

## **Oracle® Database**

バックアップおよびリカバリ・リファレンス

11g リリース 1 (11.1)

部品番号: E05703-02

2007 年 11 月

このマニュアルは、コマンド構文、互換性マトリックス、およびリカバリ・カタログ・ビューを含む、**Recovery Manager** クライアントに関する完全なリファレンス情報を提供します。

Oracle Database バックアップおよびリカバリ・リファレンス, 11g リリース 1 (11.1)

部品番号 : E05703-02

原本名 : Oracle Database Backup and Recovery Reference, 11g Release 1 (11.1)

原本部品番号 : B28273-02

原本著者 : Lance Ashdown

原本協力者 : Antonio Romero, Tammy Bednar, Anand Beldalker, Timothy Chien, Mark Dilman, Senad Dizdar, Raymond Guzman, Wei Hu, Alex Hwang, Ashok Joshi, J. William Lee, Reem Munakash, Muthu Olagappan, Cris Pedregal-Martin, Samitha Samaranayake, Francisco Sanchez, Vivian Schupmann, Mike Stewart, Steven Wertheimer, Wanli Yang

Copyright © 2004, 2007, 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載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。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、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションへの用途を目的としておりません。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があります。

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 .....	v
対象読者 .....	vi
ドキュメントのアクセシビリティについて .....	vi
関連ドキュメント .....	vi
表記規則 .....	vii
サポートおよびサービス .....	vii
<b>1 Recovery Manager コマンドの概要</b>	
Recovery Manager コマンド・エントリの構造 .....	1-2
Recovery Manager の構文図 .....	1-2
Recovery Manager 構文内のキーワード .....	1-3
Recovery Manager 構文内のプレースホルダ .....	1-3
Recovery Manager 構文内の引用符 .....	1-3
Recovery Manager コマンドの形式 .....	1-4
Recovery Manager の予約語 .....	1-4
Recovery Manager コマンドの概要 .....	1-6
Recovery Manager 副次句の概要 .....	1-9
<b>2 Recovery Manager コマンド</b>	
@ .....	2-2
@@ .....	2-4
ADVISE FAILURE .....	2-6
ALLOCATE CHANNEL .....	2-10
ALLOCATE CHANNEL FOR MAINTENANCE .....	2-13
ALTER DATABASE .....	2-17
BACKUP .....	2-19
CATALOG .....	2-55
CHANGE .....	2-60
CONFIGURE .....	2-68
CONNECT .....	2-88
CONVERT .....	2-91
CREATE CATALOG .....	2-103
CREATE SCRIPT .....	2-106
CROSSCHECK .....	2-110
DELETE .....	2-114
DELETE SCRIPT .....	2-120
DROP CATALOG .....	2-122

DROP DATABASE .....	2-124
DUPLICATE .....	2-125
EXECUTE SCRIPT .....	2-143
EXIT .....	2-146
FLASHBACK DATABASE .....	2-147
GRANT .....	2-153
HOST .....	2-156
IMPORT CATALOG .....	2-158
LIST .....	2-161
PRINT SCRIPT .....	2-183
QUIT .....	2-185
RECOVER .....	2-186
REGISTER DATABASE .....	2-200
RELEASE CHANNEL .....	2-202
REPAIR FAILURE .....	2-204
REPLACE SCRIPT .....	2-208
REPORT .....	2-211
RESET DATABASE .....	2-219
RESTORE .....	2-221
RESYNC CATALOG .....	2-238
REVOKE .....	2-242
RMAN .....	2-244
RUN .....	2-250
SEND .....	2-252
SET .....	2-254
SHOW .....	2-263
SHUTDOWN .....	2-266
SPOOL .....	2-268
SQL .....	2-269
STARTUP .....	2-271
SWITCH .....	2-273
TRANSPORT TABLESPACE .....	2-277
UNREGISTER .....	2-282
UPGRADE CATALOG .....	2-285
VALIDATE .....	2-287

### 3 Recovery Manager 副次句

allocOperandList .....	3-2
archivelogRecordSpecifier .....	3-6
completedTimeSpec .....	3-10
connectStringSpec .....	3-12
datafileSpec .....	3-14
deviceSpecifier .....	3-15
fileNameConversionSpec .....	3-16
forDbUniqueNameOption .....	3-18
foreignlogRecordSpecifier .....	3-19
formatSpec .....	3-21
keepOption .....	3-24
listObjList .....	3-27
maintQualifier .....	3-30

maintSpec .....	3-32
obsOperandList .....	3-34
recordSpec .....	3-35
tempfileSpec .....	3-37
untilClause .....	3-38

## 4 リカバリ・カタログ・ビュー

Recovery Manager のリカバリ・カタログ・ビューの概要 .....	4-2
RC_ARCHIVED_LOG .....	4-5
RC_BACKUP_ARCHIVELOG_DETAILS .....	4-7
RC_BACKUP_ARCHIVELOG_SUMMARY .....	4-8
RC_BACKUP_CONTROLFILE .....	4-9
RC_BACKUP_CONTROLFILE_DETAILS .....	4-11
RC_BACKUP_CONTROLFILE_SUMMARY .....	4-12
RC_BACKUP_COPY_DETAILS .....	4-13
RC_BACKUP_COPY_SUMMARY .....	4-15
RC_BACKUP_CORRUPTION .....	4-16
RC_BACKUP_DATAFILE .....	4-17
RC_BACKUP_DATAFILE_DETAILS .....	4-19
RC_BACKUP_DATAFILE_SUMMARY .....	4-20
RC_BACKUP_FILES .....	4-21
RC_BACKUP_PIECE .....	4-24
RC_BACKUP_PIECE_DETAILS .....	4-26
RC_BACKUP_REDOLOG .....	4-28
RC_BACKUP_SET .....	4-30
RC_BACKUP_SET_DETAILS .....	4-32
RC_BACKUP_SET_SUMMARY .....	4-34
RC_BACKUP_SPFILE .....	4-35
RC_BACKUP_SPFILE_DETAILS .....	4-36
RC_BACKUP_SPFILE_SUMMARY .....	4-37
RC_CHECKPOINT .....	4-38
RC_CONTROLFILE_COPY .....	4-39
RC_COPY_CORRUPTION .....	4-41
RC_DATABASE .....	4-42
RC_DATABASE_BLOCK_CORRUPTION .....	4-43
RC_DATABASE_INCARNATION .....	4-44
RC_DATAFILE .....	4-45
RC_DATAFILE_COPY .....	4-47
RC_LOG_HISTORY .....	4-50
RC_OFFLINE_RANGE .....	4-51
RC_PROXY_ARCHIVEDLOG .....	4-52
RC_PROXY_ARCHIVELOG_DETAILS .....	4-54
RC_PROXY_ARCHIVELOG_SUMMARY .....	4-55
RC_PROXY_CONTROLFILE .....	4-56
RC_PROXY_COPY_DETAILS .....	4-58
RC_PROXY_COPY_SUMMARY .....	4-60
RC_PROXY_DATAFILE .....	4-61
RC_REDO_LOG .....	4-63
RC_REDO_THREAD .....	4-64
RC_RESTORE_POINT .....	4-65

RC_RESYNC .....	4-66
RC_RMAN_BACKUP_JOB_DETAILS .....	4-67
RC_RMAN_BACKUP_SUBJOB_DETAILS .....	4-69
RC_RMAN_BACKUP_TYPE .....	4-71
RC_RMAN_CONFIGURATION .....	4-72
RC_RMAN_OUTPUT .....	4-73
RC_RMAN_STATUS .....	4-74
RC_SITE .....	4-76
RC_STORED_SCRIPT .....	4-77
RC_STORED_SCRIPT_LINE .....	4-78
RC_TABLESPACE .....	4-79
RC_TEMPFILE .....	4-80
RC_UNUSABLE_BACKUPFILE_DETAILS .....	4-82

## A 使用不可になった Recovery Manager コマンド

## B Recovery Manager の互換性

Recovery Manager の互換性について .....	B-2
リカバリ・カタログ・スキーマのリリースの確認 .....	B-2
Recovery Manager の互換性マトリックス .....	B-3
リカバリ・カタログのエクスポートとインポートに関するバージョン間の互換性 .....	B-4
Recovery Manager の互換性: 使用例 .....	B-4

## 索引

---

---

# はじめに

内容は次のとおりです。

- [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、次のタスクを実行するデータベース管理者を対象としています。

- Recovery Manager を使用した Oracle データベースのバックアップ、リストアおよびリカバリ
- データベース・ファイルの Recovery Manager バックアップのメンテナンス

このマニュアルを使用するには、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』で説明されています。
- Recovery Manager の概念とタスク。『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』で説明されています。
- Oracle 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・ユーザーズ・ガイド』
- 『Oracle Database リファレンス』
- 『Oracle Database ユーティリティ』

このマニュアルに記載されている多数の例は、Oracle 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。これらのスキーマがどのように作成されているかと、その使用方法については、『Oracle Database サンプル・スキーマ』を参照してください。



## 表記規則

リファレンスの本文は、次の表記規則に従っています。

- 固定幅フォントの大文字  
Recovery Manager のキーワード、SQL のキーワード、表とビューにある列ヘッダー、および初期化パラメータを示します。
- 固定幅フォントの小文字  
Recovery Manager の例中の変数テキストを示します。
- *イタリック体*  
Recovery Manager または SQL のプレースホルダ、つまり、そのとおりに入力するのではなく、ユーザーが値を入力することを示します。

**関連項目：** Recovery Manager の表記規則については、第 1 章「[Recovery Manager コマンドの概要](#)」を参照してください。

## 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.co.jp/support/>

### 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://otn.oracle.co.jp/document/>

### 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://www.oracle.co.jp/education/>

### 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.co.jp>

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

---

---

**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。

---

---



---

# Recovery Manager コマンドの概要

この章の内容は、次のとおりです。

- [Recovery Manager コマンド・エントリの構造](#)
- [Recovery Manager の構文図](#)
- [Recovery Manager コマンドの形式](#)
- [Recovery Manager の予約語](#)
- [Recovery Manager コマンドの概要](#)
- [Recovery Manager 副次句の概要](#)

## Recovery Manager コマンド・エントリの構造

第2章「Recovery Manager コマンド」では、すべての Recovery Manager コマンドを、次の項目構成で説明しています。

**用途** コマンドの簡単な説明と、コマンドが使用される最も一般的な状況を示します。

**前提条件** コマンドを使用するための前提条件を示します。

**使用上の注意** コマンドを使用するための一般的な注意事項を示します（この項はオプションです）。コマンド・オプションに固有の注意事項は、そのオプションの説明に含まれています。

**構文** 文を構成するキーワードとパラメータを構文図で示します。

---

**注意：** キーワードとパラメータは、常に有効とは限りません。構文の制約については、必ず各文および各句の「セマンティクス」の項を参照してください。

---

**セマンティクス** 構文を構成するキーワード、パラメータおよび句の用途を説明し、該当する制約事項と使用上の注意を示します。この章で使用されているキーワードおよびパラメータに関する表記規則は、vii ページの「表記規則」で説明します。

**例** 文の各句と各パラメータの使用例を示します。

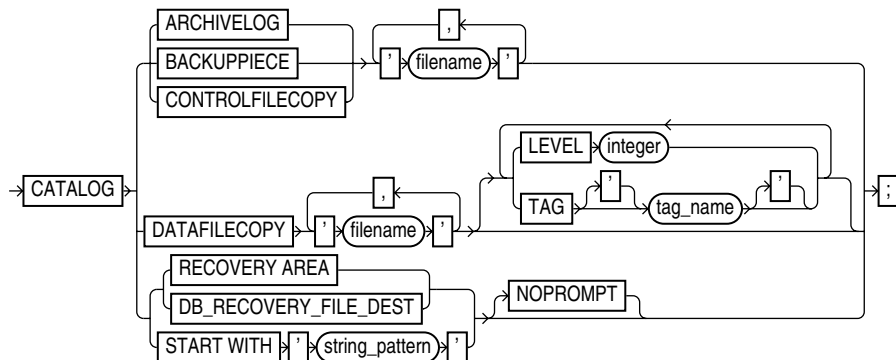
例の次の項では、文の使用法と使用時期を説明します（この項はオプションです）。

## Recovery Manager の構文図

このマニュアルでは、Recovery Manager のコマンドライン構成の構文記述に図形式またはバックス正規形（BNF）を使用しています。構文図および BNF 記法についての基本的な情報は、『Oracle Database SQL 言語リファレンス』を参照してください。この項では、Recovery Manager の表記規則についてのみ説明します。

次の CATALOG コマンドの例で示すように、Recovery Manager の構文図は、線と矢印で構文の構造を表します。

**catalog::=**



この項では、構文図の構成要素を説明し、Recovery Manager コマンドの書式の例を示します。構文図の構成要素は、次のとおりです。

- Recovery Manager 構文内のキーワード
- Recovery Manager 構文内のプレースホルダ
- Recovery Manager 構文内の引用符

## Recovery Manager 構文内のキーワード

キーワードは、Recovery Manager の構文で特殊な意味を持ちます。キーワードは、構文図では、図の例にあるワード CATALOG のように長方形で囲まれた大文字で表されます。本文とコード例で使用されている場合、Recovery Manager キーワードは CATALOG DATAFILECOPY のように固定幅フォントの大文字で表されます。Recovery Manager 文でのキーワードは、構文図に示すとおりを使用する必要があります。ただし、大文字と小文字は区別しません。

## Recovery Manager 構文内のプレースホルダ

構文図でのプレースホルダは、キーワード以外を示します。構文図では、図の例にあるワード *integer* のように楕円で囲まれて表されます。説明文の中では、Recovery Manager のプレースホルダは小文字のイタリック体で表されます。たとえば、'*filename*' となります。通常、プレースホルダは次の内容を示します。

- データベース・オブジェクト名 (*tablespace\_name*)
- Oracle データ型名 (*date\_string*)
- 副次句 (*datafileSpec*)

構文図にプレースホルダがあるときは、Recovery Manager 文中では、適切な型のオブジェクトまたは式に置き換えます。たとえば、DUPLICATE TARGET DATABASE TO '*database\_name*' コマンドを記述する場合は、構文図の *database\_name* プレースホルダを、dupdb などの作成する複製データベースの名前に置き換えて使用します。

Recovery Manager の引用符付き文字列の場合、システムに依存しない有効な環境変数は、Oracle ホームに使用する疑問符 (?) と SID に使用するアットマーク (@) のみです。ただし、ターゲット・システム上では、引用符付き文字列にオペレーティング・システム固有の環境変数を使用できます。環境変数は、Recovery Manager クライアントではなくデータベース・サーバーによって解析されます。

次の表に、構文図に表されるプレースホルダと、構文中でそれに代入する値の例を示します。

プレースホルダ	説明	例
<i>'filename'</i> 、 <i>'tablespace_name'</i> 、 <i>'channel_name'</i> 、 <i>'channel_parms'</i> など、引用符付き文字列。	一重引用符または二重引用符で囲まれた文字列。引用符で囲んだ文字列には、空白文字、句読点、Recovery Manager キーワードおよび SQL キーワードも含まれています。	"?/dbs/cf.f"  'dev1'
<i>channel_id</i> 、 <i>tag_name</i> 、 <i>date_string</i> など、引用符なしの文字列。	アルファベット文字で開始し、空白および句読点のない文字列。	ch1
<i>integer</i>	数値のみを含む文字列。	67843

## Recovery Manager 構文内の引用符

Recovery Manager 構文図で示すプレースホルダの値には、引用符で囲むことが必要なものと任意のものがあります。構文図では一重引用符を使用していますが、Recovery Manager 構文では、二重引用符も使用できます。たとえば、'*filename*' も "*filename*" も指定できます。SQL コマンドの場合は、SQL 文自体に引用符が含まれる場合があり、SQL 文で最も一般的な引用符が一重引用符であるため、二重引用符を使用することをお勧めします。SQL では、Recovery Manager の場合と異なり、一重引用符と二重引用符が別の意味を持ちます。

## Recovery Manager コマンドの形式

Recovery Manager の言語は自由区分形式です。複数のキーワードは、少なくとも 1 つの空白文字（空白、タブ、改行など）で区切る必要があります。Recovery Manager コマンドは、構文図で示すように、第 2 章「Recovery Manager コマンド」で説明するコマンドのいずれかに対応するキーワードで始まり、引数が続き、セミコロンで終了します。次の例は、Recovery Manager のバックアップ・コマンドを示しています。

```
BACKUP DATABASE;
```

コマンドは、複数行になることがあります。たとえば、前述のコマンドの各キーワードを次のように別々の行に書き直すことができます。

```
BACKUP
  DATABASE
;
```

コメントは行の任意の場所で番号記号 (#) 文字を使用して挿入することができます。# 文字に後続する行の部分は無視されます。次に例を示します。

```
# run this command once each day
BACKUP INCREMENTAL LEVEL 1
  FOR RECOVER OF COPY      # using incrementally updated backups
  WITH TAG "DAILY_BACKUP" # daily backup routine
DATABASE;
```

## Recovery Manager の予約語

Recovery Manager 言語にはいくつかの予約語があり、Recovery Manager コマンドで使用されていたか、現在も使用されているものがあります。表 1-1 に、現在予約されているすべての予約語を示します。

表 1-1 Recovery Manager の予約語

A-C	C-D	D-L	L-P	P-S	S-W
,	CHANGE	DURATION	LOGSEQ	PARALLELMEDIAR ESTORE	SECTION
#	CHANNEL	ECHO	LOGS	PARALLEL	SEND
(	CHECKSYNTAX	ENCRYPTION	LOG	PARAMETER	SEQUENCE
)	CHECK	EXCLUDE	LOGSCN	PARAMETER_ VALUE_CONVERT	SETLIMIT
\	CLEAR	EXECUTE	LOW	PARMS	SETSIZE
{	CLONENAME	EXIT	MAINTENANCE	PARTIAL	SET
}	CLONE	EXPIRED	MASK	PASSWORD	SHIPPED
<<<	CLONE_CF	EXPORT	MAXCORRUPT	PFILE	SHOW
>>>	CLOSED	FAILOVER	MAXDAYS	PIPE	SHUTDOWN
;	CMDFILE	FAILURE	MAXOPENFILES	PLATFORM	SINCE
&	COMMAND	FILESPERSET	MAXPIECESIZE	PLSQL	SIZE
_	COMMENT	FILES	MAXSEQ	PLUS	SKIP
'	COMPATIBLE	FILES	MAXSETSIZE	POINT	SLAXDEBUG
=	COMPLETED	FINAL	MAXSIZE	POLICY	SNAPSHOT
^	COMPRESSED	FLASHBACK	METHOD	POOL	SPFILE
@	COMPRESSION	FORCE	MINIMIZE	PREVIEW	SPOOL
.	CONFIGURE	FOREIGN	MISC	PRIMARY	SQL

表 1-1 Recovery Manager の予約語 (続き)

A-C	C-D	D-L	L-P	P-S	S-W
:	CONNECT	FOREVER	MOUNT	PRINT	STANDBY
ABORT	CONSISTENT	FORMAT	MSGLOG	PRIORITY	STARTUP
ACCESSIBLE	CONTROLFILECOPY	FOR	MSGNO	PRIVILEGES	START
ACTIVE	CONTROLFILE	FROM	M	PROXY	STEP
ADVISEID	CONVERT	FULL	NAMES	PUT	SUMMARY
ADVISE	COPIES	GET	NAME	QUIT	SWITCH
AES128	COPY	GLOBAL	NEED	RATE	TABLESPACES
AES192	CORRUPTION	HIGH	NEW-LINE	RCVCAT	TABLESPACE
AES256	CREATE	GRANT	NEWNAME	RCVMAN	TAG
AFFINITY	CRITICAL	GROUP	NEW	READONLY	TARGET
AFTER	CROSSCHECK	GUARANTEE	NOCATALOG	READRATE	TDES168
ALGORITHM	CUMULATIVE	G	NOCFAU	RECALL	TEMPFILE
ALLOCATE	CURRENT	HEADER	NOCHECKSUM	RECOVERABLE	TEST
ALLOW	DATABASE	HIGH	NODEVALS	RECOVERY	THREAD
ALL	DATAFILECOPY	HOST	NODUPLICATES	RECOVER	TIMEOUT
ALTER	DATAFILE	IDENTIFIED	NOEXCLUDE	REDUNDANCY	TIMES
AND	DATAPUMP	IDENTIFIER	NOFILENAMECHECK	REGISTER	TIME
APPEND	DAYS	ID	NOFILEUPDATE	RELEASE	TO
APPLIED	DBA	IMMEDIATE	NOKEEP	RELOAD	TRACE
ARCHIVELOG	DBID	IMPORT	NOLOGS	REMOVE	TRANSACTIONAL
AREA	DB_FILE_NAME_ CONVERT	INACCESSIBLE	NOMOUNT	RENORMALIZE	TRANSPORT
AS	DB_NAME	INCARNATION	NONE	REPAIRID	TYPE
AT	DB_RECOVERY_FILE_ DEST	INCLUDE	NOPARALLEL	REPAIR	UNAVAILABLE
ATALL	DB_UNIQUE_NAME	INCLUDING	NOPROMPT	REPLACE	UNCATALOG
AUTOBACKUP	DEBUG	INCREMENTAL	NOREDO	REPLICATE	UNDO
AUTOLOCATE	DECRYPTION	INPUT	NORMAL	REPORT	UNLIMITED
AUXILIARY	DEFAULT	INSTANCE	NOT	RESETLOGS	UNRECOVERABLE
AUXNAME	DEFINE	IO	NO	RESET	UNREGISTER
AVAILABLE	DELETE	JOB	NULL	RESTART	UNTIL
BACKED	DELETION	KBYTES	OBSOLETE	RESTORE	UPGRADE
BACKUPPIECE	DESTINATION	KEEP	OFFLINE	RESTRICTED	UP
BACKUPSET	DETAIL	KRB	OFF	RESYNC	USING
BACKUPS	DEVICE	K	OF	RETENTION	VALIDATE
BACKUP	DIRECTORY	LEVEL	ONLY	REUSE	VERBOSE
BEFORE	DISKRATIO	LIBPARM	ON	REVOKE	VIRTUAL
BETWEEN	DISK	LIBRARY	OPEN	RPCTEST	WINDOW
BLOCKRECOVER	DISPLAY	LIKE	OPTIMIZATION	RPC	WITH
BLOCKS	DORECOVER	LIMIT	OPTION	RUN	

表 1-1 Recovery Manager の予約語 (続き)

A-C	C-D	D-L	L-P	P-S	S-W
BLOCK	DROP	LIST	ORPHAN	SAVE	
BY	DUMP	LOAD	OR	SCHEMA	
CANCEL	DUPLEX	LOGFILE	PACKAGES	SCN	
CATALOG	DUPLICATE	LOGICAL	PARALLELISM	SCRIPT	

予約語のいずれかを Recovery Manager コマンドの引数として使用する必要がある場合 (たとえば、ファイル名、表領域名、タグ名など) は、予約語を一重引用符または二重引用符で囲んでください。そうしないと、Recovery Manager でコマンドが正しく解析されず、エラーが発生します。例 1-1 に、Recovery Manager コマンドで Recovery Manager 予約語を適切に使用している例と適切でない使用例を示します。

#### 例 1-1 予約語を Recovery Manager コマンドの引数に使用する例

```

ALLOCATE CHANNEL backup DEVICE TYPE DISK;           # incorrect
ALLOCATE CHANNEL "backup" DEVICE TYPE DISK;        # correct
BACKUP DATABASE TAG full;                           # incorrect
BACKUP DATABASE TAG 'full';                         # correct

```

通常、Recovery Manager コマンド言語の本来の意味と矛盾する方法で予約語を使用しないようにしてください。

## Recovery Manager コマンドの概要

表 1-2 に、Recovery Manager プロンプトまたは RUN コマンド、あるいはその両方で実行できる Recovery Manager コマンドの機能の概要を示します。使用できなくなっているコマンドとオプションもありますが、旧リリースの Recovery Manager のすべてのコマンドが、現行のリリースで動作します (付録 A 「使用不可になった Recovery Manager コマンド」を参照)。Recovery Manager クライアントのコマンドライン・オプションについては、2-244 ページの「[RMAN](#)」を参照してください。

表 1-2 Recovery Manager コマンド

コマンド	用途
@ (2-2 ページ)	コマンド・ファイルを実行します。
@@ (2-4 ページ)	現在実行中のコマンド・ファイルと同じディレクトリにあるコマンド・ファイルを実行します。@@ コマンドが @ コマンドと異なるのは、コマンド・ファイル内から実行される場合のみです。
<a href="#">ADVISE FAILURE</a> (2-6 ページ)	修復オプションを表示します。
<a href="#">ALLOCATE CHANNEL</a> (2-10 ページ)	チャンネルを確立します。これは、Recovery Manager とデータベース・インスタンス間の接続です。
<a href="#">ALLOCATE CHANNEL FOR MAINTENANCE</a> (2-13 ページ)	<a href="#">DELETE</a> コマンドなどのメンテナンス・コマンドの発行に備えてチャンネルを割り当てます。
<a href="#">ALTER DATABASE</a> (2-17 ページ)	データベースをマウントまたはオープンします。
<a href="#">BACKUP</a> (2-19 ページ)	データベース・ファイル、データベース・ファイルのコピー、アーカイブ・ログまたはバックアップ・セットをバックアップします。
<a href="#">CATALOG</a> (2-55 ページ)	ファイル・コピーおよびユーザー管理バックアップに関する情報をレポジトリに追加します。



表 1-2 Recovery Manager コマンド (続き)

コマンド	用途
<a href="#">CHANGE</a> (2-60 ページ)	バックアップ・ピース、イメージ・コピーまたはアーカイブ REDO ログをステータス UNAVAILABLE または AVAILABLE を持つものとしてマークします。バックアップまたはコピーのリポジトリ・レコードを削除します。バックアップまたはコピーの保存方針をオーバーライドします。DB_UNIQUE_NAME を使用してターゲット・データベースのリカバリ・カタログを更新します。
<a href="#">CONFIGURE</a> (2-68 ページ)	永続的な Recovery Manager 設定を構成します。この設定は、明示的に変更または使用不可にするまで、すべての Recovery Manager セッションに適用されます。
<a href="#">CONNECT</a> (2-88 ページ)	Recovery Manager とターゲット・データベース、補助データベースまたはリカバリ・カタログ・データベースとの接続を確立します。
<a href="#">CONVERT</a> (2-91 ページ)	プラットフォーム間で表領域とデータベースをトランスポートするために、データファイルの形式を変換します。
<a href="#">CREATE CATALOG</a> (2-103 ページ)	リカバリ・カタログのためのスキーマを作成します。
<a href="#">CREATE SCRIPT</a> (2-106 ページ)	ストアド・スクリプトを作成してリカバリ・カタログに格納します。
<a href="#">CROSSCHECK</a> (2-110 ページ)	アーカイブ・ログ、データファイルのコピーおよびバックアップ・ピースなど、Recovery Manager で管理されるファイルが、ディスクまたはテープに引き続き存在するかどうかを判断します。
<a href="#">DELETE</a> (2-114 ページ)	バックアップとコピーを削除し、その参照をリカバリ・カタログから削除し、その制御ファイル・レコードをステータス DELETED に更新します。
<a href="#">DELETE SCRIPT</a> (2-120 ページ)	ストアド・スクリプトをリカバリ・カタログから削除します。
<a href="#">DROP CATALOG</a> (2-122 ページ)	リカバリ・カタログからスキーマを削除します。
<a href="#">DROP DATABASE</a> (2-124 ページ)	ディスクからターゲット・データベースを削除して登録を解除します。
<a href="#">DUPLICATE</a> (2-125 ページ)	ターゲット・データベースのバックアップを使用して、テストに使用できる複製データベースを作成するか、スタンバイ・データベースを作成します。
<a href="#">EXECUTE SCRIPT</a> (2-143 ページ)	Recovery Manager ストアド・スクリプトを実行します。
<a href="#">EXIT</a> (2-146 ページ)	Recovery Manager 実行可能ファイルを終了します。
<a href="#">FLASHBACK DATABASE</a> (2-147 ページ)	データベースを過去のある時点または SCN での状態に戻します。
<a href="#">GRANT</a> (2-153 ページ)	リカバリ・カタログのユーザーに権限を付与します。
<a href="#">HOST</a> (2-156 ページ)	Recovery Manager からオペレーティング・システムのコマンドライン・サブシェルをコールするか、特定のオペレーティング・システム・コマンドを実行します。
<a href="#">IMPORT CATALOG</a> (2-158 ページ)	あるリカバリ・カタログから別のリカバリ・カタログに、メタデータをインポートします。
<a href="#">LIST</a> (2-161 ページ)	バックアップ・セットまたはコピーの詳細リストを生成します。
<a href="#">PRINT SCRIPT</a> (2-183 ページ)	ストアド・スクリプトを表示します。
<a href="#">QUIT</a> (2-185 ページ)	Recovery Manager 実行可能ファイルを終了します。
<a href="#">RECOVER</a> (2-186 ページ)	バックアップまたはデータファイルのコピーからリストアされたデータファイルまたはデータ・ブロックに REDO ログおよび増分バックアップを適用して、その内容を特定の時点の状態に更新します。
<a href="#">REGISTER DATABASE</a> (2-200 ページ)	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。
<a href="#">RELEASE CHANNEL</a> (2-202 ページ)	<a href="#">ALLOCATE CHANNEL</a> コマンドまたは <a href="#">ALLOCATE CHANNEL FOR MAINTENANCE</a> コマンドで割り当てたチャネルを解放します。
<a href="#">REPAIR FAILURE</a> (2-204 ページ)	自動診断リポジトリに記録された 1 つ以上の障害を修復します。

表 1-2 Recovery Manager コマンド (続き)

コマンド	用途
<a href="#">REPLACE SCRIPT</a> (2-208 ページ)	リカバリ・カタログに格納されている既存のスクリプトを置換します。既存のスクリプトがない場合は、REPLACE SCRIPT によりスクリプトが作成されます。
<a href="#">REPORT</a> (2-211 ページ)	リカバリ・カタログの内容の詳細分析を行います。
<a href="#">RESET DATABASE</a> (2-219 ページ)	SQL 文 ALTER DATABASE OPEN RESETLOGS が実行され、ターゲット・データベースの新規インカネーションが作成されたことを Recovery Manager に通知するか、またはターゲット・データベースを以前のインカネーションに再設定します。
<a href="#">RESTORE</a> (2-221 ページ)	バックアップ・セットまたはディスク・コピーから、デフォルト位置または新規の位置にファイルをリストアします。
<a href="#">RESYNC CATALOG</a> (2-238 ページ)	完全再同期化を実行します。これにより、スナップショット制御ファイルが作成され、そのファイルから新規または変更後の情報がリカバリ・カタログにコピーされます。
<a href="#">REVOKE</a> (2-242 ページ)	リカバリ・カタログ・ユーザーから権限を取り消します。
<a href="#">RMAN</a> (2-244 ページ)	オペレーティング・システムのコマンドラインから Recovery Manager を起動します。
<a href="#">RUN</a> (2-250 ページ)	1 つ以上の一連の Recovery Manager コマンドを実行します。これは、RUN の中カッコ内で実行される 1 つ以上の文です。
<a href="#">SEND</a> (2-252 ページ)	ベンダー固有の引用符付き文字列を 1 つ以上の特定チャンネルに送信します。
<a href="#">SET</a> (2-254 ページ)	RUN ブロックまたはセッションの実行時に Recovery Manager の動作に影響する様々な属性の値を設定します。
<a href="#">SHOW</a> (2-263 ページ)	現行の CONFIGURE 設定を表示します。
<a href="#">SHUTDOWN</a> (2-266 ページ)	ターゲット・データベースを停止します。このコマンドは、SQL*Plus の SHUTDOWN コマンドと同じです。
<a href="#">SPOOL</a> (2-268 ページ)	Recovery Manager 出力をログ・ファイルに書き込みます。
<a href="#">SQL</a> (2-269 ページ)	Recovery Manager 内から SQL 文を実行します。
<a href="#">STARTUP</a> (2-271 ページ)	ターゲット・データベースを起動します。このコマンドは、SQL*Plus の STARTUP コマンドと同じです。
<a href="#">SWITCH</a> (2-273 ページ)	データファイルのコピーが <b>現行のデータファイル</b> であることを指定します。つまり、制御ファイルが指すデータファイルです。このコマンドは、データファイルに適用される SQL 文 ALTER DATABASE RENAME FILE と同じです。
<a href="#">TRANSPORT TABLESPACE</a> (2-277 ページ)	1 つ以上の表領域のバックアップからトランSPORTABLE 表領域のセットを作成します。
<a href="#">UNREGISTER</a> (2-282 ページ)	リカバリ・カタログからデータベースを登録解除します。
<a href="#">UPGRADE CATALOG</a> (2-285 ページ)	リカバリ・カタログ・スキーマを、旧バージョンから Recovery Manager 実行可能ファイルに必要なバージョンにアップグレードします。
<a href="#">VALIDATE</a> (2-287 ページ)	バックアップ・セットを調べて、データの損傷がないかどうかをレポートします。Recovery Manager は、指定したバックアップ・セットのバックアップ・ピースをすべてスキャンし、チェックサムを参照して、内容が正しくリストアされるかどうかを検証します。

