、破損した制御ファイルを代替のストレージ・デバイスにリストアしてデータベースのリカバリを続行します。
3. データベースの現行の制御ファイルの多重コピーから、影響を受けていないものを選択して、破損した制御ファイルを上書きします。たとえば、bad\_cf.f を good\_cf.f で置き換えるには、次のように入力します。

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

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

```
SQL> STARTUP
```

### 多重制御ファイルのデフォルト以外の場所へのコピー

消失した制御ファイルを含むディスクおよびファイル・システムが影響を受けている場合は、正常な制御ファイルのいずれかを、欠落した制御ファイルの場所に単純にコピーできません。この場合は、CONTROL\_FILES 初期化パラメータを変更して、欠落した制御ファイルの新しい場所を示す必要があります。

#### デフォルト以外の場所に制御ファイルをリストアする手順

1. インスタンスが実行されている場合は、停止します。

```
SQL> SHUTDOWN ABORT
```

2. メディア障害の原因であるハードウェアの問題を解決できない場合は、影響を受けていない制御ファイルを代替の場所にコピーします。

たとえば、正常な control01.dbf を新しいディスクの場所にコピーするには、次のコマンドを発行します。

```
% cp /disk1/oradata/trgt/control01.dbf /new_disk/control01.dbf
```

3. CONTROL\_FILES パラメータですべての制御ファイルの現在の場所が反映され、リストアされていないすべての制御ファイルが除外されるように、データベースのパラメータ・ファイルを編集します。

初期化パラメータ・ファイルの設定が次のようになっているとします。

```
CONTROL_FILES='/disk1/oradata/trgt/control01.dbf','/bad_disk/control02.dbf'
```

次のように CONTROL\_FILES 初期化パラメータを編集します。

```
CONTROL_FILES='/disk1/oradata/trgt/control01.dbf','/new_disk/control02.dbf'
```

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

```
SQL> STARTUP
```

## 現行の制御ファイルがすべて消失した場合のリカバリ

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

```
ORA-00205: error in identifying control file, check alert log for more info
```

---

**注意：** トレース・ファイルおよびアラート・ログの場所を特定する最も簡単な方法は、SQL 問合せ `SELECT NAME, VALUE FROM V$DIAG_INFO` を実行する方法です。

---

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

表 29-1 に示すように、制御ファイルのリストアの手順は、オンライン REDO ログを利用できるかどうかによって異なります。

表 29-1 制御ファイルが消失した場合のシナリオ

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

## デフォルトの場所でのバックアップ制御ファイルを使用したりカバリ

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

### デフォルトの場所にバックアップ制御ファイルをリストアする手順

1. インスタンスが実行されている場合は、停止します。

```
SQL> SHUTDOWN ABORT
```

2. メディア障害の原因であるハードウェアの問題を解決します。

3. CONTROL\_FILES パラメータで指定されたすべての場所にバックアップ制御ファイルをリストアします。たとえば、サーバー・パラメータ・ファイルに示される制御ファイルの場所が /disk1/oradata/trgt/control01.dbf および /disk2/oradata/trgt/control02.dbf の場合は、オペレーティング・システム・ユーティリティを使用して、これらの場所にバックアップ制御ファイルをリストアします。

```
% cp /backup/control01.dbf /disk1/oradata/trgt/control01.dbf
% cp /backup/control02.dbf /disk2/oradata/trgt/control02.dbf
```

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

```
SQL> STARTUP MOUNT
```

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

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

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

たとえば、次のメッセージが表示されたとします。

```
ORA-00279: change 55636 generated at 11/08/2002 16:59:47 needed for thread 1
ORA-00289: suggestion : /oracle/work/arc_dest/arc_r_1_111.arc
ORA-00280: change 55636 for thread 1 is in sequence #111
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
```

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

```
ORACLE_HOME/oradata/redo01.dbf
Log applied.
Media recovery complete.
```

オンライン・ログにアクセスできない場合は、オンライン・ログを適用せずにリカバリを取り消すことができます。すべてのデータファイルが現行のものである場合に、リカバリに必要な REDO がオンライン・ログの中に含まれているときには、オンライン・ログを適用せずにデータベースをオープンすることはできません。オンライン・ログにアクセスできない場合は、29-6 ページの「[制御ファイルの再作成](#)」に示す手順を実行して、制御ファイルを再作成する必要があります。

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

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

## デフォルト以外の場所でのバックアップ制御ファイルを使用したリカバリ

メディアの損傷が重大なため、元の場所に制御ファイルをリストアできない場合は、サーバー・パラメータ・ファイルで制御ファイルの新しい場所を指定する必要があります。CONTROL\_FILES 初期化パラメータで指定されたすべての場所に有効な制御ファイルが存在する必要があります。存在しない場合は、データベースをマウントできません。

### デフォルト以外の場所に制御ファイルをリストアする手順

- 29-4 ページの「デフォルトの場所でのバックアップ制御ファイルを使用したリカバリ」の手順に従いますが、手順 2 の後に、次の手順を追加します。

CONTROL\_FILES 初期化パラメータで指定されたすべての場所を編集して、制御ファイルの新しい場所を反映します。たとえば、サーバー・パラメータ・ファイルに制御ファイルの場所が次のように表示され、いずれのディスクにもアクセスできないとします。

```
CONTROL_FILES='/disk1/oradata/trgt/control01.dbf',
              '/disk2/oradata/trgt/control02.dbf'
```

次の例のように初期化パラメータ・ファイルを編集して、アクセス可能な場所を指定できます。

```
CONTROL_FILES='/disk3/cf/control01.dbf','/disk4/cf/control02.dbf'
```

## バックアップ制御ファイルを使用したデータファイルの追加を伴うリカバリ

バックアップ制御ファイルを使用したデータベースのリカバリが、CREATE TABLESPACE または ALTER TABLESPACE ADD DATAFILE 操作を伴ってロールフォワードされる場合は、追加されたファイルに REDO レコードを適用するときにリカバリが停止し、ユーザーによるファイル名の確認が行われます。

たとえば、次の操作を順に行うとします。

1. データベースをバックアップします。
2. /disk1/oradata/trgt/test01.dbf および /disk1/oradata/trgt/test02.dbf というデータファイルを含む新しい表領域を作成します。
3. バックアップ制御ファイルをリストアし、CREATE TABLESPACE 操作を伴うメディア・リカバリを実行します。

CREATE TABLESPACE の REDO データの適用時に、次のエラーが表示される場合があります。

```
ORA-00283: recovery session canceled due to errors
ORA-01244: unnamed datafile(s) added to control file by media recovery
ORA-01110: data file 11: '/disk1/oradata/trgt/test02.dbf'
ORA-01110: data file 10: '/disk1/oradata/trgt/test01.dbf'
```

### ADD DATAFILE 操作を伴うリカバリを実行する手順

1. V\$DATAFILE を問い合せて、追加されたファイルを表示します。たとえば、次のように入力します。

```
SELECT FILE#,NAME
FROM V$DATAFILE;
```

```
FILE#          NAME
-----
1              /disk1/oradata/trgt/system01.dbf
.
.
.
10             /disk1/oradata/trgt/UNNAMED00001
11             /disk1/oradata/trgt/UNNAMED00002
```

2. 名前のないファイルが複数存在する場合は、次のいずれかの方法を使用して、どの名前なしファイルがどのデータファイルに対応するかを確認します。
  - alert\_SID.log を参照し、名前のない各ファイルの元のファイルの場所に関するメッセージを確認します。
  - エラー・メッセージおよび V\$DATAFILE から、名前のない各ファイルの元のファイルの場所を導出します。名前のないファイルは、エラー・メッセージの中の同じファイル番号のファイルに対応します。
3. ALTER DATABASE RENAME FILE 文を発行して、データファイル名を変更します。たとえば、次のように入力します。
 

```
ALTER DATABASE RENAME FILE '/db/UNNAMED00001' TO
                        '/disk1/oradata/trgt/test01.dbf';
ALTER DATABASE RENAME FILE '/db/UNNAMED00002' TO
                        '/disk1/oradata/trgt/test02.dbf';
```
4. 前述のリカバリ文を発行してリカバリを続行します。たとえば、次のように入力します。
 

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

## バックアップ制御ファイルを使用した読取り専用表領域のリカバリ

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

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

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

## 制御ファイルの再作成

永続的なメディア障害によってすべての制御ファイルが消失した場合は、すべてのオンライン REDO ログ・メンバーが影響を受けていなければ、新しい制御ファイルを作成した後でデータベースをリカバリできます。リカバリ後に RESETLOGS オプションを指定してデータベースをオープンする必要はありません。

制御ファイルのバックアップの有無および作成日時に応じて、CREATE CONTROLFILE 文のテキストを生成する際に次の表に示すオプションを使用できます。データベースに対する変更は alert\_SID.log に記録されるため、どのオプションを選択するかを決定する際は、このログを確認してください。



表 29-2 制御ファイルの作成のオプション

| 状況 ..  | 説明 ..  |
|--|--|
| データベースの最後の構造変更の後に ALTER DATABASE BACKUP CONTROLFILE TO TRACE NORESETLOGS を実行し、SQL コマンド・トレース出力を保存してある場合 | トレース出力の CREATE CONTROLFILE 文をそのまま使用します。  |
| データベースの構造変更を行う前に、最後の ALTER DATABASE BACKUP CONTROLFILE TO TRACE を実行した場合                                | ALTER DATABASE BACKUP CONTROLFILE TO TRACE の出力を編集して、変更を反映します。たとえば、最近データベースにデータファイルを追加した場合には、このデータファイルを CREATE CONTROLFILE 文の DATAFILE 句に追加します。   |
| ALTER DATABASE BACKUP CONTROLFILE TO filename 文を使用して制御ファイルをバックアップした場合 (TO TRACE オプションを使用しなかった場合)       | 制御ファイルのコピーを使用して SQL 出力を取得します。一時的なデータベース・インスタンスを作成し、バックアップ制御ファイルをマウントして、ALTER DATABASE BACKUP CONTROLFILE TO TRACE NORESETLOGS を実行します。制御ファイルのコピーが最新の構造変更の前の日付になっている場合は、トレースを編集して変更を反映します。 |
| TO TRACE 形式でも TO filename 形式でも制御ファイルのバックアップがない場合   | CREATE CONTROLFILE 文を手動で実行します (『Oracle Database SQL 言語リファレンス』を参照)。   |

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

### 新しい制御ファイルを作成し、データベースをリカバリする手順

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

```
STARTUP NOMOUNT
```

2. NORESETLOGS オプションを指定して、CREATE CONTROLFILE 文で制御ファイルを作成します (オプションについては表 29-2 を参照)。次の例では、キャラクタ・セットがデフォルトの US7ASCII であるとしみます。

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

制御ファイルを作成した後、インスタンスでデータベースがマウントされます。

3. データベースを通常どおりに (USING BACKUP CONTROLFILE 句を指定せずに) リカバリします。

```
RECOVER DATABASE
```

4. リカバリが終了した後、データベースをオープンします (RESETLOGS オプションは必要ありません)。

```
ALTER DATABASE OPEN;
```

5. すぐに制御ファイルをバックアップします。次の SQL 文は、/backup/control01.dbf にデータベースの制御ファイルをバックアップします。

```
ALTER DATABASE BACKUP CONTROLFILE TO '/backup/control01.dbf' REUSE;
```

**参照：** 27-11 ページの「[制御ファイルのトレース・ファイルへのバックアップ](#)」および 29-9 ページの「[バックアップが利用できない場合のデータファイルの再作成](#)」を参照してください。

## 作成された制御ファイルを使用した RESETLOGS を伴うリカバリ

次の条件を満たす場合は、OPEN RESETLOGS 操作を使用してバックアップをリカバリできます。

- 過去のインカーネーションの情報を含む現行の制御ファイル、バックアップ制御ファイルまたは作成された制御ファイルがある
- すべてのアーカイブ REDO ログが使用可能である

制御ファイルを再作成する必要がある場合は、ALTER DATABASE BACKUP CONTROLFILE TO TRACE を使用してトレース・ファイルを生成すると、そのファイルにインカーネーションの完全な履歴を再構築するために必要なコマンドが含まれます。V\$DATABASE\_INCARNATION ビューには制御ファイルに含まれている RESETLOGS の履歴が表示され、V\$LOG\_HISTORY ビューにはアーカイブ・ログの履歴が表示されます。

インカーネーションの履歴は、再作成された制御ファイルでは不完全な場合があります。たとえば、リカバリに必要なアーカイブ・ログが欠落している場合があります。この場合は、ALTER DATABASE REGISTER LOGFILE 文を使用して、インカーネーションの記録を明示的に作成できます。

次の例では、リカバリに必要なが再作成された制御ファイルには記録されていない4つのログを登録して、データベースをリカバリします。

```
ALTER DATABASE REGISTER LOGFILE '/disk1/oradata/trgt/arch/arch_1_1_42343523.arc';
ALTER DATABASE REGISTER LOGFILE '/disk1/oradata/trgt/arch/arch_1_1_34546466.arc';
ALTER DATABASE REGISTER LOGFILE '/disk1/oradata/trgt/arch/arch_1_1_23435466.arc';
ALTER DATABASE REGISTER LOGFILE '/disk1/oradata/trgt/arch/arch_1_1_12343533.arc';
RECOVER AUTOMATIC DATABASE;
```

## 再作成された制御ファイルを使用した読取り専用ファイルのリカバリ

リカバリに現行の制御ファイルまたはバックアップ制御ファイルを使用できない場合は、CREATE CONTROLFILE 文を実行できます。リカバリでスキップできるように、読取り専用ファイルは CREATE CONTROLFILE 文に含めないでください。リストアされた読取り専用データファイルのバックアップが、ファイルが読取り / 書き込みであった時点のものでないかぎり、読取り専用データファイルのリカバリは必要ありません。

新しい制御ファイルを作成し、データベースのマウントおよびオープンを試行すると、制御ファイルにリストされたファイルに対してデータ・ディクショナリのチェックが実行されます。CREATE CONTROLFILE 文に含まれていないが、データ・ディクショナリに存在する各ファイルについては、制御ファイル内にエンタリが作成されます。これらのファイルには、MISSINGnnnnn という名前が付けられます。この nnnnn は、0 (ゼロ) から開始される5桁の数字です。

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

### 制御ファイルの再作成が必要となる状況に備えるための操作

- データベースをマウントまたはオープンしているときに次の文を実行し、CREATE CONTROLFILE 構文を取得します。

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

この SQL 文によって、トレース・ファイルが作成されます。トレース・ファイルは、ユーザーが編集し、制御ファイルを再作成するためのスクリプトとして使用できます。RESETLOGS または NORESETLOGS (デフォルト) キーワードを指定して、CREATE CONTROLFILE ... RESETLOGS または CREATE CONTROLFILE ... NORESETLOGS 対応のスクリプトを生成できます。

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

**参照:** 制御ファイルのトレース・バックアップの作成方法については、27-11 ページの「[制御ファイルのトレース・ファイルへのバックアップ](#)」を参照してください。

## バックアップが利用できない場合のデータファイルの再作成

データファイルが破損し、ファイルのバックアップが利用できない場合でも、次の条件を満たしていれば、データファイルをリカバリできます。

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

---

**注意:** ALTER DATABASE 文の CREATE DATAFILE 句を使用して SYSTEM 表領域のデータファイルを再作成することはできません。これは、必要な REDO を使用できないためです。

---

### リカバリのためにデータファイルを再作成する手順

1. 新しい空のデータファイルを作成して、対応するバックアップのない、破損したデータファイルと置き換えます。たとえば、データファイル /disk1/oradata/trgt/users01.dbf が破損し、このファイルのバックアップがないとします。次の文は、元のデータファイル (同じサイズ) を disk2 上に再作成します。

```
ALTER DATABASE CREATE DATAFILE '/disk1/oradata/trgt/users01.dbf' AS
'/disk2/users01.dbf';
```

この文は、消失したファイルと同じサイズの、空のファイルを作成します。制御ファイルおよびデータ・ディクショナリ内の情報が検索され、サイズ情報が取得されます。古いデータファイルは新しいデータファイルとして名前を変更されます。

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

```
RECOVER DATAFILE '/disk2/users01.dbf'
```

3. リカバリ時には、元のデータファイルが作成された後に書き込まれたすべてのアーカイブ・ログを、消失したデータファイルと置き換えた新しい空のファイルに適用する必要があります。

## NOLOGGING 表および索引のリカバリ

表および索引は CREATE TABLE AS SELECT 文で作成できます。また、NOLOGGING オプションを使用して作成するように指定することもできます。表または索引を NOLOGGING として作成した場合、操作の REDO ログ・レコードは生成されません。このため、ARCHIVELOG モードで実行している場合でも、NOLOGGING を使用して作成されたオブジェクトのリカバリを行うことはできません。

---

**注意：** NOLOGGING を使用して作成された表または索引の破損が許容できない場合は、リカバリ不能な表または索引を作成した後で、バックアップを作成します。

---

メディア・リカバリを実行する場合、表または索引の一部が通常どおりに作成され、その他が NOLOGGING オプションを使用して作成されているときは、RECOVER 操作を行うと、NOLOGGING オブジェクトに論理的に破損しているというマークが付けられます。リカバリ不能なオブジェクトにアクセスを試みると、ORA-01578 エラー・メッセージが戻されます。NOLOGGING オブジェクトを削除し、必要に応じて再作成します。

NOLOGGING オプションを使用して表を作成し、その表の索引を LOGGING オプションを使用して作成できるため、メディア・リカバリの実行後、索引には論理的に破損しているというマークが付けられません。ただし、表はリカバリ不能であるため（リカバリ後に破損しているというマークが付けられます）、索引は破損ブロックをポイントします。索引を削除し、必要に応じて表および索引を再作成する必要があります。

**参照：** データベースへの NOLOGGING の影響の詳細は、『Oracle Data Guard 概要および管理』を参照してください。

## トランスポータブル表領域のリカバリ

Oracle Database のトランスポータブル表領域機能を使用すると、ユーザーは表領域のセットを 1 つのデータベースから別のデータベースに転送できます。表領域をデータベースに転送することは、あらかじめロードされたデータを使用して表領域を作成することに似ています。この機能を使用することには、次の理由でメリットがあります。

- データファイルのコピーとメタデータの統合のみが行われるため、データ・ポンプ・エクスポートまたは SQL\*Loader ユーティリティを使用した場合より高速です。
- 索引データを移動できるため、索引を再作成する必要がありません。

**参照：** トランスポータブル表領域機能の使用方法の詳細は、『Oracle Database 管理者ガイド』を参照してください。

トランスポータブル表領域は、通常の表領域と同じように、リカバリ可能です。ただし、通常の表領域はバックアップなしにリカバリできますが、転送された表領域のリカバリには、転送されたデータファイルの一貫性が保持されているバージョンが必要です。

### トランスポータブル表領域をリカバリする手順

1. データベースがオープンしている場合は、転送された表領域をオフライン化します。たとえば、users 表領域のリカバリの場合には、次のコマンドを発行します。

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

2. オペレーティング・システム・ユーティリティを使用して、転送されたデータファイルのバックアップをリストアします。このバックアップは、転送されたデータファイルの最初のバージョンでも、表領域が転送された後で作成されたバックアップでもかまいません。たとえば、次のように入力します。

```
% cp /backup/users.dbf $ORACLE_HOME/oradata/trgt/users01.dbf
```

- 表領域を通常どおりにリカバリします。たとえば、次のように入力します。

```
RECOVER TABLESPACE users
```

CREATE TABLESPACE 操作を伴うリカバリの場合と同じように、トランスポータブル表領域の操作を伴うリカバリでもエラー ORA-01244 が表示される場合があります。この場合は、29-5 ページの「バックアップ制御ファイルを使用したデータファイルの追加を伴うリカバリ」の手順を使用して、名前のないファイルを正しい場所の名前に変更してください。

## オンライン REDO ログ・ファイルが消失した後のリカバリ

データベースのオンライン REDO ログがメディア障害の影響を受けたときの正しいリカバリ手順は、次の考慮事項によって異なります。

- オンライン REDO ログの構成：ミラー化されているかいないか
- メディア障害のタイプ：一時的か永続的か
- メディア障害の影響を受けたオンライン REDO ログ・ファイルのタイプ：現行、アクティブ、アーカイブされていない、または非アクティブ

表 29-3 に、オンライン REDO ログをリカバリする場合に重要な、V\$LOG のステータス情報を示します。

表 29-3 V\$LOG の STATUS 列

| ステータス            | 説明   |
|------------------|--|
| UNUSED           | このオンライン REDO ログは書き込まれたことはありません。  |
| CURRENT          | このオンライン REDO ログはアクティブです。インスタンス・リカバリに必要です。また、現在書き込みが行われているログです。REDO ログはオープンまたはクローズです。                                     |
| ACTIVE           | このオンライン REDO ログはアクティブです。インスタンス・リカバリに必要です。ただし、現在書き込みが行われているログではありません。ブロック・リカバリに使用されている可能性があります。また、アーカイブされる場合とされない場合があります。 |
| CLEARING         | このログは、ALTER DATABASE CLEAR LOGFILE 文の後で、空のログとして、再作成中です。ログが消去されると、ステータスは UNUSED に変更されます。                                  |
| CLEARING_CURRENT | 現在のログはクローズ・スレッドの消去中です。新しいログ・ヘッダーへの書き込み時の I/O エラーなど、なんらかのスイッチの障害が発生した場合には、ログがこのステータスのままになることがあります。                        |
| INACTIVE         | このログはインスタンス・リカバリに必要とされなくなりました。メディア・リカバリで使用される場合があります。アーカイブされる場合とされない場合があります。   |

## 多重オンライン REDO ログ・グループの一部のメンバーが消失した後のリカバリ

データベースのオンライン REDO ログが多重化されている場合は、各オンライン REDO ログ・グループの 1 つ以上のメンバーがメディア障害の影響を受けていなければ、データベースは通常どおりに機能し続けることができます。ただし、データベースのログ・ライターのトレース・ファイルおよび alert\_SID.log にエラー・メッセージが書き込まれます。

次のいずれかのアクションを実行して、問題を解決します。

- ハードウェアの問題が一時的な場合は、問題を解決します。ログ・ライター・プロセスは、問題が発生しなかった場合と同様に、以前は使用不可であったオンライン REDO ログ・ファイルにアクセスします。
- ハードウェアの問題が永続的な場合には、次の手順に従って、破損したメンバーを削除し、新しいメンバーを追加します。