## Recovery Manager 副次句の概要

複数のコマンドで使用される副次句は、不要な重複を避けるために章を分けて説明しています。副次句を使用するコマンドの説明には、第3章「Recovery Manager 副次句」の副次句の項目への相互参照が含まれています。表 1-3 に、Recovery Manager の副次句をまとめています。

表 1-3 Recovery Manager の副次句

副次句	指定内容...
<a href="#">allocOperandList</a> (3-2 ページ)	PARMS、FORMAT などのチャンネル制御オプション。
<a href="#">archiveLogRecordSpecifier</a> (3-6 ページ)	アーカイブ REDO ログ・ファイルの範囲。
<a href="#">completedTimeSpec</a> (3-10 ページ)	バックアップまたはコピーが完了した時刻の範囲。
<a href="#">connectStringSpec</a> (3-12 ページ)	ターゲット・データベース、リカバリ・カタログ・データベースまたは補助データベースに接続するための、ユーザー名、パスワードおよびネット・サービス名。この接続は、ユーザーの認証とデータベースの識別のために必要です。
<a href="#">datafileSpec</a> (3-14 ページ)	データファイル (ファイル名または絶対ファイル番号で指定)。
<a href="#">deviceSpecifier</a> (3-15 ページ)	バックアップまたはコピーに使用するストレージ・デバイスのタイプ。
<a href="#">fileNameConversionSpec</a> (3-16 ページ)	BACKUP AS COPY、CONVERT および DUPLICATE の実行時に、ソース・ファイル名からターゲット・ファイル名への変換に使用するパターン。
<a href="#">forDbUniqueNameOption</a> (3-18 ページ)	Data Guard 環境のすべてのデータベース、または固有の DB_UNIQUE_NAME を持つデータベース。
<a href="#">foreignLogRecordSpecifier</a> (3-19 ページ)	外部アーカイブ REDO ログ・ファイルの範囲。
<a href="#">formatSpec</a> (3-21 ページ)	バックアップまたはコピーに使用するファイル名の形式。
<a href="#">keepOption</a> (3-24 ページ)	バックアップまたはコピーを現行の保存方針から除外するかどうか。
<a href="#">listObjList</a> (3-27 ページ)	LIST コマンドで表示する項目。
<a href="#">maintQualifier</a> (3-30 ページ)	DELETE や CHANGE などのメンテナンス・コマンドの追加オプション。
<a href="#">maintSpec</a> (3-32 ページ)	CHANGE、CROSSCHECK および DELETE などのメンテナンス・コマンドで操作されるファイル。
<a href="#">obsOperandList</a> (3-34 ページ)	指定条件に従って不要となるバックアップ。
<a href="#">recordSpec</a> (3-35 ページ)	メンテナンス・コマンドで操作されるオブジェクト。
<a href="#">tempfileSpec</a> (3-37 ページ)	一時ファイル (パスまたはファイル番号で指定)。
<a href="#">untilClause</a> (3-38 ページ)	上限 (時刻、SCN またはログ順序番号で指定)。通常、この句は不完全リカバリに必要な時点を指定するために使用します。



---

## Recovery Manager コマンド

この章では、Recovery Manager のコマンドをアルファベット順に説明します。Recovery Manager コマンドとコマンドライン・オプションの概要は、1-6 ページの「[Recovery Manager コマンドの概要](#)」を参照してください。

@

---

@

## 用途

@ コマンドを使用すると、指定したパス名のオペレーティング・システム・ファイルに格納されている一連の Recovery Manager コマンドを実行できます。

---

---

**注意：** ファイルには、完全な Recovery Manager コマンドを記述する必要があります。コマンドの一部のみを記述すると、構文エラーが発生します。

---

---

## 前提条件

コマンド・ファイルには、完全な Recovery Manager コマンドを記述する必要があります。

@ コマンドを RUN コマンド内で使用する場合は、@ コマンドを独自の行に記述する必要があります (2-3 ページの例 2-2 を参照)。

## 使用上の注意

Recovery Manager は、ファイルの内容が @ コマンドの位置に記述されているのと同様に処理します。2-3 ページの例 2-3 に示すように、コマンド・ファイル内に置換変数を指定し、実行時にそのコマンド・ファイルに値を渡すことができます。

**関連項目：** Recovery Manager で置換変数を使用する方法については、2-244 ページの「[RMAN](#)」を参照してください。

## 構文

**at.:=**

→ @ filename

## セマンティクス

構文の要素	説明
<i>filename</i>	たとえば、コマンド・ファイルの名前を @/oracle/dbs/cmd/cmd1.rman のように指定します。絶対パス名を指定しなかった場合 (@cmd1.rman などのように指定した場合) は、現行の作業ディレクトリが対象となります。  ファイルの拡張子は任意のものを指定できます (拡張子はなくても有効です)。文字列の前後を引用符で囲んだり、@ キーワードとファイル名との間に空白を残さないでください。

## 例

### 例 2-1 オペレーティング・システムのコマンドラインからのコマンド・ファイルの実行

次の例では、Recovery Manager コマンド・ファイルを作成し、それをオペレーティング・システムのコマンドラインから実行します。

```
% echo "BACKUP DATABASE;" > backup_db.rman
% rman TARGET / @backup_db.rman
```

### 例 2-2 Recovery Manager でのコマンド・ファイルの実行

次の例は、コマンド・ファイルを、Recovery Manager プロンプトから実行する方法と RUN コマンド内から実行する方法を示しています。ユーザー入力のテキストは太字で示しています。

```
RMAN> @backup_db.rman
RMAN> RUN {
2> @backup_db.rman
3> backup database;
4> **end-of-file**
5> }
```

### 例 2-3 置換変数の指定

テキスト・エディタを使用して、次の内容のコマンド・ファイル whole\_db を作成したとします。

```
# name: whole_db.rman
BACKUP TAG &1 COPIES &2 DATABASE;
EXIT;
```

次の例では、オペレーティング・システムのプロンプトで Recovery Manager を起動してから、ターゲット・データベースに接続します。次に、@ コマンドを実行してコマンド・ファイルに変数を渡し、タグが Q106 のデータベース・バックアップを 2 つ作成します。

```
% rman TARGET /
RMAN> @/tmp/whole_db.rman Q106 2
```

## @@

## 用途

@@ コマンドを使用すると、指定したファイル名のオペレーティング・システム・ファイルに格納されている一連の Recovery Manager コマンドを実行できます。

@@ がコマンド・ファイルに含まれている場合、@@filename によって、Recovery Manager は、指定したファイル名をコール元のコマンド・ファイルと同じディレクトリ内で検索します。@@ コマンドは、コマンド・ファイル内で使用しない場合、@ コマンドと同じです。

## 前提条件

コマンド・ファイルには、完全な Recovery Manager コマンドを記述する必要があります。

## 使用上の注意

コマンド・ファイルは、Recovery Manager クライアントに対してローカルです。ファイル名の解決は、オペレーティング・システムに依存します。たとえば、UNIX または Windows での @tmp/cmd1.rman の場合、tmp が現行のディレクトリのサブディレクトリで、ファイル cmd1.rman がそのサブディレクトリの中にあることを意味しています。

ここで、@ と @@ の違いを示します。Recovery Manager を次のように起動するとします。

```
% rman TARGET /
RMAN> @/tmp/cmd1.rman
```

cmd1.rman スクリプト内にコマンド @@cmd2.rman があるとします。この場合、@@ コマンドは Recovery Manager に対して、ディレクトリ /tmp にあるファイル cmd2.rman を検索するように指示します。

@ コマンドの場合と同様に、コマンド・ファイル内に置換変数を指定して、@@ の実行時に、そのコマンド・ファイルに値を渡すことができます (2-4 ページの例 2-4 を参照)。

## 構文

**atat::=**

→ @@ filename →

## セマンティクス

構文の要素	説明
filename	たとえば、コマンド・ファイルの名前を @@cmd2.rman のように指定します。

## 例

**例 2-4 別のコマンド・ファイル内からのコマンド・ファイルのコール**

次のオペレーティング・システム・コマンドによって、コマンド・ファイル backup\_logs.rman および backup\_db.rman が作成されます。

```
% echo "BACKUP ARCHIVELOG ALL;" > /tmp/bkup_logs.rman
% echo "BACKUP TAG &1 DATABASE;" > /tmp/bkup_db.rman
% echo "@@bkup_logs.rman" >> /tmp/bkup_db.rman
```



次の例では、コマンドラインから Recovery Manager を起動し、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。@ コマンドは、bkup\_db.rman を実行しますが、そのファイルには @@bkup\_logs.rman コマンドが記述されています。@@ コマンドは、Recovery Manager に対して、bkup\_db.rman が存在するディレクトリと同じディレクトリから bkup\_logs.rman スクリプトを検索するように指示します。この例では、置換変数を使用して、データベース・バックアップのタグ WHOLE\_DB が指定されていることに注意してください。

```
% rman TARGET /  
RMAN> @/tmp/bkup_db.rman whole_db
```

## ADVISE FAILURE

### 用途

ADVISE FAILURE コマンドを使用すると、指定した障害の修復オプションを表示できます。このコマンドは、データ・リカバリ・アドバイザーが認識している障害のサマリーを出力し、修復済の障害がオープン状態になっている場合は、暗黙的にクローズします。

このコマンドを使用する場合の推奨手順は、次のようになります。Recovery Manager セッションで、まず LIST FAILURE を実行して障害を表示し、次に ADVISE FAILURE を実行して修復オプションを表示し、REPAIR FAILURE を実行して障害を修復してください。

### 前提条件

Recovery Manager がターゲット・データベースに接続している必要があります。TARGET としてデータベースに接続する方法については、CONNECT および RMAN コマンドを参照してください。

ターゲット・データベースのインスタンスは起動されている必要があります。ターゲット・データベースは、単一インスタンス・データベースである必要があります。また、ロジカル・スタンバイ・データベースは指定できますが、フィジカル・スタンバイ・データベースは指定できません。

現行リリースでは、データ・リカバリ・アドバイザーは、単一インスタンス・データベースのみをサポートします。Oracle Real Application Clusters (Oracle RAC) データベースはサポートされていません。

### 使用上の注意

データ・リカバリ・アドバイザーは、修復の可能性を検証してから修復方法を提案します。たとえば、データ・リカバリ・アドバイザーが、メディア・リカバリに必要なすべてのバックアップとアーカイブ REDO ログ・ファイルが使用可能かどうかをチェックするとします。ADVISE FAILURE の出力には、データ・リカバリ・アドバイザーが特定の障害に対して最適とみなした修復方法が示されます。ADVISE FAILURE コマンドでは、手動と自動の両方の修復オプションを生成できます。

#### 手動修復オプション

手動修復オプションは、必須または任意のいずれかになります。任意のアクションの方が、自動修復よりも短時間で簡単に障害を修正できることがあります。他にも、自動修復が実現できないため、手動が唯一のオプションになる場合もあります。たとえば、I/O 障害は、ほとんどの場合に自動では修復できません。また、オペレーティング・システムやディスク・サブシステムから戻されるデータが不十分なために、障害を診断できない場合もあります。

#### 自動修復オプション

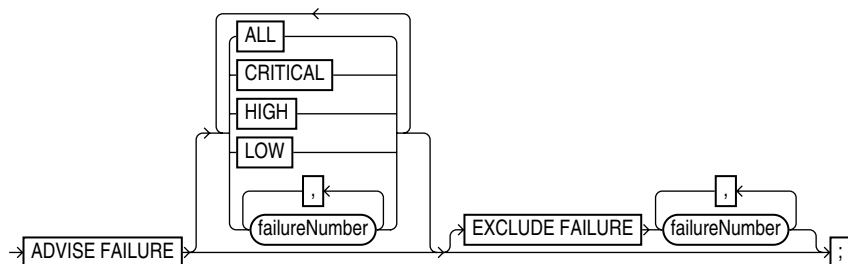
各自動修復オプションは、単一修復または修復ステップ・セットのいずれかです（コマンド出力の説明については、表 2-1 「自動修復オプション」を参照）。修復オプションに、複数の修復ステップがあるスクリプトが含まれる場合、ADVISE FAILURE は、修復ステップが正しい順序になるようにスクリプトを生成します。単一修復の場合は、常に CRITICAL な障害がまとめて修復されます。CRITICAL な障害の修復は必須ですが、CRITICAL でない障害を同時に修復することも可能です。CRITICAL でない障害は、1 つずつ任意の順序で修復することも、まとめて修復することもできます。

## Oracle RAC およびデータ・リカバリ・アドバイザー

Oracle RAC データベースのすべてのインスタンスでデータ障害が発生した場合に、単一インスタンス・モードでデータベースをマウントしてデータ・リカバリ・アドバイザーを使用すると、制御ファイル、SYSTEM データファイルおよびディスクジョナリの障害を検出して修復できます。また、ヘルス・チェックを開始して、他のデータベース・コンポーネントにデータ障害がないかをテストすることもできます。この方法では、他のクラスタ・インスタンスに対してローカルなデータ障害（アクセス不能なデータファイルなど）は検出されません。

## 構文

**advise::=**



## セマンティクス

### advise

構文の要素	説明
ADVISE FAILURE	自動診断リポジトリに記録されている障害のうち、優先順位が CRITICAL または HIGH のものに関するすべての情報を表示します。  現在のセッションで LIST FAILURE コマンドが実行済のときは、オプションを指定しない場合にのみ ADVISE FAILURE を使用できます。  <b>注意：</b> 現在の Recovery Manager セッションで、LIST FAILURE コマンドを最後に実行した後に新しい障害が診断リポジトリに記録されている場合、Recovery Manager は警告を発行してから、CRITICAL および HIGH と記録される障害に関するアドバイスを表示します。
ALL	オープン状態のすべての障害をまとめて修復するオプションを表示します。
CRITICAL	重大な障害のみを修復するオプションを表示します。
HIGH	優先順位が HIGH の障害のみを修復するオプションを表示します。
LOW	優先順位が LOW の障害のみを修復するオプションを表示します。
failureNumber	指定された障害のみを修復するオプションを表示します。
EXCLUDE FAILURE	指定された障害をリストから除外します。
failureNumber	

## ADVISE FAILURE コマンドの出力

ADVISE FAILURE の出力には、LIST FAILURE の出力も含まれます（説明は、表 2-26 「障害のリスト」を参照）。出力例については、2-8 ページの例 2-5 を参照してください。

Recovery Manager は、順序が決まっていないリスト形式で必須および任意の手動によるアクションを提示します。手動オプションが存在する場合は、自動オプションの前に表示されます。表 2-1 に、自動修復オプションの出力を示します。

表 2-1 自動修復オプション

列	指定対象
Option	自動修復オプションの識別子。
Strategy	<p>REPAIR FAILURE コマンドで障害を修復する方法。</p> <p>データ・リカバリ・アドバイザーで提示される自動修復オプションは、可能な場合には必ずデータ消失のないオプションになります。自動修復オプションは、基本的に次のカテゴリに分類されます。</p> <ul style="list-style-type: none"> <li>■ データ消失のない修復</li> <li>■ データ消失を伴う修復（フラッシュバック・データベースなど）</li> </ul> <p><b>注意:</b> ADVISE コマンドは、データ・リカバリ・アドバイザーが最適とみなした一連の修復ステップに一連の障害をマッピングします。可能な場合は、データ・リカバリ・アドバイザーが複数の修復ステップを1つの修復としてまとめます。たとえば、データベースのデータファイルが破損し、制御ファイルと現行の REDO ログ・グループがなくなった場合、データ・リカバリ・アドバイザーでは、データベースをリストアして Point-in-Time リカバリを実行するために、1つの統合された修復計画が提示されます。</p>
Repair Description	提案された修復の説明。たとえば、データファイル 17 のリストアとリカバリが、修復として提案されます。
Repair Script	すべての修復アクションとコメントが記述された編集可能なスクリプトの場所。自動修復を実行しない場合は、このスクリプトの内容を確認して、手動リカバリで使用するよう編集できます。

## 例

### 例 2-5 すべての障害の修復オプションの表示

この例は、データ・リカバリ・アドバイザーで認識されているすべての障害の修復オプションを表示します。この例では、2つの障害（データファイルの欠落と、破損ブロックのあるデータファイル）が示されています。

```
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: '/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

# ALLOCATE CHANNEL

## 用途

ALLOCATE CHANNEL コマンドを使用すると、Recovery Manager とデータベース・インスタンスを接続する **チャンネル** を手動で割り当てることができます。ALLOCATE CHANNEL は、REPAIR FAILURE ブロック内で発行する必要があります。適用されるのは、発行されたブロックのみです。

## 前提条件

ターゲット・インスタンスを事前に起動する必要があります。

## 使用上の注意

手動で割り当てたチャンネルと、CONFIGURE を指定して自動的に割り当てたチャンネルは区別されます。自動チャンネルは、手動でチャンネルを割り当てていないすべての Recovery Manager のジョブに適用されます。自動チャンネル構成を、RUN コマンド内で手動で割り当てたチャンネルでオーバーライドすることはできませんが、手動チャンネルを ALLOCATE CHANNEL で指定した後で、BACKUP DEVICE TYPE または RESTORE DEVICE TYPE を使用して、自動チャンネルを使用することはできません。

### 複数チャンネル

割り当てることができるチャンネルは最大 255 で、チャンネルあたり最大 64 ファイルを平行に読み取ることができます。ジョブ内の並列度は、割り当てるチャンネル数で制御できます。複数チャンネルを同時に割り当てると、1つのジョブで複数のバックアップ・セットやディスク・コピーを平行に読み書きできます。こうして、各チャンネルが別々のバックアップ・セットまたはコピーの操作を行います。

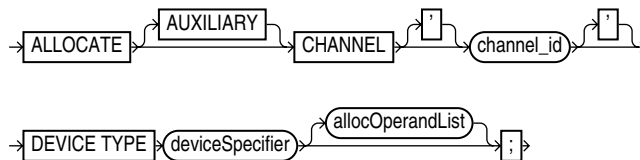
ディスクへのバックアップを行う場合、通常は、出力デバイスごとにチャンネルを1つ割り当てます (2-12 ページの例 2-7 を参照)。ただし、Recovery Manager が、ストライプ化されたファイル・システムまたは ASM ディスク・グループに対して書き込みを行う場合は、複数のチャンネルを使用することでパフォーマンスを向上できます。テープにバックアップを作成する場合は、一般に、テープ・デバイスの数を多重化されたコピー数で除算した数が、テープ・チャンネルの数に等しくなるようにします (2-12 ページの例 2-8 を参照)。

### Oracle RAC 環境のチャンネル

すべての Oracle RAC インスタンスの SYSDBA パスワードが同じであるかぎり、ALLOCATE または CONFIGURE コマンドの CONNECT オプションでパスワードを指定する必要はありません。user@database 形式の接続文字列を使用すると、Recovery Manager セッションの開始時に TARGET 接続で使用されたものと同じパスワードが自動的に使用されます。

## 構文

**allocate::=**



(deviceSpecifier::= (3-15 ページ)、 allocOperandList::= (3-2 ページ))

## セマンティクス

構文の要素	説明
AUXILIARY	<p>Recovery Manager と補助データベース・インスタンスとの接続を指定します。</p> <p>補助インスタンスは、<a href="#">DUPLICATE</a> または <a href="#">TRANSPORT TABLESPACE</a> コマンドを実行する場合、および <a href="#">RECOVER TABLESPACE</a> で TSPITR を実行する場合に使用します (2-12 ページの <a href="#">例 2-9</a> を参照)。このオプションを指定する場合は、補助インスタンスが起動されている必要がありますが、マウントされている必要はありません。</p> <p><b>関連項目</b> : データベースの複製方法は 2-125 ページの「<a href="#">DUPLICATE</a>」を、複製データベース・インスタンスへの接続方法は 2-88 ページの「<a href="#">CONNECT</a>」を参照してください。</p>
CHANNEL ' <i>channel_id</i> '	<p>Recovery Manager とターゲット・データベース・インスタンスとの接続を指定します。<i>channel_id</i> はチャンネルの名前で、大 / 小文字が区別されます。データベースでは、I/O エラーのレポートに <i>channel_id</i> が使用されます。</p> <p>接続するたびに、ターゲット・インスタンスまたは補助インスタンスでデータベース・サーバー・セッションが開始されます。このセッションで、Recovery Manager バックアップのバックアップ、リストアまたはリカバリが実行されます。共有サーバー・セッションには接続できません。</p> <p>ALLOCATE CHANNEL によってオペレーティング・システム・リソースがすぐに割り当てられるかどうかは、オペレーティング・システムによって異なります。プラットフォームによっては、コマンドの発行時に割り当てられます。別のプラットフォームでは、ファイルを読み書きのためにオープンするまで割り当てられません。</p> <p>各チャンネルは、一度に 1 つのバックアップ・セットまたはイメージ・コピーを使用します。Recovery Manager は、ジョブ終了時に自動的にチャンネルを解放します。</p> <p><b>注意</b> : チャンネル名の接頭辞に <code>ORA_</code> は使用できません。接頭辞 <code>ORA_</code> で始まるチャンネル名は、Recovery Manager 専用予約されています。</p>
DEVICE TYPE <i>deviceSpecifier</i>	<p>バックアップ用のストレージ・タイプを指定します。V\$BACKUP_DEVICE ビューへの問合せで、使用可能なデバイス・タイプと名前がわかります。</p> <p><b>注意</b> : DEVICE TYPE DISK を指定すると、サーバー・セッション作成用以外のオペレーティング・システム・リソースは割り当てられません。</p> <p><b>関連項目</b> : 3-15 ページの「<a href="#">deviceSpecifier</a>」を参照してください。</p>
<i>allocOperandList</i>	<p>割り当てたチャンネルの制御オプションを指定します。順次 I/O デバイスのチャンネル・パラメータはプラットフォームによって異なることに注意してください (2-11 ページの <a href="#">例 2-6</a> を参照)。</p> <p><b>関連項目</b> : 3-2 ページの「<a href="#">allocOperandList</a>」を参照してください。</p>

## 例

### 例 2-6 バックアップ用チャンネルの手動割当て

この例では、データベース全体およびアーカイブ REDO ログのバックアップ用に 1 つのテープ・チャンネルを割り当てます。PARMS パラメータで、`wholedb_mf` という Oracle Secure Backup メディア・ファミリを指定しています。

```

RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt
    PARMS 'ENV=(OB_MEDIA_FAMILY=wholedb_mf)';
  BACKUP DATABASE;
  BACKUP ARCHIVELOG ALL NOT BACKED UP;
}

```

**例 2-7 複数ディスクへのバックアップの分散**

ディスクにバックアップする場合は、複数のディスク・ドライブに分散したバックアップが実行できます。ディスク・ドライブごとに DEVICE TYPE DISK チャンネルを1つ割り当て、出力ファイルごとにディスクが異なるようにフォーマット文字列を指定します。

```
RUN
{
  ALLOCATE CHANNEL disk1 DEVICE TYPE DISK FORMAT '/disk1/%U';
  ALLOCATE CHANNEL disk2 DEVICE TYPE DISK FORMAT '/disk2/%U';
  BACKUP DATABASE PLUS ARCHIVELOG;
}
```

**例 2-8 テープへのバックアップの複数コピーの作成**

この例では、stape1、stape2、stape3 および stape4 の4つのテープ・ドライブを書込みで使用できます。SET BACKUP COPIES コマンドを使用して、Recovery Manager がデータベース・バックアップの同じコピーを2つ作成する必要があることを示します。一般に、テープ・チャンネルの数は、多重化されたコピー数でテープ・デバイスの数を除算した数と等しくする必要がありますため、2つのチャンネルを割り当てます。この場合、BACKUP\_TAPE\_IO\_SLAVES 初期化パラメータを TRUE に設定する必要があることに注意してください。

Oracle Secure Backup の OB\_DEVICE\_n パラメータでは、n にバックアップ・ピースのコピー数を指定します。Recovery Manager は、各バックアップ・ピースのコピー1を stape1 と stape2 のテープ・ドライブに書き込み、各バックアップ・ピースのコピー2を stape3 と stape4 のドライブに書き込みます。このようにして、データベース・バックアップの各コピーは、2つのテープ・ドライブ間で分散され、各ドライブ上にデータの一部分が格納されます。

```
RUN
{
  ALLOCATE CHANNEL t1 DEVICE TYPE sbt
    PARS 'ENV=(OB_DEVICE_1=stape1,OB_DEVICE_2=stape3)';
  ALLOCATE CHANNEL t2 DEVICE TYPE sbt
    PARS 'ENV=(OB_DEVICE_1=stape2,OB_DEVICE_2=stape4)';
  SET BACKUP COPIES 2;
  BACKUP DATABASE;
}
```

**例 2-9 データベース複製用の補助チャンネルの割当て**

この例では、バックアップから複製データベースが作成されます。複製用に構成されたチャンネルで AUXILIARY オプションが指定されていない場合でも、Recovery Manager は、そのチャンネルを使用できます。この例では、事前に構成されている SBT チャンネルがないため、補助 SBT チャンネルを手動で割り当てます。

```
RUN
{
  ALLOCATE AUXILIARY CHANNEL c1 DEVICE TYPE sbt;
  DUPLICATE TARGET DATABASE
    TO dupdb
  DB_FILE_NAME_CONVERT '/disk2/dbs/', '/disk1/'
  SPFILE
    PARAMETER_VALUE_CONVERT '/disk2/dbs/',
                              '/disk1/'
  SET LOG_FILE_NAME_CONVERT '/disk2/dbs/',
                              '/disk1/';
}
```



# ALLOCATE CHANNEL FOR MAINTENANCE

## 用途

ALLOCATE CHANNEL FOR MAINTENANCE コマンドを使用すると、**CHANGE**、**DELETE** または **CROSSCHECK** コマンドの発行に備えてチャンネルを手動で割り当てることができます。チャンネルの割当て解除は、**RELEASE CHANNEL** コマンドで実行できます。

---

**注意：** **CONFIGURE** を使用して、構成内のデバイス・タイプごとにチャンネルを 1 つ以上割り当てる場合は、ALLOCATE CHANNEL FOR MAINTENANCE を使用する必要はありません。メンテナンス・チャンネルではなく構成済チャンネルを使用することをお勧めします。構成済チャンネルは、メンテナンス・チャンネルでサポートされるメンテナンス・タスクのみでなく、指定したデバイスへのすべての Recovery Manager の I/O に使用できます。構成済チャンネルは、Recovery Manager セッション全体で保持されます。

---

## 前提条件

このコマンドは、RUN ブロック内ではなく、Recovery Manager プロンプトでのみ実行できます。ターゲット・インスタンスを事前に起動する必要があります。共有セッションにメンテナンス・チャンネルを割り当てることはできません。

## 使用上の注意

メンテナンス・チャンネルは、原則としてデバイスごとに 1 つずつ割り当ててください。手動割当てチャンネルと自動チャンネルが混在することはありません。一般に、1 つのジョブに対して複数のメンテナンス・チャンネルを割り当てるのは、次の場合に限定してください。

- 単一のコマンドを使用して、ディスクとテープの両方で、すべてのバックアップ・ピースまたはプロキシ・コピーのクロスチェックまたは削除を行えるようにする場合 (2-15 ページの例 2-11 を参照)。
- バックアップ・ピースまたはプロキシ・コピーが単一ノードにしか存在しない Oracle RAC 構成で、クロスチェックと削除を正常に機能させる場合 (2-15 ページの例 2-12 を参照)。

Recovery Manager ではメンテナンス・チャンネルのネーミング規則として

ORA\_MAINT\_devicetype\_n が使用されます。devicetype は DISK または sbt で、n はチャンネル番号です。たとえば、Recovery Manager では、手動で割り当てた 2 つのディスク・チャンネルに次の名前が使用されます。

```
ORA_MAINT_DISK_1
ORA_MAINT_DISK_2
```

**関連項目：** 複数チャンネルをクロスチェックおよび削除する方法は、『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』を参照してください。

## 構文

**allocateForMaint::=**

```
→ ALLOCATE CHANNEL FOR MAINTENANCE }
```

```
→ DEVICE TYPE {deviceSpecifier} {allocOperandList} ;
```

(deviceSpecifier::= (3-15 ページ)、allocOperandList::= (3-2 ページ))

## セマンティクス

### allocateForMaint

構文の要素	説明
DEVICE TYPE <i>deviceSpecifier</i>	バックアップ用のストレージ・タイプを指定します。V\$BACKUP_DEVICE ビューへの問合せで、使用可能なデバイス・タイプと名前がわかります。 <b>関連項目</b> : 3-15 ページの「 <i>deviceSpecifier</i> 」を参照してください。
<i>allocOperandList</i>	割り当てたチャンネルの制御オプションを指定します。順次 I/O デバイスのチャンネル・パラメータはプラットフォームによって異なることに注意してください。 <b>関連項目</b> : 3-2 ページの「 <i>allocOperandList</i> 」を参照してください。

## 例

### 例 2-10 バックアップ・セットの削除

すべての Recovery Manager バックアップを削除して、テープ・セットを再利用する必要があるとします。この例では、ディスク・チャンネルのみがデフォルトで構成されています。この例では、SBT チャンネルを手動で割り当て、すべてのバックアップをテープから削除した後で、チャンネルを解放します。

```

RMAN> ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt;

allocated channel: ORA_MAINT_SBT_TAPE_1
channel ORA_MAINT_SBT_TAPE_1: SID=135 device type=SBT_TAPE
channel ORA_MAINT_SBT_TAPE_1: Oracle Secure Backup

RMAN> DELETE NOPROMPT BACKUP;

List of Backup Pieces
BP Key   BS Key   Pc# Cp# Status      Device Type Piece Name
-----
9957     9954    1  1  AVAILABLE  SBT_TAPE   8oic41ad_1_1
9974     9972    1  1  AVAILABLE  SBT_TAPE   c-28014364-20070308-17
10024    10021   1  1  AVAILABLE  SBT_TAPE   8qic41c3_1_1
10045    10042   1  1  AVAILABLE  SBT_TAPE   c-28014364-20070308-18
10446    10443   1  1  AVAILABLE  SBT_TAPE   8uic47fg_1_1
10487    10482   1  1  AVAILABLE  SBT_TAPE   90ic47ih_1_1
10488    10483   1  1  AVAILABLE  SBT_TAPE   91ic47j1_1_1
10524    10514   1  1  AVAILABLE  SBT_TAPE   92ic47q4_1_1
10540    10538   1  1  AVAILABLE  SBT_TAPE   c-28014364-20070308-1a
deleted backup piece
backup piece handle=8oic41ad_1_1 RECID=198 STAMP=616695118
deleted backup piece
backup piece handle=c-28014364-20070308-17 RECID=199 STAMP=616695145
deleted backup piece
backup piece handle=8qic41c3_1_1 RECID=200 STAMP=616695171
deleted backup piece
backup piece handle=c-28014364-20070308-18 RECID=201 STAMP=616695188
deleted backup piece
backup piece handle=8uic47fg_1_1 RECID=204 STAMP=616701424
deleted backup piece
backup piece handle=90ic47ih_1_1 RECID=205 STAMP=616701521
deleted backup piece
backup piece handle=91ic47j1_1_1 RECID=206 STAMP=616701538
deleted backup piece
backup piece handle=92ic47q4_1_1 RECID=207 STAMP=616701764
deleted backup piece
backup piece handle=c-28014364-20070308-1a RECID=208 STAMP=616701783
Deleted 11 objects

```

```

RMAN> RELEASE CHANNEL;

released channel: ORA_MAINT_SBT_TAPE_1

```

### 例 2-11 複数のデバイス上のバックアップのクロスチェック

ディスク上とテープ上のアーカイブ REDO ログをクロスチェックする必要があるとします。また、デフォルトのデバイス・タイプがディスクに設定され、SBT チャンネルも構成されていますが、ディスクとテープの両方に別のチャンネルを使用する必要があるとします。その場合は、必要に応じた設定でメンテナンス・チャンネルを手動で割り当てることができます。

```

RMAN> SHOW DEFAULT DEVICE TYPE;

RMAN configuration parameters for database with db_unique_name PROD are:
CONFIGURE DEFAULT DEVICE TYPE TO DISK;

RMAN> SHOW CHANNEL;

RMAN configuration parameters for database with db_unique_name PROD are:
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMS
'SBT_LIBRARY=/usr/local/oracle/backup/lib/libobk.so, ENV=(OB_DEVICE_1=stape1)';