**注意：** 新しく追加されたメンバーは、ログ・グループが再利用されるまでは、冗長性を提供しません。

1. V\$LOGFILE で破損したメンバーのファイル名を検索します。ファイルにアクセスできない場合、ステータスは INVALID になります。

```
SELECT GROUP#, STATUS, MEMBER
FROM V$LOGFILE
WHERE STATUS='INVALID';
```

```
GROUP#      STATUS      MEMBER
-----
0002        INVALID    /disk1/oradata/trgt/redo02.log
```

2. 破損したメンバーを削除します。たとえば、メンバー redo02.log をグループ 2 から削除するには、次のコマンドを発行します。

```
ALTER DATABASE DROP LOGFILE MEMBER '/disk1/oradata/trgt/redo02.log';
```

3. グループに新しいメンバーを追加します。たとえば、redo02.log をグループ 2 に追加するには、次のコマンドを発行します。

```
ALTER DATABASE ADD LOGFILE MEMBER '/disk1/oradata/trgt/redo02b.log'
TO GROUP 2;
```

追加するファイルがすでに存在している場合には、グループの他のメンバーと同じサイズである必要があります。また、REUSE を指定する必要があります。たとえば、次のように入力します。

```
ALTER DATABASE ADD LOGFILE MEMBER '/disk1/oradata/trgt/redo02b.log'
REUSE TO GROUP 2;
```

## オンライン REDO ログ・グループのすべてのメンバーが消失した後のリカバリ

オンライン REDO ログ・グループのすべてのメンバーがメディア障害によって破損した場合には、障害の影響を受けたオンライン REDO ログ・グループのタイプと、データベースのアーカイブ・モードによって、想定されるシナリオが異なります。

破損したオンライン REDO ログ・グループが現行のアクティブなものである場合は、このオンライン REDO ログ・グループがクラッシュ・リカバリで必要になります。非アクティブなものである場合は、クラッシュ・リカバリで必要とされません。

**表 29-4 オンライン REDO ログ・グループが消失した後のリカバリ**

| グループの状態.. | 説明..                   | ユーザーが取るべき処置..  |
|-----------|------------------------|--|
| 非アクティブ    | クラッシュ・リカバリで必要ありません     | アーカイブされたグループまたはアーカイブされていないグループを消去します。  |
| アクティブ     | クラッシュ・リカバリが必要です        | チェックポイントを発行してログを消去します。実行できない場合は、フラッシュバック・データベースを使用するか、またはバックアップをリストアし、使用可能な最新の REDO ログの時点まで、不完全リカバリを実行する必要があります。 |
| 現行        | 現在書込みが行われている REDO ログです | ログを消去します。消去できない場合は、フラッシュバック・データベースを使用するか、またはバックアップをリストアし、使用可能な最新の REDO ログの時点まで、不完全リカバリを実行する必要があります。              |

まず、破損したグループがアクティブか非アクティブかを確認します。

1. V\$LOGFILE で消失した REDO ログのファイル名を検索し、これに対応するグループ番号を検索します。たとえば、次のように入力します。

```
SELECT GROUP#, STATUS, MEMBER FROM V$LOGFILE;
```

| GROUP# | STATUS  | MEMBER              |
|--------|---------|---------------------|
| 0001   |         | /oracle/dbs/log1a.f |
| 0001   |         | /oracle/dbs/log1b.f |
| 0002   | INVALID | /oracle/dbs/log2a.f |
| 0002   | INVALID | /oracle/dbs/log2b.f |
| 0003   |         | /oracle/dbs/log3a.f |
| 0003   |         | /oracle/dbs/log3b.f |

2. どのグループがアクティブかを確認します。

たとえば、次の SQL 問合せを実行します（出力例も示します）。

```
SELECT GROUP#, MEMBERS, STATUS, ARCHIVED
FROM V$LOG;
```

| GROUP# | MEMBERS | STATUS   | ARCHIVED |
|--------|---------|----------|----------|
| 0001   | 2       | INACTIVE | YES      |
| 0002   | 2       | ACTIVE   | NO       |
| 0003   | 2       | CURRENT  | NO       |

3. 次のいずれかの処理を実行します。

- 影響を受けたグループが非アクティブな場合には、29-13 ページの[非アクティブのオンライン REDO ログ・グループの消失](#)の手順に従ってください。
- 影響を受けたグループがアクティブな場合（前述の例のような場合）には、29-14 ページの[「アクティブなオンライン REDO ログ・グループの消失」](#)の手順に従ってください。

## 非アクティブのオンライン REDO ログ・グループの消失

INACTIVE ステータスのオンライン REDO ログ・グループのすべてのメンバーが破損した場合は、その非アクティブ REDO ログ・グループを破損させたメディアの問題を解決できるかどうかによって、手順が異なります。障害が一時的な場合は、問題を解決します。ログ・ライターは、必要な場合に REDO ログ・グループを再利用できます。障害が永続的な場合は、破損した非アクティブのオンライン REDO ログ・グループは最終的に、データベースの通常の実行を停止させます。この項の説明に従って ALTER DATABASE CLEAR LOGFILE 文を発行し、破損したグループを手動で再初期化します。

**非アクティブのアーカイブ REDO の消去** データベースをオープンまたはクローズした状態で、非アクティブの REDO ログ・グループを消去できます。手順は、破損したグループがアーカイブされているかどうかで異なります。

### アーカイブされている、非アクティブのオンライン REDO ログ・グループを消去する手順

1. データベースが停止している場合は、新しいインスタンスを起動してデータベースをマウントします。

```
STARTUP MOUNT
```

2. 破損したログ・グループを再初期化します。たとえば、REDO ログ・グループ 2 を消去するには、次の文を発行します。

```
ALTER DATABASE CLEAR LOGFILE GROUP 2;
```



**非アクティブのアーカイブされていない REDO の消去** アーカイブされていない REDO ログを消去することによって、アーカイブせずに、ログを再利用できます。ログの中の最初の変更の前にファイルがオフライン化されておらず、ログの中の最後の変更前にバックアップが開始されている場合は、このアクションによってバックアップは使用できなくなります。このため、バックアップのリカバリのために消去されたログ・ファイルが必要となった場合は、このバックアップはリカバリできません。また、ログの欠落のため、バックアップからの完全リカバリも実行できません。

### アーカイブされていない、非アクティブのオンライン REDO ログ・グループを消去する手順

1. データベースが停止している場合は、新しいインスタンスを起動してデータベースをマウントします。

```
SQL> STARTUP MOUNT
```

2. UNARCHIVED キーワードを使用してログを消去します。

たとえば、ログ・グループ 2 を消去するには、次の SQL 文を発行します。

```
SQL> ALTER DATABASE CLEAR LOGFILE UNARCHIVED GROUP 2;
```

消去されたログを使用してオンライン化する必要があるオフライン・データファイルが存在する場合は、UNRECOVERABLE DATAFILE キーワードが必要です。オンライン化するために必要な REDO が消去され、コピーもないため、そのデータファイルと、その表領域全体は削除する必要があります。たとえば、次のように入力します。

```
SQL> ALTER DATABASE CLEAR LOGFILE UNARCHIVED GROUP 2 UNRECOVERABLE DATAFILE;
```

3. オペレーティング・システム・ユーティリティを使用して、すぐにすべてのデータファイルをバックアップします。これによって、消去されたログ・グループを使用しなくても完全リカバリを実行できるバックアップを作成します。たとえば、次のように入力します。

```
% cp /disk1/oracle/dbs/*.dbf /disk2/backup
```

4. ALTER DATABASE 文を使用して、データベースの制御ファイルをバックアップします。たとえば、次のように入力します。

```
SQL> ALTER DATABASE BACKUP CONTROLFILE TO '/oracle/dbs/cf_backup.f';
```

**CLEAR LOGFILE 操作の失敗** 次の操作を実行できない場合、メディア障害による I/O エラーによって、ALTER DATABASE CLEAR LOGFILE 文が失敗することがあります。

- 現在構成されている REDO ログのファイル名を基に REDO ログ・ファイルを再作成し、REDO ログ・ファイルを代替メディアへ再配置する
- 現在構成されているログ・ファイル名が（メディア障害などによって）無効または使用できなくなったため、その名前を再利用して REDO ログ・ファイルを再作成する

これらの場合には、ALTER DATABASE CLEAR LOGFILE 文は（I/O エラーを受け取る前に）、ログが消去され、アーカイブが必要なかったことを、制御ファイルに正しく通知します。CLEAR LOGFILE 文が新しい REDO ログ・ファイルを作成して 0（ゼロ）を書き込もうとした段階で I/O エラーが発生します。この事実は V\$LOG.CLEARING\_CURRENT に反映されます。

### アクティブなオンライン REDO ログ・グループの消失

データベースがまだ実行中であり、消失したアクティブな REDO ログが現行ログでない場合は、ALTER SYSTEM CHECKPOINT 文を発行します。この文が正常に実行されると、アクティブな REDO ログがアクティブでなくなるため、29-13 ページの「[非アクティブのオンライン REDO ログ・グループの消失](#)」の手順を実行できます。正常に実行されなかったか、またはデータベースが停止した場合は、アーカイブ・モードに応じて、この項のいずれかの手順を実行します。

現行ログとは、LGWR によって現在書込みが行われているログです。LGWR の I/O が失敗すると、LGWR は終了し、インスタンスがクラッシュします。この場合には、バックアップをリストアし、不完全リカバリを実行し、RESETLOGS オプションを指定してデータベースをオープンする必要があります。



**NOARCHIVELOG モードでのアクティブなログが消失した場合のリカバリ** この例では、データベースのアーカイブ・モードは NOARCHIVELOG です。

### NOARCHIVELOG モードでアクティブなオンライン・ログ・グループが消失した場合にリカバリを行う手順

1. メディア障害が一時的な場合は、必要に応じてデータベースがグループを再利用できるように、問題を解決します。
2. 一貫性のあるデータベース全体（データファイルおよび制御ファイル）のバックアップからデータベースをリストアします。たとえば、次のように入力します。

```
% cp /disk2/backup/*.dbf $ORACLE_HOME/oradata/trgt/
```

3. データベースをマウントします。

```
STARTUP MOUNT
```

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

```
RECOVER DATABASE UNTIL CANCEL  
CANCEL
```

5. RESETLOGS オプションを使用してデータベースをオープンします。

```
ALTER DATABASE OPEN RESETLOGS;
```

6. 一貫性のある状態でデータベースを停止します。たとえば、次のように入力します。

```
SHUTDOWN IMMEDIATE
```

7. データベース全体のバックアップを実行します。

メディア障害が一時的な場合は、必要に応じてデータベースがグループを再利用できるように、問題を解決します。メディア障害が一時的なものでない場合には、次の手順を使用します。

**ARCHIVELOG モードでのアクティブなログが消失した場合のリカバリ** この例では、データベースのアーカイブ・モードは ARCHIVELOG です。

### ARCHIVELOG モードでアクティブなオンライン REDO ログ・グループが消失した場合にリカバリを実行する手順

1. 不完全メディア・リカバリを開始して、破損したログの前のログまでリカバリします。
2. 消失した REDO ログの現在の名前を、新しく作成したファイルに使用できることを確認します。使用できない場合は、破損したオンライン REDO ログ・グループのメンバーの名前を新しい場所に変更します。たとえば、次のように入力します。

```
ALTER DATABASE RENAME FILE "/disk1/oradata/trgt/redo01.log" TO "/tmp/redo01.log";  
ALTER DATABASE RENAME FILE "/disk1/oradata/trgt/redo01.log" TO "/tmp/redo02.log";
```

3. RESETLOGS オプションを使用してデータベースをオープンします。

```
ALTER DATABASE OPEN RESETLOGS;
```

---

**注意：** 不完全リカバリのエンドポイントから現在までの間に実行されたすべての更新を、再度実行する必要があります。

---

## 複数の REDO ログ・グループの消失

オンライン REDO ログの複数のグループが消失した場合には、リカバリの最も困難なログのリカバリ方法を使用します。リカバリが困難な順に示します。

1. 現行のオンライン REDO ログ
2. アクティブなオンライン REDO ログ
3. アーカイブされていないオンライン REDO ログ
4. 非アクティブのオンライン REDO ログ

## フラッシュバック機能を使用しない、削除された表のリカバリ

データベースから誤って表を削除することは少なくありません。一般に、最も高速で簡単な解決方法は、フラッシュバック・ドロップ機能を使用して、表の削除を無効にする方法です。フラッシュバック表を使用できない場合（フラッシュバック・ドロップが無効になっている場合や表が PURGE オプションで削除された場合など）は、この項の手順を実行できます。

この項では、フラッシュバック・データベース機能が無効であるために FLASHBACK DATABASE は使用できないことを想定しています。ただし、データベースの物理バックアップは存在します。可能であれば、ユーザー・エラーが発生したデータベースをオンラインのまま保ち、使用できるようにします。

---

---

**注意：** 強力な権限を適切なユーザーのみに付与することによって、リカバリを必要とするユーザー・エラーを減らすことができます。

---

---

### 誤って削除された表をリカバリする手順

1. この手順の後の手順でエラーが発生した場合に備えて、既存のデータベースのすべてのデータファイルをバックアップします。
2. データベースの部分バックアップを代替の場所にリストアします。最低限、次のものをリストアします。
  - SYSTEM 表領域および SYSAUX 表領域
  - 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
  - 取得するデータが含まれている自己完結型の表領域
3. リストアされたバックアップ制御ファイルを使用して、表が削除された直前まで、このバックアップの不完全リカバリを実行します。
4. データ・ポンプ・エクスポートを使用して、データベースの一時的にリストアされたバージョンから、消失したデータをエクスポートします。この例では、誤って削除された表がエクスポートされます。

---

---

**注意：** システム監査オプションがエクスポートされます。

---

---

5. データ・ポンプ・インポート・ユーティリティを使用して、本番データベースにデータをインポートします。
6. 領域の節約のためにデータベースの一時コピーのファイルを削除します。

**参照：** Oracle Data Pump の詳細は、『Oracle Database ユーティリティ』を参照してください。

## SQL\*Plus でのデータベースの削除

データベース（データベースを構成するデータベース・ファイル）をオペレーティング・システムから削除する必要がある場合があります。たとえば、テスト・データベースを作成した後、そのデータベースを使用しなくなった場合などがこれに当たります。SQL 文 DROP DATABASE を実行すると、データベースを削除できます。

**参照：** 同等の機能を持つ Recovery Manager コマンド DROP DATABASE の使用方法については、11-25 ページの「[データベースの削除](#)」を参照してください。

1. 管理者権限を使用してターゲット・データベースに接続した後、データベースがマウントされているか、または制限モードでユーザーが接続していない状態でオープンしていることを確認します。

たとえば、次のコマンドを入力します。

```
SQL> STARTUP RESTRICT FORCE MOUNT
```

2. データファイルおよび制御ファイルを、オペレーティング・システムから削除します。

たとえば、次のコマンドを入力します。

```
SQL> DROP DATABASE; # deletes all database files, both ASM and non-ASM
```

データベースが RAW ディスクに配置されている場合、このコマンドでは RAW ディスクの実際の特殊ファイルを削除できません。

3. オペレーティング・システム・ユーティリティを使用して、データベースに関連するすべてのバックアップおよびアーカイブ・ログを削除します。

たとえば、次のコマンドを入力します。

```
% rm /backup/* /disk1/oradata/trgt/arch/*
```



## ARCHIVELOG モード (ARCHIVELOG mode)

一杯になったオンライン REDO ログを Oracle Database でディスクにコピーするためのデータベースのモード。このモードは、データベースの作成時に指定するか、または ALTER DATABASE ARCHIVELOG 文を使用して指定する。

「[アーカイブ REDO ログ](#)」、「[NOARCHIVELOG モード](#)」を参照。

## DBID

データベースを区別するために内部で一意的に生成される番号。Oracle では、データベースの作成時にこの番号が自動的に作成される。

## LogMiner

SQL 文を使用してログ・ファイルの読取り、分析および解析ができるようにするためのユーティリティ。

「[アーカイブ REDO ログ](#)」を参照。

## MTTR

「[平均リカバリ時間](#)」を参照。

## NOARCHIVELOG モード (NOARCHIVELOG mode)

一杯になったオンライン REDO ログを上書きする前に、Oracle によってそれらのオンライン REDO ログのアーカイブを要求されないデータベースのモード。このモードは、データベースの作成時に指定するか、ALTER DATABASE NOARCHIVELOG コマンドを使用して指定する。

NOARCHIVELOG モードで実行すると、消失または破損したデータのリカバリの可能性が著しく制限されることに注意すること。

「[アーカイブ REDO ログ](#)」、「[ARCHIVELOG モード](#)」を参照。

## NORMAL モードでのオフライン (offline normal)

表領域は、ALTER TABLESPACE ... OFFLINE NORMAL 文でオフラインにされた場合、NORMAL モードでオフラインされる。この表領域内のデータファイルにはチェックポイントが設定されるため、オンラインに戻されるまでリカバリは不要となる。表領域が NORMAL モードでオフラインされていない場合、その表領域内のデータファイルは、オンラインに戻す前にリカバリする必要がある。

## Oracle Flashback Database

Recovery Manager または SQL の FLASHBACK DATABASE コマンドを使用して、データベース全体を以前の一貫性のある SCN まで戻すこと。データベース・フラッシュバックは、物理ファイルをリストアせずに、変更されたデータ・ブロックの保存イメージを使用して、現行のデータファイルの過去の状態にリストアするため、従来のメディア・リカバリとは異なる。この機能では、[フラッシュバック・ログ](#)およびアーカイブ REDO ログが使用される。

## Oracle Managed Files (OMF)

いくつかの初期化パラメータに基づいて、制御ファイル、REDO ログ・ファイル、データファイルなどのデータベース・ファイルの名前の指定、場所の設定、作成、削除を自動化するサービス。Oracle 管理ファイルは、VxFS や ODM などのホスト・オペレーティング・システムでサポートされている従来のファイル・システムに加えて使用できる。データベース管理の多くの側面について独自の方針を作成する必要をなくすことによって、それらの詳細を簡略化できる。

## Oracle Secure Backup

ファイル・システムをテープにバックアップすることによって、信頼性の高いデータ保護を提供する Oracle メディア・マネージャ。Oracle Secure Backup の **SBT** インタフェースを使用すると、Recovery Manager を使用して Oracle Database をバックアップすることもできる。SAN、ギガビット・イーサネットおよび SCSI 環境のすべての主要なテープ・ドライブおよびライブラリがサポートされている。

## Oracle VSS ライター (Oracle VSS writer)

Oracle Database インスタンスとその他の **Volume Shadow Copy Service (VSS)** のコンポーネントの間のコーディネータとして機能する Windows システム上のサービス。このサービスによって、データ・プロバイダは、Oracle インスタンスで管理されるファイルのシャドウ・コピーを作成できる。たとえば、Oracle VSS ライターでは、データファイルをホット・バックアップ・モードにして、それらのデータファイルのリカバリ可能なコピーをシャドウ・コピー・セットに作成できる。

## Oracle 管理ファイル (Oracle-managed file)

Oracle Managed Files 機能によって管理されるデータベース・ファイル。

## Oracle 推奨バックアップ計画 (Oracle-suggested backup strategy)

Oracle Enterprise Manager でウィザードを使用して実行できるバックアップ計画。この計画では、定期的にレベル 1 **増分バックアップ** をレベル 0 バックアップに適用して、**増分更新バックアップ** を作成する。この計画を毎日実行することによって、ディスクからの **Point-in-Time リカバリ** が 24 時間実行できるようになる。

## Oracle フラッシュバック技術 (Oracle Flashback Technology)

データ保護の層を追加するための Oracle Database 機能のセット。これらの機能には、Oracle Flashback Query、Oracle Flashback Version Query、Oracle Flashback Transaction Query、Oracle Flashback Transaction、Oracle Flashback Table、Oracle Flashback Drop、**Oracle Flashback Database** などがある。

フラッシュバック機能を使用して、データの過去の状態を表示したり、データベースの一部またはすべてを巻き戻すことができる。通常、フラッシュバック機能が適用されるほとんどの場合は、フラッシュバック機能の方がメディア・リカバリより効率的で簡単である。

## Point-in-Time リカバリ (point-in-time recovery)

データベース・ファイルを現在以外の時刻に不完全にリカバリすること。Point-in-Time リカバリは、**不完全リカバリ**とも呼ばれる。

「**メディア・リカバリ**」、「**リカバリする**」を参照。

## RAW デバイス (raw device)

ファイル・システムを持たないディスクまたはパーティション。このため、ls、Windows エクスプローラなどでこれらの内容を確認することはできない。RAW パーティションは、Oracle Database では単一ファイルとして表示される。

## Recovery Manager

「**Recovery Manager (RMAN)**」を参照。

## Recovery Manager (RMAN)

Oracle Database の物理バックアップおよびリカバリに使用されるプライマリ・ユーティリティ。Recovery Manager は、Recovery Manager リポジトリと呼ばれる独自の構造に Oracle Database のレコードを保持し、バックアップのストレージを管理し、バックアップを検証する。これは、**リカバリ・カタログ**と呼ばれる中央情報リポジトリとあわせて使用することも、またリカバリ・カタログなしで使用することもできる。リカバリ・カタログを使用しない場合、Recovery Manager は、バックアップおよびリカバリの操作に必要な情報を格納するために、データベースの制御ファイルを使用する。Recovery Manager をサード・パーティのメディア管理ソフトウェアとともに使用すると、ファイルを 3 次ストレージにバックアップできる。

「**バックアップ・ピース**」、「**バックアップ・セット**」、「**コピー**」、「**メディア・マネージャ**」、「**リカバリ・カタログ**」を参照。

## Recovery Manager クライアント (RMAN client)

コマンドを解釈し、それらのコマンドを実行するようにサーバー・セッションに指示し、そのアクティビティをターゲット・データベースの制御ファイルに記録する Oracle Database 実行可能ファイル。この Recovery Manager 実行可能ファイルは、データベースとともに自動的にインストールされ、通常、他のデータベース実行可能ファイルと同じディレクトリに配置される。たとえば、Linux 上の Recovery Manager クライアントは rman と名付けられ、\$ORACLE\_HOME/bin に格納される。