RMAN> ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt PARMS
'SBT_LIBRARY=/usr/local/oracle/backup/lib/libobk.so, ENV=(OB_DEVICE_1=stape2)';

allocated channel: ORA_MAINT_SBT_TAPE_1
channel ORA_MAINT_SBT_TAPE_1: SID=135 device type=SBT_TAPE
channel ORA_MAINT_SBT_TAPE_1: Oracle Secure Backup

RMAN> ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK FORMAT "/disk2/%U";

allocated channel: ORA_MAINT_DISK_2
channel ORA_MAINT_DISK_2: SID=101 device type=DISK

Finished Control File and SPFILE Autobackup at 09-MAR-07

RMAN> CROSSCHECK BACKUP OF ARCHIVELOG ALL;

crosschecked backup piece: found to be 'AVAILABLE'
backup piece handle=/disk2/95ic69jc_1_1 RECID=210 STAMP=616769132
crosschecked backup piece: found to be 'EXPIRED'
backup piece handle=/disk2/96ic69jf_1_1 RECID=211 STAMP=616769135
Crosschecked 2 objects

crosschecked backup piece: found to be 'AVAILABLE'
backup piece handle=/disk2/96ic69jf_1_1 RECID=211 STAMP=616769135
Crosschecked 1 objects

RMAN> RELEASE CHANNEL;

released channel: ORA_MAINT_SBT_TAPE_1
released channel: ORA_MAINT_DISK_2

```

### 例 2-12 Oracle Real Application Clusters (Oracle RAC) 構成でのクロスチェック

Oracle RAC 構成内のすべてのノードに、すべてのストレージ・デバイス上のすべてのバックアップに対する同じアクセス権を付与することをお勧めします（ただし、これは必須要件ではありません）。Oracle RAC 構成の 2 つのノードでバックアップのクロスチェックを実行するとします。各ノードには、ディスク・バックアップのサブセットへのアクセス権があります。すべてのバックアップは、クロスチェックで使用される 2 つのノードのうち少なくとも 1 つからアクセス可能であるものとします。いずれのノードからもアクセスできない任意のバックアップは、クロスチェック後に EXPIRED とマークされます。

次の例では、Oracle RAC インスタンス `inst1` および `inst2` へのチャンネル接続を示します。両方のチャンネル接続で、**Recovery Manager** は、ターゲット・データベース接続で入力したユーザー名とパスワードと同じものを使用します。

```
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK
CONNECT '@inst1';
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK
CONNECT '@inst2';
CROSSCHECK BACKUP;
```

## ALTER DATABASE

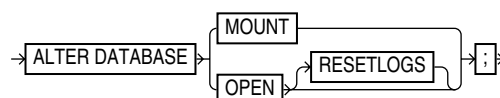
### 用途

ALTER DATABASE コマンドを使用すると、データベースをマウントまたはオープンできます。

**関連項目：** ALTER DATABASE の構文とセマンティクスについては、『Oracle Database SQL 言語リファレンス』も参照してください。

### 構文

*alterDatabase::=*



### 前提条件

このコマンドは、RUN コマンドのカッコ内または Recovery Manager プロンプトで実行してください。ターゲット・インスタンスを事前に起動する必要があります。

### セマンティクス

構文の要素	説明
MOUNT	データベースをマウントします。オープンはしません。このオプションを指定したコマンドは、SQL 文 ALTER DATABASE MOUNT と等価です。
OPEN	データベースをオープンします（例 2-13 を参照）。RECOVER DATABASE の実行後にデータベースをオープンすると、Recovery Manager のリポジトリに記録されているローカル管理のすべての一時ファイルが必要に応じて再作成されます。ただし、リカバリ・カタログを使用せずに、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た場合、制御ファイルのバックアップ後に作成された一時ファイルは Recovery Manager のリポジトリに記録されません。また、一時ファイルが自動的に再作成されることもありません。
RESETLOGS	<p>現行のオンライン REDO ログ（または、破損が検出された場合は REDO 破損前の最後の REDO レコードまで）をアーカイブし、オンライン REDO ログの内容を消去してオンライン REDO ログをログ順序 1 にリセットします。Recovery Manager コマンドの ALTER DATABASE OPEN RESETLOGS は、SQL 文の ALTER DATABASE OPEN RESETLOGS と等価です。</p> <p>リカバリ・カタログを使用する場合、Recovery Manager は、データベースがオープンされた後で RESET DATABASE を暗黙的に発行し、この新規のインカネーションをカタログ内で現行のインカネーションにします。Recovery Manager コマンドの ALTER DATABASE OPEN RESETLOGS ではなく、同じ名前の SQL 文を実行した場合は、RESET DATABASE コマンドを手動で実行する必要があります。</p>

## 例

### 例 2-13 データベースの一貫性バックアップの作成

データベースがオープンされており、そのデータベース全体の一貫性バックアップを作成する必要があるとします。この例では、一貫性を保ってデータベースを停止し、データベースをマウントし、一貫性のあるデータベース全体のバックアップを作成してから、データベースをオープンします。

```
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
BACKUP DATABASE PLUS ARCHIVELOG;
# Now that the backup is complete, open the database.
ALTER DATABASE OPEN;
```

### 例 2-14 制御ファイルのリストア後のデータベースのマウント

この例では、制御ファイルのリストアとマウントを行ってから、リカバリを実行します。最後に、オンライン REDO ログをリセットします。