## Recovery Manager ジョブ (RMAN job)

**Recovery Manager セッション**で実行される Recovery Manager コマンドのセット。たとえば、Recovery Manager クライアントを起動し、BACKUP DATABASE、BACKUP ARCHIVELOG および RECOVER COPY を実行して Recovery Manager クライアントを終了する。Recovery Manager ジョブは、2 回のバックアップ、およびデータファイルのコピーのロールフォワードで構成される。

## Recovery Manager セッション (RMAN session)

Recovery Manager セッションは、Recovery Manager クライアントの起動時に開始され、クライアントの終了時または Recovery Manager プロセスの完了時に終了する。単一の Recovery Manager セッションで複数の Recovery Manager コマンドを実行できる。

## Recovery Manager のメンテナンス・コマンド (RMAN maintenance command)

Recovery Manager のメタデータ・レコードおよびバックアップの管理に使用できるコマンド。メンテナンス・コマンドには、CATALOG、CHANGE、CROSSCHECK および DELETE がある。

## Recovery Manager バックアップ・ジョブ (RMAN backup job)

単一の **Recovery Manager セッション**内で実行される BACKUP コマンドのセット。たとえば、Recovery Manager クライアントを起動し、BACKUP DATABASE、BACKUP ARCHIVELOG および RECOVER COPY を実行して Recovery Manager クライアントを終了する。Recovery Manager バックアップ・ジョブは、データベースのバックアップおよびアーカイブ REDO ログのバックアップで構成される。

## Recovery Manager リポジトリ (RMAN repository)

ターゲット・データベース上で行われたバックアップおよびリカバリ操作に関する Recovery Manager のメタデータのレコード。Recovery Manager リポジトリの正式なコピーは、常にターゲット・データベースの制御ファイルに格納される。**リカバリ・カタログ**を使用して、Recovery Manager リポジトリを長期間保存することもできる。また、リカバリ・カタログは、データベースの制御ファイルが消失した場合に Recovery Manager リポジトリ・データの代替ソースとして使用できる。

「**リカバリ・カタログ・データベース**」、「**再同期化**」を参照。

## REDO スレッド (redo thread)

インスタンスによって生成される REDO。データベースがシングル・インスタンス構成で実行されている場合、そのデータベースには REDO のスレッドが 1 つのみ存在する。

## REDO ログ (redo log)

**オンライン REDO ログ**または**アーカイブ REDO ログ**のいずれかを指す。オンライン REDO ログは、Oracle データファイルおよび制御ファイルに行われたすべての変更が記録される 2 つ以上の REDO ログ・グループのセットである。アーカイブ REDO ログは、オフラインの転送先に書き込まれたオンライン REDO ログのコピーである。

## REDO ログ・グループ (redo log group)

オンライン REDO ログの各メンバー（それぞれがオンライン REDO ログ・ファイルに対応する）は、1 つの REDO ログ・グループに属している。REDO ログ・グループには、1 つ以上のメンバーが含まれている。2 つ以上のメンバーが含まれている REDO ログ・グループは、**多重 REDO ログ・グループ**と呼ばれる。REDO ログ・グループのすべてのメンバーの内容は同一である。

## RESETLOGS

データベースをオープンする方法の 1 つ。現行のオンライン REDO ログをアーカイブし (ARCHIVELOG モードを使用している場合)、ログ順序番号を 1 にリセットし、オンライン REDO ログをクリアする方法。ALTER DATABASE OPEN RESETLOGS 文によって、新しいデータベース・インカネーションが開始される。新しいインカネーションの開始 SCN (RESETLOGS SCN と呼ばれる) は、OPEN RESETLOGS の前に行われたメディア・リカバリの不完全リカバリ SCN に 1 を足したものになる。

不完全リカバリまたはバックアップ制御ファイルを使用したリカバリの後には、ALTER DATABASE OPEN RESETLOGS 文を実行する必要がある。OPEN RESETLOGS 操作は、データベースのリカバリ可能性には影響を与えない。OPEN RESETLOGS 操作前からのバックアップは有効なままであり、OPEN RESETLOGS 操作後に行われたバックアップとともに、データベースの損傷の修復に使用できる。

## RUN ブロック (RUN block)

順次実行される、一連の Recovery Manager コマンド。

## SBT

System Backup to Tape (テープへのシステム・バックアップ) の略称。この用語は、ディスク以外のバックアップ・デバイス・タイプを表す。通常はテープ・ライブラリまたはテープ・ドライブである。Recovery Manager では、ディスクおよび SBT のタイプのチャンネルがサポートされている。

## UNDO のバックアップの最適化 (backup undo optimization)

すでにコミットされたトランザクションを記述しているため Recovery Manager バックアップのリカバリに不要となった UNDO を除外すること。たとえば、ユーザーが USERS 表領域内の salaries 表を更新するとする。この変更は USERS 表領域に書き込まれ、データの変更前のイメージは UNDO 表領域に書き込まれる。次回、Recovery Manager で UNDO 表領域をバックアップする際に、給与の変更の UNDO は含まれない場合がある。UNDO のバックアップの最適化は、Recovery Manager に組み込まれた動作であり、無効にすることはできない。

## UNDO 表領域 (undo tablespace)

データベースが **自動 UNDO 管理モード**で実行されているときに、ロールバック情報のみが格納される専用表領域。

## UNDO 保存期間 (undo retention period)

Oracle Database で、**UNDO 表領域**内の古い UNDO データを上書きするまでの最短保存期間。現行の UNDO 保存期間より前の古い (コミットされた) UNDO データは、期限切れと呼ばれる。現行の UNDO 保存期間内の古い UNDO データは、期限内と呼ばれる。



## Volume Shadow Copy Service (VSS)

Windows サーバー・プラットフォームのインフラストラクチャの一種。これによって、リクエスト、ライターおよびプロバイダは、**シャドウ・コピー**と呼ばれる一貫性のあるスナップショットの作成に参加できる。VSS サービスでは、適切に定義された COM インタフェースが使用される。VSS とともに Recovery Manager を使用方法については、Oracle Database のプラットフォーム・ガイドを参照。

## アーカイブ (archiving)

一杯になったオンライン REDO ログ・ファイルをオフライン・ログのアーカイブ先にコピーする操作。オンライン REDO ログのオフライン・コピーは、**アーカイブ REDO ログ**と呼ばれる。REDO ログをアーカイブするには、データベースを ARCHIVELOG モードで実行する必要がある。

## アーカイブ REDO ログ (archived redo log)

オンライン REDO ログ・グループの一杯になったいずれかのメンバーのコピー。データベースが ARCHIVELOG モードの場合に作成される。LGWR プロセスによって各オンライン REDO ログが REDO レコードで一杯になると、ログは 1 つ以上の REDO ログのアーカイブ先にコピーされる。このコピーがアーカイブ REDO ログとなる。Recovery Manager は、元のアーカイブ REDO ログとアーカイブ REDO ログのイメージ・コピーを区別せず、両方をイメージ・コピーとみなす。

## アーカイブ REDO ログの削除方針 (archived redo log deletion policy)

アーカイブ REDO ログを削除できる場合を制御する構成可能で永続的な Recovery Manager の方針の 1 つ。この方針は、CONFIGURE ARCHIVELOG DELETION POLICY コマンドを使用して構成できる。

## アーカイブ REDO ログ・フェイルオーバー (archived redo log failover)

一部のアーカイブ・ログの出力先でログが欠落している場合またはログに破損ブロックが存在する場合でも、バックアップを完了できるようにする Recovery Manager の機能。たとえば、Recovery Manager は、フラッシュ・リカバリ領域にバックアップしたログが破損していると判断すると、他のアーカイブ場所にあるログを検索し、正常であればそのログをかわりにバックアップする。

## アーカイブ・バックアップ (archival backup)

通常のバックアップおよびリカバリ計画から除外されたデータベース・バックアップ。通常、このバックアップは、別のストレージ・メディアにアーカイブされ、長期間保存される。

## アクティブなデータベースの複製 (active database duplication)

**ターゲット・データベース**のバックアップをリストアせずに、ネットワーク上に**複製データベース**を作成すること。この方法は、**バックアップベースの複製**の代替方法である。

## 一時ファイル (tempfile)

一時表領域に属し、TEMPFILE オプションで作成されたファイル。一時表領域には、表などの永続的なデータベース・オブジェクトを含めることはできない。通常、一時表領域はソートの目的で使用される。一時ファイルには永続的オブジェクトが含まれていないため、Recovery Manager は一時ファイルをバックアップしない。ただし、Recovery Manager は、制御ファイル内の一時ファイルの場所を追跡し、リカバリ中、必要に応じてその場所に一時ファイルを再作成する。

## 一貫性のある状態での停止 (consistent shutdown)

文の IMMEDIATE、TRANSACTIONAL または NORMAL のオプションを使用して、データベースを停止すること。正しく停止されたデータベースにリカバリは不要である。このデータベースはすでに一貫性のある状態になっているためである。

### 一貫性バックアップ (consistent backup)

メディア・リカバリを実行することなく、RESETLOGS オプションを指定してオープンできる **データベース全体のバックアップ**。つまり、一貫性を保つためにこのバックアップに REDO を適用する必要はない。ただし、一貫性バックアップを作成した時点以降に生成された REDO を適用しないかぎり、一貫性バックアップを作成した時点以降のすべてのトランザクションが失われる。

一貫性バックアップは、データベースの**一貫性のある状態での停止**を実行した後でのみ作成できる。データベースは、バックアップが完了するまで再オープンしないようにする必要がある。

「**ファジー・ファイル**」、「**非一貫性バックアップ**」を参照。

### イメージ・コピー (image copy)

単一のデータファイル、アーカイブ REDO ログ・ファイルまたは制御ファイルのビット単位の**コピー**で、次のようなものを指す。

- そのまま使用して、リカバリを実行できるもの (**未使用ブロックの圧縮**を使用する、Recovery Manager 固有の形式のバックアップ・セットとは異なる)。
- Recovery Manager の BACKUP AS COPY コマンド、UNIX の cp などのオペレーティング・システムのコマンド、または Oracle アーカイバ・プロセスによって生成される。

### インカーネーション (incarnation)

データベースの個別のバージョン。データベースのインカーネーションは、RESETLOGS オプションでデータベースをオープンすると変更されるが、必要な REDO を使用できるかぎり、以前のインカーネーションからバックアップをリカバリできる。

### インスタンス障害 (instance failure)

ハードウェアの障害、Oracle 内部エラーまたは SHUTDOWN ABORT 文が原因で Oracle インスタンスが終了すること。インスタンス障害が発生した場合は、常にクラッシュ・リカバリまたはインスタンス・リカバリを実行する必要がある。

### インスタンス・リカバリ (instance recovery)

Oracle RAC 構成で、別のインスタンスがクラッシュしたことを検出したインスタンスによって、オープン状態のデータベースに REDO データが適用されること。

「**リカバリする**」を参照。

### エクスポート (export)

データ・ポンプ・エクスポートを使用して、論理データ（つまり、物理ファイルではないデータ）をデータベースからバイナリ・ファイルに抽出すること。その後、データ・ポンプ・インポートを使用してデータをデータベースにインポートできる。

「**論理バックアップ**」を参照。

### エクスポート・ダンプ・ファイル (export dump file)

データ・ポンプ・エクスポート・ユーティリティによって作成されるファイル。ダンプ・ファイル・セットは、表データ、データベース・オブジェクトのメタデータ、制御情報が含まれている 1 つ以上のディスク・ファイルで構成される。各ファイルは、独自のバイナリ形式で書き込まれる。

### オフサイトのバックアップ (offsite backup)

Recovery Manager によってリストアされる前に **メディア・マネージャ** で取得する必要がある **SBT** バックアップ。RESTORE ... PREVIEW を使用すると、オフサイトのバックアップを表示できる。

### オペレーティング・システムのバックアップ (operating system backups)

「**ユーザー管理バックアップ**」を参照。

## オペレーティング・システムのバックアップおよびリカバリ (operating system backup and recovery)

「[ユーザー管理のバックアップとリカバリ](#)」を参照。

## 親インカーネーション (parent incarnation)

OPEN RESETLOGS 操作後にブランチされた[現行のインカーネーション](#)の元となるデータベース・インカーネーション。

## オンライン REDO ログ (online redo log)

オンライン REDO ログは、データベースに行われたすべての変更が記録される 2 つ以上のファイルのセットである。データベースに変更が行われるたびに、Oracle によって REDO バッファに REDO レコードが生成される。LGWR プロセスによって、REDO バッファの内容がオンライン REDO ログに書き込まれる。

[現行のオンライン REDO ログ](#)とは、LGWR によって現在書込みが行われているオンライン REDO ログを指す。LGWR はファイルの最後に到達すると、[ログ・スイッチ](#)を実行し、新しいログ・ファイルへの書込みを開始する。データベースを ARCHIVELOG モードで実行している場合は、一杯になった各オンライン REDO ログ・ファイルを LGWR で上書きする前に、1 つ以上のアーカイブ場所にコピーする必要がある。

「[アーカイブ REDO ログ](#)」を参照。

## オンライン REDO ログ・グループ (online redo log group)

Oracle のオンライン REDO ログは、2 つ以上のオンライン REDO ログ・グループで構成されている。各グループには、1 つ以上の同一のオンライン REDO ログ・メンバーが含まれている。[オンライン REDO ログ・メンバー](#)は、REDO レコードを含む物理ファイルである。

## オンライン REDO ログ・メンバー (online redo log member)

[オンライン REDO ログ・グループ](#)内の物理的なオンライン REDO ログ・ファイル。各ログ・グループには、1 つ以上のメンバーが必要である。グループの各メンバーの内容は同じである。

## オンライン・バックアップ (online backup)

データベースがオープンされ、データファイルがオンラインである場合に行われる 1 つ以上のデータファイルのバックアップ。データベースをオープンした状態でユーザー管理のバックアップを作成する場合は、ALTER TABLESPACE BEGIN BACKUP コマンドを発行して、表領域を[バックアップ・モード](#)にする必要がある。(ALTER DATABASE BEGIN BACKUP を使用して、データベース内のすべての表領域を 1 回の動作でバックアップ・モードにすることもできる。)

Recovery Manager を使用してバックアップを実行する際は、表領域をバックアップ・モードにしてはいけない。

## 外部のアーカイブ REDO ログ (foreign archived redo log)

LogMiner セッションのロジカル・スタンバイ・データベースで受信されるアーカイブ REDO ログ。通常のアーカイブ・ログとは異なり、外部のアーカイブ・ログは異なる DBID を持つ。このため、ロジカル・スタンバイ・データベース上ではバックアップまたはリストアできない。

## 仮想プライベート・カタログ (virtual private catalog)

データベース・ユーザーがアクセス権限を付与されている、[リカバリ・カタログ](#)内のメタデータのサブセット。ベース・リカバリ・カタログの所有者は、他のデータベース・ユーザーに対してリカバリ・カタログへの制限付きアクセス権限の付与または取消しを行うことができる。制限付きユーザーは、それぞれ独自の仮想プライベート・カタログへの完全な読取り / 書込み権限を持っている。

## 完全再同期化 (full resynchronization)

Recovery Manager の処理の 1 つ。データベースの制御ファイル内にある変更されたメタデータを使用して、[リカバリ・カタログ](#)を更新する。Recovery Manager コマンドの RESYNC CATALOG を発行すると、完全なカタログ[再同期化](#)を開始できる。(Recovery Manager では必要に応じて再同期化が自動的に行われるため、RESYNC CATALOG を使用する必要はほとんどないことに注意。)

### 完全リカバリ (complete recovery)

バックアップのリストア後に生成されたすべての REDO に適用される 1 つ以上のデータファイルのリカバリ。通常、完全 **メディア・リカバリ** は、1 つ以上のデータファイルまたは制御ファイルが **メディア障害** によって破損した場合に実行される。損傷されたファイルは、バックアップのリストア以降に生成されたすべての REDO を使用して、完全にリカバリされる。

「**不完全リカバリ**」を参照。

### 期限切れのバックアップ (expired backup)

Recovery Manager リポジトリ内のステータスが (バックアップが検出されなかったことを意味する) EXPIRED であるバックアップ。CROSSCHECK コマンドの実行時に、バックアップおよびコピーが存在していないか、またはバックアップおよびコピーにアクセスできない場合、Recovery Manager はそれらのファイルを期限切れとマークする。

### 逆再同期化 (reverse resynchronization)

Data Guard 環境では、**リカバリ・カタログ**から取得したメタデータを使用して、プライマリ・データベースまたはスタンバイ・データベースの制御ファイルを更新すること。たとえば、接続されているターゲット・データベースではないスタンバイ・データベースに Recovery Manager の永続設定を構成すると、次回 Recovery Manager がターゲットとしてスタンバイ・データベースに接続する際に逆再同期化が実行される。このようにして、リカバリ・カタログでは Data Guard 環境での制御ファイル内のメタデータが最新に保たれる。

### クラッシュ・リカバリ (crash recovery)

シングル・インスタンス・データベースがクラッシュした後か、または Oracle Real Applications Clusters 構成のすべてのインスタンスがクラッシュした後のいずれかで、オンライン REDO レコードがデータベースに自動的に適用されること。クラッシュ・リカバリでは、オンライン・ログの REDO のみが必要となる。アーカイブ REDO ログは必要ない。

「**リカバリする**」を参照。

### クローズ状態のバックアップ (closed backup)

データベースがクローズされている場合に行われる 1 つ以上のデータベース・ファイルのバックアップ。通常、クローズ状態のバックアップはデータベース全体のバックアップである。データベースを一貫性のある状態でクローズした場合は、バックアップ内のすべてのファイルの一貫性が保たれる。そうしなかった場合は、バックアップの一貫性が保たれない。

「**一貫性のある状態での停止**」、「**一貫性バックアップ**」を参照。

### クロスチェック (crosscheck)

ディスクまたは**メディア管理カタログ**のファイルが、**Recovery Manager リポジトリ**のデータに対応しているどうかを確認するチェック。テープには**メディア・マネージャ**によって期限切れまたは使用不可のマークが付けられることがあり、ファイルはディスクから削除されたり破損することがあるため、Recovery Manager リポジトリにはバックアップに関する古い情報が含まれる場合がある。クロスチェックを行うには、CROSSCHECK コマンドを実行する。

「**検証**」を参照。

### 現行のインカーネーション (current incarnation)

データベースで REDO が現在生成されている**インカーネーション**。

### 現行のオンライン REDO ログ (current online redo log)

現時点において、LGWR バックグラウンド・プロセスで REDO レコードが記録されている**オンライン REDO ログ**・ファイル。

「**REDO ログ**」、「**REDO ログ・グループ**」を参照。

## 検証 (validation)

Recovery Manager のコンテキストでは、データベース・ファイルにブロックの破損がないかどうかをチェックするテスト、またはバックアップ・セットをチェックしてリストア可能かどうかを判断するテスト。Recovery Manager ではブロックの物理的および論理的な破損をチェックできる。

## コールド・バックアップ (cold backup)

「[クローズ状態のバックアップ](#)」を参照。

## コピー (copy)

Oracle ファイル (Oracle のデータファイル、制御ファイルおよびアーカイブ REDO ログ) のビットごとのイメージをディスクにバックアップすること。コピーには、次の 2 つの方法がある。

- オペレーティング・システムのユーティリティ (UNIX の cp や dd など) を使用する方法。
- Recovery Manager の BACKUP AS COPY コマンドを使用する方法。

「[バックアップ](#)」を参照。

## コマンド・ファイル (command file)

Recovery Manager のコンテキストでは、Recovery Manager の一連のコマンドが含まれているクライアント側のテキスト・ファイル。コマンド・ファイルは、@ コマンドまたは @@ コマンドを使用して Recovery Manager 内から実行するか、@ パラメータまたは CMDFILE パラメータを使用してオペレーティング・システム・プロンプトから実行できる。

## ごみ箱 (recycle bin)

削除されたオブジェクトに関する情報が含まれているデータ・ディクショナリ表。削除された表および関連するオブジェクト (索引、制約、ネストした表など) は、実際には削除されず、領域を占有している。フラッシュバック・ドロップ機能では、削除されたオブジェクトを取得するためにごみ箱が使用される。

## 固有の転送レート (native transfer rate)

テープ・ドライブで、圧縮なしでテープに書き込む場合の速度。この速度は、バックアップ・レートの上限を表す。

## 孤立したバックアップ (orphaned backups)

データベースの[現行のインカネーション](#)の[直系祖先パス](#)内で作成されなかったバックアップ。孤立したバックアップは、現行のインカネーションでは使用できない。

## 再開可能バックアップ (restartable backup)

Recovery Manager で、指定し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できるようにする機能。再開可能単位は最後に完了したバックアップ・セットまたはイメージ・コピーである。バックアップが失敗した場合にこの機能を使用すると、失敗したバックアップで処理されなかったデータベースの部分をバックアップできる。

## 再同期化 (resynchronization)

ターゲット・データベースの制御ファイルの現行のメタデータを使用して、[リカバリ・カタログ](#)を更新する操作。RESYNC CATALOG コマンドを発行すると、カタログの[完全再同期化](#)を開始できる。[部分再同期化](#)では、アーカイブ REDO ログ・ファイル、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リカバリ・カタログに転送する。Recovery Manager は、必要に応じてリカバリ・カタログを自動的に再同期化する。

### 差分増分バックアップ (differential incremental backup)

レベル1またはレベル0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をバックアップするタイプの**増分バックアップ**。たとえば、レベル1の差分バックアップでは、Recovery Managerは、レベル1または**レベル0の増分バックアップ**の最新のバックアップを判断し、そのバックアップ後に変更されたすべてのブロックをバックアップする。差分バックアップは、増分バックアップのデフォルト・タイプである。差分増分バックアップを使用してリカバリする場合、Recovery Managerは、リストアされたデータファイル・バックアップ以降のすべてのレベル1の差分増分バックアップを適用する必要がある。

「**累積増分バックアップ**」、「**増分バックアップ**」を参照。

### 試行リカバリ (trial recovery)

Recovery ManagerまたはSQL\*PlusのRECOVER ... TESTコマンドで開始されるリカバリのシミュレーション。**メディア・リカバリ**は、通常のメディア・リカバリと同じ方法でREDOを適用しますが、ディスクに変更を書き込むことはなく、必ず変更をロールバックする。試行リカバリはメモリー内でのみ発生する。

### システム変更番号 (system change number: SCN)

過去のある時点でのデータベースのコミットされたバージョンを定義するスタンプ。Oracleでは、コミットされたすべてのトランザクションに一意のSCNが割り当てられる。

### 自動UNDO管理モード (automatic undo management mode)

UNDOデータが専用の**UNDO表領域**に格納されるデータベース・モード。ユーザーが実行する必要があるUNDO管理操作は、UNDO表領域の作成のみである。他のすべてのUNDO管理操作は自動的に実行される。

### 自動診断リポジトリ (Automatic Diagnostic Repository: ADR)

データベース・トレース・ファイルおよびその他の診断データを格納および整理するためのシステム管理のリポジトリ。ADRでは、データベースで発生したすべての重大なエラーの包括的なビューが示され、問題の診断と最終的な解決に必要なすべての関連データが管理される。リポジトリには、インシデント、トレース、ダンプ、アラート・メッセージ、データ修復レコード、**データ整合性チェック**・レコード、SQLトレース情報、コア・ダンプなどについて記述されたデータが含まれている。

ADRベースの場所は、DIAGNOSTIC\_DEST初期化パラメータによって指定される。この場所が、1つ以上のADRホームが含まれるディレクトリとなる。各ADRホームは、適切に定義されたサブディレクトリに診断データを格納するために製品または製品インスタンスで使用される。たとえば、Oracle Databaseインスタンスの診断データは、専用のADRホームに格納される。このADRホームには、アラート・メッセージ用のalertサブディレクトリ、トレース・ファイル用のtraceサブディレクトリなどがある。トレース・ファイルおよびアラート・ログの場所を特定する最も簡単な方法は、SQL問合せSELECT NAME, VALUE FROM V\$DIAG\_INFOを実行する方法である。

### 自動ストレージ管理 (Automatic Storage Management: ASM)

Oracle Databaseファイル専用構築された、ファイル・システムとボリューム・マネージャの両方の垂直統合。ASMによって、簡単に管理可能なディスク・グループに複数のストレージ・デバイスが統合され、サード・パーティの論理的なボリューム・マネージャを必要とせずにミラー化やストライプ化などのメリットを実現できる。

### 自動チャンネル割当て (automatic channel allocation)

ALLOCATE CHANNELコマンドを使用せずに、バックアップおよびリストアのタスクを実行するRecovery Managerの機能。CONFIGUREコマンドを使用すると、ディスクおよびテープのチャンネルを指定できる。その後、手動でチャンネルを割り当てずに、Recovery Managerのコマンド・プロンプトでBACKUPやRESTOREなどのコマンドを発行できるようになる。Recovery Managerでは、コマンドを実行するために必要なすべての構成済のチャンネルが使用される。



## シャドウ・コピー (shadow copy)

Windows 上の **Volume Shadow Copy Service (VSS)** インフラストラクチャでは、コンポーネントまたはボリュームの一貫性のあるスナップショットのことである。

## 修復 (repair)

**データ・リカバリ・アドバイザー**のコンテキストでは、修復とは1つ以上の**障害**を修正するための単一または一連の処理のことである。修復の例としては、**ブロック・メディア・リカバリ**、**データファイルのメディア・リカバリ**、**Oracle Flashback Database** などがある。

## 修復オプション (repair option)

**データ・リカバリ・アドバイザー**のコンテキストでは、**障害**を修復するために使用可能な方法の1つ。同じ問題を修正するための様々な修復オプションがあるが、それぞれの方法に、修復時間およびデータ消失の点で異なるメリットおよびデメリットがある。

## 循環再利用レコード (circular reuse records)

Recovery Manager でバックアップおよびリカバリの操作に使用される情報が含まれている制御ファイルのレコード。これらのレコードは、論理的に円形に配置されている。使用可能なレコード・スロットが一杯の場合、制御ファイルを拡大して新規レコード用の領域を確保するか、あるいは最も古いレコードを上書きする。CONTROL\_FILE\_RECORD\_KEEP\_TIME 初期化パラメータで、レコードを上書きするまでの最短保管日数を制御する。CONTROL\_FILE\_RECORD\_KEEP\_TIME のデフォルト値は7日間である。

「**非循環再利用レコード**」を参照。

## 障害 (failure)

**データ・リカバリ・アドバイザー**のコンテキストでは、障害とは、データベースによって診断された永続的なデータの破損のことである。障害は、エラー・メッセージやアラートなどの目に見える症状として示される場合があるが、診断された問題を表すため、症状とは異なる。障害は、データベース外にある診断データ用のリポジトリに記録される。

障害ごとに、その障害を明確に説明する問題陳述文がデータ・リカバリ・アドバイザーによって生成される。障害の例として、アクセスできないデータファイルや破損した UNDO セグメントなどがある。データ・リカバリ・アドバイザーによって、すべての障害が1つの**修復オプション**または修復オプションのセットにマップされる。

## 障害ステータス (failure status)

**データ・リカバリ・アドバイザー**によって診断される**障害**のステータス。すべての障害のステータスは、OPEN または CLOSED のいずれかになる。

## 障害優先順位 (failure priority)

**データ・リカバリ・アドバイザー**によって診断される**障害**の優先順位。クローズされていないすべての障害のステータスは、CRITICAL、HIGH または LOW のいずれかになる。障害の HIGH および LOW のステータスは、CHANGE コマンドを使用して手動で変更できる。

## 障害リカバリ (disaster recovery)

データベースのインストールに関連するすべてのデータの消失に対する計画。たとえば、データ・センターのサーバーが火事によって破壊されたため、**Oracle Database** を新しいサーバーに再インストールして、消失したデータベースをバックアップからリカバリする必要がある場合など。

## 消失書込み (lost write)

実際には書込みが発生していないにもかかわらず、I/O サブシステムからの情報に基づいて発生がデータベースで認識された、永続ストレージへの書込み。

## 冗長性 (redundancy)

**保存方針**では、バックアップした各ファイルの保存するコピーの数を決定する設定。冗長性ベースの保存方針は、**リカバリ期間**を使用する保存方針と対比される。

### 冗長性セット (redundancy set)

任意の Oracle Database ファイルの障害または消失からのリカバリを可能にするバックアップのセット。

### ストアド・スクリプト (stored script)

**リカバリ・カタログ**に格納された一連の Recovery Manager コマンド。ストアド・スクリプトには、グローバルとローカルの 2 種類がある。グローバル・スクリプトは、リカバリ・カタログに登録されたすべてのデータベースで共有できる。

### スナップショット制御ファイル (snapshot control file)

Recovery Manager によって、オペレーティング・システム固有の場所に作成されたデータベース制御ファイルのコピー。Recovery Manager は、リカバリ・カタログの再同期化または制御ファイルのバックアップを行う際に使用する制御ファイルの一貫性のあるバージョンを確保するために、スナップショット制御ファイルを作成する。

### 制御ファイルの自動バックアップ (control file autobackup)

現在の制御ファイルおよびサーバー・パラメータ・ファイルの自動バックアップ。バックアップ後、およびデータベースが ARCHIVELOG モードである場合は構造変更後に、Recovery Manager によって作成される。

制御ファイルの自動バックアップにはデフォルトのファイル名がある。この名前によって、Recovery Manager は、制御ファイルおよびリカバリ・カタログが消失した場合でも、制御ファイルをリストアできる。デフォルトのファイル名は変更できる。

### 全体バックアップ (full backup)

増分ではない Recovery Manager のバックアップ。全体とは、バックアップされるデータベースの量を表すのではなく、バックアップが増分ではないことを表す。したがって、1 つのデータファイルに対して全体バックアップを作成できる。

### ソース・データベース (source database)

**複製データベース**の作成時にコピーするデータベース。

### ソース・プラットフォーム (source platform)

Recovery Manager の CONVERT コマンドの使用時に、ソース・データベースが実行されているプラットフォーム。ソース・データベースには、異なるプラットフォームで実行されているデータベースに転送されるデータが含まれている。

### ソース・ホスト (source host)

**ソース・データベース**が存在するホスト。

### 増分更新バックアップ (incrementally updated backup)

増分バックアップを使用して更新される、Recovery Manager のデータファイルのコピー。データファイルをコピーして増分バックアップを作成した後、その増分バックアップをイメージ・コピーにマージするバックアップ計画が効率的である。この計画では、データ・ブロックの最新の変更を使用してイメージ・コピーが更新されるため、**メディア・リカバリ**に必要な時間が削減される。

### 増分バックアップ (incremental backup)

修正されたブロックのみがバックアップされる Recovery Manager のバックアップ。増分バックアップはレベルによって分類される。**レベル 0 の増分バックアップ**は、使用されたすべてのブロックをバックアップするという点では**全体バックアップ**と同じである。ただし、全体バックアップでは後続の増分バックアップでバックアップされるブロックに影響は及ばないが、増分バックアップでは後続の増分バックアップでバックアップされるブロックに影響が及ぶ点が異なる。

レベル 1 の増分バックアップでは、前回の増分バックアップ以降に変更されたブロックのみがバックアップされる。変更されなかったブロックはバックアップされない。増分バックアップは、**差分増分バックアップ**または**累積増分バックアップ**のいずれかである。



### 祖先インカーネーション (ancestor incarnation)

**親インカーネーション**は、OPEN RESETLOGS 操作後にブランチされた**現行のインカーネーション**の元となるデータベース・**インカーネーション**である。親インカーネーションの親が祖先インカーネーションである。祖先インカーネーションのすべての親も祖先インカーネーションである。

### ターゲット・インスタンス (target instance)

Recovery Manager 環境では、**ターゲット・データベース**に関連付けられているインスタンスのことである。

### ターゲット・データベース (target database)

Recovery Manager 環境では、TARGET として接続されたデータベースのことである。ターゲット・データベースは、Recovery Manager の操作を実行するデータベースである。

### ターゲット・ホスト (target host)

**ターゲット・データベース**が存在するコンピュータ。

### タグ (tag)

Recovery Manager のバックアップの識別子。バックアップ・セットを生成した場合、タグはそのバックアップ・セットではなく、各バックアップ・ピースに割り当てられる。バックアップのタグを指定しなかった場合、Recovery Manager はタグを自動的に割り当てる。

### 多重化 (multiplexing)

この用語の意味は、多重化されるファイルによって次のように異なる。

- **オンライン REDO ログ**

オンライン REDO ログの 2 つ以上の同一コピーを自動的にメンテナンスすること。

- **制御ファイル**

データベースの制御ファイルの 2 つ以上の同一コピーを自動的にメンテナンスすること。

- **バックアップ・セット (backup set)**

データベース・ファイルをディスクから同時に読み取り、そのブロックを同一のバックアップ・ピースに書き込む Recovery Manager の機能。

- **アーカイブ REDO ログ**

Oracle アーカイバ・プロセスによって REDO ログの複数のコピーがアーカイブされること。

「**ミラー化**」を参照。

### 多重化レベル (level of multiplexing)

同時に読み取られ、Recovery Manager の同じ**バックアップ・ピース**に書き込まれる入力ファイルの数。

### 多重バックアップ・セット (duplexed backup set)

Recovery Manager では、多重**バックアップ・セット**とは、Recovery Manager によって生成される、バックアップ・セットの同一コピーのことである。元のバックアップ・セット内の各バックアップ・ピースがコピーされ、それぞれのコピーに一意のコピー番号 (0tcm8u2s\_1\_1、0tcm8u2s\_1\_2 など) が付けられる。

### 多重バックアップ・セット (multiplexed backup set)

複数の入力ファイルのブロックが含まれているバックアップ・セット。たとえば、10 個のデータファイルを 1 つのバックアップ・セットに多重化することができる。バックアップ・セットには、ファイルの一部ではなくファイル全体が含まれている。

### チェックサム (checksum)

データまたは REDO ブロックに格納されているすべてのバイトから、データベースによって計算される数値。DB\_BLOCK\_CHECKSUM 初期化パラメータが有効になっている場合は、データベースによって、すべてのデータファイルまたはオンライン REDO ログ・ブロックのチェックサムが計算され、ディスクへの書込み時にブロック・ヘッダーに格納される。データベースでは、このチェックサムの値を使用して一貫性がチェックされる。

### チェックポイント (checkpoint)

データベースの REDO スレッドに SCN を定義するデータ構造。チェックポイントは、制御ファイルおよび各データファイル・ヘッダーに記録され、リカバリに不可欠な要素である。

### チャネル (channel)

Recovery Manager のチャネルは、バックアップ・デバイスに対する 1 つの双方向ストリームを表す。チャネルは、DISK チャネル (ディスク I/O の実行に使用される) または SBT チャネル (サード・パーティの **メディア・マネージャ** を介した I/O の実行に使用される) のいずれかになる。割り当てられたチャネルごとに、新しい Oracle サーバー・セッションが開始される。このサーバー・セッションで、バックアップ、リストアおよびリカバリの操作が実行される。

「**ターゲット・データベース**」を参照。

### チャネルの平行化 (channel parallelism)

Recovery Manager の操作に対して複数のチャネルを割り当てること。

### 長期バックアップ (long-term backup)

バックアップ保存方針からは除外するが、リカバリ・カタログには記録する必要があるバックアップ。通常、長期バックアップとは、将来、レポートの生成に使用する可能性があるデータベースのスナップショットのことである。

### 直系祖先パス (direct ancestral path)

OPEN RESETLOGS 操作が複数回行われた場合の、現行のデータベース・インカネーションの **親インカネーション** および **現行のインカネーション** のそれぞれの **祖先インカネーション** を含むインカネーション・パス。

### 通常のリストア・ポイント (normal restore point)

SCN または時刻のラベル。SCN または時刻がサポートされているコマンドには、多くの場合、**リストア・ポイント** を指定できる。通常のリストア・ポイントは循環リストに含まれており、制御ファイル内で上書きできる。ただし、リストア・ポイントが **アーカイブ・バックアップ** に関連する場合は、**リカバリ・カタログ** に保存される。

### データ修復 (data repair)

**メディア・リカバリ** または **Oracle フラッシュバック技術** を使用して、消失または破損したデータをリカバリすること。

### データ整合性チェック (data integrity check)

状態モニターに登録される診断プロシージャであるチェッカの起動。

### データファイル・チェックポイント (datafile checkpoint)

特定のデータファイル用のデータベースの REDO スレッドに SCN を定義するデータ構造。すべてのデータファイルに **チェックポイント** SCN がある。この SCN は、V\$DATAFILE.CHECKPOINT\_CHANGE# で参照できる。この SCN より小さい SCN を持つすべての変更がそのデータファイル内に存在することが保証される。

### データファイルのメディア・リカバリ (datafile media recovery)

リストアされたデータファイルをより現在に近い時点までロールフォワードするために、リストアされたデータファイルに REDO レコードを適用すること。**ブロック・メディア・リカバリ** を実行している場合を除いて、リカバリ中はデータファイルをオフラインにする必要がある。

### データベース識別子 (database identifier)

「[DBID](#)」を参照。

### データベース全体のバックアップ (whole database backup)

制御ファイルおよびデータベースに属するすべてのデータファイルの[バックアップ](#)を作成する。

### データベース・チェックポイント (database checkpoint)

SCN が最も低いスレッド・チェックポイント。データベース・チェックポイント SCN より前の SCN を持つすべての有効な REDO スレッドにおけるすべての変更が、ディスクに書き込まれていることが保証される。

「[チェックポイント](#)」、「[データファイル・チェックポイント](#)」を参照。

### データベースの Point-in-Time リカバリ (database point-in-time recovery: DBPITR)

指定した過去の目標時点、SCN またはログ順序番号までデータベース全体をリカバリすること。

「[不完全リカバリ](#)」、「[表領域の Point-in-Time リカバリ](#)」を参照。

### データベースの登録 (database registration)

「[登録](#)」を参照。

### データベース領域 (database area)

Oracle によって管理されるデータファイル、制御ファイルおよびオンライン REDO ログ・ファイル用の場所。データベース領域は、DB\_CREATE\_FILE\_DEST 初期化パラメータで指定される。

### データ・リカバリ・アドバイザー (Data Recovery Advisor)

永続的なデータ障害を自動的に診断し、ユーザーに修復オプションを提示し、ユーザーの要求に応じて修復を実行する Oracle Database ツール。

### ディスク・グループ (disk group)

[自動ストレージ管理](#)によってユニットとして管理されるディスクのコレクション。ディスク・グループの構成要素には、ディスク、ファイルおよび割当て単位が含まれている。

### ディスク・コントローラ (disk controller)

1 つ以上のディスク・ドライブを制御するハードウェア・コンポーネント。

### ディスク割当て制限 (disk quota)

ユーザーが指定する[フラッシュ・リカバリ領域](#)のサイズの制限。ディスク割当て制限に達すると、Oracle によって不要なファイルが自動的に削除される。

### 転送先プラットフォーム (destination platform)

Recovery Manager の CONVERT コマンドの使用時に、転送先データベースが実行されるプラットフォーム。転送先データベースとは、データの転送先となるデータベースのことである。

### 転送先ホスト (destination host)

[複製データベース](#)が存在するコンピュータ。

### 転送スクリプト (transport script)

CONVERT DATABASE コマンドによって生成されるスクリプト。このスクリプトには、[転送先プラットフォーム](#)上の新しいデータベースの作成に使用する SQL 文が含まれている。

### 同期 I/O (synchronous I/O)

Recovery Manager がデータの読取りまたは書き込みをするとき、サーバー・プロセスは一度に 1 つのタスクのみを実行できる。

### 登録 (registration)

Recovery Manager で、REGISTER DATABASE コマンドを実行し、ターゲット・データベースの存在を **リカバリ・カタログ** に記録する。 **ターゲット・データベース** は、その **DBID** によってカタログ内で一意に識別される。2つ以上のデータベースを同じカタログに登録することも、同じデータベースを複数のカタログに登録することもできる。

### トランスポートブル表領域 (transportable tablespace)

表領域のセットを、1つのデータベースから別のデータベースに、または1つのデータベースからそのデータベース自身に転送する機能。表領域をデータベースに転送することは、あらかじめロードされたデータを使用して表領域を作成することに似ている。

### トランスポートブル表領域セット (transportable tablespace set)

トランスポートブル表領域操作での表領域のセットのデータファイル、および表領域のセットのメタデータが含まれているエクスポート・ファイル。エクスポートを実行するには、データ・ポンプ・エクスポートを使用する。

### バイナリ圧縮 (binary compression)

Recovery Manager で、バックアップ・セット内のデータに圧縮アルゴリズムを適用する方法。

### パスワード・ファイル (password file)

ORAPWD コマンドによって作成され、SYSDBA 権限または SYSOPER 権限を使用してネットワークを介して接続する場合に必要なファイル。パスワード・ファイルの詳細は、『Oracle Database 管理者ガイド』を参照。

### 破損ブロック (corrupt block)

認識されている Oracle 形式ではないか、または内容に内部的な一貫性がない Oracle ブロック。通常、破損は、ハードウェアの障害またはオペレーティング・システムの問題によって発生する。Oracle では、破損ブロックは、論理的な破損 (Oracle 内部エラー) またはメディア破損 (不正なブロック形式) のいずれかに識別される。

メディア破損ブロックは、 **ブロック・メディア・リカバリ** を使用して修復するか、または破損ブロックが別のオブジェクトで再利用されるようにそのブロックが含まれているデータベース・オブジェクトを削除することによって修復できる。メディア破損の原因がハードウェア故障の場合、前述のどちらの解決策も、そのハードウェアの故障を直さなければ効果がない。

### バックアップ (backup)

(1) データベース、表領域、表、データファイル、制御ファイル、アーカイブ REDO ログなどのデータのバックアップ・ **コピー**。バックアップには、(データベース・ファイル・レベルでの) 物理的なものと (データベース・オブジェクト・レベルでの) 論理的なものがある。物理バックアップは、Recovery Manager を使用して1つ以上のデータファイル、制御ファイルまたはアーカイブ REDO ファイルをバックアップすることによって作成することができる。論理バックアップは、データ・ポンプ・エクスポートを使用して作成することができる。

(2) Recovery Manager のコンテキストでは、BACKUP コマンドの出力。バックアップの出力形式は、 **バックアップ・セット**、 **プロキシ・コピー** または **イメージ・コピー** のいずれかである。データベースによってアーカイブされたログは、バックアップではなくコピーとみなされる。

### バックアップおよびリカバリ (backup and recovery)

メディア障害またはユーザー・エラーによるデータの消失からデータベースを保護するために必要な概念、手順および計画のセット。

### バックアップ期間 (backup window)

バックアップ・アクティビティを完了する必要がある時間の長さ。

### バックアップ制御ファイル (backup control file)

制御ファイルのバックアップ。制御ファイルは、Recovery Manager の backup コマンドまたは SQL 文の ALTER DATABASE BACKUP CONTROLFILE TO 'ファイル名' を使用してバックアップすることができる。

## バックアップ・セット (backup set)

1つ以上のデータファイル、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 REDO ログ・ファイルのバックアップ。各バックアップ・セットは、1つ以上のバイナリ・ファイルで構成される。各バイナリ・ファイルは、**バックアップ・ピース**と呼ばれる。バックアップ・ピースは、Recovery Manager によってのみ作成またはリストアできる独自の形式で書き込まれる。

バックアップ・セットは、Recovery Manager の BACKUP コマンドで作成する。バックアップ・セットは、通常は1つのバックアップ・ピースのみで構成される。Recovery Manager は、ユーザーが ALLOCATE CHANNEL または CONFIGURE CHANNEL コマンドの MAXPIECESIZE オプションを使用してバックアップ・ピースのサイズを制限した場合にのみ、バックアップ・セットの内容を複数のバックアップ・ピースに分割する。

「**未使用ブロックの圧縮**」、「**多重化**」、「**Recovery Manager**」を参照。

## バックアップの暗号化 (backup encryption)

V\$RMAN\_ENCRYPTION\_ALGORITHMS に表示されているアルゴリズムの1つを使用して行われるバックアップ・セットの暗号化。Recovery Manager は、バックアップ・セットに書き込まれるデータを透過的に暗号化し、これらのバックアップ・セットが RESTORE 操作で必要になると復号化する。Recovery Manager には、透過モード、パスワード保護モードおよびデュアル・モードの3つの暗号化モードがある。