```
STARTUP FORCE NOMOUNT;
RESTORE CONTROLFILE FROM AUTOBACKUP;
ALTER DATABASE MOUNT;
# you must run the RECOVER command after restoring a control file even if no datafiles
# require recovery
RECOVER DEVICE TYPE DISK DATABASE;
ALTER DATABASE OPEN RESETLOGS;
```

## BACKUP

### 用途

BACKUP コマンドを使用すると、データベース（プライマリまたはスタンバイ）、表領域、データファイル（現行またはコピー）、制御ファイル（現行またはコピー）、サーバー・パラメータ・ファイル、アーカイブ REDO ログ・ファイルまたはバックアップ・セットのバックアップを作成できます。

### 追加トピック

- [前提条件](#)
- [使用上の注意](#)
- [構文](#)
- [セマンティクス](#)
- [例](#)

### 前提条件

Recovery Manager がターゲット・データベースに接続されている必要があります。TARGET としてデータベースに接続する方法については、[CONNECT](#) および [RMAN](#) コマンドを参照してください。

#### データベースのアーカイブ・モード

ターゲット・データベースのモードが ARCHIVELOG の場合は、データベースが、現行の制御ファイルでマウントされているかまたはオープンされている必要があります。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。一貫性のないバックアップをリストアした後は、データベースに一貫性を持たせるために、REDO ログを適用する必要があります。

ターゲット・データベースのモードが NOARCHIVELOG の場合は、バックアップの作成時に一貫性のある停止を行った後で、データベースがマウントされている必要があります。停止の一貫性が保たれるのは、NORMAL、IMMEDIATE または TRANSACTIONAL オプションを指定して、SHUTDOWN コマンドを正しく実行できた場合のみです。インスタンス障害が発生した場合や SHUTDOWN ABORT を実行した後で、Recovery Manager を使用して、NOARCHIVELOG のデータベースをバックアップすることはできません。

#### バックアップ・メディア

ファイルをバックアップできるのは、有効なメディアに対してのみです。DEVICE TYPE DISK を指定すると、ランダム・アクセス・ディスクにバックアップが作成されます。バックアップは、データファイルを格納できる任意のデバイスに作成できます。文 CREATE TABLESPACE *tablespace\_name* DATAFILE '*filename*' が正しく動作すれば、'*filename*' は有効なバックアップ・パス名です。DEVICE TYPE sbt を指定した場合は、メディア・マネージャがサポートするメディアであれば、どのメディアにでもファイルをバックアップできます。

Oracle データベースのファイルをディスクにバックアップする場合、そのファイルの論理ブロック・サイズは、バックアップ先デバイスの物理ブロック・サイズの偶数倍である必要があります。たとえば、ブロック・サイズが 2KB のディスク・デバイスは、論理ブロック・サイズが 2KB、4KB、6KB などの Oracle ファイルのバックアップ先としてのみ使用できます。実際には、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 バイトのため、この制限事項がバックアップに影響することはほとんどありません。ただし、BACKUP ... DEVICE TYPE DISK コマンドを使用して、書込み可能 CD や DVD、またはより大容量の物理ブロック・サイズを持つその他のデバイスにデータベースをバックアップする場合は、この制限事項を考慮する必要があります。

## チャンネル

指定したデバイス・タイプに自動チャンネルが構成されていない場合は、BACKUP を実行するたびにチャンネルをデバイスに手動で割り当てる必要があります。手動チャンネルを割り当てていない場合、Recovery Manager では CONFIGURE コマンドで設定されたデフォルトのチャンネルが使用されます。Recovery Manager には事前構成済の DISK チャンネルがありますが、事前構成済の sbt チャンネルはありません。

---

**注意：** ディスク・テスト API を使用するバックアップは本番バックアップではサポートされません。かわりに、事前構成済の DISK チャンネルを使用するか、DISK チャンネルを手動で割り当ててください。

---

## 使用上の注意

Recovery Manager でバックアップできるのは、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、アーカイブ REDO ログ・ファイル、およびこれらのファイルの Recovery Manager バックアップのみです。その他のデータベース関連ファイル（ネットワーク構成ファイル、パスワード・ファイル、ブロック・チェンジ・トラッキング・ファイルなど）および Oracle データベース・ホームの内容は、バックアップできません。また、外部表や BFILE データ型などの Oracle データベースの一部の機能についても、同様に、前述のファイル以外のファイルにデータが格納されます。Recovery Manager では、これらのファイルをバックアップできません。

BACKUP コマンドは、Recovery Manager では独立した複数のバックアップ手順に分けられています。独立した各手順は、特定のデバイスに割り当てられたチャンネルで実行できます。割り当てられているチャンネルが複数ある場合に、1つのチャンネルで障害が発生するか、またはバックアップ手順の実行中に問題が発生すると、Recovery Manager は、別のチャンネルで作業の完了を試行します。Recovery Manager は、チャンネルでフェイルオーバーが発生すると、V\$RMAN\_OUTPUT、対話方式セッションまたはログ・ファイルへの出力にメッセージをレポートします。

あるプラットフォームで作成された Recovery Manager バックアップを、別のプラットフォームにトランスポートすることはできません。

以前のリリースの Oracle データベースで作成された Recovery Manager バックアップは、データベースの移行またはアップグレードの実行後に使用できます。

データベースの DBID ではなく、DB\_NAME を変更すると、Recovery Manager は以前の DB\_NAME で作成されたデータベースのバックアップをリストア可能とみなします。

### バックアップ・セットの暗号化

Recovery Manager では、バックアップ・セットに書き込まれるデータを透過的に暗号化し、RESTORE 操作で必要な際に復号化できます。暗号化したバックアップをディスク上に作成するには、データベースで Advanced Security Option を使用している必要があります。暗号化したバックアップをテープ上に直接作成するには、Recovery Manager で、Oracle Secure Backup の SBT インタフェースを使用する必要がありますが、Advanced Security Option を使用する必要はありません。Oracle Secure Backup 以外の SBT ライブラリを使用して暗号化された Recovery Manager バックアップを作成しようとする、Recovery Manager により ORA-19916 エラーが発行されます。

Recovery Manager では、バックアップの暗号化に複数の暗号化アルゴリズムを使用できます（アルゴリズムのリストは V\$RMAN\_ENCRYPTION\_ALGORITHMS にあります）。Recovery Manager は、次の3つの暗号化モードによるバックアップをサポートしています。

- 透過的暗号化：Recovery Manager で、必要な Oracle 公開鍵インフラストラクチャ (PKI) を使用できるかぎり、DBA の介入なく、暗号化されたバックアップを作成およびリストアできます。
- パスワード・ベースの暗号化：バックアップ時にパスワードが指定され、バックアップをリストアする場合に同じパスワードを適用する必要があります。



- デュアル・モードの暗号化: 透過的暗号化またはパスワード・ベースの暗号化の場合と同様に使用してバックアップを作成できます。また、Oracle Wallet または復号化時に DBA が指定するパスワードのいずれかに基づいて、復号化を実行できます。

---

**注意:** Wallet ベースの暗号化は、パスワードが含まれないため、パスワード・ベースの暗号化よりも安全です。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可能な必要があるため、必要な場合のみ使用してください。

---

データベース・バックアップの暗号化設定の管理には、CONFIGURE および SET コマンドを使用します。詳細は、これらのコマンドのリファレンス・エントリを参照してください。アーカイブ・ログが含まれているバックアップ・セットは、次の条件に該当する場合に暗号化されます。

- バックアップの作成時に SET ENCRYPTION ON が有効になっている。
- 暗号化が、データベース全体または 1 つ以上の表領域に対して構成されている。

#### 関連項目:

- バックアップの暗号化の概要、その使用方法、および様々なモードの暗号化の選択方法については、『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』を参照してください。
- PKI および Oracle Wallet については、『Oracle Database Advanced Security 管理者ガイド』を参照してください。

### Data Guard 環境での Recovery Manager のバックアップ

Data Guard 環境で Recovery Manager の操作を行う場合は、リカバリ・カタログが必要です。カタログを使用することにより、すべてのプライマリ・データベースまたはスタンバイ・データベースで、Recovery Manager のすべての操作を透過的に実行できるようになります。この環境では、プライマリ・データベースのバックアップを、任意のスタンバイ・データベースにオフロードできます。Recovery Manager バックアップは、交換可能です。Recovery Manager を NOCATALOG モードで使用する場合、Recovery Manager は、マウントされている制御ファイル内のメタデータのみを使用します。

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ます。たとえば、Recovery Manager がデータベース prod に TARGET として接続し、そのデータベースをバックアップする場合、このデータベースのバックアップは prod 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されたデータベースとの関連付けを維持します。

バックアップの関連付けとアクセス可能性は異なります。リカバリ・カタログでは、ディスク・バックアップはデータベースが作成された Data Guard 環境のデータベースのみからアクセス可能とみなされますが、あるデータベース上で作成されたテープ・バックアップの場合は、すべてのデータベースからアクセス可能とみなされます。バックアップ・ファイルがど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・ビューでは、そのバックアップ・ファイルを示す行の SITE\_KEY 列に null が示されます。デフォルトでは、SITE\_KEY が null のファイルは、Recovery Manager が TARGET として接続されているデータベースに関連付けられます。

Data Guard 環境では、Recovery Manager のコマンドはアクセス可能ないずれのバックアップに対しても操作を実行できます。たとえば、データベース prod と standby1 が異なるホスト上に存在するとします。Recovery Manager が、prod 上のデータファイル 1 を本番ホスト上の /prodhst/disk1/df1.dbf とテープに対してバックアップします。さらに、standby1 上のデータファイル 1 をスタンバイ・ホスト上の /sby1hst/disk2/df1.dbf とテープに対してバックアップします。Recovery Manager がデータベース prod に TARGET として接続されている場合は、スタンバイ・ホスト上にある /sby1hst/disk2/df1.dbf のバックアップに対して Recovery Manager の操作は実行できません。ただし、Recovery Manager は、standby1 で作成されたテープ・バックアップはリストア可能とみなします。

---

**注意：** スタンバイ・ホストからプライマリ・ホストへ（またはその逆方向に）バックアップを FIP すると、そのバックアップに CATALOG を実行できます。ファイルは、ターゲット・データベースでカタログ化された後、ターゲット・ベースと関連付けられます。

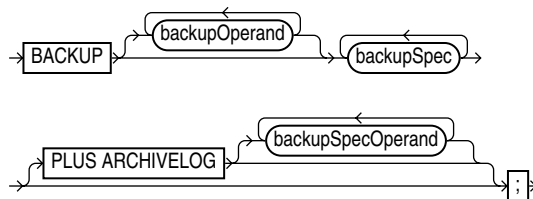
---

Recovery Manager からバックアップにアクセス可能であるかぎり、プライマリまたはスタンバイ・データベースに接続していれば、Recovery Manager のメンテナンス・コマンド (CHANGE、CROSSCHECK、DELETE など) をバックアップに使用できます。

**関連項目：** Data Guard 環境で、Recovery Manager を使用してファイルをバックアップおよび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。

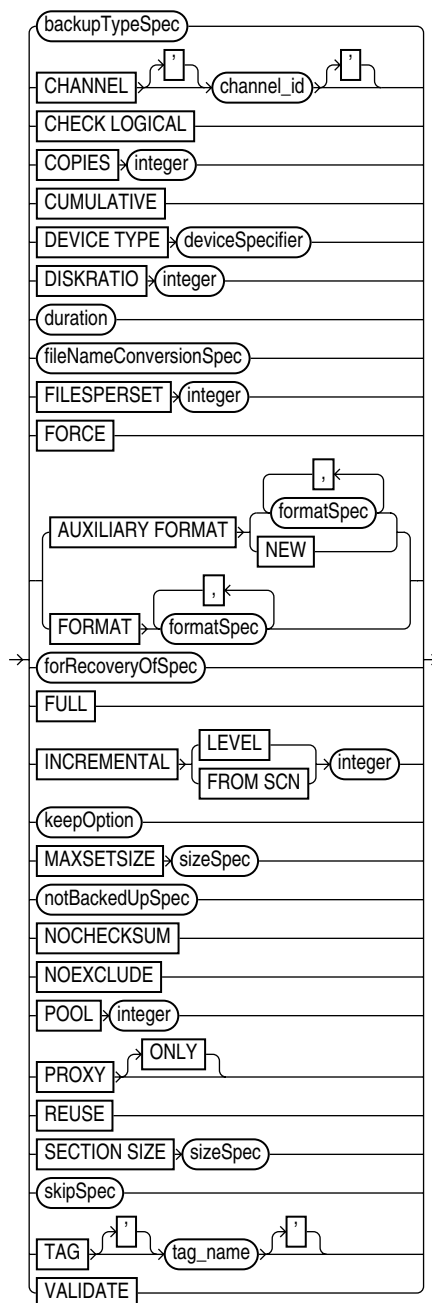
## 構文

**backup::=**



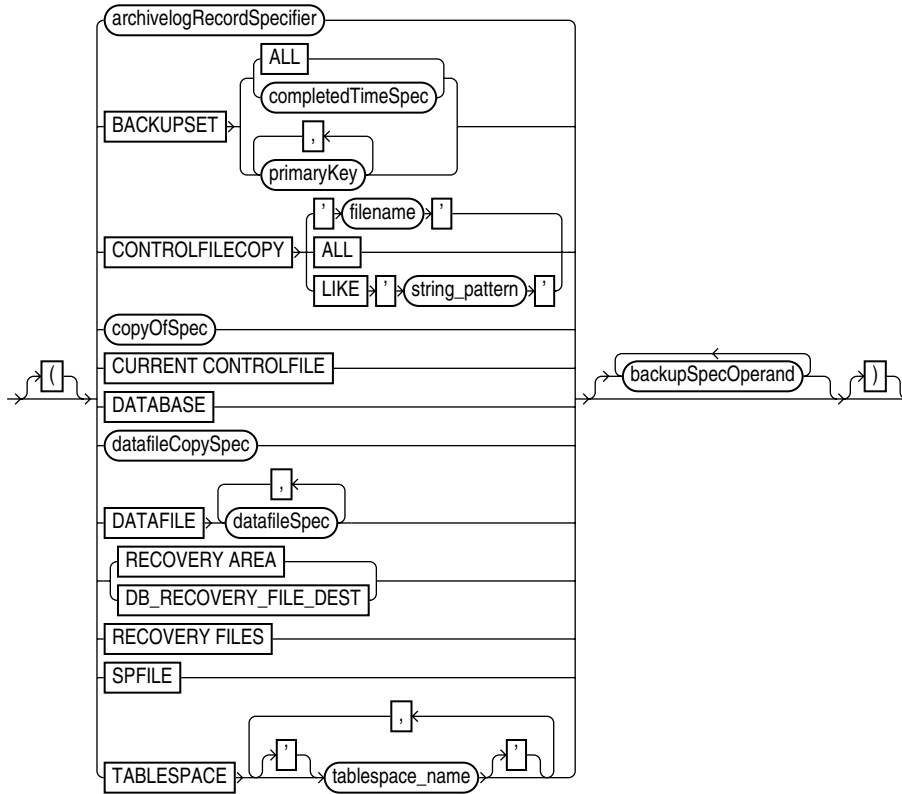
(backupOperand::= (2-23 ページ)、 backupSpec::= (2-24 ページ)、  
 backupSpecOperand::= (2-25 ページ))

**backupOperand::=**



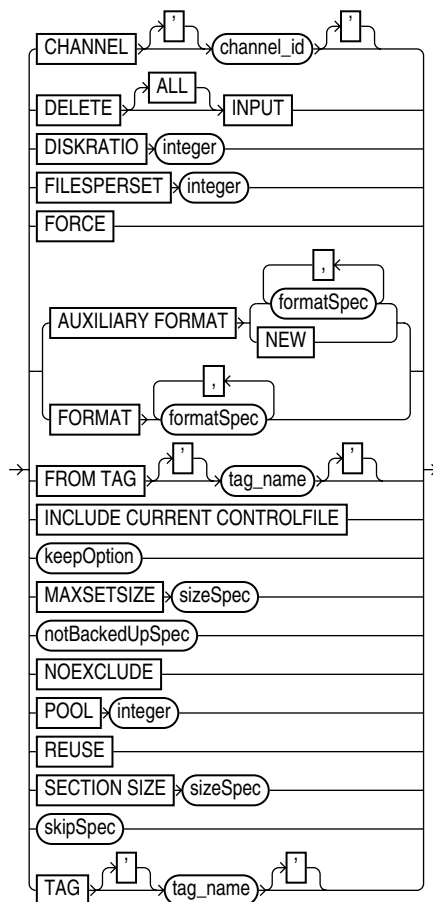
(deviceSpecifier::= (3-15 ページ)、 fileNameConversionSpec::= (3-16 ページ)、  
 formatSpec::= (3-21 ページ)、 forRecoveryOfSpec::= (2-26 ページ)、  
 keepOption::= (3-24 ページ)、 notBackedUpSpec::= (2-26 ページ)、 sizeSpec::= (2-26 ページ)、  
 skipSpec::= (2-26 ページ))

**backupSpec::=**



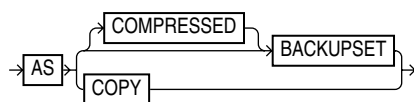
(archivelogRecordSpecifier::= (3-6 ページ)、 completedTimeSpec::= (3-10 ページ)、  
copyOfSpec::= (2-25 ページ)、 datafileCopySpec::= (2-26 ページ)、  
datafileSpec::= (3-14 ページ)、 backupSpecOperand::= (2-25 ページ))

**backupSpecOperand::=**

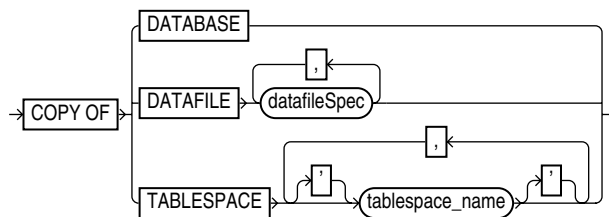


(formatSpec::= (3-21 ページ)、 keepOption::= (3-24 ページ)、  
notBackedUpSpec::= (2-26 ページ)、 sizeSpec::= (2-26 ページ)、 skipSpec::= (2-26 ページ))

**backupTypeSpec::=**

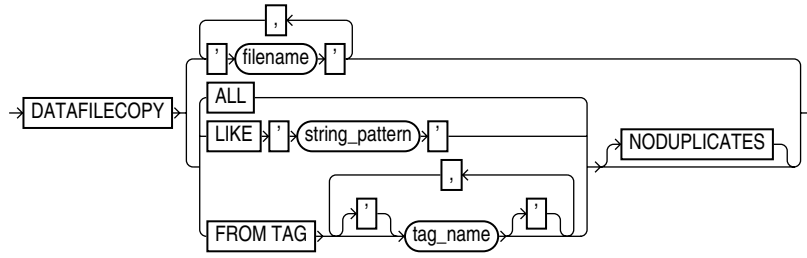


**copyOfSpec::=**

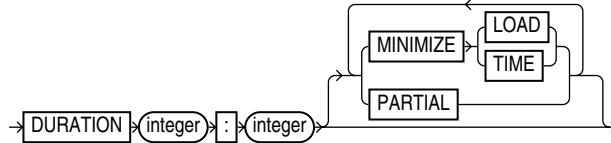


(datafileSpec::= (3-14 ページ))

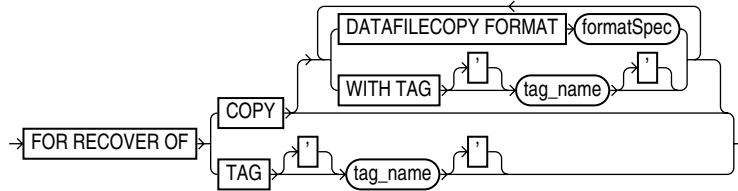
**datafileCopySpec::=**



**duration::=**

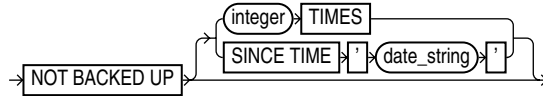


**forRecoveryOfSpec::=**

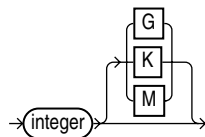


(formatSpec::= (3-21 ページ))

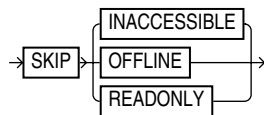
**notBackedUpSpec::=**



**sizeSpec::=**



**skipSpec::=**



## セマンティクス

### backup

この句は、バックアップするオブジェクトと、バックアップの制御オプションを指定します。構文図は、2-22 ページの「[backup::=](#)」を参照してください。

構文の要素	説明
<a href="#">backupOperand</a>	BACKUP コマンドの各種オプションを指定します。
<a href="#">backupSpec</a>	バックアップするオブジェクトを1つ以上指定します。  <i>backupSpec</i> 句ごとに、1つ以上のバックアップ・セット ( <b>AS BACKUPSET</b> ) またはイメージ・コピー ( <b>AS COPY</b> ) が生成されます。 <b>AS BACKUPSET</b> の場合は、オブジェクト・リストで指定されているデータファイルの数またはオブジェクト・リストで自動的に選択されたデータファイルの数が <b>FILESPERSET</b> の制限を超えていると、 <i>backupSpec</i> 句で複数のバックアップ・セットが生成されます。
PLUS ARCHIVELOG	アーカイブ REDO ログもバックアップの対象にします (2-49 ページの例 2-15 を参照)。Recovery Manager によって次の手順が実行されます。 <ol style="list-style-type: none"> <li>ALTER SYSTEM ARCHIVE LOG CURRENT 文が実行されます。</li> <li>BACKUP ARCHIVELOG ALL コマンドが実行されます。バックアップの最適化が有効になっている場合は、まだバックアップされていないログのみがバックアップされます。</li> <li>BACKUP コマンドで指定したファイルのバックアップが作成されます。</li> <li>ALTER SYSTEM ARCHIVE LOG CURRENT 文が実行されます。</li> <li>残りのアーカイブ REDO ログのバックアップが作成されます。バックアップの最適化が有効になっていない場合は、手順 1 で生成されたログの他に、バックアップ中に生成されたすべてのログもバックアップされます。</li> </ol> <p><b>注意:</b> PLUS ARCHIVELOG を BACKUP ARCHIVELOG コマンドで指定することはできません。また、BACKUP AS COPY INCREMENTAL コマンド (デフォルト・バックアップ・タイプが COPY の場合は BACKUP INCREMENTAL コマンド) で指定することもできません。INCREMENTAL FROM SCN を指定しているときに、PLUS ARCHIVELOG も指定することはできません。</p> <p><b>注意:</b> バックアップの最後にオンライン REDO ログがアーカイブされていない場合、そのバックアップに DUPLICATE は使用できません。</p>
<a href="#">backupSpecOperand</a>	<i>backupSpec</i> 句に影響する様々なオプションとパラメータを指定します。

### backupOperand

この副次句は、デバイス・タイプ、出力形式などのオプションを指定します。構文図は、2-23 ページの「[backupOperand::=](#)」を参照してください。

構文の要素	説明
<a href="#">backupTypeSpec</a>	作成するバックアップのタイプ (バックアップ・セット ( <b>AS BACKUPSET</b> ) またはイメージ・コピー ( <b>AS COPY</b> ) のいずれか) を指定します。  <b>関連項目:</b> 詳細は、2-43 ページの「 <a href="#">backupTypeSpec</a> 」を参照してください。
CHANNEL <i>channel_id</i>	バックアップの作成時に使用するチャンネルの名前を指定します。この名前には大 / 小文字の区別があります。たとえば ch1、dev1 のように、わかりやすい名前を付けてください。データベースでは、このチャンネル ID が I/O エラーのレポートに使用されます。このパラメータを設定しない場合、Recovery Manager は実行中に使用できるチャンネルに動的にバックアップ・セットを割り当てます。  2-52 ページの例 2-23 に示すように、CHANNEL を使用して、バックアップに使用するチャンネルとバックアップするファイルを指定できます。  <b>注意:</b> <i>backupSpec</i> 句でもこのパラメータを指定できます。

構文の要素	説明
CHECK LOGICAL	<p>物理的な破損チェックを通過したデータ・ブロックと索引ブロックについて、論理的な破損がないかどうかをテストします (2-52 ページの例 2-25 を参照)。</p> <p>たとえば、行ピースまたは索引エントリの論理的な破損がないかどうかを調べます。Recovery Manager は論理的な破損を見つけると、アラート・ログとサーバー・セッション・トレース・ファイルにそのブロックのログを書き込みます。SET MAXCORRUPT コマンドによって、データファイルに許容される物理的および論理的な破損の合計数が指定されます。</p> <p>デフォルトでは、BACKUP コマンドによって各ブロックのチェックサムが計算され、バックアップに格納されます。NOCHECKSUM オプションを指定すると、バックアップの書き込み時にブロックのチェックサムは実行されません。</p> <p>SET MAXCORRUPT および NOCHECKSUM が設定されていない場合、CHECK LOGICAL は、バックアップ時に検出される可能性があるすべてのタイプの破損を検出します。</p>
COPIES <i>integer</i>	<p>Recovery Manager で作成する同一バックアップの数 (1 ~ 4) を設定します。デフォルト値は 1 です。</p> <p>複数のフォーマット文字列を使用して、コピーに異なる名前と場所を指定できます。2-52 ページの例 2-22 に、ディスクの様々な場所に多重化されたバックアップを示します。</p> <p>Recovery Manager は、バックアップをディスクまたはテープのいずれかに多重化できますが、テープとディスクに同時に多重化することはできません。テープにバックアップを行う場合は、コピー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ます。また、COPIES が 2 以上の場合、ターゲット・データベースで BACKUP_TAPE_IO_SLAVES 初期化パラメータを有効にする必要があります。</p> <p>複数のコマンドを多重化するように指定できます。優先順位は次のとおりで、リストの上位にある設定で下位にある設定がオーバーライドされます。</p> <ol style="list-style-type: none"> <li>1. BACKUP COPIES</li> <li>2. SET BACKUP COPIES</li> <li>3. CONFIGURE ... BACKUP COPIES</li> </ol> <p><b>注意:</b> このオプションは、AS COPY では適用されないため、エラー・メッセージが戻されます。</p> <p><b>注意:</b> フラッシュ・リカバリ領域にファイルを作成する場合、多重化は使用できません。</p>
CUMULATIVE	<p>最新のレベル 0 バックアップ以降に使用されたデータ・ブロックをコピーします (2-49 ページの例 2-16 を参照)。</p> <p><b>注意:</b> このオプションは、AS COPY には適用されません。使用すると、エラー・メッセージが戻されます。</p>
DEVICE TYPE <i>deviceSpecifier</i>	<p>指定したデバイス・タイプ専用の自動チャンネルを割り当てます。たとえば、ディスクおよびテープ・チャンネルを構成してから、sbt をデフォルトのデバイス・タイプとして構成すると、次のコマンドではディスク・チャンネルのみが割り当てられます。</p> <pre>BACKUP DEVICE TYPE DISK DATABASE;</pre> <p>DEVICE TYPE オプションが有効なのは自動チャンネルに対してのみであり、手動で割り当てられたチャンネルには無効です。DISK 以外のデバイスについては、そのデバイスに対してまだ CONFIGURE DEVICE TYPE を実行していない場合、DEVICE TYPE オプションは使用できません。</p> <p><b>注意:</b> BACKUP RECOVERY AREA コマンドを実行する場合は、DEVICE TYPE DISK を指定できません。</p> <p><b>関連項目:</b> 3-15 ページの「<i>deviceSpecifier</i>」を参照してください。</p>



構文の要素	説明
DISKRATIO <i>integer</i>	<p>各バックアップ・セットに、<i>integer</i> で指定する台数以上のディスクからのデータファイルに移入するように Recovery Manager に指示します。</p> <p>このパラメータは、データファイルまたは制御ファイルのバックアップ時に、オペレーティング・システムから Recovery Manager にディスク競合情報およびノードのアフィニティ・データを送信可能な場合にのみ有効です。この機能を手動で無効にするには、DISKRATIO を 0 に設定します。</p> <p>たとえば、データファイルが 10 台のディスクに分散されるとします。データがディスクから毎秒 10 バイトで送信され、テープ・ドライブでストリームを維持するために毎秒 50 バイトが必要な場合は、DISKRATIO を 5 に設定して、各バックアップ・セットに 5 台以上のディスクからのデータファイルを含めるように Recovery Manager に指示します。</p> <p>FILESERSET <i>integer</i> を設定して、DISKRATIO を設定しない場合、DISKRATIO はデフォルトで FILESERSET と同じ値になります。いずれのパラメータも指定しない場合、DISKRATIO はデフォルトで 4 になります。Recovery Manager は、DISKRATIO の値を、バックアップに関連するデバイスの実数の数と比較して、小さいほうの値を使用します。たとえば、DISKRATIO が 4 で、データファイルが 3 台のディスクに格納されている場合、Recovery Manager は、各バックアップ・セットに 3 台のディスクからのデータファイルを含めようとはします。</p> <p>DISKRATIO パラメータは、データファイルがストライプ化されているか、または別々のディスク・スピンドルに格納されていて、次の条件のいずれかを満たす場合、データファイルのバックアップでより有効になります。</p> <ul style="list-style-type: none"> <li>■ テープ・ドライブのストリームを維持するために、いくつかのデータファイルを多重化する必要がある高帯域幅のテープ・ドライブを使用している場合。</li> <li>■ データベースのオープン中にバックアップを作成し、I/O の負荷をいくつかのディスク・スピンドルに分散して、オンライン操作の帯域幅を確保する必要がある場合。</li> </ul> <p><b>注意:</b> I/O は、テープ・ストリームの維持に必要なディスクの最小台数を越えて分散させないでください。必要以上に分散させた場合、パフォーマンスは向上せず、ファイルのリストア時間が増加します。</p>
<i>duration</i>	<p>バックアップ・コマンドの最長実行時間に関連するオプションを指定します。</p> <p><b>関連項目:</b> 2-46 ページの「<i>duration</i>」を参照してください。</p>
<i>fileNameConversionSpec</i>	<p>このオプションは、BACKUP を使用したイメージ・コピーの作成時にのみ有効です。作成されるファイルは、指定したパターンに従って名前を変更されます。バックアップされるファイルの名前が、指定した名前変更パターンのいずれにも一致しない場合、Recovery Manager は、<b>FORMAT formatSpec</b> を使用して出力イメージ・コピーに名前を付けます。FORMAT が指定されていない場合は、デフォルトのフォーマットの %U が使用されます。</p> <p><b>関連項目:</b> ファイルの名前変更パターンについては、3-16 ページの「<i>fileNameConversionSpec</i>」を参照してください。</p>

構文の要素	説明
FILESERSET <i>integer</i>	<p>各出力バックアップ・セットに含める入力ファイルの最大数を指定します。このパラメータは、BACKUP でバックアップ・セットを生成する場合にのみ関係します。</p> <p>各 <i>backupSpec</i> のファイルは、1 つ以上のバックアップ・セットとしてバックアップされます。各 <i>backupSpec</i> のファイル数が、FILESERSET の設定値より大きい場合は、その制限に合うように、複数のバックアップ・セットにファイルが分割されます。FILESERSET のデフォルト値は 64 です。</p> <p>次の BACKUP コマンドで、Recovery Manager の動作を説明します。</p> <pre>BACKUP AS BACKUPSET (DATAFILE 3, 4, 5, 6, 7) (DATAFILE 8, 9); BACKUP AS BACKUPSET DATAFILE 3, 4, 5, 6, 7, 8, 9; BACKUP AS BACKUPSET DATAFILE 3, ... 72;</pre> <p>最初のコマンドでは、データファイル 3、4、5、6 および 7 が 1 つのバックアップ・セットに入れられ、データファイル 8 および 9 がもう 1 つのバックアップ・セットに入れられます。2 番目のコマンドでは、すべてのデータファイルが 1 つのバックアップ・セットに入れられます。3 番目のコマンドの場合、省略記号はデータファイル 3 ~ 72 を表します。この場合は、70 のデータファイルをバックアップすることになるため、64 ファイルが 1 つのバックアップ・セットに入れられ、6 ファイルがもう 1 つのバックアップ・セットに入れられます。</p> <p>デフォルトでは、チャンネル・リソースを最適に使用するために、Recovery Manager によって、ファイルがバックアップ・セットに分割されます。バックアップされるファイルの数が、チャンネル数で除算されます。その結果が 64 未満の場合は、その値が FILESERSET 値になります。それ以外の場合、FILESERSET はデフォルトで 64 になります。</p> <p><b>注意:</b> バックアップ・セット内のバックアップ・ピースの数は指定できません。</p>
FORCE	<p>Recovery Manager にバックアップの最適化を無視させます。つまり、CONFIGURE BACKUP OPTIMIZATION の設定が ON になっている場合でも、指定したすべてのファイルと、アクティブなトランザクションの UNDO データがバックアップされます。</p> <p><b>注意:</b> <i>backupSpecOperand</i> 句でもこのオプションを指定できます。</p>
AUXILIARY FORMAT	<p>ターゲット・データベース上のファイルを、補助インスタンス上の指定された場所にコピーします。AUXILIARY FORMAT を指定した場合は、イメージ・コピーのみを生成できます。Recovery Manager は、TARGET と AUXILIARY の両方のインスタンスに接続し、補助チャンネルにアクセスする必要があります。</p> <p>BACKUP AUXILIARY FORMAT コマンドを使用すると、プライマリ・データベースとスタンバイ・データベースとの間で、データファイルをネットワーク経由でコピーできます。たとえば、プライマリ・データベース上のデータファイルが消失した場合は、TARGET としてスタンバイ・データベースに接続し、AUXILIARY としてプライマリ・データベースに接続して、完全な状態のデータファイルをスタンバイ・ホストからプライマリ・ホストにコピーできます。</p> <p>補助インスタンス上の出力イメージのコピーに名前を付けるパターンを指定します。このパスは、補助ホスト上で有効である必要があります。</p> <p><b>関連項目:</b> 有効な置換変数については、3-21 ページの「<i>formatSpec</i>」を参照してください。</p>
<i>formatSpec</i>	
NEW	<p>補助インスタンスの DB_CREATE_FILE_DEST 初期化パラメータで指定したディレクトリに、イメージ・コピーを作成します。イメージ・コピーは、Oracle 管理ファイルです。</p>

## 構文の要素

## 説明

FORMAT *formatSpec*

出力バックアップ・ピースまたはイメージ・コピーに名前を付けるパターンを指定します (2-50 ページの例 2-17 を参照)。AS COPY の場合、指定した形式で設定されたディレクトリが 1 つ以上存在しないと、Recovery Manager によってエラーが発行されます。

ディスク・バックアップのデフォルトの場所は、フラッシュ・リカバリ領域が有効かどうかと、FORMAT が指定されているかどうかによって決まります。

- フラッシュ・リカバリ領域が有効で、FORMAT を指定した場合は、FORMAT の設定に従って出力ファイルに名前が付けられます。FORMAT で場所が指定されていない場合は、(リカバリ領域にではなく) プラットフォームに固有な場所にバックアップが作成されます。
- フラッシュ・リカバリ領域が有効で、FORMAT を指定しなかった場合は、リカバリ領域にバックアップが作成され、置換変数 %U を使用してバックアップに名前が付けられます。
- フラッシュ・リカバリ領域が有効でなく、FORMAT を指定しなかった場合は、プラットフォーム固有の場所にバックアップが作成され、%U を使用してバックアップに名前が付けられます。

Recovery Manager バックアップを、Oracle Managed Files 形式の名前でフラッシュ・リカバリ領域に作成するには、BACKUP コマンドまたはチャネルで FORMAT 句を指定しないでください。

**注意:** Oracle Managed Files のファイル名は、バックアップ用の形式として指定できません。たとえば、+DISK1/datafile/system.732.609791431 が OMF ファイル名の場合、そのファイル名を FORMAT パラメータに指定することはできません。

バックアップ・ピースにはそれぞれ一意の名前を付ける必要があります。バックアップ・ピースのファイル名の最大長はプラットフォームによって異なります。メディア・マネージャへのバックアップの場合は、サポートされている Media Management API のバージョンの制限によっても長さが制限されます。SBT 1.1 をサポートしているベンダーは、14 文字までのファイル名をサポートしている必要があります。SBT 1.1 のベンダーによってはさらに長いファイル名をサポートしている場合もあります。SBT 2.0 をサポートしているベンダーは、512 文字までのファイル名をサポートする必要があります。SBT 2.0 のベンダーによってはさらに長いファイル名をサポートしている場合もあります。

1 つの *backupSpec* に、同じ FORMAT 文字列を複数回指定することはできません (たとえば、BACKUP DATAFILE 3 TO '/tmp/df3.f', DATAFILE 4 TO '/tmp/df4.f' のような指定はできません)。ただし、Recovery Manager では、複数の *backupSpec* 句に単一の FORMAT 文字列を指定できます。

**注意:** KEEP オプションを指定して、アーカイブ・バックアップを作成する (2-53 ページの例 2-26 を参照) 場合は、フォーマット文字列に %U が含まれている必要があります。このフォーマット文字列は、自動バックアップでも使用されます。

**関連項目:** 有効な置換変数については、3-21 ページの「*formatSpec*」を参照してください。

*forRecoveryOfSpec*

イメージ・コピーのロールフォワード時に使用する増分バックアップとして作成するバックアップを識別します。

**関連項目:** 2-47 ページの「*forRecoveryOfSpec*」を参照してください。

FULL

バックアップに含まれているデータファイルのすべてのブロックのバックアップを作成します。FULL は、INCREMENTAL の逆です。FULL または INCREMENTAL を指定しなければ、Recovery Manager ではデフォルトで全体バックアップが実行されます。

全体バックアップは、その後の増分バックアップに影響せず、増分バックアップ計画の一部とはみなされません。ただし、イメージ・コピーの全体バックアップについては、RECOVER コマンドで増分バックアップを適用して、増分更新が可能です。

**注意:** 未使用ブロックの圧縮 (BACKUP AS BACKUPSET の項を参照) を行うと、全体バックアップの際に一部のデータファイル・ブロックがスキップされます。

## 構文の要素

## 説明

INCREMENTAL LEVEL *integer*

最後の増分 *integer* バックアップ以降に変更されたデータ・ブロックのみをコピーします。ここで、*integer* は 0 または 1 です (2-49 ページの例 2-16 を参照)。

INCREMENTAL を指定する場合は、`DATAFILE datafileSpec`、`DATAFILECOPY`、`TABLESPACE tablespace_name`、`DATABASE` のいずれかのパラメータを `backupSpec` 句で指定する必要があります。Recovery Manager は、制御ファイル、アーカイブ REDO ログまたはバックアップ・セットの増分バックアップをサポートしません。

**注意:** NOARCHIVELOG データベースがオープンされて使用中の場合は、増分バックアップを作成できません (ただし、データベースが一貫性のある停止の後でマウントされている場合には、増分バックアップを作成できます)。

レベル 0 の増分バックアップでは、バックアップ対象のデータファイルのすべてのデータ・ブロックがバックアップされます。レベル 0 の増分バックアップの内容は FULL バックアップと同じですが、全体バックアップとは異なり、増分バックアップ方法の一部分とみなされます。

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バックアップされます。レベル 1 の増分バックアップは、差分または CUMULATIVE のいずれかです。CUMULATIVE の場合は、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がバックアップされます。差分の場合は、最新のレベル 0 またはレベル 1 の増分バックアップ以降に更新されたブロックがバックアップされます。スタンバイ・データベースのレベル 1 バックアップは、プライマリ・データベースのレベル 0 バックアップに適用できます。また、プライマリ・データベースのレベル 1 バックアップは、スタンバイ・データベースのレベル 0 バックアップに適用できます。

レベル 0 の増分バックアップは、バックアップ・セットまたはイメージ・コピーのいずれかにできますが、レベル 1 の増分バックアップは、バックアップ・セットのみが可能です。

レベル 1 の増分バックアップを作成しようとすると、データベースでチェックが実行されます。このチェックによって、増分バックアップがその後の RECOVER コマンドで使用できることが確認されます。チェックの内容は、次のとおりです。

- レベル 0 バックアップが、BACKUP コマンド内のデータファイルごとに、増分方法の基本バックアップとして存在していること。レベル 0 バックアップのステータスは UNAVAILABLE でないことが必要です。レベル 0 バックアップが存在しない場合は、レベル 0 バックアップが自動的に作成されます。
- レベル 0 以降の十分な増分バックアップがあり、これから作成する増分バックアップで使用できること。

**注意:** 増分バックアップの作成時、Recovery Manager では、親インカネーションからのバックアップが有効であるとみなされます。たとえば、レベル 0 バックアップを作成した後、OPEN RESETLOGS を実行するとします。レベル 1 の増分バックアップを作成すると、Recovery Manager によって、RESETLOGS より前のレベル 0 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。レベル 1 バックアップを作成する場合は、Recovery Manager によって、現行のデータベース・インカネーションまたは親データベース・インカネーションでレベル 0 が使用できない場合のみ、新しいレベル 0 バックアップが作成されます。

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** を有効にすると、増分バックアップのパフォーマンスを向上できます。この場合、Recovery Manager では、変更されたブロックがブロック・チェンジ・トラッキング・ファイルに記録されます。

チェンジ・トラッキング・ファイルは、データファイルの変更をマークするビットマップをバックアップ間で保持します。データベースでは、各バックアップを行う前にビットマップの切替えが行われます。Oracle Database では、最新の 8 回のバックアップを網羅するブロック・チェンジ・データが保持されるように、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。ビットマップが 8 個まで作成されると、最新のビットマップが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。

## 構文の要素

## 説明

最初のレベル 0 の増分バックアップでは、データファイル全体がスキャンされます。その後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の後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。増分バックアップの最適化は、ブロック・チェンジ・トラッキング・ファイルの最も古いビットマップ以降に作成された親バックアップに基づいてのみ行われます。

増分バックアップを計画するときは、ビットマップの制限が 8 個であることに注意してください。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、差分増分バックアップを 7 回実行すると、ブロック・チェンジ・トラッキング・ファイルには 8 個のビットマップが含まれます。次に、累積レベル 1 の増分バックアップを作成すると、現在の変更を追跡するビットマップによって親（レベル 0）のバックアップに対応するビットマップが上書きされるため、Recovery Manager はバックアップを最適化できなくなります。

**関連項目：**ブロック・チェンジ・トラッキングの詳細は、『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』を参照してください。

INCREMENTAL FROM SCN  
integer

指定した SCN 以上の SCN で変更されたすべてのデータファイル・ブロックが含まれている、指定したすべてのデータファイルの増分バックアップを作成します。

このオプションを使用するのは、プライマリ・データベースの変更に基づいてスタンバイ・データベースをリフレッシュするような場合です（2-52 ページの例 2-24 および『Oracle Data Guard 概要および管理』の Recovery Manager バックアップに関する章を参照）。このバックアップには、スタンバイ・データベースが作成された時点または最後に同期されたとき以降に変更されたすべてのブロックが含まれます。スタンバイ・データベースでは、NOREDO を指定して RECOVER を使用し、増分バックアップを適用できます。増分バックアップに取得されたすべての変更ブロックが、スタンバイ・データベースに適用され、プライマリ・データベースによって最新の状態になります。

ボリューム・シャドウ・コピー・サービス（VSS）のスナップショットに基づいて増分バックアップを作成していない場合は、INCREMENTAL FROM SCN を指定するときに *formatSpec* を指定する必要があります。Recovery Manager は制御ファイルのバックアップを作成するため、FORMAT 文字列には %U などの置換変数を含める必要があります。

Windows 環境では、NOKEEP オプションを使用して FROM SCN を指定する場合、および INCREMENTAL FROM SCN を指定するときに *formatSpec* を指定しない場合は、VSS スナップショットに基づいて増分バックアップを作成できるように、Recovery Manager によってフラッシュ・リカバリ領域に増分バックアップが作成されます。その場合、増分バックアップ・セットと VSS のシャドウ・コピーを同時に使用できます。FROM SCN パラメータに指定されたチェックポイント SCN 値は、VSS バックアップ・メタデータ・ドキュメントの BACKUP\_CHECKPOINT 値と同じである必要があります。ブロック・チェンジ・トラッキングが有効な場合、バックアップにチェンジ・トラッキングのメカニズムが使用され、増分バックアップの作成にかかる時間が大幅に短縮されます。Recovery Manager は、リカバリ時に、フラッシュ・リカバリ領域の増分バックアップを透過的に適用できます。

**注意：**INCREMENTAL FROM SCN を指定しているときに、PLUS ARCHIVELOG も使用することはできません。

**関連項目：**VSS を使用してバックアップを作成する方法については、『Oracle Database Platform Guide for Microsoft Windows』を参照してください。

構文の要素	説明
<i>keepOption</i>	<p>バックアップが不要とみなされないように、そのバックアップについて構成されている保存方針をオーバーライドします (2-53 ページの例 2-26 を参照)。</p> <p>KEEP 構文を使用すると、ビジネス要件または法的要件を満たすアーカイブ・データベース・バックアップを生成できます。KEEP の設定は、バックアップ・セット (個々のバックアップ・ピースの属性ではありません) またはイメージ・コピーの属性です。</p> <p><b>注意:</b> KEEP は BACKUP BACKUPSET と併用できません。</p> <p>KEEP 構文を使用すると、バックアップを、指定時間後に不要とみなされるようにしたり (KEEP UNTIL)、不要にならないようにすることができます (KEEP FOREVER)。KEEP FOREVER を指定する場合は、2-53 ページの例 2-27 に示すように、リカバリ・カタログに接続しておく必要があります。</p> <p><b>注意:</b> CHANGE を使用すると、KEEP を指定して生成されたバックアップのステータスを変更できます。</p> <p><b>関連項目:</b> KEEP オプションを指定して作成されるバックアップの詳細は、3-24 ページの「<i>keepOption</i>」を参照してください。</p>
MAXSETSIZE <i>sizeSpec</i>	<p>バックアップ・セットの最大サイズを指定します (2-50 ページの例 2-17 を参照)。すべてのバックアップ・セットは、このサイズに制限されます。</p> <p>バックアップ・セットは複数のテープにわたって作成可能なため、各データファイルのブロックは複数のテープに書き込まれる場合があります。複数ボリュームのバックアップ・セットの 1 つのテープにエラーが発生した場合、すべてのテープのデータを失うこととなります。バックアップ・セットには、必ず、ファイルの一部ではなく 1 つのファイル全体が含まれるため、MAXSETSIZE を使用して、各バックアップ・セットを 1 つのテープに収まるように指定することができます。</p> <p>サイズはバイト単位 (デフォルト)、KB 単位 (K)、MB 単位 (M) または GB 単位 (G) で指定します。たとえば、バックアップ・セットを 3MB に制限するには、MAXSETSIZE 3M と指定します。デフォルト・サイズはバイト単位で、下位の KB 数に丸められます。たとえば、MAXSETSIZE 3000 であれば、2KB (2048 バイト) に丸められます。最小値には、データベースのブロック・サイズ以上の値を指定する必要があります。</p> <p>各バックアップ・セット内のデフォルトのファイル数は、FILESERSET によって決定されます。デフォルトは 64 です。MAXSETSIZE を指定すると、Recovery Manager は、MAXSETSIZE パラメータに従ってバックアップ・セットのサイズをバイト単位で制限します。バックアップ・セット内のファイル数の制限は、結果のバックアップ・セットの合計サイズが MAXSETSIZE より少ない場合でも適用されません。</p> <p><b>注意:</b> このオプションを BACKUP AS COPY で使用すると、エラー・メッセージが戻されます。MAXSETSIZE が設定されているチャンネルに対して BACKUP AS COPY を実行すると、MAXSETSIZE は暗黙的に無視されます。</p>
<i>notBackedUpSpec</i>	<p>指定した数のバックアップがすでに存在している (かつ不要になっていない) かどうか、または指定した日付以降にログがバックアップされているかどうかによって、バックアップするアーカイブ REDO ログ・ファイルのセットを制限します。</p> <p><b>関連項目:</b> 2-48 ページの「<i>notBackedUpSpec</i>」を参照してください。</p>

構文の要素	説明
NOCHECKSUM	<p>バックアップ時にブロックに対するチェックサムを抑止します。</p> <p>チェックサムとは、データ・ブロックの内容によって計算した数字のことです。DB_BLOCK_CHECKSUM は、データファイルのブロックのチェックサムを、(バックアップではなく) データベースに書き込むかどうかを制御する、データベースの初期化パラメータです。DB_BLOCK_CHECKSUM が typical の場合、データベースでは、通常の操作中に各ブロックのチェックサムが計算され、計算結果がブロックに格納されてから、ブロックがディスクに書き込まれます。その後、データベースによって、ディスクからブロックが読み取られ、チェックサムが再計算されて格納されている値と比較されます。これらの値が一致しない場合は、ブロックが破損しています。</p> <p><b>注意:</b> SYSTEM 表領域のデータファイルに対するチェックサムは、DB_BLOCK_CHECKSUM=false の場合でも無効にできません。</p> <p>デフォルトでは、BACKUP コマンドによって各ブロックのチェックサムが計算され、バックアップに格納されます。DB_BLOCK_CHECKSUM が適用されるのは、バックアップではなくデータベースのデータファイルであるため、この初期化パラメータの値は、BACKUP コマンドでは無視されます。NOCHECKSUM オプションを指定すると、バックアップの書き込み時にブロックのチェックサムは実行されません。</p> <p>バックアップ・データファイルのリストア時には、DB_BLOCK_CHECKSUM 初期化パラメータの設定が考慮されます。DB_BLOCK_CHECKSUM が false に設定されている場合は、チェックサムが消去されます。typical に設定されている場合は、バックアップからリストアしてデータファイルに書き込む際に、チェックサムが検証されます。</p> <p><b>注意:</b> チェックサムのチェックは NOCHECKSUM を指定して無効にできますが、他の物理的な整合性チェック (ブロックのヘッダーとフッターのチェックなど) は無効にできません。</p> <p><b>関連項目:</b> DB_BLOCK_CHECKSUM 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。</p>
NOEXCLUDE	<p>このパラメータを BACKUP DATABASE または BACKUP COPY OF DATABASE コマンドで指定すると、CONFIGURE EXCLUDE コマンドで指定されているものも含め、すべての表領域がバックアップされます。このオプションにより、SKIP OFFLINE または SKIP READONLY がオーバーライドされることはありません。</p>
POOL integer	<p>バックアップを格納するメディア・プールを指定します。POOL がサポートされているかどうかは、メディア管理ソフトウェアのドキュメントで確認してください。</p> <p><b>注意:</b> このオプションは、AS COPY では機能しません。使用すると、エラー・メッセージが戻されます。</p>