## バックアップの最適化 (backup optimization)

Recovery Manager で、すでにバックアップされているファイルのバックアップを自動的にスキップできるようにする構成。バックアップの最適化を有効または無効にするには、CONFIGURE コマンドを使用する。

## バックアップの保存方針 (backup retention policy)

メディア・リカバリを実行するためにバックアップおよびアーカイブ・ログを保存しておく必要がある期間を決定するユーザー定義の方針。保存方針は、バックアップ冗長性または**リカバリ期間**で定義できる。Recovery Manager は、現行の保存方針を満たすために必要なデータファイル・バックアップ、およびこれらのデータファイル・バックアップの完全なリカバリに必要なすべてのアーカイブ REDO ログを保持する。

## バックアップ・ピース (backup piece)

Recovery Manager の**バックアップ・セット**の格納に使用される物理ファイルの形式。各論理バックアップ・セットには、1つ以上の物理バックアップ・ピースが含まれている。

## バックアップベースの複製 (backup-based duplication)

**ターゲット・データベース**のバックアップをリストアおよびリカバリすることによって、**複製データベース**を作成すること。この方法は、**アクティブなデータベースの複製**の代替方法である。

## バックアップ・モード (backup mode)

**オンライン・バックアップ**を実行する前に ALTER TABLESPACE ... BEGIN BACKUP または ALTER DATABASE BEGIN BACKUP コマンドを発行すると開始されるデータベースのモード (**ホット・バックアップ・モード**とも呼ばれる)。ALTER TABLESPACE ... END BACKUP または ALTER DATABASE END BACKUP コマンドを発行する場合は、表領域のバックアップ・モードを終了する。

オンラインの表領域のデータファイルのユーザー管理バックアップを実行する場合は、**分裂ブロック**が作成されないように、表領域をバックアップ・モードにする必要がある。バックアップ・モードでは、データベースへの更新によって通常より多くの REDO が作成される。データベースでは、バッファ・キャッシュ内のブロックが使用済になるたびに、データに対する変更の記録に加えて、変更されたブロックのイメージを REDO ログ・ファイルに書き込む必要がある。Recovery Manager では、データベースをバックアップ・モードにする必要はない。

「**破損ブロック**」を参照。

### パラレル・リカバリ (parallel recovery)

複数のプロセスによって、REDO ログ・ファイルからの変更が同時に適用されるリカバリの形式。RECOVERY\_PARALLELISM 初期化パラメータによって、インスタンスおよびクラッシュ・リカバリのパラレル化のレベルを決定する。RECOVER コマンドの PARALLEL および NOPARALLEL オプションを使用して、メディア・リカバリのパラレル化を制御できる。

リカバリの最適な並列度は、Oracle Database によって自動的に選択される。ほとんどの場合、インスタンス・リカバリ、クラッシュ・リカバリまたはメディア・リカバリのパラレル化のレベルを手動で設定することは推奨されないか、または不要である。

### 非一貫性バックアップ (inconsistent backup)

バックアップ内のファイルの一部に、ファイルのチェックポイント後に行われた変更が含まれているバックアップ。このタイプのバックアップは、一貫性を持たせるためにリカバリが必要である。通常、非一貫性バックアップは、オンライン・データベース・バックアップによって作成される。次のいずれかの理由でデータベースがクローズされている際にデータファイルをバックアップした場合も、非一貫性バックアップが作成されることがある。

- Oracle インスタンスの1つ (RAC 構造ではすべてのインスタンス) がクラッシュした直後
- SHUTDOWN ABORT を使用してデータベースを停止した後

非一貫性バックアップは、データベースが ARCHIVELOG モードで、バックアップ以降に作成されたすべてのアーカイブ REDO ログを使用できる場合にのみ有効である。

「[一貫性バックアップ](#)」、「[オンライン・バックアップ](#)」、「[システム変更番号](#)」、「[データベース全体のバックアップ](#)」を参照。

### 非循環再利用レコード (noncircular reuse records)

Oracle Database で必要とされる重要な情報が含まれている制御ファイルのレコード。これらのレコードが自動的に上書きされることはない。非循環再利用レコードに含まれている情報の例としては、データファイルの場所およびオンライン REDO ログがある。

「[循環再利用レコード](#)」を参照。

### 非同期 I/O (asynchronous I/O)

Recovery Manager がデータの読取りまたは書込みを行っているとき、サーバー・プロセスは、1つの I/O を開始し、その I/O が完了するまで待機している間に別の作業を実行できる。また、1つ目の I/O の完了の待機に入る前に、複数の I/O 操作を開始することもできる。

### 表領域の Point-in-Time リカバリ (tablespace point-in-time recovery: TSPITR)

SYSTEM 以外の1つ以上の表領域を現在以外の時刻にリカバリすること。TSPITR の実行には Recovery Manager を使用する。

### 表領域の格納場所 (tablespace destination)

[トランスポートابل表領域](#)の操作では、表領域の転送コマンドの完了時にデータファイルのコピーおよびその他の出力ファイルが (デフォルトで) 含まれるディスク上の場所。

### ファイル・セクション (file section)

データファイル内の連続するブロックの範囲。[マルチセクション・バックアップ](#)は、各セクションを個別の[バックアップ・ピース](#)にコピーすることによって、大規模なファイルをパラレルで処理する。

### ファジー・ファイル (fuzzy file)

データファイル・ヘッダーのチェックポイント SCN 以上の SCN を持つブロックが1つ以上含まれているデータファイル。データベース・ライターでは、ファイル・ブロックが書込まれるたびにファイル・ヘッダーの SCN が更新されないため、ファジー・ファイルは使用できない。たとえば、[バックアップ・モード](#)のデータファイルが Oracle によって更新されると、このような状況が発生する。リストアされたファジー・ファイルには、常に[メディア・リカバリ](#)を実行する必要がある。



### フィジカル・スタンバイ・データベース (physical standby database)

障害保護の目的で使用できる本番データベースのコピー。

### 不完全リカバリ (incomplete recovery)

データベースの Point-in-Time リカバリ の同義語。

「完全リカバリ」、「メディア・リカバリ」、「リカバリする」を参照。

### 複製データベース (duplicate database)

Recovery Manager の DUPLICATE コマンドを使用して、ターゲット・データベースのバックアップから作成されるデータベース。

「補助データベース」を参照。

### 物理スキーマ (physical schema)

任意の時点でデータベースに存在するデータファイル、制御ファイルおよび REDO ログ。表領域およびデータファイルのリストを取得するには、Recovery Manager の REPORT SCHEMA コマンドを発行する。

### 物理的な破損 (physical corruption)

破損ブロックがデータベースで認識されない破損。チェックサムが無効であるか、ブロックの内容がすべて 0 (ゼロ) であるか、またはブロックのヘッダーとフッターが一致していないため、ブロックがデータベースで認識されない場合がある。

### 物理バックアップ (physical backup)

物理ファイルのバックアップ。物理バックアップは、表のエクスポートなどの論理バックアップと対比される。

### 部分再同期化 (partial resynchronization)

再同期化のタイプ。この同期化では、Recovery Manager はアーカイブ・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データを、ターゲット制御ファイルから **リカバリ・カタログ** に転送する。

### 不要なバックアップ (obsolete backup)

現行のバックアップ **保存方針** を満たす必要がないバックアップ。たとえば、データファイルごとに 1 つのバックアップを保持する必要があると保存方針で定められている場合に、データファイル 1 のバックアップが 2 つ存在すると、データファイル 1 の 2 つ目のバックアップは不要であるとみなされる。

### フラッシュバック・データ・アーカイブ (flashback data archive)

表内のすべてのレコードに対するトランザクション関連の変更を、それらのレコードの存続期間中保存する履歴リポジトリ。フラッシュバック・データ・アーカイブでは、いくつかの **論理フラッシュバック機能** を使用して、非常に古い履歴データに透過的にアクセスできる。

### フラッシュバック・データベース・ウィンドウ (flashback database window)

FLASHBACK DATABASE コマンドをサポートするための十分なフラッシュバック・ログ・データが現在存在する SCN の範囲。フラッシュバック・データベース・ウィンドウは、使用可能な **フラッシュバック・ログ** にある最も古い SCN より前には拡張できない。

### フラッシュバック保存目標 (flashback retention target)

データベースのフラッシュバックで戻ることができる最も古い過去の時点を示す、ユーザーが指定する時刻または SCN。

### フラッシュバック・ログ (flashback logs)

フラッシュバック・データベース操作の実行に使用する、Oracle で生成されるログ。データベースでは、フラッシュ・リカバリ領域へのフラッシュバック・ログの書込みのみを実行できる。フラッシュバック・ログは、順次書き込まれ、アーカイブは行われない。また、ディスクへのバックアップはできない。

### フラッシュ・リカバリ領域 (flash recovery area)

制御ファイルのコピー、オンライン REDO ログのコピー、アーカイブ REDO ログ・ファイル、**フラッシュバック・ログ**、Recovery Manager バックアップなどのリカバリ関連ファイルの格納のために使用可能なオプションのディスクの場所。フラッシュ・リカバリ領域内のファイルは、Oracle Database および Recovery Manager によって自動的に管理される。フラッシュ・リカバリ領域の最大サイズは、**ディスク割当て制限**で指定できる。

### プロキシ・コピー (proxy copy)

Recovery Manager のバックアップ処理とリストア処理時に、**メディア・マネージャ**がメディア・ストレージ・デバイスとディスク間のデータ転送を管理するバックアップ。

### ブロック間の破損 (interblock corruption)

ブロックの破損の一種。破損がブロック自体内ではなく、ブロック間で発生する。このタイプの破損は、**論理的な破損**のみである。

### ブロック・チェンジ・トラッキング (block change tracking)

各データベース更新によって影響を受けたデータファイル・ブロックを追跡するためのデータベース・オプション。追跡情報は、ブロック・チェンジ・トラッキング・ファイルに格納される。ブロック・チェンジ・トラッキングが有効になっている場合、Recovery Manager は、チェンジ・トラッキング・ファイルに含まれている変更されたブロックのレコードを使用して、データファイル全体を読み取るのではなく、変更されたことがわかっているブロックのみを読み取ることによって、増分バックアップのパフォーマンスを向上させる。

### ブロック・チェンジ・トラッキング・ファイル (block change tracking file)

Recovery Manager で、**増分バックアップ**のパフォーマンスを向上させるために、変更されたブロックの記録に使用されるバイナリ・ファイル。このファイルの作成および名前の変更には、ALTER DATABASE 文を使用する。

### ブロック内の破損 (intrablock corruption)

ブロックの破損の一種。破損がブロック自体内で発生する。このタイプの破損には、**物理的な破損**と**論理的な破損**がある。

### ブロック・メディア・リカバリ (block media recovery)

Recovery Manager の RECOVER ...BLOCK コマンドを使用して、データファイル内の指定したブロックをリカバリすること。ブロック・メディア・リカバリでは、影響を受けたデータファイルはオンラインのまま、損傷または破損したブロックのみがリストアおよびリカバリされる。

### 分裂ブロック (fractured block)

任意の SCN でヘッダーとフッターが一貫性していないブロック。**ユーザー管理バックアップ**では、DBWR がファイルを更新するときに、オペレーティング・システム・ユーティリティを使用してデータファイルをバックアップできる。オペレーティング・システム・ユーティリティは、更新途中の状態のブロックを読み取ることができるため、バックアップ・メディアにコピーされるブロックの前半は更新されているが、後半には古いデータが含まれていることがある。この場合、ブロックは分裂している。

Recovery Manager を使用しないバックアップの場合は、ALTER TABLESPACE ... BEGIN BACKUP または ALTER DATABASE BEGIN BACKUP コマンドが、分裂ブロック問題の解決策となる。表領域が**バックアップ・モード**のときにデータ・ブロックを変更した場合は、ブロック・イメージ全体のコピーをログに記録してから変更が行われる。このため、メディア・リカバリによって、このブロックが分裂していることが検出された場合でも、データベースはブロックを再構築できる。

### ベース・リカバリ・カタログ (base recovery catalog)

**リカバリ・カタログ**・スキーマ全体。ベース・リカバリ・カタログは、リカバリ・カタログのサブセットである**仮想プライベート・カタログ**と区別される。



## 平均リカバリ時間 (Mean Time To Recover: MTTR)

リカバリを実行するために必要な時間。

## 変換スクリプト (convert script)

CONVERT DATABASE コマンドで生成されるスクリプト。転送先ホスト上でデータファイル形式を変換するために使用できる。

## 補助インスタンス (auxiliary instance)

補助データベース、あるいは**表領域の Point-in-Time リカバリ**または**トランスポータブル表領域**の操作で使用される一時インスタンスに関連付けられている Oracle インスタンス。

## 保証付きリストア・ポイント (guaranteed restore point)

**Oracle Flashback Database** の操作を行うために**フラッシュバック・ログ**がデータベースで保持されていることが保証された**リストア・ポイント**。**通常のリストア・ポイント**とは異なり、保証付きリストア・ポイントは制御ファイルからエージ・アウトされないため、明示的に削除する必要がある。保証付きリストア・ポイントでは**フラッシュ・リカバリ領域**内の領域が使用されるが、この領域は定義する必要がある。

## 補助セット (auxiliary set)

TSPITR において、リカバリ・セットにはないファイルのセット。ただし、TSPITR 操作を正しく実行するには、これらのファイルを**補助データベース**でリストアする必要がある。トランスポータブル表領域の操作では、データファイル以外に、表領域の転送には必要であるが、それ自体は**リカバリ・セット**の一部ではないその他のファイルが補助セットに含まれる。

## 補助チャネル (auxiliary channel)

補助インスタンスに接続されている Recovery Manager **チャネル**。補助チャネルは、ALLOCATE CHANNEL または CONFIGURE CHANNEL コマンドの AUXILIARY キーワードで指定する。

## 補助データベース (auxiliary database)

- (1) Recovery Manager の DUPLICATE コマンドを使用して、ターゲット・データベースのバックアップから作成されたデータベース。
- (2) **表領域の Point-in-Time リカバリ**の実行中に新しい場所にリストアされ、その後新しいインスタンス名で起動される一時データベース。TSPITR 補助データベースは、**リカバリ・セット**と**補助セット**で構成される。

## 補助の格納場所 (auxiliary destination)

**トランスポータブル表領域**の操作で、補助インスタンスのパラメータ・ファイル、データファイル (転送する表領域のデータファイル以外)、制御ファイル、オンライン・ログなどの補助セット・ファイルを格納できるディスク上の場所。

## 保存方針 (retention policy)

「**バックアップの保存方針**」を参照。

## ホット・バックアップ (hot backup)

「**オンライン・バックアップ**」を参照。

## ホット・バックアップ・モード (hot backup mode)

「**バックアップ・モード**」を参照。

## マルチセクション・バックアップ (multisection backup)

**バックアップ・ピース**ごとに1つの**ファイル・セクション** (データファイル内の連続するブロックの範囲) が含まれている Recovery Manager のバックアップ・セット。マルチセクション・**バックアップ・セット**には複数のバックアップ・ピースが含まれているが、バックアップ・セットにはデータファイルの一部のみが含まれるのではないことに注意すること。

マルチセクション・バックアップは、BACKUP コマンドで SECTION SIZE パラメータを指定して作成する。Recovery Manager の **チャンネル** は、各ファイル・セクションをシリアルまたはパラレルのいずれかで個別に処理できる。このため、マルチセクション・バックアップでは、複数のチャンネルで単一のファイルをバックアップできる。

### 未使用ブロックの圧縮 (unused block compression)

Recovery Manager で、データ・ブロックをスキップすることによってデータファイル・バックアップ・セットのサイズを小さくする機能。Recovery Manager では、一度も使用されていないブロックが常にスキップされる。特定の条件 (『Oracle Database バックアップおよびリカバリ・リファレンス』の BACKUP AS BACKUPSET エントリに関する項を参照) では、以前に使用されていたが、現在は使用されていないブロックもスキップされる。

### ミラー化 (mirroring)

1 つ以上のディスクにデータの同一のコピーを保持すること。通常、ミラー化は複製ハード・ディスク上でオペレーティング・システム・レベルで実行されるため、一方のディスクが使えない場合でも、もう一方のディスクで要求の処理を中断せずに続行できる。ファイルのミラー化では、Oracle Database によって 1 回の書き込みが行われるが、オペレーティング・システムによって複数のディスクへの書き込みが行われる。ファイルの **多重化** では、Oracle Database によって同じデータが複数のファイルに書き込まれる。

### ミラー化の解除 (breaking a mirror)

ディスク・ミラー化プロセスの中断。ミラー・イメージは最新の状態ではなくなる。

### ミラーの復元 (resilvering a mirror)

ミラーを管理しているオペレーティング・システムまたはハードウェアに、破損したミラーを最新のミラーから更新する必要があることを通知し、両方のミラーを保持する。

### ミラーの分割によるバックアップ (split mirror backup)

以前にミラー化されたデータベース・ファイルのバックアップ。サード・パーティ・ツールには、ディスクまたは論理デバイスのセットを **ミラー化** (プライマリ・データの正確な複製を別の場所に保持) できるものがある。ミラーを分割すると、ファイルのコピーが分離されるため、それぞれを別々に使用できる。データベース機能の ALTER SYSTEM SUSPEND/RESUME を使用すると、データベースに対する I/O を一時停止した後、ミラーを分割し、分割されたミラーのバックアップを作成できる。

### メディア管理カタログ (media management catalog)

**メディア・マネージャ** によって管理されるレコードのカタログ。このカタログは、Recovery Manager の **リカバリ・カタログ** とは完全に無関係である。メディア管理カタログの例としては、**Oracle Secure Backup** カタログがある。

### メディア管理ライブラリ (media management library)

Recovery Manager で 3 次ストレージへのバックアップに使用できるソフトウェア・ライブラリ。**SBT** インタフェースは、公開されている API に準拠しており、メディア管理ベンダーによって提供される。**Oracle Secure Backup** には、Recovery Manager で使用するための SBT インタフェースが含まれている。

### メディア障害 (media failure)

Oracle で使用されるファイル (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など) のいずれかが含まれているディスクの損傷。Oracle では、メディア障害が検出されると、影響を受けているファイルがオフラインになる。

「**メディア・リカバリ**」を参照。

### メディア・マネージャ (media manager)

データベース・バックアップを 3 次ストレージに直接書き込むことができるように Recovery Manager に統合可能な、サード・パーティのネットワーク型バックアップ・システム。

## メディア・マネージャによる多重化 (media manager multiplexing)

Recovery Manager によるバックアップ中に、Recovery Manager ではなくメディア・マネージャによってブロックの組合せが管理される多重化。メディア・マネージャによる多重化のタイプの1つとして、メディア・マネージャが、複数の Recovery Manager チャンネルからの同時出力を単一のシーケンシャル・デバイスに書き込んだ場合に発生するものがある。また、別のタイプとして、バックアップによって同じテープ上にデータベース・ファイルとデータベース以外のファイルが混在した場合に発生するものもある。

## メディア・リカバリ (media recovery)

リストアされたバックアップ・データファイルまたは個々のデータ・ブロックに REDO または増分バックアップを適用すること。

メディア・リカバリを実行すると、データベース、表領域、データファイル、またはデータファイル内の一連のブロックをリカバリできる。メディア・リカバリは、**完全リカバリ** (REDO ログ内のすべての変更が適用される) または**不完全リカバリ** (指定した時点までの変更のみが適用される) のいずれかで実行できる。メディア・リカバリは、データベースが ARCHIVELOG モードの場合にのみ実行できる。

「**ブロック・メディア・リカバリ**」、「**リカバリする**」を参照。

## 問題 (problem)

**自動診断リポジトリ** に記録されているデータベースの重大エラー。重大エラーには、内部エラーおよび他のサーバー・エラーが含まれている。それぞれの問題には、その問題を記述する属性のセットである問題キーがある。問題キーには、ORA エラー番号、エラー・パラメータ値および他の情報が含まれている。

## ユーザー管理のバックアップとリカバリ (user-managed backup and recovery)

Recovery Manager を使用しない Oracle Database のバックアップおよびリカバリ計画。**オペレーティング・システムのバックアップおよびリカバリ**とも呼ばれる。オペレーティング・システムのユーティリティ (UNIX の cp コマンドなど) を使用してデータベース・ファイルをバックアップおよびリストアしてから、SQL\*Plus の RECOVER コマンドを使用してリカバリできる。

## ユーザー管理バックアップ (user-managed backup)

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方法などの Recovery Manager 以外の方法を使用して行われるバックアップ。たとえば、Linux で cp コマンドを実行するか、または Windows で COPY コマンドを実行することによって、ユーザー管理バックアップを作成できる。ユーザー管理のバックアップは、**オペレーティング・システムのバックアップ**とも呼ばれる。

## リカバリ (recovery)

データベース・ファイルまたはデータベースに関して使用する場合は、消失した変更を再構築するために、REDO データまたは増分バックアップをデータベース・ファイルに適用すること。リカバリには、**インスタンス・リカバリ**、**クラッシュ・リカバリ**および**メディア・リカバリ**の3つのタイプがある。Oracle では、最初の2つのリカバリはオンライン REDO レコードを使用して自動的に実行される。メディア・リカバリのみは、ユーザーがバックアップをリストアしてコマンドを発行する必要がある。

「**完全リカバリ**」、「**不完全リカバリ**」を参照。

## リカバリ・カタログ (recovery catalog)

1つ以上の Oracle Database に関する Recovery Manager のリポジトリ情報を格納するために、Recovery Manager で使用される Oracle の表およびビューのセット。Recovery Manager は、このメタデータを使用して、Oracle Database のバックアップ、リストアおよびリカバリを管理する。

Data Guard 環境で Recovery Manager を使用しない場合は、リカバリ・カタログの使用は任意である。Data Guard 環境で使用する場合は必須である。データベースに関する Recovery Manager のリポジトリ情報の 1 次ストレージは、常にデータベースの制御ファイル内となる。リカバリ・カタログは、制御ファイルからの Recovery Manager のリポジトリ・データで定期的に更新される。制御ファイルが消失した場合は、データベースのリストアおよびリカバリに必要となる、消失したメタデータの大部分またはすべてがリカバリ・カタログによって提供される。リカバリ・カタログには、アーカイブ・バックアップのレコード、およびターゲット・データベースで使用する Recovery Manager ストアド・スクリプトのレコードも記録できる。

「[リカバリ・カタログ・データベース](#)」を参照。

### リカバリ・カタログ・スキーマ (recovery catalog schema)

[リカバリ・カタログ](#)の表およびビューが含まれている、[リカバリ・カタログ・データベース](#)のスキーマ。

### リカバリ・カタログ・データベース (recovery catalog database)

リカバリ・カタログ・スキーマが含まれている Oracle Database。リカバリ・カタログはターゲット・データベースには格納できない。

### リカバリ期間 (recovery window)

Recovery Manager バックアップ [保存方針](#) のタイプの 1 つ。DBA が期間を指定し、Recovery Manager が、リカバリ期間内の任意の時点までの Point-in-Time リカバリに必要なバックアップおよびアーカイブ REDO ログが保存されることを保証する。この期間は常に、現在の時刻で終了し、ユーザーが指定した日数さかのぼる。

たとえば、保存方針が 7 日間のリカバリ期間で設定されている場合、現在の時刻が火曜日の午前 11 時であれば、Recovery Manager は前の週の火曜日の午前 11 時までの Point-in-Time リカバリに必要なバックアップを保持する。

### リカバリする (recover)

データベース・ファイルまたはデータベースのリカバリとは、通常、[メディア・リカバリ](#)、[クラッシュ・リカバリ](#)または[インスタンス・リカバリ](#)を実行することを指す。この用語は、消失したデータをなんらかの方法で再構築または再作成する操作の総称として使用される場合もある。

「[完全リカバリ](#)」、「[不完全リカバリ](#)」を参照。

### リカバリ・セット (recovery set)

[表領域の Point-in-Time リカバリ](#)の実行時、以前の時点までのリカバリが行われる 1 つ以上の表領域。TSPITR の実行後は、リカバリ・セット内のすべてのデータベース・オブジェクトが同じ時点までリカバリされている。

「[補助セット](#)」を参照。

### リストア (restore)

消失または損傷したファイルをバックアップと置き換えること。ファイルをリストアするには、UNIX の cp などのコマンドまたは Recovery Manager の RESTORE コマンドを使用する。

### リストアの最適化 (restore optimization)

Recovery Manager で、可能な場合、バックアップからデータファイルがリストアされないようにするデフォルトの動作。

### リストア・フェイルオーバー (restore failover)

破損したバックアップまたはアクセスできないバックアップが検出された場合にリストア操作で利用できるバックアップを Recovery Manager で自動的に検索すること。

### リストア・ポイント (restore point)

データベースの SCN に関連付けられているユーザー定義の名前。SCN はリストア・ポイントが作成された時刻に対応する。リストア・ポイントには、**保証付きリストア・ポイント**または**通常のリストア・ポイント**がある。

### 累積増分バックアップ (cumulative incremental backup)

レベル 0 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をバックアップする**増分バックアップ**。累積増分バックアップを使用してリカバリするときは、最新の累積増分バックアップを 1 つのみ適用する必要がある。

「**差分増分バックアップ**」、**「増分バックアップ」**を参照。

### レベル 0 の増分バックアップ (level 0 incremental backup)

バックアップ対象のデータファイル内のすべてのデータ・ブロックをバックアップする、Recovery Manager の**増分バックアップ**。レベル 0 の増分バックアップの内容は**全体バックアップ**と同じである。ただし、レベル 0 のバックアップは、全体バックアップと異なり、増分バックアップ計画の一部とみなされる。

### ロールバック (rolling back)

**リカバリするのロールフォワード**段階でデータベースに適用される、コミットされていない変更を、ロールバック・セグメントを使用して取り消すこと。

### ロールバック・セグメント (rollback segments)

データベースに対する変更の変更前のイメージが記録されたデータベース・セグメント。

### ロールフォワード (rolling forward)

データファイルおよび制御ファイルに行われた変更の**リカバリする**を行うために、それらのファイルに対して REDO レコードまたは増分バックアップを適用すること。

「**ロールバック**」を参照。

### ログ順序番号 (log sequence number)

REDO ログ・ファイル内の一連の REDO レコードを一意に識別する番号。Oracle では、1 つのオンライン REDO ログ・ファイルが一杯になり、別のオンライン REDO ログ・ファイルに切り替わると、その新しいファイルにログ順序番号が自動的に割り当てられる。

「**ログ・スイッチ**」、**「REDO ログ」**を参照。

### ログ・スイッチ (log switch)

LGWR によって、アクティブな REDO ログ・ファイルへの書込みが停止され、使用可能な次の REDO ログ・ファイルに切り替えられる時点。LGWR での切替えは、アクティブなログ・ファイルが REDO レコードで一杯になるか、ユーザーが手動で強制的に切り替えた場合に行われる。

「**REDO ログ**」を参照。

### 論理的な破損 (logical corruption)

ブロックのチェックサムが有効で、ヘッダーとフッターは一致しているなどの点は正常だが、内容に論理的な一貫性がない破損。

### 論理バックアップ (logical backups)

表などのデータベース・スキーマ・オブジェクトのバックアップ。論理バックアップは、Oracle Data Pump Export ユーティリティによって作成およびリストアされる。オブジェクトを論理バックアップからリストアするには、データ・ポンプ・インポート・ユーティリティを使用する。

### 論理フラッシュバック機能 (logical flashback features)

**Oracle Flashback Database** 以外の **Oracle フラッシュバック技術**機能のセット。この論理機能を使用すると、過去の時点の個々のデータベース・オブジェクトまたはトランザクションを表示したり、個々のデータベース・オブジェクトまたはトランザクションを過去の時点まで巻き戻すことができる。

## 記号

---

%U 置換変数, 8-5

## A

---

ABORT オプション

SHUTDOWN 文, 29-2

ABORT オプション、SHUTDOWN 文, 28-18

Advanced Security Option, 6-8, 7-5

ADVISE FAILURE コマンド, 14-5, 14-10

ALLOCATE CHANNEL コマンド, 6-2, 8-5

MAXPIECESIZE オプション, 6-5

ALLOCATE コマンド, 5-5

ALLOW ...CORRUPTION 句、RECOVER コマンド,  
28-24

ALTER DATABASE 文

CLEAR LOGFILE 句, 29-14

END BACKUP 句, 27-8

OPEN RESETLOGS 句, 12-28

RECOVER 句, 28-7, 28-11, 28-13

RESETLOGS オプション, 28-19

ALTER SYSTEM 文

KILL SESSION 句, 22-14

RESUME 句, 27-14

SUSPEND 句, 27-13

ALTER TABLESPACE 文

BEGIN BACKUP 句, 27-5, 27-7

END BACKUP オプション, 27-7

ARCHIVELOG モード

バックアップ, 2-5

AS SELECT 句

CREATE TABLE 文, 29-10

AUTORECOVERY オプション

SET 文, 28-5

AVAILABLE オプション

CHANGE コマンド, 11-15

## B

---

BACKUP CONTROLFILE 句

ALTER DATABASE 文, 27-2

BACKUP COPIES パラメータ

CONFIGURE コマンド, 6-5

Backup Solutions Program (BSP), 3-7

BACKUP コマンド, 2-4, 2-5, 3-5, 3-9, 5-21, 5-23,

6-2, 6-4, 6-6, 7-2, 7-3, 8-2, 8-14

ARCHIVELOG オプション, 8-13

AS BACKUPSET オプション, 7-4

AS COMPRESSION BACKUPSET オプション, 8-6

AS COPY オプション, 2-4, 7-9

BACKUPSET オプション, 6-8, 7-10, 7-11, 8-28,  
8-30

CHANNEL オプション, 5-7

COMPRESSED BACKUPSET オプション, 8-6

COPIES パラメータ, 7-10

COPY OF オプション, 7-10, 7-12, 8-28, 8-31

CURRENT CONTROLFILE オプション, 8-9, 8-10

DATABASE オプション, 8-8

DATAFILE オプション, 8-8

DB\_FILE\_NAME\_CONVERT パラメータ, 7-9

DELETE INPUT オプション, 8-14, 11-21

DELETE オプション, 8-12

DEVICE TYPE 句, 5-4, 5-24, 8-3, 8-10

DURATION パラメータ, 9-14

FILESERSET パラメータ, 7-7

FOR RECOVER OF COPY オプション, 8-18

FORMAT パラメータ, 2-6, 2-7, 5-10, 5-13, 7-5,  
7-11, 8-4

INCREMENTAL オプション, 2-6, 2-7, 2-8, 8-15,  
8-17, 8-18

KEEP オプション, 8-24, 8-26, 8-27

MAXSETSIZE パラメータ, 9-2

NOT BACKED UP 句, 8-14

PLUS ARCHIVELOG オプション, 8-12, 8-13

PROXY ONLY オプション, 7-8

PROXY オプション, 7-8

RECOVERY AREA オプション, 8-28

SECTION SIZE パラメータ, 7-4, 9-3

SPFILE オプション, 8-11

TABLESPACE オプション, 8-8

TAG パラメータ, 2-6, 2-7, 8-5

VALIDATE オプション, 2-8, 14-3, 14-9, 15-4

BEGIN BACKUP 句

ALTER TABLESPACE 文, 27-5

BSP 「Backup Solutions Program (BSP)」を参照

BZIP2 圧縮アルゴリズム, 6-7, 7-4

## C

---

CATALOG コマンド, 11-16

START WITH パラメータ, 12-9

CHANGE FAILURE コマンド, 14-15

CHANGE コマンド

AVAILABLE オプション, 11-15

DB\_UNIQUE\_NAME パラメータ, 12-25

RESET DB\_UNIQUE\_NAME オプション, 3-9  
UNCATALOG オプション, 11-19  
CLEAR LOGFILE 句  
ALTER DATABASE, 29-14  
COMPATIBLE 初期化パラメータ, 6-7  
CONFIGURE コマンド  
AUXNAME オプション, 6-11  
BACKUP OPTIMIZATION オプション, 5-26  
CHANNEL オプション, 5-5, 6-2  
CONTROLFILE AUTOBACKUP オプション, 7-12, 8-25  
DB\_UNIQUE\_NAME オプション, 5-36  
ENCRYPTION オプション, 6-9  
EXCLUDE オプション, 6-6  
FOR DB\_UNIQUE\_NAME オプション, 3-8  
MAXPIECESIZE オプション, 6-5  
MAXSETSIZE オプション, 6-4  
RETENTION POLICY 句, 7-18  
保存方針オプション, 5-22  
CONTROL\_FILE\_RECORD\_KEEP\_TIME 初期化パラメータ, 11-4, 11-5, 12-24  
CONTROL\_FILES 初期化パラメータ, 5-20, 19-6, 20-19, 29-2  
CONVERT コマンド  
表領域およびデータファイル, 25-1  
COPIES オプション  
BACKUP コマンド, 9-9  
CREATE CATALOG コマンド, 12-7, 12-12  
CREATE DATAFILE 句、ALTER DATABASE 文, 29-9  
CREATE SCRIPT コマンド, 12-16  
CREATE TABLESPACE 文, 29-5  
CREATE TABLE 文  
AS SELECT 句, 29-10  
CROSSCHECK コマンド, 11-12

## D

---

Data Guard 環境, 3-10  
DB\_UNIQUE\_NAME の変更, 12-25  
Recovery Manager の構成, 5-36  
Recovery Manager の使用方法, 3-8  
Recovery Manager バックアップ, 8-2  
Recovery Manager バックアップ、アクセシビリティ, 3-9  
Recovery Manager バックアップ、関連付け, 3-9  
Recovery Manager バックアップ、互換性, 3-9, 8-9  
アーカイブ・ログの削除方針, 5-28  
メンテナンス・コマンド, 11-3  
レポート, 10-3  
DB\_BLOCK\_CHECKSUM 初期化パラメータ, 15-2  
DB\_CREATE\_FILE\_DEST 初期化パラメータ, 5-20, 8-23, 17-11  
DB\_FILE\_NAME\_CONVERT 初期化パラメータ, 20-19  
Recovery Manager の DUPLICATE コマンドの使用, 23-19  
DB\_FLASHBACK\_RETENTION\_TARGET 初期化パラメータ, 5-17, 5-18, 5-19  
DB\_LOST\_WRITE\_PROTECT 初期化パラメータ, 6-13  
DB\_NAME 初期化パラメータ, 20-18  
DB\_RECOVERY\_FILE\_DEST\_SIZE 初期化パラメータ, 2-2, 5-17  
DB\_RECOVERY\_FILE\_DEST 初期化パラメータ, 2-2, 5-17, 5-20

DB\_UNIQUE\_NAME 初期化パラメータ, 3-8, 3-9, 5-36, 10-3  
DBA\_DATA\_FILES ビュー, 27-4, 27-5, 27-10  
DBID  
DBNEWID を使用した設定, 12-8  
確認, 17-6  
コピー・データベースの登録時の問題, 12-3  
DBMS\_PIPE パッケージ, 4-13  
DBNEWID ユーティリティ, 12-8, 23-2  
DBPITR 「データベースの Point-in-Time リカバリ」を参照  
DBVERIFY ユーティリティ, 27-18  
DELETE コマンド, 11-17, 11-20, 11-23  
EXPIRED オプション, 11-12, 11-24  
OBSOLETE オプション, 7-20, 11-24  
DROP DATABASE コマンド, 11-25  
DUPLICATE コマンド, 23-1  
DURATION パラメータ、BACKUP コマンド, 9-14

## E

---

EXECUTE SCRIPT コマンド, 12-18  
EXIT コマンド, 2-4  
EXPIRED オプション  
DELETE コマンド, 11-24

## F

---

FLASHBACK DATABASE コマンド, 5-29, 16-12  
FLASHBACK TABLE 文, 16-4, 16-5

## G

---

GRANT コマンド, 12-11

## I

---

IMPORT CATALOG コマンド, 12-32  
INCARNATION オプション  
LIST コマンド, 10-10, 12-29  
RESET DATABASE コマンド, 12-28, 12-29  
INCLUDE CURRENT CONTROLFILE オプション  
BACKUP コマンド, 8-10  
I/O エラー  
バックアップへの影響, 9-13

## K

---

KEEP オプション  
BACKUP コマンド, 11-16

## L

---

LIST コマンド, 2-9, 10-2, 10-4  
FAILURE オプション, 14-7  
INCARNATION オプション, 12-29  
LOCK\_NAME\_SPACE 初期化パラメータ, 20-18  
LOG\_ARCHIVE\_DEST<sub>n</sub> 初期化パラメータ, 5-21, 5-22, 5-27, 17-9, 28-5, 28-6, 28-10, 28-15, 28-17  
LOG\_ARCHIVE\_FORMAT 初期化パラメータ, 28-6  
LOG\_FILE\_NAME\_CONVERT 初期化パラメータ, 20-19



LOGSOURCE 変数  
SET 文, 28-7, 28-11, 28-13

## M

MAXPIECESIZE パラメータ  
SET コマンド, 5-13  
MAXSETSIZE パラメータ  
BACKUP コマンド, 6-4, 9-2  
CONFIGURE コマンド, 6-4  
MTTR, 14-2

## N

NLS\_DATE\_FORMAT 環境変数, 4-3  
NLS\_LANG 環境変数, 4-3  
NOARCHIVELOG モード  
デメリット, 28-18  
バックアップ, 8-11  
リカバリ, 28-18

## O

OPEN RESETLOGS 句  
ALTER DATABASE 文, 12-28, 13-6, 16-12, 16-17  
ORA-01578 エラー・メッセージ, 29-10  
Oracle Data Pump, 1-2, 16-14  
Oracle Encryption Wallet  
バックアップ, 6-8  
Oracle Flashback Database 「フラッシュバック・データベース」を参照  
Oracle Flashback Drop, 1-7  
Oracle Flashback Query, 1-6  
Oracle Flashback Table, 1-6  
Oracle Flashback Transaction, 1-6  
Oracle Flashback Transaction Query, 1-6  
Oracle Flashback Version Query, 1-6  
Oracle Managed Files  
フラッシュ・リカバリ, 5-16  
Oracle Real Application Clusters (Oracle RAC)  
Recovery Manager チャンネル, 6-3  
Oracle Secure Backup, 3-7, 5-9  
Oracle VSS ライター, 5-16  
Oracle ウォレット, 6-8

## P

Point-in-Time リカバリ, 28-14  
実行  
現行の制御ファイルの使用, 16-15  
表領域, 16-2  
PREVIEW オプション、RESTORE コマンド, 10-3  
PRINT SCRIPT コマンド, 12-20  
PROXY オプション  
BACKUP コマンド, 7-8

## Q

QUIT コマンド, 2-4

## R

RAW デバイス  
UNIX でのバックアップ, 27-14  
Windows のバックアップ, 27-16  
バックアップ先, 27-14  
RC\_ARCHIVED\_LOG ビュー, 10-19  
RC\_BACKUP\_FILES ビュー, 10-21  
RC\_BACKUP\_PIECE ビュー, 10-19  
RC\_BACKUP\_SET ビュー, 11-20  
Recovery Manager  
DBMS\_PIPE パッケージ, 4-13  
Recovery Manager コマンドの使用, 3-3  
アーカイブ REDO ログ  
バックアップ, 8-12  
イメージ・コピー・バックアップ, 7-8  
エラー, 22-2, 22-3  
解釈, 22-5  
エラー・コード  
メッセージ番号, 22-3  
概要, 2-2, 3-3  
監視, 22-8  
起動, 2-3  
クライアント, 2-2  
コマンドの終了, 22-13  
時間パラメータの設定, 4-3  
ジョブ、詳細の問合せ, 10-16  
ジョブ、進捗状況の監視, 21-11  
スナップショット制御ファイルの場所, 6-11  
整合性チェック, 15-2  
切断, 2-4  
増分バックアップ  
差分, 7-14  
累積, 7-15  
レベル 0, 7-14  
チャンネル, 3-4  
ネーミング規則, 3-5  
定義, 2-2  
データベース・キャラクタ・セット, 4-3  
データベース接続, 4-8  
カタログの指定なし, 4-8  
ターゲットに必要な SYSDBA, 4-8  
パスワードの保護, 4-12  
複製データベース, 4-11  
補助データベース, 4-11  
データベースに接続, 2-3  
テープ・バッファの割当て, 21-7  
テスト・ディスク API, 5-11  
同期 I/O と非同期 I/O, 21-6, 21-8  
認証, 2-3  
パイプ・インタフェース, 4-13  
データファイル・ブロックの破損  
I/O エラー, 9-13  
バックアップ, 8-2  
アーカイブ REDO ログ, 8-12  
イメージ・コピー, 7-8  
検証, 15-4, 15-6  
最適化, 5-23, 8-14  
制御ファイル, 8-9  
増分, 8-15, 9-8, 9-9  
多重化, 7-10  
データファイル, 8-8, 8-30, 8-31  
データベース全体, 8-8

- テスト, 15-4, 15-6
- バックアップ, 7-11, 8-28
- 表領域, 8-30, 8-31
- 不要のバッチ削除, 7-20
- バックアップのタイプ, 7-8
- パフォーマンス
  - 監視, 22-8
- ファイルの削除, 11-21
- 複製データベース、概要, 23-3
- プロキシ・コピー, 3-7
- 分裂ブロックの検出, 7-3
- 保存方針
  - 構成, 5-22
- メタデータ, 3-5, 10-1, 11-1, 12-1
- メディア管理
  - Backup Solutions Program (BSP), 3-7
  - クロスチェック, 11-12
  - ファイルのバックアップ, 3-6
  - メディア・マネージャ、リンク, 5-10
- メンテナンス・コマンド, 2-11
- リカバリ
  - メディア全体の障害後, 19-9
- リカバリ・カタログ, 12-2
  - アップグレード, 12-30
  - クロスチェック, 11-13
  - サイズの管理, 12-24
  - 再同期化, 12-22
  - 使用する操作, 3-6
  - ターゲット・データベースの登録, 12-3, 12-7, 12-9
  - 同期化, 12-22
  - バックアップ, 12-13
  - リカバリ, 12-15
- リカバリ・カタログのクロスチェック, 11-13
- リスト, 10-4
- リストア
  - アーカイブ REDO ログ, 17-9
  - データファイル, 13-3
- リターン・コード, 22-7
- リポジトリ, 3-5
- レポート, 10-12
  - データベース・スキーマ, 10-15
  - バックアップが必要なオブジェクト, 10-12
  - 不要なバックアップ, 10-14
- Recovery Manager 「Recovery Manager」を参照
- Recovery Manager エラー・スタックの解釈, 22-5
- Recovery Manager 構文のコメント, 4-4
- Recovery Manager コマンドの終了, 22-13
- Recovery Manager コマンドの取消し, 22-13
- Recovery Manager セッション, 2-13, 3-4
- Recovery Manager の監視, 22-8
- Recovery Manager の構成
  - 概要, 5-2
  - 共有サーバー, 6-12
  - 自動バックアップ, 5-7, 5-8, 7-12
  - スナップショット制御ファイルの場所, 6-11
  - デフォルト・デバイス, 5-3
  - デフォルトのバックアップ・タイプ, 5-4
  - 特定のチャネル, 6-2
  - バックアップから除外する表領域, 6-6
  - バックアップ・セットのサイズ, 6-4
  - バックアップの最適化, 5-23
  - バックアップ保存方針, 5-22

- Recovery Manager の終了, 2-4
- Recovery Manager のチューニング
  - V\$ ビュー, 22-8
- Recovery Manager のテスト
  - Media Management API, 22-11
  - バックアップ, 15-4, 15-6
- Recovery Manager バックアップ時のファイルのスキップ, 9-7
- Recovery Manager メタデータの管理, 10-1, 11-1
- Recovery Manager リポジトリ, 1-4, 2-2
- RECOVERY WINDOW パラメータ
  - CONFIGURE コマンド, 5-23
- RECOVERY\_CATALOG\_OWNER ロール, 12-11
- RECOVER 句
  - ALTER DATABASE 文, 28-7, 28-11, 28-13
- RECOVER コマンド, 13-5
  - COPY オプション, 8-18
  - PARALLEL および NOPARALLEL オプション, 28-8
  - TEST オプション, 15-7
  - UNTIL TIME オプション, 28-17
  - USING BACKUP CONTROLFILE 句, 29-6
  - リカバリ不能なオブジェクトおよびスタンバイ・データベース, 29-10
- REDO レコード
  - 適用時の問題, 28-20
- REDO ログ
  - 互換性のないフォーマット, 28-20
  - 名前の指定, 28-6
  - パラレル REDO, 28-20
- REGISTER コマンド, 12-9
- REPAIR FAILURE コマンド, 14-13, 14-15
- REPLACE SCRIPT コマンド, 12-17
- REPORT コマンド, 2-10, 10-2, 10-12
  - NEED BACKUP オプション, 10-12
  - OBSOLETE オプション, 7-20
- RESET DATABASE コマンド
  - INCARNATION オプション, 12-29
- RESETLOGS オプション
  - ALTER DATABASE, 28-19
- RESETLOGS 操作
  - 必要な場合, 13-6
- RESTORE コマンド, 13-3, 17-4
  - FORCE オプション, 13-5
  - PREVIEW オプション, 10-3, 17-6
  - VALIDATE HEADER オプション, 10-3, 17-6
- RESUME 句
  - ALTER SYSTEM 文, 27-14
- RESYNC CATALOG コマンド, 12-22, 12-25
  - FROM CONTROLFILECOPY オプション, 12-15
- REVOKE コマンド, 12-12