PROXY	<p>プロキシ・コピー機能を使用して、指定したファイルをバックアップします。これによって、メディア管理ソフトウェアはストレージ・デバイスとディスク上のデータファイルとの間のデータ転送を制御できます。メディア・マネージャ (Recovery Manager ではなく) がデータ移動の方法と時期を決めます。</p> <p>PROXY オプションを指定して BACKUP を実行すると、Recovery Manager では次の手順が実行されます。</p> <ol style="list-style-type: none"> <li>1. 指定したデバイス・タイプでプロキシが可能なチャンネルを検索します。このタイプのチャンネルが見つからない場合、Recovery Manager は警告を表示し、指定したファイルの従来型 (非プロキシ) バックアップを試みます。</li> <li>2. Recovery Manager は、プロキシ可能なチャンネルが見つかった場合、メディア・マネージャをコールして、ファイルのプロキシ・コピーが可能かどうかをチェックします。メディア・マネージャがプロキシ・コピーを行うことができない場合、Recovery Manager は従来のバックアップ・セットを使用してファイルをバックアップします。</li> </ol> <p><b>注意:</b> PROXY を指定した場合は、%p 変数を FORMAT 文字列内の %U に明示的または暗黙的に含める必要があります。</p> <p><b>注意:</b> このオプションは、AS COPY では機能しません。使用すると、エラー・メッセージが戻されます。</p>

構文の要素	説明
ONLY	<p>プロキシ・コピーを実行できない場合は、従来のバックアップ・セットを作成しないで、データベースからエラー・メッセージを発行します。プロキシ・コピーで障害が発生したときに Recovery Manager で従来型コピーを試行しない場合は、ONLY オプションを使用します。</p>
REUSE	<p>Recovery Manager で、BACKUP によって現在作成されているファイルと同じファイル名を持つ既存のバックアップまたはコピーを上書きできます。</p>
SECTION SIZE <i>sizeSpec</i>	<p>データファイルまたはデータファイル・コピーのバックアップ時に作成される各バックアップ・セクションのサイズを指定します。</p> <p>このパラメータを設定すると、Recovery Manager でマルチセクション・バックアップを作成できます。マルチセクション・バックアップの場合は、ファイル・セクション（ファイル内の連続したブロック範囲）を1つ含むバックアップ・ピースが作成されます。マルチセクション・バックアップのセクションは、すべて同じサイズになります。</p> <p>ファイル・セクションを使用すると、複数のステップで、1つの大きなデータファイルのバックアップを処理できます。Recovery Manager のチャンネルは、各ステップを個々にパラレルで処理することが可能で、各チャンネルではマルチセクション・バックアップ・セットの1つのセクションが生成されます。</p> <p>ファイルのサイズより大きいセクション・サイズを指定すると、そのファイルに対してはマルチセクション・バックアップは使用されません。256より多くのセクションが作成される小さなセクション・サイズを指定すると、Recovery Manager は、セクション・サイズをちょうど 256 セクションが作成されるような値に増やします。</p> <p><b>注意：</b>このパラメータを Recovery Manager 構文のどこに指定するかによって、同じバックアップ・ジョブの中でも、ファイルごとに異なるセクション・サイズを指定できます。</p> <p><b>注意：</b>SECTION SIZE と MAXPIECESIZE を一緒に使用することはできません。</p>
<i>skipSpec</i>	<p>データファイルまたはアーカイブ REDO ログがアクセス不能、オフラインまたは読み取り専用である場合はバックアップから除外します。</p> <p><b>関連項目：</b>詳細は、2-49 ページの「<i>skipSpec</i>」を参照してください。</p>



構文の要素	説明
TAG <i>tag_name</i>	<p>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・コピーに対してユーザー指定のタグ名を指定します。タグは、BACKUP コマンドによって生成された出力ファイルに適用されます。</p> <p>タグ名には、大 / 小文字の区別はありません。名前は 30 文字以下にしてください。使用する文字は、ターゲット・ファイル・システムのファイル名に使用できる有効な文字に限定されています。たとえば、ASM が内部的に使用するファイル名では、ハイフン (-) 文字がサポートされていません。このため、<code>weekly-incremental</code> は、ASM ディスク・グループのバックアップに対するタグ名としては有効ではありません。</p> <p>一般的に、タグ名は <code>MON_PM_BKUP</code>、<code>WEEKLY_FULL_BKUP</code> などのように、わかりやすい名前にします。タグは再使用できます。たとえば、ある週にはバックアップ・セット 100 が <code>MON_PM_BKUP</code> のタグを使用し、翌週にはバックアップ・セット 105 が同じタグを使用できます。</p> <p>タグ名を指定しない場合、デフォルトでは、バックアップ用のタグが作成されます (制御ファイルの自動バックアップを除く)。デフォルトのタグは、<code>TAGYYYYMMDDTHHMMSS</code> 書式を使用します。ここで、<code>YYYY</code> は年、<code>MM</code> は月、<code>DD</code> は日、<code>HH</code> は時 (24 時間書式)、<code>MM</code> は分および <code>SS</code> は秒です。たとえば、データファイル 1 のバックアップのタグは、<code>TAG20070208T133437</code> となります。日付と時刻は、Recovery Manager によるバックアップの開始日時を指します。1 つの <code>BACKUP AS BACKUPSET</code> 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は、各バックアップ・ピースに同じデフォルト・タグが割り当てられます。</p> <p><code>backupSpec</code> レベルでもタグを指定できます。どのレベルでタグを指定するかによって、次のようになります。</p> <ul style="list-style-type: none"> <li>■ コマンド・レベルで指定した場合は、このコマンドによって作成されるすべてのバックアップ・セットに、このタグが与えられます。</li> <li>■ <code>backupSpec</code> レベルで指定した場合は、異なるバックアップ指定で作成されたバックアップ・セットに、それぞれ異なるタグが与えられます。</li> <li>■ 両方のレベルに指定すると、<code>backupSpec</code> のタグが優先されます。</li> </ul> <p><b>注意:</b> タグは、バックアップ・セット (<code>AS BACKUPSET</code> の場合) の特定のコピーの各バックアップ・ピースの属性、または各イメージ・コピー (<code>AS COPY</code> の場合) の属性です。たとえば、<code>BACKUP AS BACKUPSET COPIES 1 DATABASE TAG TUE_PM</code> を実行した場合、存在するバックアップ・セットは 1 つのみで、それぞれのバックアップ・ピースのタグは <code>TUE_PM</code> になります。このバックアップ・セットの主キーが 1234 であるとします。<code>BACKUP BACKUPSET 1234 TAG WED_PM</code> を実行すると、バックアップ・セットの最初のコピーのタグは <code>TUE_PM</code>、2 番目のコピーのタグは <code>WED_PM</code> となります。</p>
VALIDATE	<p>指定されたファイルをスキャンして内容を検査し、そのファイルがバックアップ可能かどうか、およびデータ・ブロックが破損していないかどうかをテストします。</p> <p>出力ファイルは作成されません。このオプションは、バックアップで指定されたデータベース・ファイルに対して <code>VALIDATE</code> コマンドを使用するのと同じです。</p> <p><code>CHECK LOGICAL</code> が指定されていない場合、<code>BACKUP VALIDATE</code> は物理的な破損のみをチェックします。<code>CHECK LOGICAL</code> が指定されている場合は、物理的な破損と論理的な破損の両方が <code>BACKUP VALIDATE</code> によってチェックされます。破損が見つかった場合は、その情報が <code>V\$DATABASE_BLOCK_CORRUPTION</code> ビューに移入されます。</p> <p><code>SET MAXCORRUPT</code> コマンドを使用して、バックアップの検証中に許容可能な破損ブロック数に制限を設定できます。デフォルトは 0 (ゼロ) です。</p> <p><code>VALIDATE</code> を指定して <code>BACKUP INCREMENTAL</code> を実行する場合、ブロック・チェンジ・トラッキングが有効かどうかによって動作が異なります。チェンジ・トラッキングが有効な場合は、変更されたブロックのみが検証されます。無効な場合は、バックアップに含まれるファイルのすべてのブロックが検証されます。</p> <p><b>注意:</b> バックアップ・セットのバックアップは検証できません。</p>

**backupSpec**

この副次句は、バックアップの対象とする 1 つ以上のオブジェクトのリストを指定します。backupSpec 句ごとに、1 つ以上のバックアップ・セット (**AS BACKUPSET**) またはイメージ・コピー (**AS COPY**) が生成されます。AS BACKUPSET では、オブジェクト・リストで指定したか自動的に選択されたデータファイルの数が、各バックアップ・セットでデフォルトの制限の 4 個のデータファイルまたは 16 個のアーカイブ・ログを超えている場合は、*backupSpec* 句で複数のバックアップ・セットが作成されます。構文図は、2-24 ページの「**backupSpec::=**」を参照してください。

**構文の要素****説明***archivelogRecordSpecifier*

バックアップ対象となるアーカイブ REDO ログの範囲を指定します。

アーカイブ REDO ログのバックアップ作成時に、Recovery Manager でアーカイブ・ログのフェイルオーバーを自動的に実行できます。Recovery Manager は、指定されたログ順序番号およびスレッドに対応する 1 つ以上のアーカイブ・ログが使用可能な場合に、ログのバックアップを作成します。また、Recovery Manager がバックアップ中のコピーに破損ブロックが含まれている場合は、同じアーカイブ・ログの他のコピー内で該当ブロックの正常なコピーが検索されます。このコマンドでバックアップ対象のログが見つからなくても、Recovery Manager はエラーを発行しません。この状況になるのは、前回の BACKUP ARCHIVELOG ALL **DELETE INPUT** コマンド以降に新規ログが生成されていないためです。

BACKUP ARCHIVELOG ALL を指定すると、Recovery Manager は個々のログ順序番号ごとに単一コピーのバックアップのみを作成します。たとえば、複数のアーカイブ先にアーカイブする場合、Recovery Manager は、各ログ順序番号のそれぞれのアーカイブ・コピーではなく、1 つのコピーをバックアップします。**DELETE** など、他のコマンドの場合、ALL はログ順序が重複する場合にも各ログを参照します。

BACKUP ARCHIVELOG の実行時にデータベースがオープンしていて UNTIL 句または SEQUENCE パラメータが指定されていない場合、Recovery Manager は ALTER SYSTEM ARCHIVE LOG CURRENT を実行します。

**注意：**BACKUP ARCHIVELOG ALL を実行する場合、または指定したログ範囲に以前のインカネーションからのログが含まれている場合、Recovery Manager は、以前のインカネーションからのログをバックアップして、OPEN RESETLOGS によるリカバリに必要なすべてのログの可用性を確認します。

**関連項目：**構文については 3-6 ページの「*archivelogRecordSpecifier*」を、ログ・バックアップ・フェイルオーバーと自動的なログ・スイッチについては『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』を参照してください。

構文の要素	説明
BACKUPSET	<p>バックアップ・セットのバックアップを指定します。このパラメータを <b>DEVICE TYPE deviceSpecifier sbt</b> 句と併用すると、ディスク上のバックアップをテープにオフロードできます (例 2-21 を参照)。テープ間またはテープからディスクへはバックアップを作成できず、ディスク間またはディスクからテープへのバックアップのみが可能です。</p> <p>BACKUP BACKUPSET コマンドで <b>DELETE INPUT</b> オプションを指定すると、ディスクに存在するバックアップ・セットのコピーがすべて削除されます。たとえば、単一バックアップを 4 つの位置に多重化した場合、<b>Recovery Manager</b> では 4 つのバックアップ・セットがすべて削除されます。ALL オプションで追加される機能はありません。</p> <p><b>Recovery Manager</b> では、バックアップ・セットのバックアップ時に、<b>バックアップ・セット・フェイルオーバー</b> が実行されます。バックアップ対象となるコピーが破損または欠落している場合は、使用可能なバックアップ・コピーがすべて検索されます。この動作は、複数のアーカイブ先に存在しているアーカイブ・ログのバックアップを作成する場合の、<b>Recovery Manager</b> の動作と同じです。</p> <p>バックアップ・セットのバックアップ時に<b>バックアップの最適化</b>が有効な場合、同じバックアップ・セットが同じデバイス・タイプにすでにバックアップされていると、そのバックアップ・セットのバックアップはスキップされます。</p> <p><b>注意</b> : BACKUP COPIES および <b>SET BACKUP COPIES</b> を使用すると、バックアップ・セットのバックアップを多重化できます。</p> <p><b>注意</b> : 暗号化されたバックアップ・セットに対して BACKUP BACKUPSET コマンドを使用すると、バックアップ・セットはその暗号化形式でバックアップされます。BACKUP BACKUPSET は、暗号化されたバックアップ・セットを単にディスクまたはテープにコピーするのみであるため、BACKUP BACKUPSET 操作に暗号化鍵は不要です。この操作のどの段階でも、データの復号化が行われることはありません。バックアップ・セットは、BACKUP BACKUPSET コマンドによって暗号化されることも復号化されることもありません。</p>
ALL	すべてのバックアップ・セットを指定します。
<i>completedTimeSpec</i>	<p>完了時刻に基づいてバックアップ・セットを指定します。</p> <p><b>関連項目</b> : 3-10 ページの「<i>completedTimeSpec</i>」を参照してください。</p>
<i>integer</i>	主キーに基づいてバックアップ・セットを指定します。バックアップ・セットの主キーは、 <b>LIST BACKUP</b> コマンドの出力から得られます。
CONTROLFILECOPY	<p>バックアップのための制御ファイル・コピーを 1 つ以上指定します。</p> <p>制御ファイルのコピーは、BACKUP AS COPY CURRENT CONTROLFILE コマンドまたは SQL 文 ALTER DATABASE BACKUP CONTROLFILE TO '...' で作成できます。</p> <p><b>注意</b> : 制御ファイルの自動バックアップは、制御ファイルのコピーではありません。</p>
'filename'	ファイル名で制御ファイルのコピーを指定します。
ALL	制御ファイルのすべてのコピーを指定します。
LIKE 'string_pattern'	ファイル名のパターンで制御ファイルのコピーを指定します。パーセント記号 (%) は 0 文字以上を示すワイルド・カードで、アンダースコア ( ) は 1 文字を示すワイルド・カードです。
<i>copyOfSpec</i>	<p>データファイルの前のイメージ・コピーのバックアップを作成します (制御ファイルに対しても行われる場合があります)。</p> <p><b>関連項目</b> : 2-45 ページの「<i>copyOfSpec</i>」を参照してください。</p>
CURRENT CONTROLFILE	<p>現行の制御ファイルを指定します。</p> <p><b>注意</b> : 現行の制御ファイルのバックアップにタグを割り当てることはできません。</p>

構文の要素	説明
DATABASE	<p>データベース内のすべてのデータファイルのバックアップを作成します。バックアップ・セットを生成すると、Recovery Manager にデータファイルおよび制御ファイルのみを含めることができます。アーカイブ REDO ログを含めることはできません。</p> <p><code>backupSpec</code> にデータファイル 1 が含まれている場合に <code>CONFIGURE CONTROLFILE AUTOBACKUP</code> を OFF に設定すると、制御ファイルがバックアップに自動的に組み込まれます。インスタンスの起動にサーバー・パラメータ・ファイルが使用される場合は、このパラメータ・ファイルもバックアップに組み込まれます。</p> <p><code>backupSpec</code> にデータファイル 1 が含まれている場合に <code>CONFIGURE CONTROLFILE AUTOBACKUP</code> を ON に設定すると、制御ファイルは出力に自動的に組み込まれません。かわりに、別個の制御ファイルの自動バックアップ・ピースが生成されます。インスタンスの起動にサーバー・パラメータ・ファイルが使用される場合は、このパラメータ・ファイルが自動バックアップ・ピースに組み込まれます。</p> <p>データベースの全体バックアップは、通常、イメージ・コピーまたは圧縮されたバックアップ・セットのいずれかである必要があります。イメージ・コピーは、作成時に発生する CPU のオーバーヘッドが許容範囲内である場合、いくつかの用途（増分更新バックアップ計画での使用など）でバックアップ・セットより高い柔軟性を示します。また、圧縮されたバックアップ・セットによってストレージをより有効に使用できます。</p> <p><b>注意：</b> <code>CONTROLFILE AUTOBACKUP</code> が ON のときに、現行の制御ファイルを強制的にバックアップに組み込むには、<code>INCLUDE CURRENT CONTROLFILE</code> 句を指定します。</p> <p><b>関連項目：</b> データベースに <code>BIGFILE</code> の表領域が含まれる場合のバックアップ動作については、<code>TABLESPACE tablespace_name</code> の説明を参照してください。</p>
<code>datafileCopySpec</code>	<p>1 つ以上のデータファイル・イメージ・コピーのファイル名を指定します。</p> <p><b>関連項目：</b> 詳細は、2-46 ページの「<code>datafileCopySpec</code>」を参照してください。</p>
DATAFILE <code>datafileSpec</code>	<p>1 つ以上のデータファイルのリストを指定します。データファイル 1 をバックアップするときの Recovery Manager の動作については、BACKUP DATABASE の説明を参照してください。</p> <p><b>関連項目：</b> 3-14 ページの「<code>datafileSpec</code>」を参照してください。</p>
RECOVERY AREA	<p>現行および前回のすべてのフラッシュ・リカバリ領域の指定先に作成されたリカバリ・ファイルをバックアップします。バックアップは、SBT に作成する必要があります。</p> <p>リカバリ・ファイルには、全体および増分のバックアップ・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 ログが含まれます。アーカイブ REDO ログ・ファイルが存在しないか破損している場合、Recovery Manager はバックアップに使用できるログの正常なコピーが、リカバリ領域の外にないかどうかを確認します。フラッシュバック・ログ、現行の制御ファイルおよびオンライン REDO ログはバックアップされません。</p> <p><code>CONFIGURE BACKUP OPTIMIZATION</code> の設定が OFF の場合でも、このコマンドでのバックアップの最適化は、デフォルトで有効になっています。FORCE を指定すると、BACKUP RECOVERY AREA に対するバックアップの最適化を無効にできません。</p> <p><b>注意：</b> フラッシュ・リカバリ領域が現在有効でなくても、以前に有効にされていたことがある場合は、前回のフラッシュ・リカバリ領域の場所に作成されたファイルがバックアップされます。</p>

構文の要素	説明
DB_RECOVERY_FILE_DEST	RECOVERY AREA と DB_RECOVERY_FILE_DEST はシノニムです。
RECOVERY FILES	<p>ディスク上のすべてのリカバリ・ファイルを、フラッシュ・リカバリ領域に格納されているか、ディスク上の別の場所に格納されているかに関係なくバックアップします。バックアップは、SBT に作成する必要があります。</p> <p>リカバリ・ファイルには、全体および増分のバックアップ・セット、制御ファイルの自動バックアップ、アーカイブ REDO ログおよびデータファイルのコピーが含まれます。</p> <p><b>CONFIGURE BACKUP OPTIMIZATION</b> の設定が OFF の場合でも、このコマンドでのバックアップの最適化は、デフォルトで有効になっています。FORCE を指定すると、RECOVERY FILES に対するバックアップの最適化を無効にできます。</p>
SPFILE	<p>サーバー・パラメータ・ファイルをバックアップ・セットに入れるように指定します。サーバー・パラメータ・ファイルのバックアップでは、AS COPY オプションはサポートされていません。</p> <p>Recovery Manager は、ターゲット・データベースで使用中のサーバー・パラメータ・ファイルをバックアップします。サーバー・パラメータ・ファイルは、インスタンスが初期化パラメータ・ファイルによって起動された場合にはバックアップされません。SPFILE の増分バックアップは作成できません。</p>
TABLESPACE <i>tablespace_name</i>	<p>1 つ以上の表領域の名前を指定します。Recovery Manager は、その内部で表領域名をデータファイルのリストに変換してから、表領域を現在構成しているデータファイルをすべてバックアップします。SYSTEM 表領域（およびデータファイル 1）がバックアップに含まれていて <b>CONTROLFILE AUTOBACKUP</b> が設定されていない場合は、制御ファイルのコピーも作成されます。</p> <p>ローカル管理の一時表領域のバックアップを作成することはできません（ディクショナリ管理表領域のバックアップは作成できます）。</p> <p>次の条件が満たされる場合は、トランスポート後に読み書き両用に設定されなかったトランスポート表領域についてもバックアップを作成できます。</p> <ul style="list-style-type: none"> <li>■ COMPATIBLE 初期化パラメータが 11.0.0 以上に設定されている。</li> <li>■ Oracle Database 11g Recovery Manager クライアントを使用している。</li> </ul> <p>前述の条件のいずれかが満たされない場合、Recovery Manager は、読み書き両用に設定されていないトランスポート表領域を自動的にスキップします。条件のいずれかが満たされない場合にトランスポート表領域を明示的に指定すると、表領域が存在しないというエラーが Recovery Manager によって発行されます。</p> <p><b>注意：</b>ユーザーが表領域の名前を変更すると、その変更が Recovery Manager によって検出され、次の再同期化時にリカバリ・カタログが更新されます。</p>
<i>backupSpecOperand</i>	<i>backupSpec</i> の後に続く <i>backupSpecOperand</i> は、 <i>backupSpec</i> に適用されるオプションを指定します。

**backupSpecOperand**

この副次句は、[backupSpec](#) 句に影響する様々なオプションとパラメータを指定します。また、多くの副次句は [backupOperand](#) でも使用されます。ここでは、[backupOperand](#) では通常共有されないオプションを示します。構文図は、2-24 ページの「[backupSpecOperand::=](#)」を参照してください。

構文の要素	説明
DELETE [ALL] INPUT	<p>バックアップが正常に実行された後で、入力ファイルを削除します。</p> <p>このオプションを指定できるのは、アーカイブ REDO ログ、データファイルのコピー（COPY OF または DATAFILECOPY）またはバックアップ・セットのバックアップを作成するときのみです。BACKUP ARCHIVELOG コマンドの場合にバックアップされるのは、個々のログ順序番号ごとに1つのコピーのみです。したがって、DELETE INPUT オプションを指定しても、ALL キーワードを付けなければ、Recovery Manager はバックアップしたファイルのコピーのみを削除します。</p> <p>DELETE INPUT オプションを指定すると、入力ファイルに対して DELETE コマンドを発行するのと同じ効果があります。アーカイブ REDO ログをバックアップする場合、Recovery Manager は、構成済の設定（<a href="#">CONFIGURE ARCHIVELOG DELETION POLICY TO BACKED UP</a>）を使用して、アーカイブ REDO ログが削除可能かどうかを判断します。</p> <p>ALL オプションはアーカイブ REDO ログにのみ適用されます。DELETE ALL INPUT を実行すると、対応するアーカイブ REDO ログまたはデータファイルのコピーのうち、BACKUP コマンドの選択条件と一致するもののコピーがすべて削除されます（2-50 ページの <a href="#">例 2-19</a> を参照）。たとえば、SEQUENCE <i>n</i> 句を指定すると、Recovery Manager は同じ順序番号 <i>n</i> を持つアーカイブ REDO ログをすべて削除します。</p>
FROM TAG 'tag_name'	<p>タグ名でファイルを指定します（2-50 ページの <a href="#">例 2-18</a> を参照）。他のいくつかのコマンドとの関係で定義されます。</p>
INCLUDE CURRENT CONTROLFILE	<p>現行の制御ファイルのスナップショットを作成し、BACKUP コマンドで作成されるいずれかのバックアップ・セットに組み込みます。</p> <p><b>注意：</b>このオプションは、AS COPY では適用されないため、エラー・メッセージが戻されます。</p>

**backupTypeSpec**

この副次句は、BACKUP コマンドの出力形式（バックアップ・セットまたはイメージ・コピー）を指定します。構文図は、2-25 ページの「**backupTypeSpec::=**」を参照してください。

**構文の要素****説明**

AS BACKUPSET

指定されたデバイス上にバックアップ・セットを作成します。これがデフォルトのバックアップ・タイプです。

テープにバックアップする場合およびレベル 1 の増分バックアップを任意のバックアップ先に作成する場合に使用できるのは、AS BACKUPSET のみです。バックアップ・セットは、Recovery Manager に固有の論理構造です。バックアップ・セットは、バックアップの最小単位です。

BACKUP コマンドの **FILESERSET integer** パラメータは、各バックアップ・セット内のファイルの最大数を決定します。アーカイブ・ログとデータファイルは、1 つのバックアップ・セットに組み合わせて入れられることはありません。

暗号化されたバックアップを使用している場合は（2-20 ページの「**バックアップ・セットの暗号化**」を参照）、異なる暗号化設定が使用されている表領域からのデータファイルは、同じバックアップ・セットに書き込むことができません。

ブロック・サイズの異なるファイルのバックアップを同じバックアップ・セット内に作成することはできません。Recovery Manager では、ブロック・サイズの異なる表領域のバックアップを作成できますが、それぞれ異なるサイズのデータファイルが専用バックアップ・セットに入れられます。

Recovery Manager のバックアップ・セットでは、**未使用ブロックの圧縮**が自動的に使用されます。可能な場合に未使用のデータ・ブロックをスキップすると、Recovery Manager で、データファイルのバックアップにより少ない領域を使用し、より効率的に I/O を行うことができます。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場合、割り当てられたことがないデータ・ブロックはその対象になりません。次のすべての条件に該当する場合は、現在データが含まれていないデータファイル・ブロックもスキップされます。

- COMPATIBLE 初期化パラメータが 10.2 以上に設定されている。

COMPATIBLE が 10.2 の場合は、10.2 互換で作成された表領域のみが、現在データを含まないブロックを除外するように最適化されることに注意してください。ただし、COMPATIBLE が 11.0.0 以上の場合は、COMPATIBLE が 11.0.0 以上に設定された後でバックアップ・セットを作成する最初のバックアップが、すべてのローカル管理データファイルを最適化できるように、それらのデータファイルのヘッダーを更新します。

- 保証付きリストア・ポイントがデータベースに対して現在定義されていない。
- データベースがローカル管理されている。
- データファイルが、全体バックアップの一部分またはレベル 0 の増分バックアップとしてバックアップ・セットにバックアップされている。
- バックアップ・セットがディスクに作成されているか、メディア・マネージャが Oracle Secure Backup である。つまり、Oracle Secure Backup 以外のメディア・マネージャにバックアップしている場合、Recovery Manager は、現在データを含まないデータファイル・ブロックをスキップしません。

各バックアップ・セットには、1 つ以上の**バックアップ・ピース**が含まれます。これは、バックアップ対象のデータを含む Recovery Manager 固有の物理ファイルです。BACKUP コマンドを使用して**プロキシ・コピー**を作成することもできます。プロキシ・コピーは、データ転送全体がメディア・マネージャによって実行されるバックアップです。

Recovery Manager では、完全バックアップ・セットのみが Recovery Manager リポジトリに記録されます。部分バックアップ・セットは記録されません。BACKUP コマンドによってバックアップ・ピースが作成されても、完全なバックアップ・セットは生成されなかった場合、作成されたバックアップ・ピースは破棄されます。

**注意：**1 つのバックアップ・セットを複数のチャネルに分散することはできません。また、1 つの入力ファイルを複数のバックアップ・セットに分散することもできません。

**関連項目：**Recovery Manager で Oracle Secure Backup を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。

構文の要素	説明
COMPRESSED	<p>バイナリ圧縮を有効にします。</p> <p>Recovery Manager は、バックアップ・セットに書き込まれたデータを圧縮するため、バックアップ・セット全体のサイズは小さくなります。バックアップ・セットを作成するすべてのバックアップで、圧縮されたバックアップ・セットを作成できます。圧縮されたバックアップ・セットのリストア方法と圧縮されていないバックアップ・セットのリストア方法に違いはありません。</p> <p>Recovery Manager は、データをバックアップ・セットに書き込む際にバイナリ圧縮アルゴリズムを適用します。この圧縮方法は、多くのメディア・マネージャ・ベンダーが提供している圧縮方法に類似しています。ローカル接続されたテープ・デバイスにバックアップする場合は、通常、メディア管理ベンダーが提供する圧縮のほうが BACKUP AS COMPRESSED BACKUPSET で実行されるバイナリ圧縮より適しています。そのため、ローカル接続されたテープ・デバイスにバックアップする場合は、圧縮されていないバックアップ・セットを使用し、メディア管理ベンダーが提供する圧縮をオンにします。Recovery Manager のバイナリ圧縮とメディア・マネージャの圧縮を同時に使用しないでください。</p> <p>バックアップ・セットを圧縮する場合、ある程度の CPU オーバーヘッドが発生します。ターゲット・データベースが最大またはそれに近い負荷で実行されていると、このオーバーヘッドが許容できないほど大きくなる場合があります。他のほとんどの環境では、バックアップ・セットを圧縮すると、CPU のオーバーヘッドに値するだけのディスク領域が確保されます。</p>
AS COPY	<p>(バックアップ・セットではなく) イメージ・コピーを作成します。</p> <p>イメージ・コピーは、元のファイルのバイト単位の同一コピーです。データファイル、データファイルのコピーおよびアーカイブ REDO ログ・ファイルのイメージ・コピー・バックアップを作成できます。イメージ・コピー・ファイルは、ディスクにのみ格納できます。増分更新バックアップを使用している場合、レベル 0 の増分はイメージ・コピー・バックアップである必要があります。</p> <p>デフォルトでは、BACKUP はバックアップ・セットを生成します。CONFIGURE DEVICE TYPE ...BACKUP TYPE TO COPY コマンドを使用すると、ディスク・バックアップでのデフォルトのバックアップ・タイプをイメージ・コピーに変更できます。</p> <p>Recovery Manager では、次の規則に従ってコピーの作成場所を選択します (優先順位の高い順に示しています)。</p> <ol style="list-style-type: none"> <li>1. バックアップするオブジェクトに対する BACKUP コマンドで指定された FORMAT</li> <li>2. BACKUP コマンドに指定された FORMAT</li> <li>3. BACKUP コマンドの <i>fileNameConversionSpec</i> 設定</li> <li>4. CONFIGURE CHANNEL <i>integer</i> ...FORMAT</li> <li>5. CONFIGURE CHANNEL DEVICE TYPE ...FORMAT</li> <li>6. プラットフォーム固有のデフォルトの FORMAT (一意のファイル名を生成する %U が含まれる)</li> </ol> <p>イメージ・コピー・バックアップの作成およびリストアは、Recovery Manager を使用するか、またはファイルをコピーするためのオペレーティング・システム固有のコマンドを使用して実行できます。Recovery Manager を使用する場合は、コピーが Recovery Manager リポジトリに記録され、リストアおよびリカバリで簡単に使用できます。Recovery Manager を使用しない場合は、CATALOG コマンドでユーザー管理コピーを Recovery Manager リポジトリに追加して、Recovery Manager で使用できるようにする必要があります。</p> <p>イメージ・コピーのイメージ・コピーは作成できますが、バックアップ・セットのコピーは作成できません。バックアップ・セットのバックアップを作成するには、BACKUP BACKUPSET コマンドを実行します。</p>



**copyOfSpec**

この副次句は、BACKUP コマンドの出力形式（バックアップ・セットまたはイメージ・コピー）を指定します。構文図は、2-25 ページの「copyOfSpec::=」を参照してください。

構文の要素	説明
COPY OF DATABASE	<p>データベース内のすべてのデータファイルおよび制御ファイルの前のイメージ・コピーのバックアップを作成します。BACKUP DATABASE によって通常組み込まれるすべてのデータファイルにコピーが含まれている必要があります。含まれていない場合は、Recovery Manager によってエラーが発行されます。すべてのコピーが 1 回の BACKUP コマンドで作成されている必要はありません。データファイルの複数のコピーが存在する場合は、Recovery Manager によって最新のコピーがバックアップされます。オプションで、タグ名（FULL_COLD_COPY など）でコピーを指定します。</p> <p><b>注意:</b> このコマンドの出力は、イメージ・コピーまたはバックアップ・セットにできます。</p>
COPY OF DATAFILE <i>datafileSpec</i>	<p>1 つ以上のデータファイルの前のイメージ・コピーのバックアップを作成します。ファイル番号 (DATAFILE 3) またはファイル名 (DATAFILE '*/oradata/trgt/users01.dbf') でデータファイルを指定します。データファイルのコピーのファイル名ではなく、データファイルのファイル名を指定します。指定したデータファイルの複数のコピーが存在する場合は、Recovery Manager によって最新のコピーがバックアップされます。</p> <p><b>注意:</b> バックアップ対象のイメージ・コピーが 1 回の BACKUP コマンドで作成されている必要はありません。</p> <p><b>注意:</b> このコマンドの出力は、イメージ・コピーまたはバックアップ・セットにできます。</p> <p><b>関連項目:</b> 3-14 ページの「<i>datafileSpec</i>」を参照してください。</p>
COPY OF TABLESPACE <i>tablespace_name</i>	<p>指定した 1 つ以上の表領域内のデータファイルの前のイメージ・コピーのバックアップを作成します。BACKUP TABLESPACE <i>tablespace_name</i> によって通常組み込まれるすべてのデータファイルにコピーが含まれている必要があります。含まれていない場合は、Recovery Manager によってエラーが発行されます。すべてのコピーが 1 回の BACKUP コマンドで作成されている必要はありません。データファイルの複数のコピーが存在する場合は、Recovery Manager によって最新のコピーがバックアップされます。</p> <p>表領域名 (users など) でリストに表領域を指定するか、またはタグ名 (0403_CPY_OF_USERS など) で特定のコピーを指定します。TAG を指定しない場合は、Recovery Manager によって、表領域内の各データファイルの最新のデータファイルのコピーがバックアップされます。</p> <p><b>注意:</b> このコマンドの出力は、イメージ・コピーまたはバックアップ・セットにできます。</p>

**datafileCopySpec**

この副次句は、データファイルのコピーを指定します。構文図は、2-26 ページの「[datafileCopySpec::=](#)」を参照してください。

構文の要素	説明
'filename'	1つ以上のデータファイル・イメージ・コピーのファイル名を指定します。
ALL	データファイルのすべてのイメージ・コピーをバックアップするように指定します。
LIKE 'string_pattern'	ファイル名のパターンを指定します。パーセント記号 (%) は 0 文字以上を示すワイルド・カードで、アンダースコア ( _ ) は 1 文字を示すワイルド・カードです。
FROM TAG tag_name	データファイルの 1 つ以上のコピーのリストをタグ名で指定します。このタグの付いたデータファイルのコピーが複数存在している場合、 <b>Recovery Manager</b> は個々のデータファイルの最新のデータファイルのみをバックアップします。タグには、大 / 小文字区別はありません。
NODUPLICATES	バックアップ操作で同一のデータファイルのコピーが組み込まれないようにします (2-54 ページの例 2-29 を参照)。データファイルのコピーが重複している場合は、最新のタイムスタンプを持つファイルが選択されます。

**duration**

この副次句は、データファイルのコピーを指定します。構文図は、2-26 ページの「[duration::=](#)」を参照してください。

構文の要素	説明
DURATION hh:mm	バックアップ・コマンドの最長実行時間を指定します。指定した実行時間内にバックアップ・コマンドが完了しなかった場合、バックアップは停止します。  PARTIAL オプションを指定しないと、バックアップが指定した実行時間で完了しなかった場合、バックアップ・コマンドが正常に実行されなかったとみなされ、 <b>Recovery Manager</b> によってエラーがレポートされます。バックアップ・コマンドが RUN ブロックの一部である場合、その RUN ブロック内の後続コマンドは実行されません。
MINIMIZE {LOAD TIME}	ディスク・バックアップでは、MINIMIZE TIME を使用してバックアップを最大速度で実行 (デフォルト) したり、システムの負荷を軽減するために MINIMIZE LOAD を使用してバックアップの速度を遅くできます。MINIMIZE LOAD を指定すると、バックアップに指定した実行時間が最大限に使用されます。  TIME を指定すると、最も新しくバックアップされたファイルに対するバックアップの優先順位が最も低くなります。このスケジュール・メカニズムによって、異なるデータファイルがラウンドロビン法でバックアップされるため、一連のバックアップ期間内で最終的に完全なデータベースのバックアップが行われます。
PARTIAL	PARTIAL オプションを指定すると、バックアップ全体が指定した実行時間で完了しなかった場合でも、コマンドが正常に完了したと判断され、 <b>Recovery Manager</b> によってエラーはレポートされません。  PARTIAL オプションを指定しないと、バックアップが指定した実行時間で完了しなかった場合、バックアップ・コマンドが正常に実行されなかったとみなされ、 <b>Recovery Manager</b> によってエラーがレポートされます。バックアップ・コマンドが RUN ブロックの一部である場合、その RUN ブロック内の後続コマンドは実行されません。  PARTIAL を指定しているかどうかに関係なく、バックアップが中断される前に完了したすべてのバックアップ・セットは保存され、RESTORE および RECOVER の操作で使用できます。

**forRecoveryOfSpec**

この副次句は、バックアップを増分更新バックアップ計画で使用することを指定します。FOR RECOVER OF を指定する際に、INCREMENTAL LEVEL 1 を指定する必要があります。構文図は、2-26 ページの「forRecoveryOfSpec:=」を参照してください。

構文の要素	説明
FOR RECOVER OF COPY	<p>ターゲット・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以降に行われたすべての変更を増分バックアップに含めるように指定します（2-51 ページの例 2-20 を参照）。</p> <p>BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY コマンドを使用すると、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。増分レベル0のイメージ・コピーがすでに存在する場合はバックアップ・セットが作成されますが、存在しない場合はイメージ・コピーが作成されます。</p> <p>データファイルのコピーは、DATAFILE COPY または WITH TAG 句のいずれかで指定して、このバックアップで使用する増分バックアップ計画と、残りのバックアップ計画とを区別する必要があります。データファイルのコピーを指定しない場合は、各データファイルの最新コピーが増分バックアップのベースとして使用されます。</p> <p><b>関連項目：</b>増分更新バックアップの作成方法は、『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』を参照してください。</p>
WITH TAG <i>tag_name</i>	<p>増分バックアップの基礎として使用されるレベル0の増分バックアップのタグを指定します。</p> <p>WITH TAG パラメータで指定したタグを持つレベル0バックアップが、現行または親のデータベース・インカネーションで検出されなかった場合は、WITH TAG パラメータで指定したタグを持つレベル0のデータファイル・コピーが FOR RECOVER OF COPY オプションによって作成されます。</p> <p>BACKUP INCREMENTAL ... FOR RECOVER OF COPY WITH TAG 構文を使用して、データベースのレベル0の増分イメージ・コピー・バックアップのロールフォワードに適したレベル1の増分バックアップを作成できます。次に、RECOVER COPY OF ... WITH TAG を使用して、バックアップの増分更新を実行できます。この方法は、Enterprise Manager のディスクへの推奨バックアップで使用されません。</p>
DATAFILECOPY	出力イメージ・コピーに名前を付けるパターンを指定します。
FORMAT <i>formatSpec</i>	
FOR RECOVER OF TAG <i>tag_name</i>	<p><i>tag_name</i> で指定したレベル0の増分バックアップをリカバリするためのアーカイブ REDO ログ・ファイルまたは増分バックアップをバックアップします。</p> <p>たとえば、BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF TAG wholedb ARCHIVELOG ALL コマンドによって、wholedb というタグが付いているレベル0の増分バックアップのリカバリに必要なアーカイブ REDO ログ・ファイルをすべてバックアップできます。</p>

**notBackedUpSpec**

この副次句は、まだ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ように指定します。構文図は、2-26 ページの「notBackedUpSpec::=」を参照してください。

構文の要素	説明
NOT BACKED UP	<p>BACKUP コマンドで指定されたファイルの中で、バックアップが実行されていないファイルのみをバックアップします (2-53 ページの例 2-28 を参照)。</p> <p>この副次句を使用すると、データベースに新規に追加したデータファイルを簡単にバックアップできます。Recovery Manager は、データファイル・チェックポイントを検査せずに、まだバックアップされていないデータファイルをすべてバックアップすることに注意してください。バックアップ・セットをバックアップするときに、NOT BACKED UP を指定することもできます。</p> <p>BACKUP をこの句とともに使用すると、アーカイブ REDO ログの削除方針で自動的に実行されることを手動で実行することになります。CONFIGURE ARCHIVELOG DELETION POLICY TO BACKED UP integer TIMES コマンドを指定すると、指定したデバイス・タイプに integer 回のバックアップが存在しないかぎり、BACKUP ARCHIVELOG ALL コマンドによってすべてのログがコピーされます。ログの integer 回のバックアップが存在する場合は、BACKUP コマンドはログをスキップします。このように、アーカイブ・ログの削除方針は、BACKUP コマンドのデフォルトの NOT BACKED UP integer TIMES 句として機能します。</p> <p><b>注意:</b> この句は、バックアップの最適化 (CONFIGURE BACKUP OPTIMIZATION) およびアーカイブ・ログの削除方針 (CONFIGURE ARCHIVELOG DELETION POLICY) をオーバーライドします。</p>
integer TIMES	<p>integer 回以上バックアップされていないアーカイブ REDO ログ・ファイルのみをバックアップします。</p> <p><b>注意:</b> アーカイブ REDO ログ・ファイルをバックアップ・セットにバックアップしている場合のみ、NOT BACKED UP integer TIMES を指定できます。</p> <p>ファイルのバックアップ数を判断する場合は、現行のバックアップと同じデバイス・タイプで作成されたバックアップのみが考慮されます。</p> <p>このオプションは、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便利です。たとえば、各ログの3つ以上のコピーをテープに保存できます。</p>
SINCE TIME 'date_string'	<p>その日付を過ぎると Recovery Manager が、バックアップされていないファイルのバックアップを開始する日付を指定します。</p> <p>date_string は、現行の NLS_DATE_FORMAT の日付、または 'SYSDATE-1' などの SQL DATE 式です。ファイルのバックアップ数の計算では、現行のバックアップと同じデバイス・タイプで作成されたバックアップのみが考慮されます。</p> <p>このオプションを使用すると、前回失敗したバックアップ中に処理されなかったデータファイルをバックアップできます。たとえば、データベースをバックアップしても、インスタンスの途中で障害が発生する場合があります。その場合は、NOT BACKED UP SINCE TIME 句を使用してバックアップを再開し、バックアップ済のファイルを対象から除外できます。AS BACKUPSET を指定すると、この機能が役立つのは、Recovery Manager がバックアップ中に複数のバックアップ・セットを生成する場合のみです。</p> <p>ファイルがバックアップされたかどうかを判断するときには、SINCE 日付が最新バックアップの完了時刻と比較されます。BACKUP AS BACKUPSET の場合、バックアップ・セット内のファイルの完了時刻は、バックアップ・セット全体の完了時刻です。つまり、同じバックアップ・セット内のすべてのファイルの完了時刻は同じです。</p>

**sizeSpec**

この副次句は、データのサイズを指定します。構文図は、2-26 ページの「[sizeSpec::=](#)」を参照してください。

**構文の要素****説明**

構文の要素	説明
<code>integer [G   K   M]</code>	データのサイズを GB 単位 (G)、KB 単位 (K) または MB 単位 (M) で指定します。

**skipSpec**

この副次句は、スキップするファイルを指定します。構文図は、2-26 ページの「[skipSpec::=](#)」を参照してください。

**構文の要素****説明**

構文の要素	説明
SKIP	次のキーワードで指定する基準に従って、データファイルまたはアーカイブ REDO ログをバックアップから除外します。 <b>注意:</b> <code>backupSpec</code> 句でもこのオプションを指定できます。
INACCESSIBLE	I/O エラーのために読み取ることができないデータファイルまたはアーカイブ REDO ログをバックアップに含めないように指定します。 データファイルは、読み取りが不可能な場合にのみアクセス不能とみなされます。一部のオフライン・データファイルは、ディスク上に残っているために読み取りが可能です。他のデータファイルは削除または移動されたためにアクセス不可となり、読み取り不可となります。
OFFLINE	オフライン・データファイルをバックアップに含めないように指定します。
READONLY	読み取り専用データファイルをバックアップに含めないように指定します。

**例****例 2-15 データベースのバックアップ**

この例では、オペレーティング・システム・コマンドラインから Recovery Manager クライアントを起動した後、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。BACKUP コマンドによって、すべてのデータファイル、現行の制御ファイル、サーバー・パラメータ・ファイルおよびアーカイブ REDO ログ・ファイルがデフォルトのストレージ・デバイスにバックアップされます。