## S

- SBT, 3-4, 5-13
- sbtio.log
  - Recovery Manager, 22-2
- sbttest プログラム, 22-11
- SET コマンド
  - DBID オプション, 3-8
  - ENCRYPTION オプション, 6-9
  - MAXCORRUPT オプション, 15-3
- SET 文
  - AUTORECOVERY オプション, 28-5

LOGSOURCE 変数, 28-7, 28-11, 28-13  
SHOW コマンド, 2-4, 5-2  
SHUTDOWN 文  
  ABORT オプション, 28-18, 29-2  
SPOOL コマンド, 14-13  
SUSPEND 句  
  ALTER SYSTEM 文, 27-13  
SWITCH コマンド, 17-16  
SYSDBA 権限, 2-3

## U

UNAVAILABLE オプション  
  CHANGE, 11-15  
UNCATALOG オプション  
  CHANGE コマンド, 11-19  
  リポジット・レコードの削除, 11-19  
UNDO の最適化、バックアップ, 5-24, 7-5  
UNREGISTER DATABASE コマンド, 12-26  
UNTIL TIME オプション  
  RECOVER コマンド, 28-17

## V

V\$ARCHIVED\_LOG ビュー, 5-19, 10-19, 16-16  
  すべてのアーカイブ・ログの表示, 27-12  
V\$BACKUP\_ASYNC\_IO ビュー, 21-13  
V\$BACKUP\_DATAFILE ビュー, 8-16, 11-18  
V\$BACKUP\_FILES ビュー, 5-23, 11-13, 11-17  
V\$BACKUP\_PIECE ビュー, 10-19, 11-18  
V\$BACKUP\_REDOLOG ビュー, 11-18  
V\$BACKUP\_SET ビュー, 11-18, 11-20  
V\$BACKUP\_SPCFILE ビュー, 11-18  
V\$BACKUP\_SYNC\_IO ビュー, 21-13  
V\$BACKUP ビュー, 27-2  
V\$BLOCK\_CHANGE\_TRACKING ビュー, 8-23  
V\$CONTROLFILE ビュー, 8-10  
V\$DATABASE\_BLOCK\_CORRUPTION ビュー, xxiii,  
  1-5, 2-17, 15-4, 18-1, 18-2, 18-3, 18-4, 18-5  
V\$DATABASE\_INCARNATION ビュー, 12-28  
V\$DATABASE ビュー, 10-20, 16-5, 16-17  
V\$DATAFILE\_HEADER ビュー, 10-2, 17-5  
V\$DATAFILE ビュー, 17-6, 20-19, 27-2  
  バックアップ用のファイルの表示, 27-2  
V\$DIAG\_INFO ビュー, 2-17, 18-5  
V\$EVENT\_NAME ビュー, 22-8  
V\$FLASH\_RECOVERY\_AREA\_USAGE ビュー, 11-7  
V\$FLASHBACK\_DATABASE\_LOG ビュー, 5-20,  
  16-12, 16-17  
V\$FLASHBACK\_DATABASE\_STAT ビュー, 11-11  
V\$INSTANCE ビュー, 17-5  
V\$LOG\_HISTORY ビュー  
  すべてのアーカイブ・ログの表示, 28-10  
V\$LOGFILE ビュー, 20-19, 29-12, 29-13  
V\$PARAMETER ビュー, 16-6  
V\$PROCESS ビュー, 10-2, 22-8, 22-9  
V\$PROXY\_ARCHIVEDLOG ビュー, 7-8  
V\$PROXY\_DATAFILE ビュー, 7-8  
V\$RECOVER\_FILE ビュー, 17-5, 28-9  
V\$RECOVERY\_FILE\_DEST, 11-7  
V\$RECOVERY\_FILE\_DEST ビュー, 11-7  
V\$RECOVERY\_LOG ビュー  
  リカバリに必要なログの表示, 28-10

V\$RESTORE\_POINT ビュー, 10-11, 16-5  
V\$RMAN\_BACKUP\_JOB\_DETAILS ビュー, 10-16  
V\$RMAN\_BACKUP\_SUBJOB\_DETAILS ビュー, 10-16  
V\$RMAN\_ENCRYPTION\_ALGORITHMS ビュー, 6-7,  
  6-10, 21-7  
V\$RMAN\_OUTPUT ビュー, 10-20  
V\$RMAN\_STATUS ビュー, 22-2  
V\$SESSION\_LONGOPS ビュー, 21-11  
V\$SESSION\_WAIT ビュー, 22-8  
V\$SESSION ビュー, 6-12, 10-2, 22-8, 22-9  
V\$SGASTAT ビュー, 21-16  
V\$SYSSTAT ビュー, 11-11  
V\$TABLESPACE ビュー, 17-6, 27-2  
VALIDATE HEADER オプション、RESTORE コマンド,  
  10-3  
VALIDATE コマンド, 14-3, 14-9, 15-4, 17-4  
  SECTION SIZE パラメータ, 15-6  
Volume Shadow Copy Service (VSS), 5-16, 8-18

## Z

ZLIB 圧縮アルゴリズム, 6-7, 7-4

## あ

アーカイブ REDO ログ・ファイル  
  Recovery Manager を使用したリストア, 17-9  
  カタログに追加, 11-16  
  互換性のないフォーマット, 28-20  
  削除, 13-5, 28-12  
  消失, 28-14  
  デフォルトの場所の変更, 28-7  
  破損, 28-20  
  バックアップ, 8-13  
    Recovery Manager を使用, 8-12  
    他のバックアップ, 8-13  
  バックアップ後の削除, 8-12  
  フェイルオーバー, 8-12  
  メディア・リカバリ時の適用, 28-4, 28-6, 28-7  
  リカバリ時の位置, 28-4  
アーカイブ REDO ログの削除方針, 5-27, 5-28, 8-14  
アーカイブ・バックアップ, 1-3, 8-24, 8-25, 11-16  
  DUPLICATE でのリストア, 23-21  
アクティブなデータベースの複製, 4-11, 23-3  
圧縮バックアップ, 5-4, 8-6  
  アルゴリズム, 6-7  
アプリケーション・エラー, 1-3  
アラート・ログ, 11-6, 22-2  
暗号化バックアップ, 9-11, 13-4  
  復号化, 17-11

## い

一時停止 / 再開モード, 27-12  
一貫性バックアップ, 7-2  
  Recovery Manager を使用, 7-2  
  データベース全体, 27-3  
イメージ・コピー, 2-4, 7-2, 7-8  
  定義, 7-8  
  リストアのテスト, 17-9  
インカネーション、データベース, 10-10, 13-6, 16-12,  
  16-18

インスタンス障害  
バックアップ・モード, 27-8

## う

ウォレット, 6-8

## え

エラー・コード  
Recovery Manager, 22-2, 22-3  
メディア・マネージャ, 22-4  
エラー・スタック、Recovery Manager  
解釈, 22-5  
エラー・メッセージ、Recovery Manager  
解釈, 22-5

## お

オフサイト・バックアップ, 17-7  
オンライン REDO ログ, 29-13  
アーカイブ・グループ, 29-12, 29-13  
アクティブ・グループ, 29-12, 29-13  
アクティブ・ログの確認, 29-13  
グループの消失, 29-15  
現行グループ, 29-12, 29-13  
消去  
障害, 29-14  
消失, 29-13  
アクティブ・グループ, 29-15  
グループ, 29-12  
すべてのメンバー, 29-12  
ミラー・メンバー, 29-11  
リカバリ, 29-11  
場所の構成, 5-20  
破損したメンバーの置換, 29-11  
非アクティブのグループ, 29-12, 29-13  
非アクティブのログの消去  
アーカイブ, 29-13  
アーカイブされていない, 29-14  
複数のグループの消失, 29-16  
メディア・リカバリ時の適用, 28-4  
メンバーのステータス, 29-12, 29-13

## か

仮想プライベート・カタログ, 3-6  
削除, 12-13  
作成, 12-10  
可用性  
Recovery Manager バックアップ, 11-15  
仮の API, 5-11  
環境変数  
NLS\_DATE\_FORMAT, 4-3  
NLS\_LANG, 4-3  
完全リカバリ  
概要, 17-2  
手順, 28-9

## き

期限切れのバックアップ, 7-18, 11-13  
削除, 11-24

機能、新規, xxi  
キャラクタ・セット  
Recovery Manager で使用される設定, 4-3  
共有サーバー  
Recovery Manager とともに使用するための構成,  
6-12  
Recovery Manager の構成, 6-12

## く

クライアント、Recovery Manager, 2-2, 3-2, 3-6  
グループ、REDO ログ, 29-12, 29-13  
クロスチェック、Recovery Manager, 2-11, 11-3, 11-12  
定義, 11-12  
リカバリ・カタログとメディア・マネージャ, 11-13  
クロス・プラットフォーム・トランスポートダブル表領  
域, 25-1

## け

形式、Recovery Manager バックアップ用, 8-4  
検証、Recovery Manager, 14-9  
データベース・ファイル, 2-8, 15-4  
バックアップ, 2-8, 15-4, 15-6  
リストア操作, 17-9

## こ

コールド・フェイルオーバー・クラスタ  
定義, 27-8  
コマンド、Recovery Manager  
ADVISE FAILURE, 14-5, 14-10  
ALLOCATE CHANNEL, 5-5, 6-2, 6-5, 8-5  
BACKUP, 2-4, 2-5, 2-6, 2-7, 2-8, 3-5, 3-9, 5-4,  
5-7, 5-10, 5-13, 5-21, 5-23, 5-24, 6-2, 6-4,  
6-6, 6-8, 7-2, 7-3, 7-4, 7-7, 7-8, 7-9, 7-10,  
7-11, 7-12, 8-2, 8-3, 8-5, 8-6, 8-8, 8-9,  
8-10, 8-11, 8-12, 8-13, 8-14, 8-15, 8-17,  
8-18, 8-24, 8-26, 8-27, 8-28, 8-30, 8-31  
PROXY ONLY オプション, 7-8  
PROXY オプション, 7-8  
BACKUP CURRENT CONTROLFILE, 8-10  
CATALOG, 11-16  
CHANGE, 3-9, 11-12  
CHANGE FAILURE, 14-15  
CONFIGURE, 3-8, 5-5, 5-22, 5-36, 6-2, 6-5, 6-9,  
6-11  
CREATE CATALOG, 12-7, 12-12  
CREATE SCRIPT, 12-16  
CROSSCHECK, 11-12  
DELETE, 11-12, 11-17, 11-20  
DROP CATALOG, 12-34  
DROP DATABASE, 11-25  
DUPLICATE, 23-1  
EXECUTE SCRIPT, 12-16, 12-18  
EXIT, 2-4  
FLASHBACK DATABASE, 5-29, 12-28  
GRANT, 12-11  
IMPORT CATALOG, 12-32  
LIST, 2-9, 10-2, 10-4, 12-29, 14-7  
INCARNATION オプション, 10-10, 12-29  
MAXSETSIZE, 6-4  
PRINT SCRIPT, 12-20

RECOVER, 13-5  
Recovery Manager による解析方法, 3-3  
REPAIR FAILURE, 14-13, 14-15  
REPLACE SCRIPT, 12-17  
REPORT, 2-10, 10-12  
    NEED BACKUP オプション, 10-12  
RESET DATABASE  
    INCARNATION オプション, 12-29  
RESTORE, 17-4  
RESYNC CATALOG, 12-15, 12-22, 12-25  
    FROM CONTROLFILECOPY オプション, 12-15  
REVOKE, 12-12  
SET, 6-9  
SHOW, 2-4, 5-2  
SPOOL, 14-13  
SWITCH, 17-16  
UNREGISTER DATABASE, 12-26  
UPGRADE CATALOG, 12-30  
VALIDATE, 14-3, 14-9, 15-4  
終了, 22-13  
取消し, 22-13  
パイプ, 4-13  
コマンド、SQL\*Plus  
    RECOVER  
        UNTIL TIME オプション, 28-17  
        SET, 28-5, 28-7, 28-11, 28-13  
    コマンド・ファイル、Recovery Manager, 2-8  
    コマンド・インタフェース  
        Recovery Manager, 3-3  
    ごみ箱, 16-4, 16-8  
    オブジェクトのリストア, 16-8  
孤立したバックアップ, 13-8

## さ

サーバー・セッション、Recovery Manager, 3-4  
サーバー・パラメータ・ファイル  
    自動バックアップ, 7-12  
    自動バックアップの構成, 5-7, 5-8, 7-12  
    バックアップ, 8-11  
    リストア, 19-2  
再開可能バックアップ, 9-13  
削除された表、取得, 16-7  
削除方針、アーカイブ REDO ログ, 5-27  
    有効化, 5-28  
差分増分バックアップ, 2-6, 7-14

## し

時間パラメータ  
    Recovery Manager で使用される設定, 4-3  
時間ベースのリカバリ, 28-17  
試行リカバリ, 15-7, 28-25  
時刻書式  
    RECOVER DATABASE UNTIL TIME 文, 28-17  
システム時刻  
    変更  
        リカバリへの影響, 28-17  
実行可能性チェック、データ・リカバリ・アドバイザー,  
    14-5  
自動修復  
    データ・リカバリ・アドバイザー, 1-8

自動診断リポジトリ (ADR), 5-12, 7-13, 11-5, 14-3,  
    15-3, 15-4, 22-2  
自動ストレージ管理 (ASM)  
    データベースの複製, 23-15  
    バックアップ, 8-4  
自動チャネル, 3-4, 3-5  
    上書き, 6-2  
    構成, 6-2  
    ネーミング規則, 3-5  
自動チャネル割当て, 6-2  
自動バックアップ、制御ファイル, 7-12, 8-9, 8-25  
    構成, 5-7  
    書式, 5-8  
自動ワークロード・リポジトリ (AWR), 11-11  
シャドウ・コピー, 8-18  
修復、データ・リカバリ・アドバイザー, 14-3  
    手動および自動, 14-5  
    統合, 14-5  
修復オプション、データ・リカバリ・アドバイザー, 14-10  
循環再利用レコード, 11-4  
障害  
    定義, 1-2  
    メディア, 1-3  
    「リカバリ」も参照  
障害、データ・リカバリ・アドバイザー, 14-3, 14-4  
    統合, 14-5  
    優先順位, 14-4  
障害リカバリ, 1-3  
    定義, 1-3  
消失  
    非アクティブのログ・グループ, 29-13  
消失書込み、検出, 6-13  
状態モニター, 14-3  
ショート・ウェイト  
    定義, 21-14  
初期化パラメータ  
    CONTROL\_FILES, 19-6, 29-2  
    DB\_FILE\_NAME\_CONVERT, 20-19  
    DB\_NAME, 20-18  
    LARGE\_POOL\_SIZE, 21-16  
    LOCK\_NAME\_SPACE, 20-18  
    LOG\_ARCHIVE\_DEST\_n, 28-6  
    LOG\_ARCHIVE\_FORMAT, 28-6  
    LOG\_FILE\_NAME\_CONVERT, 20-19  
初期化パラメータ・ファイル, 13-5  
ジョブ、Recovery Manager  
    詳細の問合せ, 10-16  
    進捗状況の監視, 21-11  
新機能, xxi

## す

スクリプト、Recovery Manager, 2-8  
    置換変数, 8-26  
スタック・リカバリ, 28-20  
スタンバイ・データベース, 3-2  
    DUPLICATE での作成, 23-2  
ストアド・スクリプト, 3-6, 8-26, 12-3, 12-16, 12-32  
Recovery Manager の作成, 12-17  
    管理, 12-16  
    削除, 12-21  
    実行, 12-21  
    出力, 12-20

- 置換変数, 12-19
  - 動的, 12-19
  - 名前の表示, 12-20
- ストレージ・スナップショット, 5-31
- スナップショット制御ファイル, 6-11, 12-23
  - 場所の指定, 6-11

## せ

- 制御ファイル
  - 再作成, 29-6
  - 循環再利用レコード, 11-4
  - スナップショット, 12-23
    - 場所の指定, 6-11
  - すべてのコピーが消失した後の作成, 29-7
  - すべてのコピーが消失した後のユーザー管理リストア, 29-6
  - 多重, 5-15, 5-20, 11-6, 17-3, 27-2, 28-4, 29-2
    - 消失, 29-2
  - 多重化, 11-6
  - 場所の構成, 5-20
  - バックアップ, 27-2, 27-11
    - Recovery Manager を使用, 8-9
    - データベース・バックアップに含める, 8-10
    - バイナリ, 27-11
    - リカバリ, 19-5
  - ファイル名の検索, 27-2
  - 複製データベース, 23-17
  - リストア, 19-6, 29-2
- 制御ファイルの自動バックアップ, 11-6
  - 構成, 5-7, 5-8, 7-12
  - 書式, 5-8
  - データベースの構造変更時, 7-12
  - デフォルトの書式, 7-13
- 整合性チェック, 15-2
- セッション・アーキテクチャ, Recovery Manager, 3-4
- 切断
  - Recovery Manager, 2-4
- 全体バックアップ, 7-13
  - 増分バックアップ, 2-6

## そ

- 増分バックアップ, 2-6, 8-15
  - Recovery Manager での適用方法, 13-5
  - Recovery Manager を使用, 9-8, 9-9
  - 作成, 8-15
  - 差分, 7-14
  - ブロック・チェンジ・トラッキング, 8-21
  - マルチレベル, 7-14

## た

- ターゲット・データベース
  - 接続, 2-3
  - 定義, 2-2, 3-2
- 多重制御ファイル, 5-15, 5-20, 11-6, 17-3, 27-2, 28-4, 29-2
- 多重バックアップ・セット, 6-4, 6-5, 7-7, 7-10, 8-6, 9-7, 21-5

## ち

- 置換変数、FORMAT パラメータ, 5-13, 7-5, 7-9
- 置換変数、ストアド・スクリプト, 12-19
- チャンネル、Recovery Manager, 3-4
  - Oracle RAC 環境, 6-3
  - 構成, 5-5
  - 高度なオプションの構成, 6-2
  - 定義, 2-4, 3-4
  - ネーミング規則, 3-5
  - パラレル, 5-6
  - 一般, 5-5
  - 補助, 23-3
- 中断後のリカバリの再開, 28-8
- 直系祖先パス, 13-7, 16-12, 16-19

## て

- ディスク API, 5-11
- ディスク障害, 1-3
- ディスク使用量
  - 監視, 11-7
- データ修復
  - 概要, 13-2
  - 方法, 13-2
- データ整合性チェック, 1-8, 14-3, 14-9
- データ送信、Recovery Manager, 1-3
- データ・ディクショナリ・ビュー, 27-4, 27-5, 27-10
- データファイル
  - 再作成, 29-9
  - 消失, 28-3
  - ステータスの確認, 27-2
  - 名前の変更
    - リカバリ後, 29-6
  - バックアップ, 8-8, 8-30, 8-31, 27-4
  - 表示, 27-2
  - 複製データベース, 23-18
  - リカバリ
    - バックアップを利用しない, 29-9
  - リストア, 13-3
- データ・ブロック、破損, 1-3, 1-4, 2-12, 2-17, 13-2, 14-9, 14-15, 15-5, 18-2, 28-20, 28-21
- データベース
  - 一時停止, 27-12
  - スキーマに関するレポート, 10-15
  - バックアップ用の表示, 27-2
  - メディア・リカバリの例, 29-1
  - ユーザー管理のメディア・リカバリ手順, 28-1
  - リカバリ
    - 制御ファイルの破損後, 29-2
    - リカバリ・カタログからの登録解除, 12-26
    - リカバリ・カタログへの登録, 12-8, 12-9
- データベース接続
  - Recovery Manager
    - カタログの指定なし, 4-8
    - パスワードの保護, 4-12
    - 補助データベース, 4-11
  - Recovery Manager でのタイプ, 4-8
  - Recovery Manager に必要な SYSDBA, 4-8
- データベース全体のバックアップ
  - ARCHIVELOG モード, 27-3
  - NOARCHIVELOG モード, 27-3
  - Recovery Manager を使用, 8-8

- 準備, 27-4
- 非一貫性, 27-3
- データベースの Point-in-Time リカバリ, 16-15
  - 前提条件, 16-15
  - 定義, 16-2
  - フラッシュバック・データベース, 5-29, 16-2
  - ユーザー管理, 28-14
- データベースの一時停止, 27-12
- データベースの削除, 11-25
- データベースの登録解除, 12-26
- データベースの複製
  - アクティブなデータベースの複製, 23-3
  - バックアップベースの複製, 23-3
- データ保護
  - 定義, 1-2
- データ保存、定義, 1-3
- データ・リカバリ・アドバイザー, 2-12, 10-5, 13-2
  - 概要, 1-8
  - サポートされる構成, 14-6
  - 実行可能性チェック, 14-5
  - 自動修復, 1-8
  - 修復, 14-3, 14-5
  - 修復オプション, 14-10
  - 障害, 14-3, 14-4
  - 障害の修復, 14-13
  - 障害の統合, 14-5
  - 障害優先順位, 14-4
  - データ整合性チェック, 14-3, 14-9
  - ユーザー・インタフェース, 14-2
  - 用途, 14-2
- テープ・デバイス, 3-7
- テープへのシステム・バックアップ「SBT」を参照
- テスト・ディスク API, 5-11
- デバイス、デフォルトの構成, 5-3
- デュアル・モードのバックアップ暗号化, 6-9, 9-12