```
% rman
RMAN> CONNECT TARGET /
RMAN> BACKUP DATABASE PLUS ARCHIVELOG;
```

**例 2-16 累積増分バックアップの実行**

この例では、最後に実行されたレベル 0 の増分バックアップ以降にデータベース上で変更されたすべてのブロックをバックアップします。レベル 1 バックアップの実行時にレベル 0 バックアップが存在しない場合は、レベル 0 バックアップが自動的に作成されます。アクセスできないファイルは、スキップされます。

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

### 例 2-17 複数ディスクへのバックアップの分散

この例では、2つの異なるディスクに表領域をバックアップし、Recovery Manager にバックアップの平行化を自動的に実行させます。FORMAT 文字列の %U は、出力するイメージ・コピーごとに一意のファイル名を生成する置換変数です。

```
RUN
{
  ALLOCATE CHANNEL dev1 DEVICE TYPE DISK FORMAT '/disk1/%U';
  ALLOCATE CHANNEL dev2 DEVICE TYPE DISK FORMAT '/disk2/%U';
  BACKUP AS COPY
    TABLESPACE SYSTEM, tools, users, undotbs;
}
```

### 例 2-18 タグによるデータファイル・コピーの指定

この例では、データファイルのイメージ・コピーをテープにバックアップします。この BACKUP コマンドは、LATESTCOPY というタグが付いているすべてのデータファイル・コピーを検出してテープにバックアップし、そのバックアップに置換変数を使用した名前を付けます。変数 %f は、絶対ファイル番号を指定します。また、%d は、データベースの名前を指定します。データファイルのコピーをテープ上に作成したら、LATESTCOPY というタグが付いているすべてのイメージ・コピーが削除されます。

```
BACKUP
  DEVICE TYPE sbt
  DATAFILECOPY
    FROM TAG 'LATESTCOPY'
  FORMAT 'Datafile%f_Database%d';
DELETE COPY TAG 'LATESTCOPY';
```

### 例 2-19 アーカイブ REDO ログのバックアップと削除

この例では、2つのアーカイブ先 (/disk2/PROD/archivelog/ および /disk1/arch/) が設定されているとします。このコマンドは、一意の順序番号ごとにアーカイブ REDO ログを1つバックアップします。たとえば、アーカイブ REDO ログ 1000 が両方のディレクトリにある場合、Recovery Manager は、このログの1つのコピーのみをバックアップします。ALL キーワードが指定された DELETE INPUT 句によって、バックアップの終了後に、両方のアーカイブ・ディレクトリからすべてのアーカイブ REDO ログを削除するように指示しています。

```
BACKUP DEVICE TYPE sbt
  ARCHIVELOG LIKE '/disk%arc%'
  DELETE ALL INPUT;
```

このコマンドでは、次のような出力が表示されます。

```
Starting backup at 12-MAR-07
allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=150 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: Oracle Secure Backup
channel ORA_SBT_TAPE_1: starting archived log backup set
channel ORA_SBT_TAPE_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=4 RECID=4 STAMP=616789551
input archived log thread=1 sequence=5 RECID=5 STAMP=616789551
input archived log thread=1 sequence=6 RECID=6 STAMP=616789554
input archived log thread=1 sequence=7 RECID=7 STAMP=616789731
input archived log thread=1 sequence=8 RECID=8 STAMP=616789825
input archived log thread=1 sequence=9 RECID=10 STAMP=616789901
input archived log thread=1 sequence=10 RECID=12 STAMP=616789985
channel ORA_SBT_TAPE_1: starting piece 1 at 12-MAR-07
channel ORA_SBT_TAPE_1: finished piece 1 at 12-MAR-07
piece handle=0vice0g7_1_1 tag=TAG20070312T105917 comment=API Version 2.0,MMS Version 10.1.0.3
channel ORA_SBT_TAPE_1: backup set complete, elapsed time: 00:00:25
channel ORA_SBT_TAPE_1: deleting archived log(s)
```

```

archived log file name=/disk2/PROD/archivelog/2007_03_09/o1_mf_1_4_2z45sgrc_.arc RECID=4
STAMP=616789551
archived log file name=/disk2/PROD/archivelog/2007_03_09/o1_mf_1_5_2z45sgrc_.arc RECID=5
STAMP=616789551
archived log file name=/disk2/PROD/archivelog/2007_03_09/o1_mf_1_6_2z45sl3g_.arc RECID=6
STAMP=616789554
archived log file name=/disk2/PROD/archivelog/2007_03_09/o1_mf_1_7_2z45z2kt_.arc RECID=7
STAMP=616789731
archived log file name=/disk2/PROD/archivelog/2007_03_09/o1_mf_1_8_2z4620sk_.arc RECID=8
STAMP=616789825
archived log file name=/disk1/arch/archiver_1_8_616789153.arc RECID=9 STAMP=616789825
archived log file name=/disk2/PROD/archivelog/2007_03_09/o1_mf_1_9_2z464dhk_.arc RECID=10
STAMP=616789901
archived log file name=/disk1/arch/archiver_1_9_616789153.arc RECID=11 STAMP=616789901
archived log file name=/disk2/PROD/archivelog/2007_03_09/o1_mf_1_10_2z4670gr_.arc RECID=12
STAMP=616789985
archived log file name=/disk1/arch/archiver_1_10_616789153.arc RECID=13 STAMP=616789985
Finished backup at 12-MAR-07

```

```

Starting Control File and SPFILE Autobackup at 12-MAR-07
piece handle=c-28643857-20070312-02 comment=API Version 2.0,MMS Version 10.1.0.3
Finished Control File and SPFILE Autobackup at 12-MAR-07

```

## 例 2-20 増分更新バックアップのスキプト作成

バックアップを増分更新することによって、データベースのイメージ・コピーの全体バックアップに伴うオーバーヘッドを避けると同時に、データベースのメディア・リカバリに必要な時間を最小限にすることもできます。たとえば、日次バックアップ用のスキプトを実行する場合に、メディア・リカバリに適用する REDO を 1 日分より多く持つことはありません。

次のスキプトを毎日実行するとします。初回実行時には、スキプトによって、指定したタグを使用してディスク上にデータベースのイメージ・コピーのバックアップが作成されます。2 回目の実行では、データベースのレベル 1 のバックアップが作成されます。以降のすべての実行では、レベル 1 の増分がデータファイルのコピーに適用され、新しいレベル 1 のバックアップが作成されます。

```

RUN
{
  RECOVER COPY OF DATABASE
    WITH TAG 'incr_update';
  BACKUP
    INCREMENTAL LEVEL 1
    FOR RECOVER OF COPY WITH TAG 'incr_update'
    DATABASE;
}

```

## 例 2-21 テープへのディスクベースのバックアップ・セットのバックアップ

最近のバックアップ・セットをディスク上に保持し、古いバックアップ・セットをテープ上に置くことを目標とします。また、同じバックアップ・セットのコピーを、ディスクとテープに同時に保持することは避けるものとします。この例では、2 週間より前に作成されたバックアップ・セットはテープにバックアップされ、そのバックアップ・ピースがディスクから削除されます。

```

BACKUP
  DEVICE TYPE sbt
  BACKUPSET
  COMPLETED BEFORE 'SYSDATE-14'
  DELETE INPUT;

```

### 例 2-22 データベース・バックアップの多重化

この例では、**COPIES integer** パラメータを使用して、圧縮されたデータベース・バックアップを2つ（別々のディスクに1つずつ）作成します。出力場所は、**FORMAT** パラメータで指定します。

```
BACKUP AS COMPRESSED BACKUPSET
  DEVICE TYPE DISK
  COPIES 2
  DATABASE
  FORMAT '/disk1/db_%U', '/disk2/db_%U';
```

### 例 2-23 チャンネルでのワークロードの分割方法の指定

この例では、**CHANNEL** パラメータで、どのチャンネルでどのファイルをどこにバックアップするかを指定し、バックアップを明示的にパラレル化します。

```
RUN
{
  ALLOCATE CHANNEL ch1 DEVICE TYPE sbt
    PARMS 'ENV=(OB_DEVICE_1=stape1)';
  ALLOCATE CHANNEL ch2 DEVICE TYPE sbt
    PARMS 'ENV=(OB_DEVICE_1=stape2)';
  BACKUP
    (DATABASE          # channel ch1 backs up database to tape drive stape1
     CHANNEL ch1)
    (ARCHIVELOG ALL
     CHANNEL ch2); # channel ch2 backs up archived redo logs to tape drive stape2
}
```

### 例 2-24 スタンバイ・データベースのリフレッシュ用の増分バックアップの作成

この例では、プライマリ・データベースの増分バックアップを作成し、それを使用して関連付けられたスタンバイ・データベースを更新することを目標とします。**Recovery Manager** クライアントを起動し、**CONNECT TARGET** としてプライマリ・データベースに接続してから、リカバリ・カタログに接続します。**BACKUP** コマンドでは、スタンバイ・データベースで適用可能なプライマリ・データベースの増分バックアップを作成し、指定した SCN 以降の変更を反映して更新します。

```
RMAN> CONNECT TARGET /

connected to target database: PROD (DBID=39525561)

RMAN> CONNECT CATALOG rman@catdb

recovery catalog database Password: password
connected to recovery catalog database

RMAN> BACKUP DEVICE TYPE DISK
2> INCREMENTAL FROM SCN 404128 DATABASE
3> FORMAT '/disk1/incr_standby_%U';
```

### 例 2-25 データファイル・バックアップの破損許容度の指定

この例では、データベースに5つのデータファイルが含まれているとします。**SET MAXCORRUPT** コマンドを使用して、各データファイル内に1つの破損も許容されないことを指定します。**BACKUP** コマンドで**CHECK LOGICAL** オプションが指定されているため、**Recovery Manager** は、物理的な破損と論理的な破損の両方をチェックします。

```
RUN
{
  SET MAXCORRUPT FOR DATAFILE 1,2,3,4,5 TO 1;
  BACKUP CHECK LOGICAL
    DATABASE;
}
```



## 例 2-26 アーカイブ目的での一貫性バックアップの作成

この例では、`keepOption` を使用して、1 年間は不要とみなされることがないアーカイブ・バックアップ・セットを作成します。この例では、データベースをバックアップし、REDO を現行のオンライン・ログにアーカイブしてこの新しいバックアップに一貫性があることを保証し、データファイル・バックアップを一貫性がある状態にリストアするために必要なアーカイブ・ログのみをバックアップします。

この BACKUP コマンドでは、リストア・ポイントも作成されます。そのリストア・ポイントは、このバックアップが一貫性を持つ SCN に対応します。FORMAT パラメータは、複数のバックアップ・セット内に複数のバックアップ・ピースを作成できるように指定する必要があります。

```
BACKUP DATABASE
  FORMAT '/disk1/archival_backups/db_%U.bck'
  TAG quarterly
  KEEP UNTIL TIME 'SYSDATE + 365'
  RESTORE POINT Q1FY06;
```

## 例 2-27 保存方針からのコピーの除外

次の例では、2 つのデータファイルをコピーして、保存方針から永久に除外します。(KEEP FOREVER にはリカバリ・カタログが必要です。) 自動バックアップがオフの場合でも、制御ファイルおよびサーバー・パラメータ・ファイルもバックアップされます。

```
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
BACKUP
  KEEP FOREVER
  FORMAT '?/dbs/%U_longterm.cpy'
  TAG LONGTERM_BCK
  DATAFILE 1 DATAFILE 2;
ALTER DATABASE OPEN;
```

## 例 2-28 バックアップが必要なファイルのバックアップ

データベースおよびアーカイブ・ログをテープに毎晩バックアップするために、次のコマンドを実行するとします。

```
BACKUP
  MAXSETSIZE 500M
  DATABASE PLUS ARCHIVELOG;
```

複数のバックアップ・セットが生成されるように、先行するコマンドで各バックアップ・セットのサイズの上限が設定されています。バックアップ処理の途中でメディア管理デバイスに障害が発生し、そのデバイスが再起動されるとします。翌日、バックアップ・セットの半分しか完了していないことに気付きます。その場合は、次のコマンドを夕方に実行できます。

```
BACKUP
  NOT BACKED UP SINCE TIME 'SYSDATE-1'
  MAXSETSIZE 500M
  DATABASE PLUS ARCHIVELOG;
```

先行するコマンドによって、過去 24 時間内にバックアップされていないファイルのみがバックアップされます。指定された時間枠のバックアップが使用可能であると Recovery Manager が判断すると、次のような出力が表示されます。

```
skipping datafile 1; already backed up on 18-JAN-07
skipping datafile 2; already backed up on 18-JAN-07
skipping datafile 3; already backed up on 18-JAN-07
```

BACKUP コマンドのすぐ後に NOT BACKED UP SINCE 句を置くと、バックアップするすべてのオブジェクトに影響します。また、その句は、個々の `backupSpec` 句の後に置くこともできます。その場合は、その句の制限を受ける `backupSpec` によって指定されたオブジェクトのバックアップのみが作成されます。

**例 2-29 NODUPLICATES を使用したデータファイルのコピーのバックアップ**

この例では、/disk2/df2.cpy というデータファイル 2 のデータファイル・コピーを作成します。次に、そのデータファイル・コピーを /disk1 および /disk3 ディレクトリにバックアップします。最後の BACKUP コマンドで指定されている NODUPLICATES オプションは、データファイル 2 のコピーを 1 つのみバックアップする必要があることを示しています。

```
BACKUP AS COPY
  DATAFILE 2
  FORMAT '/disk2/df2.cpy' TAG my_tag;
BACKUP AS COPY
  DATAFILECOPY '/disk2/df2.cpy'
  FORMAT '/disk1/df2.cpy';
BACKUP AS COPY
  DATAFILECOPY '/disk1/df2.cpy'
  FORMAT '/disk3/df2.cpy';
BACKUP
  DEVICE TYPE sbt
  DATAFILECOPY ALL NODUPLICATES; # backs up only copy of datafile 2
```

## CATALOG

### 用途

CATALOG コマンドは、次の目的に使用します。

- ディスク上のバックアップ・ピースおよびイメージ・コピーを Recovery Manager リポジトリに追加します。
- データファイルのコピーをレベル 0 の増分バックアップとして Recovery Manager リポジトリに記録する場合。これによって、コピーを増分バックアップ計画の一部として使用できます。

**関連項目：** カタログに格納されているターゲット・データベース・レコードを管理する方法は、『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』を参照してください。

### 前提条件

ターゲット・データベースに接続していて、そのデータベースがマウントまたはオープン状態である必要があります。Recovery Manager がリカバリ・カタログに接続されている場合は、カタログ・データベースをオープンする必要があります。

カタログ化するファイルは、次の条件を満たしている必要があります。

- SBT のデバイスに存在していないこと。
- ファイルがユーザー管理コピーの場合は、データファイルのコピー、制御ファイルのコピー、アーカイブ REDO ログまたはバックアップ・ピースであること。

### 使用上の注意

Recovery Manager では、すべてのユーザー管理バックアップがイメージ・コピーとみなされません。カタログ化中は、ファイルがオペレーティング・システム・ユーティリティにより正常にコピーされたかどうか Recovery Manager ではチェックされず、ヘッダーがチェックされるのみであるため注意してください。

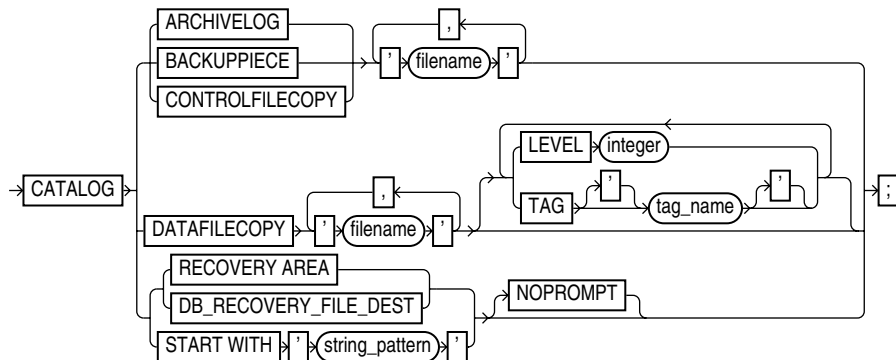
Data Guard 環境で Recovery Manager を使用する場合は、リカバリ・カタログが必要です。リカバリ・カタログは、DBID が同じ（ただし、DB\_UNIQUE\_NAME 値は異なる）すべてのプライマリ・データベースおよびスタンバイ・データベースに対して、統一されたファイルの名前空間をサポートしています。したがって、リカバリ・カタログでは、すべてのプライマリ・データベースまたはスタンバイ・データベースのデータベース・ファイル名が、オンライン REDO ログ、スタンバイ REDO ログ、一時ファイル、アーカイブ REDO ログ、バックアップ・セットおよびイメージ・コピーが作成された場所とともに追跡されます。

異なるプライマリ・データベースおよびスタンバイ・データベースで作成されたバックアップが Recovery Manager でどのように処理されるかについては、2-21 ページの「[Data Guard 環境での Recovery Manager のバックアップ](#)」を参照してください。通常、1 つのデータベースで作成されたテープ・バックアップはその環境のすべてのデータベースにアクセス可能ですが、ディスク・バックアップは作成元であるデータベースにのみアクセス可能です。

接続されているターゲット・データベースにバックアップからアクセス可能であれば、**RESTORE** や **RECOVER** などの **Recovery Manager** コマンドは、異なるデータベース間で透過的に動作します。ディスク・バックアップを環境内のホスト間で手動で転送して、そのバックアップをカタログ化することができます。バックアップが共有ディスク上に作成されている場合は、**CHANGE RESET DB\_UNIQUE\_NAME** を使用すると、バックアップを新しいデータベースに関連付けることができます。

## 構文

**catalog::=**



## セマンティクス

構文の要素	説明
ARCHIVELOG 'filename'	Recovery Manager リポジトリに追加するアーカイブ REDO ログのファイル名を指定します。  <b>注意:</b> このコマンドは、外部アーカイブ REDO ログをカタログ化しません。外部アーカイブ REDO ログは、LogMiner セッション中にロジカル・スタンバイ・データベースで受け取った REDO ログです。通常のアーカイブ・ログとは異なり、外部アーカイブ・ログには別の DBID が使用されています。

構文の要素	説明
BACKUPPIECE ' <i>filename</i> '	<p>Recovery Manager リポジトリに追加するバックアップ・ピースの名前を指定します (2-58 ページの例 2-30 を参照)。</p> <p>バックアップ・ピースがディスクに存在している必要があります。Recovery Manager は、バックアップ・ピースのヘッダーを検証した後でカタログ化します。Recovery Manager は、以前のデータベース・インカネーションからのバックアップ・ピースをカタログ化できます。</p> <p>バックアップ・ピースは、次のような状況の場合にカタログ化します。</p> <ul style="list-style-type: none"> <li>■ オペレーティング・システムのユーティリティでコピーまたは移動したバックアップ・ピースを、Recovery Manager で使用できるようにする場合。</li> <li>■ バックアップ・ピースの Recovery Manager メタデータは削除されていても、バックアップ・ピースがまだ存在している場合。このような状況は、一時的に使用できなかったバックアップ・ピースに対して DELETE コマンドを実行すると発生します。</li> <li>■ Data Guard 環境のデータベース・ホストで NOCATALOG バックアップを作成し、そのバックアップ・ピースを別のデータベース・ホストの同じ場所に移動した場合。このような場合のリカバリ・カタログには、元のバックアップ・ピースのレコードは存在しません。</li> <li>■ リカバリ・カタログを使用していないときに制御ファイルの再作成が必要になり、その結果、すべての Recovery Manager リポジトリ・データが消失する場合。バックアップをカタログ化すると、再度使用可能になります。</li> <li>■ 制御ファイルの自動バックアップが無効になっているときに、制御ファイルをバックアップしてからアーカイブ REDO ログをバックアップした場合。制御ファイルのリストアとマウントは実行できますが、制御ファイルの後でバックアップされたアーカイブ REDO ログを含むバックアップ・ピースをカタログ化する必要があります。</li> </ul> <p>バックアップ・ピースのリストを指定すると、一部のバックアップ・ピースの処理で障害が発生した場合でも、Recovery Manager によって、すべてのピースのカタログ化が試行されます。バックアップ・ピースをカタログ化すると、V\$BACKUP_PIECE に新しい行が作成されます。バックアップ・セットは、すべてのバックアップ・ピースがカタログ化された場合にのみ使用可能です。一部でもカタログ化が完了していない場合は、部分的に使用可能な状態であるにすぎません。</p> <p><b>注意:</b> データベースの COMPATIBLE パラメータが 11.0.0 以上に設定されている場合、Recovery Manager でサーバー・パラメータ・ファイルのバックアップを作成すると、バックアップはこのデータベースに関連付けられます。この場合、Recovery Manager を別のデータベースに接続して、バックアップ・ピースを明示的にカタログ化しても、このバックアップに関連付けられている DB_UNIQUE_NAME は変更されません。たとえば、COMPATIBLE が 11.0.0 の場合に DB_UNIQUE_NAME 'NEWYORK' でデータベースのサーバー・パラメータ・ファイルのバックアップを作成すると、データベース NEWYORK で作成されたサーバー・パラメータ・ファイルのバックアップを使用して、データベース BOSTON のサーバー・パラメータ・ファイルをリストアすることはできません。</p>
CONTROLFILECOPY ' <i>filename</i> '	<p>Recovery Manager リポジトリに追加する制御ファイルのコピーのファイル名を指定します。指定できる制御ファイル・コピーは、次のいずれかのコマンドで作成される、通常の制御ファイルまたはスタンバイ制御ファイルのコピーです。</p> <ul style="list-style-type: none"> <li>■ Recovery Manager コマンドの BACKUP AS COPY CURRENT CONTROLFILE</li> <li>■ SQL 文の ALTER DATABASE BACKUP CONTROLFILE</li> <li>■ SQL 文の ALTER DATABASE CREATE STANDBY CONTROLFILE</li> </ul> <p><b>注意:</b> プライマリ・データベースの制御ファイルのバックアップは、リストア操作時に自動的にスタンバイ制御ファイルに変換されます。</p>

構文の要素	説明
DATAFILECOPY ' <i>filename</i> '	Recovery Manager リポジトリに追加するデータファイルのコピーのファイル名を指定します (2-58 ページの例 2-30 を参照)。データファイルのコピーを作成するには、Recovery Manager の BACKUP AS COPY コマンドを使用する方法、またはオペレーティング・システムのユーティリティと ALTER TABLESPACE BEGIN/END BACKUP を併用する方法があります。
LEVEL <i>integer</i>	データファイル・コピーがレベル 0 の増分バックアップとして記録されるように指定します (LEVEL の値として有効なのは、0 のみです)。  このデータファイルのコピーを基本レベル 0 のバックアップとして使用して、増分バックアップを実行できます。
TAG <i>tagname</i>	データファイルのコピーのタグを指定します。
RECOVERY AREA	フラッシュ・リカバリ領域内のすべての有効なバックアップ・セット、データファイルのコピーおよびアーカイブ REDO ログをカタログ化します (2-59 ページの例 2-32 を参照)。  Recovery Manager は、TARGET としてデータベースに接続されている必要があります。ターゲット・データベースはマウントまたはオープン状態である必要があります。キーワード RECOVERY AREA と DB_RECOVERY_FILE_DEST は完全に同一のシノニムです。  <b>注意:</b> フラッシュ・リカバリ領域に外部アーカイブ・ログ (LogMiner セッション中にロジカル・スタンバイ・データベースで受け取る外部アーカイブ REDO ログ) が存在する場合、このコマンドではそれらの外部アーカイブ・ログもカタログ化されます。
DB_RECOVERY_FILE_DEST	キーワード RECOVERY AREA と DB_RECOVERY_FILE_DEST は完全に同一のシノニムです。
START WITH ' <i>string_pattern</i> '	名前が <i>string_pattern</i> で始まる有効なバックアップ・セット、データファイルと制御ファイルのコピー、およびアーカイブ REDO ログをすべてカタログ化します。文字列パターンには、ASM ディスク・グループ、Oracle Managed Files ディレクトリ、またはファイル名の一部分を指定できます (2-59 ページの例 2-31 を参照)。  ディスクの場所にカタログ化できないファイルがある場合は、すべてレポートされます。Recovery Manager がターゲット・データベースに接続している必要があります。  文字列パターンによってファイル名を指定すると、その文字列パターンはファイル名パターンの左側に一致します。たとえば、/tmp/arc は、ディレクトリ /tmp/arc_dest および /tmp/archive/january 内のすべてのファイルと、ファイル /tmp/arc.cpy に一致します。  <b>注意:</b> 文字列パターンにワイルド・カード文字は使用できません。正確な接頭辞のみを使用してください。
NOPROMPT	確認のプロンプトを抑制します。デフォルトでは、一致するたびにプロンプトが表示されます。

## 例

### 例 2-30 増分バックアップとしてのカタログに対するデータファイル・コピーの追加

Linux ユーティリティを使用して、users01.dbf データファイルを /disk2/backup/users01.bak にバックアップするとします。この例では、データファイル・コピーがレベル 0 の増分バックアップとしてカタログ化されてから、すべてのコピーがリストされます。

```
CATALOG DATAFILECOPY '/disk2/backup/users01.bak' LEVEL 0;
LIST COPY;
```

**例 2-31 ディレクトリ内の複数のコピーのカタログ化**

この例では、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 /disk2/archlog ディレクトリにコピーされたアーカイブ REDO ログで一杯のディレクトリをカタログ化します。この例では、出力例も示します。

```
CATALOG START WITH '/disk2/archlog' NOPROMPT;

searching for all files that match the pattern /disk2/archlog

List of Files Unknown to the Database
=====
File Name: /disk2/archlog/o1_mf_1_10_24trtc7s_.arc
File Name: /disk2/archlog/o1_mf_1_11_24trtg7s_.arc
File Name: /disk2/archlog/o1_mf_1_12_24trtk84_.arc
File Name: /disk2/archlog/o1_mf_1_13_24trtn85_.arc
File Name: /disk2/archlog/o1_mf_1_14_24trtq84_.arc
File Name: /disk2/archlog/o1_mf_1_15_24trtt84_.arc
File Name: /disk2/archlog/o1_mf_1_16_24trtx84_.arc
File Name: /disk2/archlog/o1_mf_1_17_24trv085_.arc
File Name: /disk2/archlog/o1_mf_1_18_24trv385_.arc
File Name: /disk2/archlog/o1_mf_1_19_24trv685_.arc
cataloging files...
cataloging done

List of Cataloged Files
=====
File Name: /disk2/archlog/o1_mf_1_10_24trtc7s_.arc
File Name: /disk2/archlog/o1_mf_1_11_24trtg7s_.arc
File Name: /disk2/archlog/o1_mf_1_12_24trtk84_.arc
File Name: /disk2/archlog/o1_mf_1_13_24trtn85_.arc
File Name: /disk2/archlog/o1_mf_1_14_24trtq84_.arc
File Name: /disk2/archlog/o1_mf_1_15_24trtt84_.arc
File Name: /disk2/archlog/o1_mf_1_16_24trtx84_.arc
File Name: /disk2/archlog/o1_mf_1_17_24trv085_.arc
File Name: /disk2/archlog/o1_mf_1_18_24trv385_.arc
File Name: /disk2/archlog/o1_mf_1_19_24trv685_.arc
```

**例 2-32 フラッシュ・リカバリ領域内のファイルのカタログ化**

この例では、現在有効になっているフラッシュ・リカバリ領域内のすべてのファイルを、ファイルごとにユーザーにプロンプトを表示せずにカタログ化します。出力例に示すとおり、カタログ化するファイルが見つからない場合はメッセージが表示されます。

```
CATALOG RECOVERY AREA;

searching for all files in the recovery area
no files found to be unknown to the database
```

**例 2-33 バックアップ・ピースのカタログ化**

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、ある場所のバックアップ・ピースを別の場所にコピーするとします。この例では、新しい場所のバックアップ・ピースがカタログ化されます（例には出力例も含まれます）。

```
RMAN> CATALOG BACKUPPIECE '/disk1/c-874220581-20061128-01';

using target database control file instead of recovery catalog
cataloged backup piece
backup piece handle=/disk1/c-874220581-20061128-01 RECID=12 STAMP=607695990
```

## CHANGE

### 用途

CHANGE コマンドを使用すると、次のタスクを実行できます。

- **Recovery Manager** リポジトリに記録されているバックアップおよびコピーの可用性ステータスを更新します。
- 自動診断リポジトリに記録されている障害について、その優先順位を変更したり障害をクローズします。
- リカバリ・カタログに記録されている、ターゲット・データベースの `DB_UNIQUE_NAME` を更新します。
- **Data Guard** 環境にあるデータベースのバックアップを同じ環境内の別のデータベースに関連付けます。

**関連項目：** バックアップまたはコピーの可用性ステータスについては、『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』を参照してください。

### 前提条件

**Recovery Manager** は **TARGET** としてデータベース・インスタンスに接続され、そのインスタンスが起動されている必要があります。

### 使用上の注意

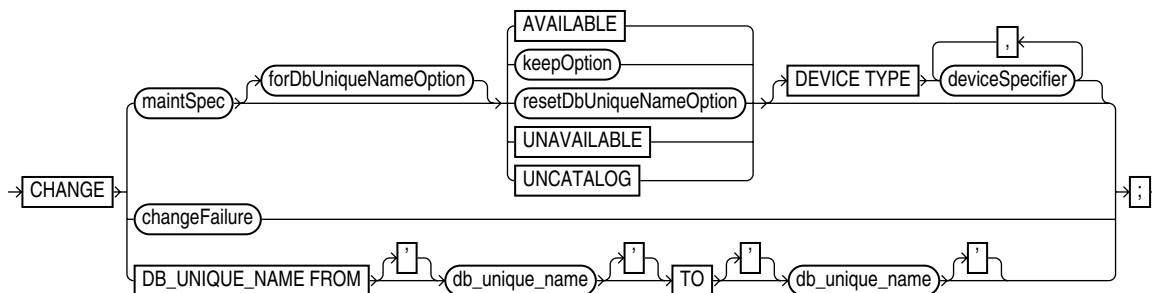
バックアップの関連付けとアクセス可能性の違いについては、2-21 ページの「[Data Guard 環境での Recovery Manager のバックアップ](#)」を参照してください。Data Guard 環境では、バックアップまたはコピーを作成するデータベースはファイルに関連付けられます。Data Guard 環境のどのデータベースに接続されていても、バックアップに接続可能であれば、CHANGE、DELETE、CROSSCHECK などのメンテナンス・コマンドをバックアップに使用できます。通常、Recovery Manager では、データベース上で作成されたテープ・バックアップはその環境のすべてのデータベースにアクセス可能であるとみなされますが、ディスク・バックアップは作成元のデータベースのみにアクセス可能であるとみなされます。

Recovery Manager は、バックアップと関連付けられたデータベースに **TARGET** として接続されている場合のみ、バックアップのステータスを **AVAILABLE** から **EXPIRED** または **DELETED** に更新できます。バックアップがターゲット・データベースに関連付けられていないために Recovery Manager が削除を実行できない場合は、そのバックアップが関連付けられているデータベースで、同じ **CROSSCHECK** 操作をそのファイルに実行するように求めるプロンプトが表示されます。このように、Recovery Manager では、不適切な **SBT** 構成による意図しないステータス変更を防止しています。

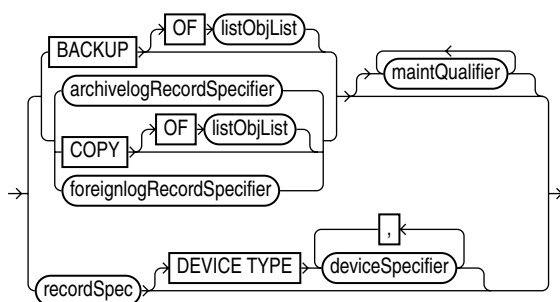
たとえば、Recovery Manager をスタンバイ・データベース `standby1` に **TARGET** として接続し、そのデータベースをテープとディスクにバックアップするとします。テープ・ドライブが使用できなくなった場合は、Data Guard 環境のプライマリ・データベースまたはスタンバイ・データベースに Recovery Manager を **TARGET** として接続して、テープ・バックアップのステータスを **UNAVAILABLE** に変更できます。テープ・ドライブが修復された後は、Recovery Manager を **TARGET** としてデータベースに接続して、テープ・バックアップのステータスを **AVAILABLE** に戻すことができます。ただし、オペレーティング・システムのユーティリティによって、ディスク・バックアップが誤って削除された場合、Recovery Manager は `standby1` に **TARGET** として接続されているときにのみ、ディスク・バックアップのステータスを変更できます。



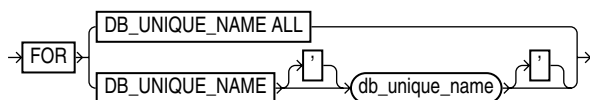
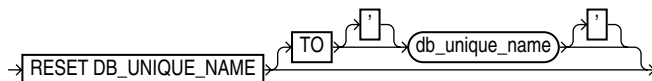
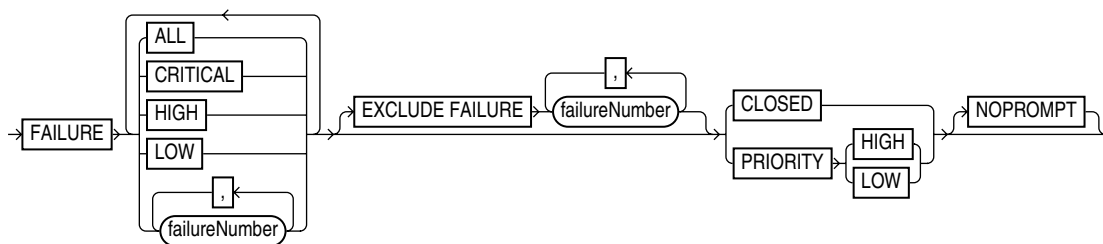
## 構文

**change::=**

(maintSpec::= (2-61 ページ)、 forDbUniqueNameOption::= (3-18 ページ)、  
keepOption::= (3-24 ページ)、 deviceSpecifier::= (3-15 ページ))

**maintSpec::=**

(listObjList::= (3-27 ページ)、 archivelogRecordSpecifier::= (3-6 ページ)、  
maintQualifier::= (3-30 ページ)、 recordSpec::= (3-35 ページ)、  
deviceSpecifier::= (3-15 ページ))

**forDbUniqueNameOption::=****resetDbUniqueNameOption::=****changeFailure::=**

## セマンティクス

### change

この句を使用すると、Recovery Manager リポジトリのレコードのステータスを変更できます。ステータスを変更する Recovery Manager リポジトリ・レコードの主キーを取得するには、[LIST](#) コマンドを実行するか、リカバリ・カタログ・ビューを検索します。

構文の要素	説明
<a href="#">maintSpec</a>	CHANGE を実行するファイルを指定します。  <b>関連項目:</b> この句のオプションについては、3-32 ページの「 <a href="#">maintSpec</a> 」を参照してください。
<a href="#">forDbUniqueNameOption</a>	Data Guard 環境において、指定した DB_UNIQUE_NAME にのみ関連付けられたオブジェクトのメタデータを変更します。  <b>関連項目:</b> この句のオプションについては、3-18 ページの「 <a href="#">forDbUniqueNameOption</a> 」を参照してください。
AVAILABLE	リポジトリ内でバックアップまたはコピーのステータスを AVAILABLE に変更します。Recovery Manager によってファイルが検索され、存在しているかどうかを確認されます。  この機能は、使用不可のファイルが再度使用できるようになった場合に有効です。また、このオプションを使用して、以前のインカネーションのバックアップおよびコピーのリポジトリ・ステータスを変更することもできます。  これは、手動または自動のメンテナンス・チャンネルを必要とする唯一の CHANGE オプションです。ただし、ディスク専用（つまり、ARCHIVELOG、DATAFILECOPYY または CONTROLFILECOPY）のファイルに CHANGE ... AVAILABLE を使用する場合、メンテナンス・チャンネルは不要です。CHANGE ... AVAILABLE をディスク専用でないファイルに使用する場合に、自動チャンネル用に構成されていないデバイス・タイプでオブジェクトを作成しているときは、これらのチャンネルに対して手動メンテナンス・コマンドを発行します。たとえば、sbt チャンネルでバックアップを作成したが、自動的に構成されているのが DISK チャンネルのみであれば、バックアップに対して CHANGE ... AVAILABLE 操作を行う前に、sbt チャンネルを手動で割り当てる必要があります。  Data Guard 環境のファイルに対して CHANGE ... AVAILABLE を実行すると、Recovery Manager は、ファイルのステータスを AVAILABLE に更新する前にクロスチェックの実行を試みます。ファイルにアクセスできない場合は、ファイルに関連付けられたデータベースに TARGET として接続してから、同じ操作を実行するように求めるプロンプトが表示されます。ファイルがアクセス可能な場合は、要求に応じてステータスが更新されます。  <b>注意:</b> バックアップのステータスは、 <a href="#">LIST</a> 出力またはリカバリ・カタログ・ビューで確認できます。  <b>注意:</b> CHANGE ... AVAILABLE は、LogMiner セッション中にロジカル・スタンバイ・データベースによって受け取られる外部アーカイブ REDO ログに対しては無効です。通常のアーカイブ・ログとは異なり、外部アーカイブ・ログには別の DBID が使用されています。
<a href="#">keepOption</a>	バックアップまたはコピーのステータスを、構成済の保存方針に基づいて変更します。たとえば、CHANGE ... NOKEEP を指定すると、バックアップの KEEP 属性が削除され、そのバックアップはバックアップ保存方針の対象になります。  KEEP FOREVER 句を指定するには、リカバリ・カタログを使用する必要があります (2-67 ページの例 2-36 を参照)。RESTORE POINT オプションは、CHANGE では有効ではありません。フラッシュ・リカバリ領域に格納されているファイルの CHANGE ... UNAVAILABLE または KEEP 属性は使用できません。  <b>関連項目:</b> 3-24 ページの「 <a href="#">keepOption</a> 」を参照してください。

構文の要素	説明
<a href="#">resetDbUniqueNameOption</a>	<p><code>maintSpec</code> のファイルを Data Guard 環境の別のデータベースに関連付けます。</p> <p><b>関連項目:</b> 2-65 ページの「<a href="#">resetDbUniqueNameOption</a>」を参照してください。</p>
UNAVAILABLE	<p>リポジトリ内でバックアップまたはコピーのステータスを UNAVAILABLE に変更します (2-66 ページの例 2-34 を参照)。ステータスは、LIST 出力またはリカバリ・カタログ・ビューで確認できます。</p> <p>このオプションは、ファイルが見つからない場合や、別のサイトに移された場合に有効です。Recovery Manager では、UNAVAILABLE マークを付けたファイルは、RESTORE または RECOVER コマンドでは使用されません。後でそのファイルが見つかるか、メイン・サイトに戻った場合は、AVAILABLE オプションを使用して、このステータスを更新します。UNAVAILABLE オプションは、特定のバックアップまたはコピーをリストア対象にせず、削除もしない場合や、以前のインカネーションのバックアップおよびコピーのリポジトリ・ステータスを変更する場合などにも有効です。</p> <p>フラッシュ・リカバリ領域のファイルでは、CHANGE ... UNAVAILABLE は無効です。このコマンドは、LogMiner セッション中にロジカル・スタンバイ・データベースで受け取られる外部アーカイブ REDO ログに対しても無効です。通常のアーカイブ・ログとは異なり、外部アーカイブ・ログには別の DBID が使用されています。</p> <p><b>注意:</b> Data Guard 環境のファイルに対して CHANGE ... UNAVAILABLE を実行すると、ステータスを UNAVAILABLE に更新するまで、ファイルのクロスチェックは試行されません。ファイルが物理的に存在しているかどうかに関係なく、ステータスは要求に応じて更新されます。</p>
UNCATALOG	<p>リカバリ・カタログから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 REDO ログの参照を削除し、ターゲット制御ファイル内のレコードをステータス DELETED に更新します (2-66 ページの例 2-35 を参照)。CHANGE ... UNCATALOG コマンドでは、物理バックアップおよびコピーは処理されないことに注意してください。ファイルが DELETE コマンド以外の手段で削除されたときは、このコマンドを使用して Recovery Manager に通知します。</p> <p>Data Guard 環境のファイルに対して CHANGE ... UNCATALOG を実行すると、リカバリ・カタログからそのファイルのメタデータが削除されるまで、ファイルのクロスチェックは試行されません。ファイルが物理的に存在しているかどうかに関係なく、メタデータは要求に応じて削除されます。</p> <p><b>注意:</b> バックアップ制御ファイルから再同期化するか、またはリカバリ・カタログをアップグレードすると、CHANGE ... UNCATALOG で Recovery Manager リポジトリから以前削除されたレコードがリカバリ・カタログに再表示される場合があります。</p>
DEVICE TYPE <a href="#">deviceSpecifier</a>	<p>指定したデバイス・タイプにのみ CHANGE を実行します (3-15 ページの「<a href="#">deviceSpecifier</a>」を参照)。このオプションが有効になるのは、構成済の自動チャンネルがあり、チャンネルを手動で割り当てていない場合のみです。たとえば、CHANGE UNCATALOG ... DEVICE TYPE DISK を実行すると、Recovery Manager ではディスク上のファイルのみがカタログから削除されます。</p>
<a href="#">changeFailure</a>	<p>データ・リカバリ・アドバイザによって記録された障害の変更内容を指定します。</p>

構文の要素	説明
<pre>DB_UNIQUE_NAME FROM db_unique_name TO db_unique_name</pre>	<p>リカバリ・カタログ内のメタデータを更新して、Data Guard 環境でデータベースに新しい DB_UNIQUE_NAME を反映します。最初の値には、リカバリ・カタログに現在記録されている、データベースの古い DB_UNIQUE_NAME を指定します。2 番目の値には、新しい DB_UNIQUE_NAME を指定します。</p> <p>Recovery Manager は、リカバリ・カタログおよびマウント済のターゲット・データベースに接続している必要があります。ターゲット・データベースでは、FROM db_unique_name パラメータに DB_UNIQUE_NAME を指定できません。指定すると、Recovery Manager によってエラーが発行されます。</p> <p>通常、このコマンドを使用するのは、データベースの DB_UNIQUE_NAME 初期化パラメータを変更した後で、リカバリ・カタログのメタデータを更新する必要があるときです。一般に、このコマンドは、名前を変更したデータベースに対して他の Recovery Manager 操作を行う前に実行してください。LIST DB_UNIQUE_NAME を実行した後に CHANGE DB_UNIQUE_NAME を実行することをお勧めします。</p> <p>スタンバイ・データベースの DB_UNIQUE_NAME 初期化パラメータを、standby_old から standby_new に変更したとします。通常、CHANGE DB_UNIQUE_NAME は場合に実行します。</p> <ul style="list-style-type: none"> <li>■ LIST DB_UNIQUE_NAME コマンドで、DB_UNIQUE_NAME の古い値は表示されるが、新しい値が表示されない (例 2-39 の 2-67 ページを参照)。</li> <li>■ LIST DB_UNIQUE_NAME コマンドで、DB_UNIQUE_NAME の古い値と新しい値の両方が表示される。Recovery Manager を TARGET として、DB_UNIQUE_NAME が認識されていないデータベースに接続すると、そのインスタンスは暗黙的に新しいデータベースとして登録されます。そのため、LIST DB_UNIQUE_NAME コマンドを実行すると、DB_UNIQUE_NAME 初期化パラメータが変更されたデータベースに対して、古い名前と新しい名前 (この例では standby_old と standby_new) が表示されることとなります。</li> </ul> <p>古い名前のみが表示される場合は、CHANGE DB_UNIQUE_NAME FROM standby_old TO standby_new を実行して、リカバリ・カタログの DB_UNIQUE_NAME の standby_old を standby_new に変更します。</p> <p>古い名前と新しい名前の両方が表示される場合は、CHANGE DB_UNIQUE_NAME FROM standby_old TO standby_new を実行すると、Recovery Manager によって次のコマンドが自動的に実行されます。</p> <ol style="list-style-type: none"> <li>1. CHANGE ARCHIVELOG ALL FOR DB_UNIQUE_NAME standby_old RESET DB_UNIQUE_NAME</li> <li>2. CHANGE BACKUP FOR DB_UNIQUE_NAME standby_old RESET DB_UNIQUE_NAME</li> <li>3. UNREGISTER DB_UNIQUE_NAME standby_old</li> </ol> <p>このように、FROM 句に指定された DB_UNIQUE_NAME のすべてのバックアップの関連付けが、TO 句に指定された DB_UNIQUE_NAME に変更されます。</p>

**resetDbUniqueNameOption**

この句を使用すると、Data Guard 環境の 1 つのデータベースで作成されたバックアップを、同じ環境の別のデータベースに関連付けることができます。次の表では、RESET DB\_UNIQUE\_NAME で各種オプションを指定したときの Recovery Manager の動作について説明します。

**表 2-2 RESET DB\_UNIQUE\_NAME のオプション**

TO db_unique_name	FOR DB_UNIQUE_NAME	Recovery Manager の動作
実行しない	実行しない	Recovery Manager は、 <i>maintSpec</i> ファイルをターゲット・データベースに関連付けます。また、どのデータベースにも関連付けられていないすべてのバックアップの関連付けも変更します。 通常、Oracle Database 11g のリカバリ・カタログ・スキーマにアップグレードした後で、これらのオプションを指定して CHANGE を実行します。これにより、バックアップをターゲット・データベースに関連付けることができます。
実行する	実行しない	Recovery Manager は、 <i>maintSpec</i> ファイルを TO db_unique_name で示されたデータベースに関連付けます。また、どのデータベースにも関連付けられていないすべてのバックアップの関連付けも変更します。
実行しない	実行する	Recovery Manager は、その操作を FOR DB_UNIQUE_NAME 句でデータベースに関連付けられている <i>maintSpec</i> ファイルに限定し、それらのファイルをターゲット・データベースに関連付けます。
実行する	実行する	Recovery Manager は、その操作を FOR DB_UNIQUE_NAME 句でデータベースに関連付けられている <i>maintSpec</i> ファイルに限定し、それらのファイルを TO db_unique_name で指定されたデータベースに関連付けます。

**構文の要素****説明**

RESET DB\_UNIQUE\_NAME

*maintSpec* のファイルをターゲット・データベースと関連付けます (2-67 ページの例 2-38 を参照)。各種オプションを指定したときの Recovery Manager の動作については、表 2-2 を参照してください。

データベース間でファイルの関連付けを変更すると、Recovery Manager は、リカバリ・カタログから重複した名前を削除します。たとえば、データファイル・コピー /d1/df1.bak の関連付けをデータベース standby1 からデータベース prod に変更すると、リカバリ・カタログが持つこのファイルのレコードは 2 つではなく、1 つのみになります。

このコマンドの実行結果は元に戻せないため、RESET DB\_UNIQUE\_NAME オプションを指定する際は注意が必要です。たとえば、データベース standby1 に関連付けられたファイルの関連付けをデータベース prod に変更すると、これらのファイルが以前に関連付けられていたデータベースに関するメタデータの履歴情報はリカバリ・カタログに保持されなくなります。ただし、データベース standby1 の登録を解除して、Recovery Manager を再度 standby1 に接続することは可能です。しかし、この場合は、リカバリ・カタログが standby1 制御ファイルのすべてのメタデータで更新されます。

TO db\_unique\_name

*maintSpec* のファイルを Data Guard 環境の指定されたデータベースに関連付けます。

**changeFailure**

この句を使用すると、障害のステータスを変更できます。障害のリストを表示するには、`LIST FAILURE` コマンドを使用します。

構文の要素	説明
<code>FAILURE</code>	自動診断リポジトリに記録されている障害について、その優先順位を変更したり、障害をクローズすることができます。デフォルトでは、変更を実行する前に Recovery Manager によって確認のプロンプトが表示されます。  Recovery Manager が接続しているデータベースは、単一インスタンス・データベースである必要があります。また、フィジカル・スタンバイ・データベース以外である必要があります。
<code>ALL</code>	オープン状態の障害のみを変更します。
<code>CRITICAL</code>	CRITICAL な障害のみを変更します。
<code>HIGH</code>	優先順位が HIGH の障害のみを変更します。
<code>LOW</code>	優先順位が LOW の障害のみを変更します。
<code>failnum</code>	指定した障害のみを変更します。
<code>EXCLUDE FAILURE failnum</code>	指定した障害を変更しないようにします。

**例****例 2-34 UNAVAILABLE ステータスへのバックアップの更新**

ディスクに領域の問題があるため、バックアップ・セット 4 を一時的に別の場所に移動したとします。キー 4 を持つバックアップは、まだ使用可能としてリストされます。

```
RMAN> LIST BACKUP SUMMARY;
```

```
List of Backups
=====
Key      TY LV S Device Type Completion Time #Pieces #Copies Compressed Tag
-----
1        B A A DISK      24-FEB-07      1      1      NO      TAG20070427T115348
3        B A A DISK      24-MAR-07      1      1      NO      TAG20070427T115452
4        B F A DISK      24-APR-07      1      1      NO      TAG20070427T115456
```

このバックアップは、ディスク領域が空いたときに元の場所に戻す予定なので、カタログから削除しません。したがって、次の手順でバックアップを使用不可能にします（例には出力例も含まれます）。

```
RMAN> CHANGE BACKUPSET 4 UNAVAILABLE;
```

```
changed backup piece unavailable
backup piece handle=/disk2/backup/c-3257893776-20070424-00 RECID=4 STAMP=588858897
Changed 1 objects to UNAVAILABLE status
```

**例 2-35 カタログ内のアーカイブ REDO ログの削除と追加**

この例では、すべてのアーカイブ REDO ログを新しいディレクトリに移動し、カタログから削除した後で、新しい場所でカタログ化します。

```
HOST '/bin/mv $ORACLE_HOME/dbs/*.arc /disk2/archlog/';
CHANGE ARCHIVELOG ALL UNCATALOG;
CATALOG START WITH '/disk2/archlog' NOPROMPT;
```

**例 2-36 アーカイブ・バックアップへのデータベース・バックアップの変更**

別のサイトに保存するために、データベース・バックアップをアーカイブ・バックアップに変更することを目標とします。このバックアップには一貫性があるため、リカバリは必要なく、アーカイブ REDO ログをバックアップとともに保存する必要もありません。この例では、CHANGE ... KEEP FOREVER を使用して、バックアップが不要とみなされないことがないように指定します。

```

RMAN> CONNECT TARGET /
RMAN> CONNECT CATALOG rman@catdb

recovery catalog database Password: password

RMAN> CHANGE BACKUP TAG 'consistent_db_bkup' KEEP FOREVER;

```

**例 2-37 障害のステータスの変更**

次の例では、LIST FAILURE コマンドによって、データファイルに破損ブロックがあることが示されます。障害の番号は 5 で、優先順位は HIGH です。この障害の優先順位を LOW に変更することにします。

```

RMAN> LIST FAILURE;

List of Database Failures
Failure ID Priority Status    Time Detected Summary
-----
5          HIGH    OPEN    11-DEC-06    datafile 8 contains corrupt blocks

RMAN> CHANGE FAILURE 5 PRIORITY LOW;

List of Database Failures
Failure ID Priority Status    Time Detected Summary
-----
5          HIGH    OPEN    11-DEC-06    datafile 8 contains corrupt blocks

Do you really want to change the above failures (enter YES or NO)? YES
changed 1 failures to LOW priority

```

**例 2-38 Data Guard 環境での新しいデータベースへのバックアップの関連付け**

プライマリ・データベース prod と関連付けられたスタンバイ・データベースが、standby1、standby2 および standby3 であるとし、この例では、Recovery Manager はターゲット・データベース prod およびリカバリ・カタログに接続されているとします。

環境から standby1 を削除する予定のため、standby1 バックアップをプライマリ・データベースに関連付ける必要があります。環境から standby3 も削除する予定のため、standby3 バックアップを standby2 に関連付ける必要があります。次のコマンドを実行します。

```

CHANGE BACKUP FOR DB_UNIQUE_NAME standby1 RESET DB_UNIQUE_NAME;
CHANGE BACKUP FOR DB_UNIQUE_NAME standby3 RESET DB_UNIQUE_NAME TO standby2;

```

**例 2-39 リカバリ・カタログの DB\_UNIQUE\_NAME の更新**

スタンバイ・データベースの DB\_UNIQUE\_NAME 初期化パラメータが dgrdbms4 に設定されており、これを sfrdbms4 に変更するとします。スタンバイ・インスタンスを停止し、DB\_UNIQUE\_NAME 初期化パラメータを sfrdbms4 に変更して、スタンバイ・インスタンスを再起動します。

その後、リカバリ・カタログを更新して、スタンバイ・データベースの変更された一意の名前を反映させるには、次のように Recovery Manager をプライマリ・データベースとリカバリ・カタログに接続し、CHANGE コマンドを実行します。