## と

- 透過モードのバックアップ暗号化, 9-12
- トランスポート表領域
  - Recovery Manager での作成, 24-2
    - 過去の時点, 24-9
    - 共有ブールのサイズ, 24-6
    - 初期化パラメータ, 20-16, 24-6
    - データ・ポンプ・エクスポート, 24-10
    - ファイルの場所, 24-11
    - 補助インスタンスのパラメータ・ファイル, 24-5, 24-7
    - 補助の格納場所, 24-4
    - クロス・プラットフォーム, 25-1
    - リカバリ, 29-10
- 取消しベースのメディア・リカバリ, 28-17
- トレース・ファイル、Recovery Manager, 22-2

## に

- 認証、Recovery Manager, 2-3

## は

- パイプ・インタフェース、Recovery Manager, 4-13
- パスワード
  - Recovery Manager への接続, 4-12

- パスワード・モードの暗号化, 9-12
- 破損ブロック, 13-2, 15-2, 28-20
  - Recovery Manager, 9-13
  - リカバリ, 18-2, 18-3
- バックアップ
  - DBVERIFY ユーティリティ, 27-18
  - NOARCHIVELOG モード, 8-11
  - Recovery Manager, 8-2
  - Recovery Manager チャンネルの関連付け, 22-10
  - Recovery Manager のテスト, 15-4, 15-6
  - メディア・マネージャの使用, 5-12
  - Recovery Manager のデフォルト・タイプ, 5-4
  - RESETLOGS の実行前のリカバリ, 16-20
  - アーカイブ, 1-3, 8-25
  - アーカイブ REDO ログ
    - Recovery Manager を使用, 8-12
  - 一貫性, 27-3
    - Recovery Manager を使用した作成, 7-2
  - イメージ・コピー, 7-8
  - オフサイト, 17-7
  - オフライン, 27-4
  - 可用性, 11-15
  - 管理, 11-2
  - 期限切れ、削除, 11-24
  - クローズ, 27-4
  - クロスチェック, 11-12
  - 検証, 15-4, 15-6, 27-18
  - 孤立, 13-8
  - サーバー・パラメータ・ファイル, 8-11
  - 再開可能, 9-13
  - 最適化, 5-23, 8-14
  - ストアド・スクリプト, 12-3, 12-16
  - 制御ファイル, 8-9, 27-11
    - バイナリ, 27-11
  - 増分, 7-14, 8-15, 9-8, 9-9
  - 増分更新, 8-18
  - 多重化, 6-5, 9-7
  - 長期, 1-3
  - データファイル
    - Recovery Manager を使用, 8-30, 8-31
    - データファイルのステータスの確認, 27-2
    - データベース全体, 8-8, 27-4
    - バックアップ, 7-11
    - バックアップが必要なオブジェクトのレポート, 10-12
    - バックアップからの表領域の除外, 6-6
    - バックアップ・セット, 8-28
    - 非一貫性, 27-3
      - Recovery Manager を使用した作成, 7-2
    - 必要なファイルの表示, 27-2
    - 表領域, 27-6
      - Recovery Manager を使用, 8-8, 8-30, 8-31
    - ファイルのスキップ, 9-7
    - 物理, 1-2
    - 不要, 7-20, 11-24
    - プレビュー, 17-6
    - 分割されたミラー, 7-10
      - Recovery Manager を使用, 9-9
    - 保存方針からの除外, 11-16
    - マルチセクション, 3-5, 7-4, 15-6
    - ユーザー管理, 27-1
    - ユーザー管理バックアップのリストア, 28-3
    - 読取り専用表領域, 27-10

- リカバリ・カタログ, 12-13
- 累積増分, 7-15
- レポートの生成, 10-2, 10-12
- 論理, 1-2
- バックアップおよびリカバリ
  - VSS 対応, xxiv
  - 概要, 1-1
  - 計画, 1-2
  - 新機能, xxi
  - ソリューション, 1-4
  - 定義, 1-2
  - ユーザー管理, 1-4
- バックアップ期間, 9-14
- バックアップ計画
  - フラッシュ・リカバリ領域, 5-14
- バックアップ・セット, 2-4, 7-2
  - Recovery Manager での生成方法, 7-6
  - 圧縮, 5-4, 6-7, 7-4, 8-6
  - 概要, 7-3
  - 数の指定, 7-6
  - クロスチェック, 11-13
  - サイズの制限, 7-6
  - 最大サイズ, 6-4, 9-2
  - 最大サイズの構成, 6-4
  - 最大サイズの指定, 7-6
  - 多重, 2-4, 6-4, 7-7, 8-6, 21-5
  - 多重化, 9-7
  - デフォルトの構成, 5-4
  - 名前の指定, 7-5
  - バックアップ, 7-11, 8-28
  - 未使用ブロックの圧縮, 7-4
  - リストアのテスト, 17-9
- バックアップ・セットのサイズ、設定, 7-6
- バックアップ・セット名の指定, 7-5
- バックアップ・タグ、Recovery Manager, 8-5
- バックアップの暗号化, 6-8, 7-5, 13-4
  - 概要, 9-11
  - デフォルト・アルゴリズム, 6-7
  - デュアル・モード, 6-9, 9-12
  - 透過, 6-8, 9-12
  - パスワード, 6-9, 9-12
  - バックアップの復号化, 17-11
- バックアップの最適化, 8-14
  - 構成, 5-23, 9-4
  - 冗長性, 5-25
  - 定義, 5-23, 8-14
  - 保存方針および, 5-25
  - 無効化, 5-24, 5-26
  - 有効化, 5-24, 5-26
- バックアップの削除, 2-11, 11-20, 11-21, 11-23
- バックアップの透過的暗号化, 6-8
- バックアップのバイナリ圧縮, 8-6
- バックアップのパスワード暗号化, 6-9
- バックアップのプレビュー, 17-6
- バックアップ・ピース, 7-4
  - 最大サイズ, 6-5
  - 定義, 2-4
  - テープ上の名前, 5-13
  - 名前, 7-5
- バックアップベースの複製, 23-3
- バックアップ方法、比較, 1-5
- バックアップ保存方針, 1-3, 3-8, 5-15
  - 構成, 5-22

- 冗長性, 7-18, 7-20
- 冗長性の構成, 5-22
- 定義, 7-18
- バックアップの最適化への影響, 5-25
- バックアップの除外, 8-25, 11-16
- 無効化, 5-23
- リカバリ期間, 5-23, 7-18
- バックアップ・モード, 7-10
  - ALTER DATABASE END BACKUP を使用した終了, 27-8
  - インスタンス障害, 27-8
  - オンラインのユーザー管理バックアップ, 7-3, 27-6
- パッケージ
  - DBMS\_PIPE, 4-13
- パフォーマンス・チューニング
  - ショート・ウェイト
  - 定義, 21-14
- パフォーマンス・チューニング、Recovery Manager
  - LARGE\_POOL\_SIZE 初期化パラメータ, 21-16
  - バックアップのパフォーマンス, 21-15
  - ロング・ウェイト, 21-14
- 汎用チャネル
  - 定義, 5-5

## ひ

- 非一貫性バックアップ, 7-2
  - Recovery Manager を使用, 2-5, 7-2
- ビュー、リカバリ・カタログ, 10-2, 10-18
- 表、削除のリカバリ, 29-16
- 表領域
  - Recovery Manager バックアップからの除外, 6-6
  - Recovery Manager を使用した転送, 24-1
  - Recovery Manager を使用したバックアップ, 8-8
  - オープン状態のデータベースでのオフラインのリカバリ, 28-12
  - バックアップ, 8-30, 8-31, 27-6
    - オフライン, 27-4
    - オンライン, 27-6
  - バックアップからの除外, 6-6
  - 読取り / 書込み
    - バックアップ, 27-5
  - 読取り専用
    - バックアップ, 27-10
  - リカバリのアクセス性
    - データベースがオープン状態の場合, 17-14
- 表領域の Point-in-Time リカバリ, 16-2
  - 計画, 20-5
  - 実行理由, 20-2
  - 制限事項, 20-5
  - データファイル名の構成, 6-10
  - 補助インスタンスの準備, 20-18
- 表領域の転送, 24-1

## ふ

- ファイル・セクション, 7-6, 7-7, 9-3, 15-6
- ファイル名、バックアップ用の表示, 27-2
- フェイルオーバー、ファイルのリストア時, 13-4
- 不完全メディア・リカバリ, 28-14
- 不完全リカバリ
  - Oracle Real Application Clusters 構成, 28-6
  - 概要, 13-5



- 時間ベース, 28-17
- 定義, 16-15
- バックアップ制御ファイルを使用, 28-6
- 複製データベース, 3-2
- OFFLINE NORMAL オプションによる表領域のスキップ, 23-20
- Recovery Manager による作成方法, 23-3
- アーカイブ・バックアップのリストア, 23-21
- アクティブなデータベースの複製, 4-11
- 概要, 23-2
- 作成
  - CONFIGURE AUXNAME の使用, 23-24
  - 同じファイル・システムを持つリモート・ホスト, 23-12
  - 異なるファイル・システムを持つリモート・ホスト, 23-13
  - ローカル・ホスト, 23-14
- 制御ファイルの生成, 23-17
- データファイル, 23-18
- ファイル名の生成, 23-6
- 複製の準備, 23-8
- 読取り専用表領域のスキップ, 23-20
- 物理的なブロック破損, 15-3
- 物理バックアップ, 1-2
- 不要なバックアップ, 7-18
- 削除, 2-11, 7-20, 11-24
- 定義, 7-18
- フラッシュバック技術, 16-2
  - 概要, 1-6
  - 論理機能, 16-3
- フラッシュバック削除
  - オブジェクトのリストア, 16-8
- フラッシュバック・データ・アーカイブ
  - 定義, 1-7
- フラッシュバック・データベース, 2-14, 13-2
  - 概要, 1-7
  - 監視, 11-11
  - 構成, 5-29
  - 前提条件, 16-11
  - パフォーマンスのチューニング, 5-35
  - フラッシュバック・データベース・ウィンドウの確認, 16-12
  - フラッシュバック・ログ, 1-7, 5-32
  - 有効化, 5-34
  - 要件, 5-33
  - 用途, 16-2
  - 領域管理, 11-8
    - ディスク領域要件の見積り, 5-20
- フラッシュバック・データベース・ウィンドウ, 5-29
- フラッシュバック・ドロップ, 16-4, 16-7
- フラッシュバック表, 16-3
  - 使用, 16-4, 16-5
- フラッシュバック保存目標, 5-29
- フラッシュバック・ログ, 1-7, 2-14, 5-29, 11-8, 16-3
- 保証付きリストア・ポイント, 5-31
- フラッシュ・リカバリ領域, 3-2, 3-7, 16-3
- Oracle Managed Files, 5-16
- Recovery Manager ファイル, 5-21
- 永続的なファイルおよび一時的なファイル, 5-15
- 構成, 5-14
- サイズの設定, 5-18
- 自動バックアップ, 5-8

- 使用状況の監視, 11-7
- スナップショット制御ファイル, 6-11
- 定義, 2-2
- ディスク使用量の監視, 11-7
- 場所の設定, 5-18
- 場所の変更, 11-10
- フラッシュバック・データベース・ウィンドウ, 5-29
- 保存方針の効果, 7-21
- 無効化, 11-10
- メンテナンス, 11-7
- 有効化, 5-17
- 領域管理, 5-16
- フラッシュ・リカバリ領域の使用状況の監視, 11-7
- プロキシ・コピー, 3-7, 7-8
- ブロック・チェンジ・トラッキング, 1-4, 7-16, 8-21
  - 使用されるディスク領域, 8-22
  - チェンジ・トラッキング・ファイルの移動, 8-24
  - 有効化と無効化, 8-22, 8-23
- ブロックの破損, 1-3
  - V\$DATABASE\_BLOCK\_CORRUPTION に格納, 15-4
- ブロック・メディア・リカバリ, 1-3, 15-4
- 文、SQL
  - ALTER DATABASE, 28-7, 28-11, 28-13
- 分割されたミラー
  - 一時停止 / 再開モード, 27-12
  - バックアップとして使用, 9-9
- 分裂ブロック, 7-3
  - 検出, 7-3

---

## ほ

- 補助インスタンスのパラメータ・ファイル
  - TRANSPORT TABLESPACE を使用, 24-5
- 保証付きリストア・ポイント, 1-7, 5-18, 5-27
  - 作成, 5-35
  - ストレージ・スナップショットとの比較, 5-31
  - ストレージ・スナップショットにかわる機能, 5-31
  - フラッシュバック・ログ, 5-31
  - フラッシュ・リカバリ領域の領域使用状況, 10-11
  - 要件, 5-33
- 補助チャンネル, 23-3
- 保存方針「バックアップ保存方針」を参照
- ホット・バックアップ・モード
  - オンラインのユーザー管理バックアップ, 27-6
  - 失敗したバックアップ, 27-8

---

## ま

- マルチセクション・バックアップ, 3-5, 7-4, 7-6, 7-7, 9-3, 15-6

---

## み

- 未使用ブロックの圧縮, 7-4
- ミラー化ファイル
  - オンライン REDO ログ
    - 消失, 29-11
  - バックアップ, 9-9
  - 分割, 27-12
    - Recovery Manager を使用, 9-9
    - 一時停止 / 再開モード, 27-12

## め

メタデータ、Recovery Manager, 3-5, 10-1, 11-1, 12-1  
メディア管理レイヤー (MML) API, 3-6, 6-5  
メディア障害, 1-2  
NOARCHIVELOG モード, 28-18  
アーカイブ REDO ログ・ファイルの消失, 28-14  
オンライン REDO ログ・グループの消失, 29-12  
オンライン REDO ログの消失, 29-11  
オンライン REDO ログ・メンバーの消失, 29-11  
完全リカバリ, 28-9  
制御ファイルの消失, 29-6  
定義, 1-3  
データファイルの消失, 28-3  
ユーザー管理の完全リカバリ, 28-8  
リカバリ, 28-9  
リカバリ手順  
例, 28-3  
メディア・マネージャ, 3-2, 3-4, 3-6, 3-7  
API のテスト, 22-11  
Backup Solutions Program, 3-7  
Recovery Manager とともに使用するための構成,  
5-10  
sbttest プログラム, 22-11  
インストール, 5-9  
エラー・コード, 22-4  
カタログに追加, 3-2  
クロスチェック, 11-12  
構成の前提条件, 5-9  
サード・パーティ, 5-9  
ソフトウェアへのリンク, 3-7, 5-10  
定義, 2-2  
テスト, 5-11  
トラブルシューティング, 5-12  
バックアップの多重化, 7-8  
バックアップのテスト, 5-12  
バックアップ・ピース名, 5-13  
ファイル制限, 5-13  
ファイルのバックアップ, 3-6  
ライブラリの場所, 5-10  
リンク  
テスト, 5-11  
メディア・マネージャの構成, 5-10  
インストール, 5-9  
前提条件, 5-9  
メディア・リカバリ, 7-17  
ADD DATAFILE 操作, 29-5  
NOARCHIVELOG モード, 28-18  
Recovery Manager を使用, 13-5  
アーカイブ REDO ログの適用, 28-4  
エラー, 28-20  
オープン状態のデータベースのオフライン表領域,  
28-12  
オンライン REDO ログ・ファイル, 29-11  
オンラインの破損していない表領域, 28-12  
完全, 28-8  
クローズ状態のデータベース, 28-9  
再開, 28-8  
時間ベース, 28-14  
試行, 28-25  
消失したファイル  
消失したアーカイブ REDO ログ・ファイル,  
28-14

消失したデータファイル, 28-3  
消失したミラー化制御ファイル, 29-2  
制御ファイルの破損後, 29-2  
中断, 28-8  
中断後の再開, 28-8  
データファイル  
バックアップを利用しない, 29-9  
トラブルシューティング, 28-19, 28-21  
トランスポータブル表領域, 29-10  
取消しベース, 28-14, 28-17  
破損  
許可, 28-23  
パラレル, 28-8  
不完全, 28-14  
問題, 28-19, 28-20, 28-21, 28-22  
ユーザー管理, 28-1  
ユーザー管理の完全リカバリ, 28-8  
リストア  
データベース全体のバックアップ, 28-18  
例, 29-1  
ロールフォワード・フェーズ, 28-4  
メディア・リカバリの中断, 28-8  
メンテナンス・コマンド、Recovery Manager, 2-11,  
3-5, 11-2  
Data Guard 環境, 11-3

## ゆ

ユーザー・エラー  
定義, 1-3  
リカバリ, 29-16  
ユーザー管理のリストア操作, 28-3  
ユーザー管理バックアップ, 27-1, 27-3  
オフライン表領域, 27-4  
検証, 27-18  
制御ファイル, 27-11  
定義, 7-10  
データファイルのステータスの確認, 27-2  
データベース全体, 27-4  
バックアップ・モード, 27-6, 27-8  
表領域, 27-6  
ファイルの事前の表示, 27-2  
ホット・バックアップ, 7-3, 27-8  
読取り専用表領域, 27-10  
ユーザー管理リカバリ, 28-14  
ADD DATAFILE 操作, 29-5  
完全, 28-8  
中断, 28-8  
不完全, 28-14  
例, 29-1

## よ

読取り専用表領域  
バックアップ, 27-10

## り

リカバリ  
ADD DATAFILE 操作, 29-5  
Recovery Manager を使用した障害リカバリ, 19-9  
アーカイブ・ログの自動適用, 28-5

- エラー, 28-20
- オンライン REDO ログ, 29-11
  - グループの消失, 29-12
  - メンバーの消失, 29-11
- 完全, 17-2, 28-8
  - オフライン表領域, 28-12
  - クローズ状態のデータベース, 28-9
- 削除された表, 29-16
- 時間ベース, 28-17
- 試行, 28-25
  - 概要, 28-25
  - 説明, 28-25
- 準備, 17-4
- 消失または破損したリカバリ・カタログ, 12-15
- スタック, 28-20
- 中断, 28-8
- データファイル, 28-3
- データベース
  - NOARCHIVELOG モード, 19-2
  - データベースの Point-in-Time, 16-15
  - データベース・ファイル
    - Recovery Manager での変更の適用方法, 13-5
    - 概要, 13-5
  - デフォルト以外の場所のログの使用, 28-7
  - デフォルトの場所のログの使用, 28-6
  - トラブルシューティング, 28-19
  - トランスポータブル表領域, 29-10
  - 取消しベース, 28-17
  - 破損
    - 意図的な許可, 28-23
  - バックアップ制御ファイルの使用, 19-5
    - リカバリ・カタログを使用しない, 19-7
  - パラレル, 28-8
  - 必要な障害, 1-2
  - 複数の REDO スレッド, 28-6
  - メディア, 28-1, 28-19, 29-1
  - 問題, 28-19
    - 解決, 28-22
    - 調査, 28-21
  - ユーザー・エラー, 29-16
  - ユーザー管理, 28-1, 28-19, 29-1
  - リカバリ・カタログを使用しない, 11-6
- リカバリ・カタログ, 3-6, 12-2
  - DBID の問題, 12-3
  - アップグレード, 12-30
  - 移動, 12-33
  - インポート, 12-32
  - 仮想プライベート・カタログ, 3-6, 12-10
  - カタログへのバックアップの追加, 11-16, 12-9
  - クロスチェック, 11-13
  - 更新
    - オペレーティング・システムの削除後, 11-20
  - サイズの管理, 12-24
  - 再同期化, 12-22
  - 削除, 12-34
  - 作成, 12-5
  - 使用する操作, 3-6
  - ストアド・スクリプト, 12-16
    - 作成, 12-17
  - 定義, 2-2, 3-2
  - データベースの登録, 12-3, 12-7, 12-8, 12-9
  - データベースの登録解除, 12-26
  - 同期化, 12-22
    - バックアップ, 12-13
    - バックアップの削除, 11-20
    - ビュー、問合せ, 10-18
    - ベース・リカバリ・カタログ, 12-10
    - メタデータの集中化, 12-3
    - 目的, 12-2
    - リカバリ, 12-15
    - リフレッシュ, 12-22
    - 領域要件, 12-6
    - レコードの削除, 11-23
    - ログ・スイッチ・レコード, 11-16
    - リカバリ・カタログのアップグレード, 12-30
    - リカバリ・カタログの再同期化, 3-8, 12-22, 12-25
    - リカバリ・カタログの削除, 12-34
    - リカバリ可能ポイント
      - リカバリ期間, 7-19
    - リカバリ期間, 5-22
      - 定義, 7-19
      - 保存方針の構成, 5-23
      - リカバリ可能ポイント, 7-19
    - リカバリ不能オブジェクト
      - リカバリ, 29-10
  - リストア
    - サーバー・パラメータ・ファイル, 19-2
    - 制御ファイル, 19-6
      - デフォルト以外の場所, 29-2
      - デフォルトの場所, 29-2
    - データベース
      - デフォルトの場所, 28-18
    - データベース・ファイル, 13-3, 13-4, 13-5
    - テスト, 15-7, 17-9
    - ユーザー管理バックアップ, 28-3
  - リストアの最適化, 13-5
  - リストアの妥当性検査, 17-9
  - リストア・ポイント, 1-7, 2-14
    - 削除, 11-9
    - 作成, 5-35
    - 表示, 10-10
    - フラッシュバック先, 16-19
    - 保証付き, 1-7, 5-31
      - ストレージ・スナップショットとの比較, 5-31
    - 要件, 5-33
  - リターン・コード
    - Recovery Manager, 22-7
  - リポジトリ、Recovery Manager, 3-5

## る

累積増分バックアップ, 2-6, 2-7, 7-14, 7-15

## れ

- 例、Recovery Manager
  - NOARCHIVELOG バックアップ, 8-11
  - RESETLOG の実行前のバックアップのリカバリ, 16-20, 19-2
    - メディア全体の障害後のリカバリ, 19-9
- レベル0の増分バックアップ, 2-6, 7-14, 7-16
- レベル1の増分バックアップ, 7-14, 7-15
- レポート、Recovery Manager, 2-9, 10-2, 10-12
  - データベース・スキーマ, 10-15
  - バックアップが必要なファイル, 10-12
  - バックアップ・ジョブ, 10-16

不要なバックアップ, 10-14  
リカバリ不能なバックアップ, 10-14

## ろ

---

ログ順序番号, 28-4  
ロング・ウェイト, 21-14  
論理的なブロック破損, 15-3  
論理バックアップ, 1-2