```

CHANGE DB_UNIQUE_NAME FROM dgrdbms4 TO sfrdbms4;

```

## CONFIGURE

### 用途

CONFIGURE コマンドを使用すると、特定のデータベースに対する Recovery Manager のバックアップ、リストア、複製およびメンテナンス・ジョブに影響する永続構成を作成または変更できます。構成は、明示的に消去または変更するまで、そのデータベースに対するすべての Recovery Manager セッションに有効です。SHOW コマンドを使用すると、1 つ以上のデータベースの構成を表示できます。

**関連項目：** Recovery Manager 環境の構成方法は、『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』を参照してください。

### 追加トピック

- [前提条件](#)
- [使用上の注意](#)
- [構文](#)
- [セマンティクス](#)
- [例](#)

### 前提条件

このコマンドは、Recovery Manager プロンプトでのみ実行してください。

FOR DB\_UNIQUE\_NAME 句を指定しない場合は、Recovery Manager をターゲット・データベースに接続する必要があります。ターゲット・データベースはマウントまたはオープン状態である必要があります。

### 使用上の注意

CONFIGURE コマンドは、ターゲット・データベース構成を、常にターゲット・データベースの制御ファイル内に格納します。Recovery Manager をリカバリ・カタログとともに使用すると、登録されているデータベースごとに永続的な構成設定がカタログ内にも保存されます。

#### Recovery Manager のデフォルト構成の設定

Recovery Manager の CONFIGURE 設定には、デフォルト値があります。CLEAR オプションを指定して CONFIGURE コマンドを再実行すると、このコマンドのあらゆる設定をデフォルトに戻すことができますが、この方法で個々のパラメータをクリアすることはできません。たとえば、CONFIGURE CHANNEL DEVICE TYPE sbt CLEAR は実行できますが、CONFIGURE CHANNEL DEVICE TYPE sbt MAXPIECESIZE 5M CLEAR は実行できません。

#### Data Guard 環境での Recovery Manager の構成

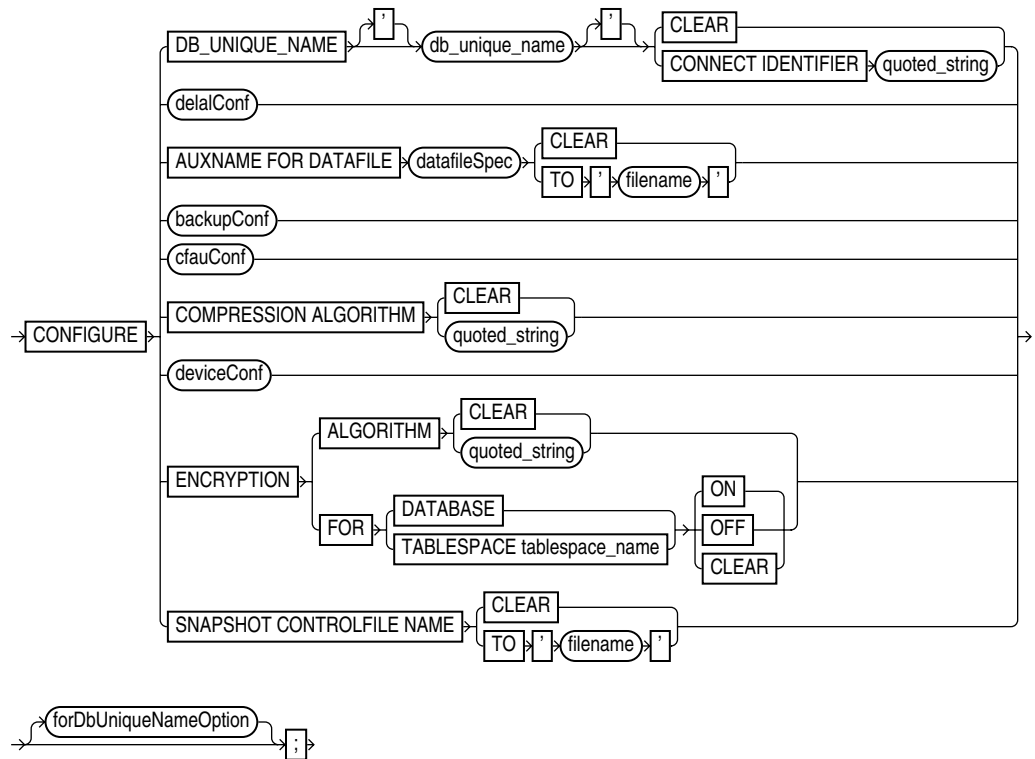
Data Guard 環境で Recovery Manager を使用する場合は、常にリカバリ・カタログを使用することをお勧めします。CONFIGURE コマンドを使用すると、Data Guard 環境の個々のプライマリまたはスタンバイ・データベースに対する永続的な Recovery Manager 構成（バックアップの保存方針、表領域の除外および補助名の設定は除く）を作成できます。このように、プライマリ・データベースおよびスタンバイ・データベースに、様々なチャンネル構成、自動バックアップの場所などを設定できます。

FOR DB\_UNIQUE\_NAME 句を使用すると、Recovery Manager が TARGET として接続されていないデータベースを構成できます。CONFIGURE DB\_UNIQUE\_NAME を使用すると、新しいフィジカル・スタンバイ・データベースをリカバリ・カタログに認識させて、暗黙的に登録できます。



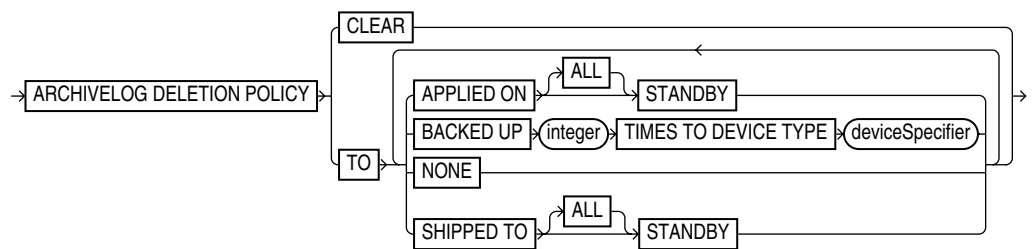
構文

**configure::=**



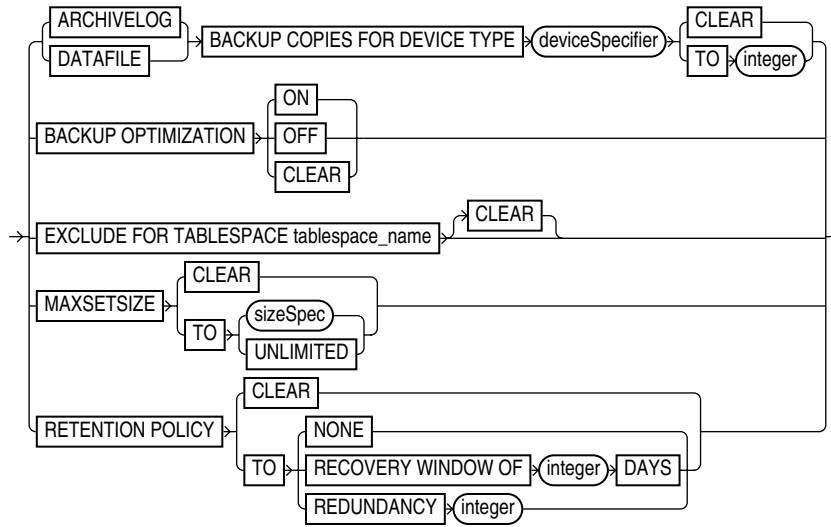
(datafileSpec::= (3-14 ページ)、 backupConf::= (2-70 ページ)、 cfauConf::= (2-70 ページ)、 deviceConf::= (2-70 ページ)、 forDbUniqueNameOption::= (3-18 ページ))

**delalConf::=**



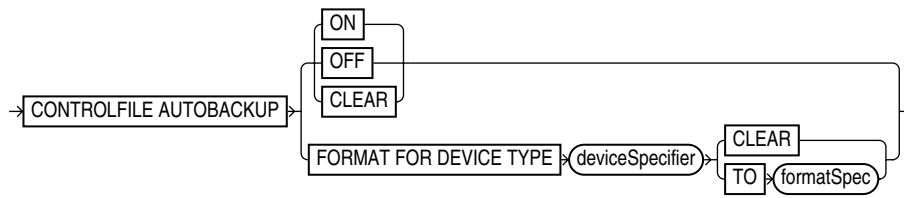
(deviceSpecifier::= (3-15 ページ))

**backupConf::=**



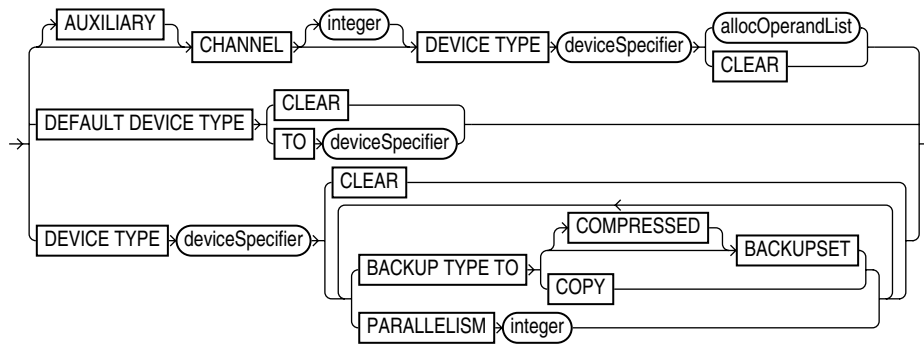
(*deviceSpecifier::=* (3-15 ページ)、*sizeSpec::=* (2-70 ページ))

**cfauConf::=**



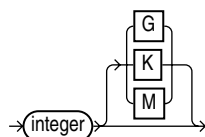
(*deviceSpecifier::=* (3-15 ページ)、*formatSpec::=* (3-21 ページ))

**deviceConf::=**

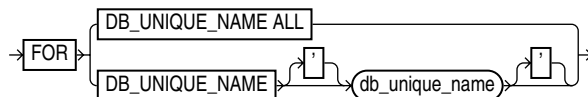


(*deviceSpecifier::=* (3-15 ページ)、*allocOperandList::=* (3-2 ページ))

**sizeSpec::=**



## forDbUniqueNameOption::=



## セマンティクス

## configure

構文の要素	説明
DB_UNIQUE_NAME	DB_UNIQUE_NAME で指定されたフィジカル・スタンバイ・データベースのネット・サービス名を指定します。CONNECT IDENTIFIER 文字列には、データベースのユーザー名とパスワードを指定しないでください。
db_unique_name	
{CLEAR	また、Recovery Manager は TARGET としてプライマリ・データベースにも接続している必要があります。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。
CONNECT IDENTIFIER 'connect_string' }	<p><b>RESYNC CATALOG FROM DB_UNIQUE_NAME</b> コマンドを実行すると、Data Guard 環境のデータベースは、ネット・サービス名を使用して db_unique_name データベースと接続します。たとえば、スタンバイ・データベースの一意の名前が standby1 で、ネット・サービス名が sby1 であるとし、Recovery Manager を TARGET としてプライマリ・データベースに接続し、CONFIGURE DB_UNIQUE_NAME 'standby1' CONNECT IDENTIFIER 'sby1' を実行します。Data Guard 環境のすべてのプライマリ・データベースおよびスタンバイ・データベースで standby1 に Oracle Net 接続を確立する必要がある場合は、ネット・サービス名 sby1 が使用されます。</p> <p><b>注意:</b> ターゲット・データベースが他のスタンバイ・データベースまたはプライマリ・データベースに接続する必要がある場合は、既存の Data Guard 認証メカニズムを使用して、SYS ユーザーとして接続します。</p> <p>例として、最近、Recovery Manager を TARGET としてプライマリ・データベースに接続し、CONFIGURE ...FOR DB_UNIQUE_NAME standby_new を使用してスタンバイ・データベース standby_new のバックアップ設定を構成したとします。ただし、Recovery Manager は TARGET として standby_new に接続されていません。このような場合に、RESYNC CATALOG FROM DB_UNIQUE_NAME standby_new を実行できます。プライマリ・データベースでは、スタンバイ・データベースへの Oracle Net 接続を確立するために接続識別子を使用します。後で Recovery Manager をスタンバイ・データベースに接続する場合は、Recovery Manager によってリカバリ・カタログからマウント済の制御ファイルに構成が送信されます。</p> <p><b>注意:</b> CONFIGURE DB_UNIQUE_NAME で指定された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は、Recovery Manager によって暗黙的に登録されます。</p>
delalConf	アーカイブ REDO ログの削除方針を構成します。

構文の要素	説明
AUXNAME FOR DATAFILE <code>datafileSpec TO 'filename'</code>	<p>指定したターゲット・データファイルの補助ファイル名を 'filename' に構成します (2-86 ページの例 2-44 を参照)。補助ファイル名の指定を解除するには、CLEAR を指定します。</p> <p>TSPITR を実行しているか、DUPLICATE を使用している場合は、AUXNAME を設定すると、プロシージャ中に手動で補助ファイル名を指定しなくても、補助データベースで使用するファイル名を事前に構成できます。</p> <p>たとえば、TSPITR 中に、データファイルがロー・ディスクにあってパフォーマンスの理由で補助データファイルをロー・ディスクにリストアする必要がある場合は、このコマンドを使用します。一般に、TSPITR で AUXNAME パラメータを設定するのは、SYSTEM および SYSAUX 表領域のデータファイルと、ロールバック・セグメントまたは UNDO セグメントが含まれる表領域を対象とした場合です。本番データベースで使用中のファイルを上書きしないでください。このファイルは、TSPITR の完了後に廃棄されます。本質的には、データファイルの AUXNAME とは、TSPITR がデータファイルの一時コピーを作成できる場所です。</p> <p>DUPLICATE コマンドでファイル名を変更する場合は、SET NEWNAME のかわりに CONFIGURE AUXNAME を使用します。違いは、最初に AUXNAME を設定すると、別の 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ことです。AUXNAME 設定は、CONFIGURE AUXNAME...CLEAR を発行するまで有効になっています。これとは対照的に、SET NEWNAME コマンドの場合は、DUPLICATE コマンドを実行するたびにコマンドを再発行する必要があります。</p> <p><b>関連項目:</b> Recovery Manager の TSPITR を実行する方法、および Recovery Manager を使用してデータベースを複製する方法は、『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』を参照してください。</p>
<code>backupConf</code>	<p>デフォルトのバックアップ・オプション (多重化、最適化、表領域の除外、バックアップ・セットのサイズ、保存方針など) を構成します。</p>
<code>cfauConf</code>	<p>制御ファイルの自動バックアップ設定を構成します。</p>
COMPRESSION ALGORITHM <code>'algorithm_name'</code>	<p>圧縮されたバックアップ・セットの作成時に、Recovery Manager が使用するアルゴリズムを指定します。デフォルトの圧縮アルゴリズムは、BZIP2 です。</p> <p>サポートされているアルゴリズムは、ZLIB および BZIP2 です。ZLIB は、CPU 効率を基準に最適化されています。BZIP2 は、圧縮率が最大になるように最適化されています。BZIP2 は、ZLIB より多くの CPU リソースを消費しますが、通常はサイズがより小さいバックアップを得られます。ZLIB 圧縮 (Oracle Advanced Compression オプションが必要) を使用する場合は、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。</p> <p><b>注意:</b> サポートされているアルゴリズムは、V\$RMAN_COMPRESSION_ALGORITHM ビューに示されています。</p>
<code>deviceConf</code>	<p>デバイスのデフォルトのバックアップ設定 (デフォルトのバックアップ・デバイス、デバイスのチャンネル構成、各デバイスのデフォルトのバックアップ・タイプ、並列度など) を構成します。</p>
ENCRYPTION	<p>データベースまたは表領域に透過モードの暗号化設定を指定します。</p> <p>この構成は、SET ENCRYPTION コマンドによるオーバーライドがないかぎり適用されます。個々の表領域に指定されているオプションは、データベース全体に指定されているオプションより優先されます。</p> <p><b>関連項目:</b> 様々なモードのバックアップの暗号化については 2-20 ページの「バックアップ・セットの暗号化」を、透過的データ暗号化については『Oracle Database Advanced Security 管理者ガイド』を参照してください。</p>

構文の要素	説明
ALGORITHM 'algorithm_name'	バックアップ・セットの書き込み時に使用するデフォルトの暗号化アルゴリズムを指定します。V\$RMAN_ENCRYPTION_ALGORITHMS に、使用可能な値が示されています。CLEAR オプションを指定すると、データベースがデフォルトにリセットされます。
FOR DATABASE [ON   OFF   CLEAR]	データベース全体に対して透過的暗号化を有効にするかどうかを指定します。オプションは次のとおりです。 <ul style="list-style-type: none"> <li>■ ON を指定すると、すべてのデータベース・ファイルの暗号化が有効になります。</li> <li>■ OFF を指定すると、すべてのデータベース・ファイルの暗号化が無効になります。</li> <li>■ CLEAR を指定すると、OFF のデフォルト設定がリストアされます。</li> </ul> <b>注意:</b> パスワードの暗号化を有効にするには、SET ENCRYPTION IDENTIFIED BY コマンドを使用する必要があります。
FOR TABLESPACE tablespace_name [ON   OFF   CLEAR]	1 つ以上の表領域に対して透過的暗号化を有効にするかどうかを指定します。表領域に対して構成された設定は、常に、データベース・レベルで設定された構成によってオーバーライドされます。オプションは次のとおりです。 <ul style="list-style-type: none"> <li>■ ON を指定すると、指定した表領域の暗号化が有効になります（ただし、SET ENCRYPTION OFF FOR ALL TABLESPACES が使用されている場合を除く）。</li> <li>■ OFF を指定すると、指定した表領域の暗号化が無効になります（ただし、SET ENCRYPTION ON FOR ALL TABLESPACES が使用されている場合を除く）。</li> <li>■ CLEAR を指定すると、指定した表領域の暗号化は、データベース全体に対する現行のデフォルトによって決定されます。</li> </ul> <b>注意:</b> パスワードの暗号化を有効にするには、SET ENCRYPTION IDENTIFIED BY コマンドを使用する必要があります。
SNAPSHOT CONTROLFILE NAME TO 'filename'	スナップショット制御ファイルの名前と場所を 'filename' に設定します。CONFIGURE SNAPSHOT CONTROLFILE NAME CLEAR を実行すると、Recovery Manager ではスナップショット制御ファイル名がデフォルトに設定されます。スナップショット制御ファイル名のデフォルト値はプラットフォーム固有であり、Oracle ホームに依存します。たとえば、一部の UNIX システムでは、デフォルトは ?/dbs/snapsf_@.f です。制御ファイル名を消去し、Oracle ホームを変更すると、スナップショット制御ファイルのデフォルト位置も変更されません。スナップショット制御ファイルの名前は、そのデータベースにのみ有効なことに注意してください。プライマリ・データベースで、スナップショット制御ファイル名をデフォルト値以外に設定したとします。DUPLICATE を使用してスタンバイ・データベースを作成すると、スタンバイ・データベース上のスナップショット制御ファイルの場所は、デフォルト値に設定されます。必要であれば、スタンバイ・データベース上のスナップショットの場所をデフォルト値以外に設定できます。 <b>関連項目:</b> スナップショット制御ファイルの詳細は、『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』を参照してください。

構文の要素	説明
<code>forDbUniqueNameOption</code>	<p>Data Guard 環境の DB_UNIQUE_NAME で指定されたデータベースのリカバリ・カタログ内に Recovery Manager 構成を作成します。カタログ内のデータベースは、db_unique_name を使用して 1 つのみ指定することも、ターゲット・データベースの DBID を共有するすべてのデータベースを ALL で指定することもできます（後者の場合、DBID は SET DBID コマンドで指定することもできます）。</p> <p>Data Guard 環境で操作を実行する場合は、リカバリ・カタログが必要です。Recovery Manager は、マウント済またはオープン状態のデータベース（プライマリとスタンバイのいずれでも可）に TARGET として接続されているか、ターゲット・データベースを SET DBID コマンドで指定する必要があります。したがって、この句を使用すると、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 TARGET として接続しなくても、スタンバイ・データベースの永続的な構成を作成できます。たとえば、スタンバイ・データベースの作成後に構成を適用できるように、データベースの作成前にこの構成を作成することができます（2-86 ページの例 2-46 を参照）。</p> <p>FOR DB_UNIQUE_NAME を指定すると、リカバリ・カタログ内の構成メタデータが直接更新されます。Recovery Manager が TARGET として接続するデータベースの構成が FOR DB_UNIQUE_NAME で変更されていた場合、Recovery Manager は、リカバリ・カタログの構成メタデータに基づいてマウントされている制御ファイルを更新します。</p> <p><b>注意：</b>このデータベースに TARGET として接続していないときに、CONFIGURE をスタンバイ・データベースでローカルに実行し、次に同じデータベースに対して CONFIGURE FOR DB_UNIQUE_NAME を実行できます。この場合、リカバリ・カタログ内の構成が、そのデータベースの制御ファイル内の構成をオーバーライドします。</p>

**delalConf**

この副次句は、アーカイブ REDO ログの削除方針の永続構成を管理します。

構文の要素	説明
ARCHIVELOG DELETION POLICY	<p>アーカイブ REDO ログが削除可能になる条件を決定します。</p> <p>アーカイブ・ログの削除方針は、ログのすべてのアーカイブ先（フラッシュ・リカバリ領域も含む）に適用されます。この方針は、バックアップ・セット内のアーカイブ・ログには適用されません。</p> <p>フラッシュ・リカバリ領域内のアーカイブ REDO ログのみが自動的に削除されます。BACKUP ... DELETE INPUT、DELETE ARCHIVELOG または DELETE OBSOLETE コマンドを実行すると、ログのアーカイブ先（リカバリ領域も含む）からログを手動で削除できます。FORCE が削除コマンドに指定されていない場合、これらの削除コマンドはアーカイブ・ログの削除方針に従います。FORCE が指定されていると、削除コマンドはアーカイブ・ログの削除方針を無視します。</p> <p>リカバリ領域では、削除可能なログは可能なかぎり長く保持されます。ディスク領域が必要になると、まず最も古いログから削除されます。リカバリ領域のディスク容量が厳しくなると、Oracle Streams で必要なアーカイブ REDO ログが削除される場合があります。</p> <p><b>注意：</b>この削除方針は、外部アーカイブ REDO ログには適用されません。外部アーカイブ REDO ログは、LogMiner セッション中にロジカル・スタンバイ・データベースで受け取ったログです。これらのログは、プライマリ・データベースから転送されていますが、通常のアーカイブ・ログとは異なり、別の DBID が使用されています。外部アーカイブ REDO ログは、ロジカル・スタンバイ・データベースでバックアップおよびリストアすることはできません。</p>

構文の要素	説明
TO APPLIED ON [ALL] STANDBY	<p>次の条件が両方とも満たされている場合に、アーカイブ REDO ログが削除可能であることを指定します。</p> <ul style="list-style-type: none"> <li>■ そのアーカイブ REDO ログが、必要なスタンバイ・データベースに適用済であること。</li> <li>■ そのアーカイブ REDO ログが、BACKED UP ...TIMES TO DEVICE TYPE 削除方針で不要とされていること。BACKED UP 方針が設定されていない場合、この条件が常に満たされます。</li> </ul> <p>どのリモートの宛先が考慮されるかは、次の条件によって異なります。</p> <ul style="list-style-type: none"> <li>■ ALL を指定しなかった場合は、必須であるすべてのリモートの宛先に適用した後、アーカイブ・ログは削除可能になります。</li> <li>■ ALL を指定した場合、アーカイブ・ログは、すべてのリモートの宛先で適用済または消費済であれば削除可能です（リモートの宛先が必須かどうかは関係しません）。</li> </ul> <p>たとえば、スタンバイ・データベース <code>sby1</code> のみがログを受け取るリモートの宛先でも、他のリモートの宛先が、<code>sby1</code> の同じ場所を参照してログを適用するとします。ALL を指定すると、<code>sby1</code> は、プライマリ・データベース上のログが <code>sby1</code> で必要なくなるとすぐに消費済のマークを付けますが、同じ場所を参照する他のリモートの宛先で適用または消費されるまで、このログの削除は許可されません。</p> <p><b>注意:</b> TO APPLIED 句を、NONE 句または TO SHIPPED 句のいずれかと同時に指定することはできません。</p> <p><b>関連項目:</b> 詳細は、『Oracle Data Guard 概要および管理』を参照してください。</p>
BACKED UP <i>integer</i> TIMES TO DEVICE TYPE <i>deviceSpecifier</i>	<p>次の条件が両方とも満たされている場合に、アーカイブ REDO ログが削除可能であることを指定します。</p> <ul style="list-style-type: none"> <li>■ 指定されたデバイス・タイプ上に、指定された数のアーカイブ・ログ・バックアップが存在すること。</li> <li>■ そのログが、TO SHIPPED ON ...STANDBY または TO APPLIED ON ...STANDBY の削除方針で不要とされていること。いずれの削除方針も設定されていない場合、この条件は常に満たされます。</li> </ul> <p>この句を指定して削除方針を構成すると、指定したデバイス・タイプに <i>integer</i> 回のバックアップが存在しないかぎり、BACKUP ARCHIVELOG コマンドによってログがコピーされます。ログの <i>integer</i> 回のバックアップが存在する場合は、BACKUP ARCHIVELOG コマンドはログをスキップします。このように、アーカイブ・ログの削除方針は、BACKUP ARCHIVELOG コマンドのデフォルトの NOT BACKED UP <i>integer</i> 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、この削除方針をオーバーライドできます。</p> <p><b>関連項目:</b> 3-15 ページの「<i>deviceSpecifier</i>」を参照してください。</p>

構文の要素	説明
TO NONE	<p>アーカイブ・ログの削除方針を無効にします。これは、デフォルトの設定です。アーカイブ REDO ログは、フラッシュ・リカバリ領域の内側または外側に配置できます。ログは、それがどこに配置されていても、手動コマンドで削除できます。フラッシュ・リカバリ領域内のログのみがデータベースで自動的に削除できます。</p> <p>削除方針が NONE に設定されている場合は、次の両方の条件が満たされる場合に、アーカイブ REDO ログ・ファイルは削除可能であるとみなされます。</p> <ul style="list-style-type: none"> <li>■ アーカイブ REDO ログ・ファイルが、フラッシュ・リカバリ領域の内部に格納されているか、外部に格納されているかにかかわらず、LOG_ARCHIVE_DEST_n で指定されたリモートの宛先に転送されていること。</li> <li>■ フラッシュ・リカバリ領域に格納されているアーカイブ REDO ログ・ファイルが、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、またはバックアップ保存方針に従って不要とされている。</li> </ul> <p>バックアップ保存方針でログが不要であるとみなされるのは、そのログが保証付きリストア・ポイントで必要とされていないことに加え、フラッシュバック・データベースからも必要とされていない場合のみです。</p> <p>SYSDATE-'DB_FLASHBACK_RETENTION_TARGET' より後に作成されたアーカイブ REDO ログは、フラッシュバック・データベースに必要です。</p> <p>たとえば、アーカイブ・ログが、必須であるリモートの宛先に転送済であるとし、このログは、リカバリ期間の保存方針によると不要ですが、まだバックアップされていません。その場合、ログは削除可能です。または、ログが不要になり、SBT にバックアップ済であるとし、ただし、必須であるリモートの宛先には転送されていません。その場合は、ログは削除可能ではありません。</p> <p>削除方針が NONE に設定されているときに、フラッシュ・リカバリ領域外にあるアーカイブ REDO ログに対して削除コマンドを実行したとすると、Recovery Manger は、削除コマンドに指定された条件のみに基づいて処理を行います。</p>
TO SHIPPED TO [ALL] STANDBY	<p>次の条件が両方も満たされている場合に、アーカイブ REDO ログが削除可能であることを指定します。</p> <ul style="list-style-type: none"> <li>■ そのアーカイブ REDO ログが、必須であるリモートの宛先に転送済であること。</li> <li>■ そのアーカイブ REDO ログが、BACKED UP ...TIMES TO DEVICE TYPE 削除方針で不要とされていること。BACKED UP の削除方針が設定されていない場合、この条件が常に満たされます。</li> </ul> <p>どのリモートの宛先が考慮されるかは、次の条件によって異なります。</p> <ul style="list-style-type: none"> <li>■ ALL を指定しなかった場合は、必須であるリモートの宛先のみ転送された後、アーカイブ REDO ログは削除可能です。</li> <li>■ ALL を指定した場合は、ログは、すべてのリモートの宛先に転送されれば削除可能です（リモートの宛先が必須かどうかは、関係しません）。</li> </ul> <p><b>注意:</b> TO SHIPPED 句を、NONE 句または TO APPLIED 句のいずれかと同時に指定することはできません。</p> <p><b>関連項目:</b> 詳細は、『Oracle Data Guard 概要および管理』を参照してください。</p>



**backupConf**

この副次句は、**BACKUP** コマンドに関する永続構成を管理します。構成の1つに、バックアップの最適化があります。バックアップの最適化を有効にした場合、ファイルがデバイス・タイプにバックアップ済であれば、それと同じファイルが同じデバイス・タイプにバックアップされることはありません。

バックアップの最適化を行う場合に、ファイルが同じかどうかおよびファイルがスキップされる可能性があるかどうかを判断する基準を表 2-3 に示します。また、この表では、バックアップの最適化が有効で、同一ファイルのバックアップをスキップするかどうかを決定する必要がある場合に、**Recovery Manager** によって使用されるアルゴリズムも説明します。**Recovery Manager** によってバックアップがスキップされない場合は、指定されたそのとおりにバックアップが作成されます。

表 2-3 バックアップの最適化アルゴリズム

ファイル・タイプ	同一ファイル条件	バックアップの最適化が有効な場合のバックアップ・アルゴリズム
データファイル	データファイルの DBID、チェックポイント SCN、作成 SCN および RESETLOGS の SCN と時刻が、すでにバックアップ内にあるデータファイルと同じである必要があります。データファイルは、NORMAL モードでオフラインされ、読取り専用で、正常にクローズされている必要があります。	<p>リカバリ期間ベースの保存方針が有効な場合、<b>Recovery Manager</b> がデータファイルをスキップするかどうかはバックアップ・メディアによって決まります。</p> <p>テープへのバックアップの場合、最新のバックアップがリカバリ期間よりも古ければ、同一データファイルのバックアップが存在していても、<b>Recovery Manager</b> によってデータファイルのバックアップがもう 1 つ取られます。こうして、期限切れのテープのリサイクルが可能になります。</p> <p>ディスクへのバックアップの場合は、同一データファイルがディスク上で使用可能であれば、そのバックアップがリカバリ期間の開始時よりも古くても、<b>Recovery Manager</b> によってバックアップはスキップされません。期間ベースの保存方針では、古いほうのバックアップも必要とされるかぎり保持されます。</p> <p>保存方針が CONFIGURE RETENTION POLICY TO REDUNDANCY <math>r</math> を使用して有効になっている場合は、指定したデバイス上に同一ファイルのバックアップが <math>n</math> 個以上存在するときのみ、<b>Recovery Manager</b> によってバックアップがスキップされます（ここで、<math>n=r+1</math> です）。</p> <p>有効になっている保存方針がない場合は、指定したデバイス上に同一ファイルのバックアップが <math>n</math> 個以上存在するときのみ、<b>Recovery Manager</b> によってバックアップがスキップされます。この <math>n</math> の値は、<b>Recovery Manager</b> によって次の優先順位で検索されます（リストの上位にある値が、下位にある値よりも優先されます）。</p> <ol style="list-style-type: none"> <li>1. BACKUP ... COPIES <math>n</math></li> <li>2. SET BACKUP COPIES <math>n</math></li> <li>3. CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE ... TO <math>n</math></li> <li>4. <math>n=1</math></li> </ol>
アーカイブ REDO ログ	アーカイブ REDO ログのスレッド、順序番号および RESETLOGS の SCN と時刻が、すでにバックアップ内にあるアーカイブ・ログと同じである必要があります。	<p>指定したデバイス上に同一ファイルのバックアップが <math>n</math> 個以上存在するときのみ、バックアップがスキップされます。この <math>n</math> の値は、<b>Recovery Manager</b> によって次の優先順位で検索されます（リストの上位にある値が、下位にある値よりも優先されます）。</p> <ol style="list-style-type: none"> <li>1. BACKUP ... COPIES <math>n</math></li> <li>2. SET BACKUP COPIES <math>n</math></li> <li>3. CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE ... TO <math>n</math></li> <li>4. <math>n=1</math></li> </ol>
バックアップ・セット	バックアップ・セットのレコード ID およびタイムスタンプが、既存のバックアップ・セットと同じである必要があります。	<p>指定したデバイス上に同一ファイルのバックアップが <math>n</math> 個以上存在するときのみ、バックアップがスキップされます。デフォルトでは、<math>n=1</math> です。<math>n</math> の値が他にある場合は、次の優先順位で検索されます（リストの上位にある値が、下位にある値よりも優先されます）。</p> <ol style="list-style-type: none"> <li>1. BACKUP ... COPIES <math>n</math></li> <li>2. SET BACKUP COPIES <math>n</math></li> <li>3. <math>n=1</math></li> </ol>

構文の要素	説明
<pre>{ARCHIVELOG   DATAFILE} BACKUP COPIES FOR DEVICE TYPE <i>deviceSpecifier</i> TO integer</pre>	<p>指定したデバイス・タイプ上での、DATAFILE（データファイルと制御ファイルの両方）または ARCHIVELOG ファイルのバックアップ・セットごとのコピー数を指定します（2-84 ページの例 2-41 を参照）。作成できるコピー数は、1（デフォルト）～4 です。</p> <p>Recovery Manager は、バックアップをディスクまたはテープのいずれかに多重化できますが、テープとディスクに同時に多重化することはできません。テープにバックアップを行う場合は、コピー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ます。また、COPIES が 2 以上の場合、ターゲット・データベースで BACKUP_TAPE_IO_SLAVES 初期化パラメータを有効にする必要があります。</p> <p>制御ファイルの自動バックアップが多重化されることはありません。また、フラッシュ・リカバリ領域では多重化が許可されません。</p> <p>BACKUP コマンドまたは SET BACKUP COPIES コマンドで多重化が指定された場合、CONFIGURE の設定はオーバーライドされます。</p>
<pre>BACKUP OPTIMIZATION [ON   OFF   CLEAR]</pre>	<p>バックアップの最適化を ON または OFF（デフォルト）にします。CLEAR を指定すると、最適化はデフォルト値の OFF に戻ります。</p> <p>バックアップの最適化が使用可能になるのは、次の条件がすべて満たされている場合です。</p> <ul style="list-style-type: none"> <li>■ CONFIGURE BACKUP OPTIMIZATION ON コマンドを実行済の場合。</li> <li>■ BACKUP DATABASE、BACKUP ARCHIVELOG（ALL または LIKE オプションを指定）BACKUP BACKUPSET ALL、BACKUP RECOVERY AREA、BACKUP RECOVERY FILES または BACKUP DATAFILECOPY を実行する場合。</li> <li>■ Recovery Manager のジョブに、単一のデバイス・タイプのチャンネルのみが使用される場合。</li> </ul> <p>最適化によって、ファイルがデバイス・タイプにバックアップ済である場合は、同じファイルが同じデバイス・タイプにバックアップされないようになります。Recovery Manager では、バックアップ中に、バックアップの最適化によってすべてのファイルがスキップされてもエラーは発行されません。バックアップ保存方針は、バックアップの最適化でどのファイルがスキップされるかに影響することに注意してください。</p> <p>2 つのファイルが同一の場合、その内容が正確に同一である必要があります。ファイルが同一とみなされる条件の詳細は、2-77 ページの表 2-3 を参照してください。バックアップ・ピースをディスク上または Oracle Secure Backup で管理されているメディア上に作成する場合、UNDO データがアクティブなトランザクションに属していなければ、最適化によって、そのデータはバックアップから除外されます。</p> <p><b>注意：</b>BACKUP ...DELETE INPUT を実行すると、バックアップ中に最適化によってスキップされるかどうかに関係なく、指定したアーカイブ REDO ログがすべて削除されます。</p> <p><b>注意：</b>BACKUP コマンドの FORCE オプションを使用すると、バックアップの最適化をオーバーライドできます。</p> <p><b>関連項目：</b>Recovery Manager によってファイルのバックアップをスキップできるかどうかを判断する方法は、『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』を参照してください。</p>
<pre>EXCLUDE FORTABLESPACE <tablespace_name [clear]<="" pre=""> </tablespace_name></pre>	<p>指定した表領域を BACKUP DATABASE コマンドから除外します（2-85 ページの例 2-43 を参照）。SYSTEM 表領域は除外できないため注意してください。</p> <p>デフォルトでは、各表領域は除外されません。つまり、除外機能は無効化されています。除外は個々のデータファイルではなく表領域の属性として格納されるため、将来この表領域に追加されるすべてのファイルに適用されます。表領域を除外した後、その表領域に対して CONFIGURE ...CLEAR を実行すると、除外されていないデフォルトの構成に戻ります。</p> <p>その場合も、BACKUP コマンドで明示的に指定するか、BACKUP DATABASE コマンドで NOEXCLUDE オプションを指定すると、除外した表領域をバックアップできます。</p>

構文の要素	説明
MAXSETSIZE	<p>チャンネル上で作成される各バックアップ・セットの最大サイズを指定します。CLEAR オプションを指定すると、MAXSETSIZE をデフォルト値の UNLIMITED に戻すことができます。</p> <p><b>注意:</b> このオプションは、BACKUP AS COPY では無視されます。</p>
TO <i>sizeSpec</i>	各バックアップ・セットの最大サイズを、 <i>integer</i> (GB、KB または MB) で指定します。
TO UNLIMITED	バックアップ・セットのサイズを制限しません。
RETENTION POLICY	<p>Recovery Manager で不要マークが付けられた、つまり不要となり削除可能になっているバックアップ・セットおよびコピーについて、現行の永続的な方針を指定します。</p> <p>時間が経過すると、保存方針で指定された条件に従ってバックアップ・セットとコピーに不要マークが付けられます。リカバリ領域内の不要なバックアップ・セットおよびコピーは、領域が必要になると Recovery Manager によって自動的に削除されます。リカバリ領域外にある不要なファイルが自動的に削除されることはありません。削除するには、DELETE OBSOLETE コマンドを手動で実行する必要があります。</p> <p>バックアップの場合、保存方針の基本単位はバックアップ・セット (バックアップ・ピースではない) またはイメージ・コピーです。たとえば、BACKUP AS BACKUPSET COPIES 4 TABLESPACE <i>users</i> を実行すると、4つの同一バックアップ・ピースに多重化されるバックアップ・セットが1つ作成されます。保存方針では、これは4つの別々のバックアップではなく、1つのバックアップとみなされません。</p> <p><b>注意:</b> CLEAR オプションを使用すると、RETENTION POLICY をデフォルトの REDUNDANCY 1 に戻すことができます。</p>
TO NONE	保存方針機能を使用禁止にします。Recovery Manager では、バックアップ・セットおよびコピーは不要とみなされません。
TO RECOVERY WINDOW	Recovery Manager でデータベースをリカバリ可能な時間枠を指定します。
OF <i>integer</i> DAYS	<p>時間枠は、現在の時刻 (SYSDATE) からリカバリを必要とする最も過去の日付であるリカバリ可能ポイントまでです。リカバリ可能ポイントは、SYSDATE - <i>integer</i> で指定された日数分過去の時点を示します。</p> <p><b>注意:</b> REDUNDANCY オプションと RECOVERY WINDOW オプションを同時に指定することはできません。一度に有効な保存方針のタイプは1つのみです。</p>
TO REDUNDANCY <i>integer</i>	<p>データファイルおよび制御ファイルごとに、<i>integer</i> で指定した数のバックアップを保存するように指定します。保存方針のデフォルト設定は、REDUNDANCY 1 です。</p> <p><i>integer</i> で指定した数よりも多くのデータファイルまたは制御ファイルが存在する場合、その余分なファイルには不要マークが付けられます。次に、Recovery Manager では保存されているバックアップのうち最も古いものが判断され、そのバックアップより古いすべてのアーカイブ REDO ログおよびログのバックアップに不要マークが付けられます。DELETE OBSOLETE コマンドを実行すると、不要なデータファイル・バックアップ (全体または増分)、制御ファイルのバックアップ、およびアーカイブ・ログのバックアップまたはイメージ・コピーが削除されます。</p> <p><b>注意:</b> REDUNDANCY オプションと RECOVERY WINDOW オプションを同時に指定することはできません。一度に有効な保存方針のタイプは1つのみです。</p>

**cfauConf**

この副次句は、制御ファイルの自動バックアップに関する永続構成を作成します。

構文の要素	説明
CONTROLFILE AUTOBACKUP	<p>制御ファイルの自動バックアップ機能を制御します。</p> <p><b>注意:</b> リカバリ・カタログを使用せずに Recovery Manager を使用する場合は、制御ファイルの自動バックアップ機能を有効にすることをお勧めします。</p>
ON	<p>次のような状況で、制御ファイルの自動バックアップを実行するように指定します。</p> <ul style="list-style-type: none"> <li>■ Recovery Manager プロンプトで <b>BACKUP</b> または <b>CREATE CATALOG</b> コマンドが発行された後。 <ul style="list-style-type: none"> <li><b>RUN</b> ブロック内の <b>BACKUP</b> コマンドの後に、<b>BACKUP</b> 以外のコマンドが続いている場合。</li> </ul> </li> <li>■ 各 <b>RUN</b> ブロックの終わり（そのブロックの最後のコマンドが <b>BACKUP</b> だった場合）。</li> <li>■ <b>ARCHIVELOG</b> モードでのデータベース構造の変更後。<b>NOARCHIVELOG</b> モードでは、構造の変更後にデータベースの自動バックアップは実行されません。 <p>表領域の追加、表領域やデータファイルの状態の変更（オンライン化など）、新規オンライン <b>REDO</b> ログの追加、ファイル名の変更、新規 <b>REDO</b> スレッドの追加、フラッシュバック・データベースの有効化または無効化などの、構造の変更。前述の状況で発生する自動バックアップとは異なり、このタイプの自動バックアップはディスクにのみ行われます。<b>CONFIGURE CONTROLFILE AUTOBACKUP FOR DEVICE TYPE DISK</b> を実行すると、デフォルト以外のディスクの場所を設定できます。</p> <p>リカバリ・カタログを使用せずに Recovery Manager を使用する場合は、制御ファイルの自動バックアップを行うことをお勧めします。</p> <p>バックアップ・ジョブまたはコピー・ジョブで割り当てた最初のチャネルによって自動バックアップが作成され、独自のバックアップ・セットに格納され、構造の自動バックアップ後にデフォルトのディスク・チャネルによってバックアップが作成されます。Recovery Manager は、制御ファイルとサーバー・パラメータ・ファイルと同じバックアップ・ピースに書き込みます。制御ファイルの自動バックアップが完了すると、データベースによってバックアップ・ピースへのフルパスとデバイス・タイプを含むメッセージがアラート・ログに書き込まれます。</p> <p>ディスク上の自動バックアップのデフォルトの場所は、フラッシュ・リカバリ領域（構成されている場合）またはプラットフォーム固有の場所（構成されていない場合）です。現行の制御ファイルは、デフォルトのフォーマット <b>%F</b> を使用して自動的にバックアップされます。<b>CONFIGURE CONTROLFILE AUTOBACKUP FORMAT</b> および <b>SET CONTROLFILE AUTOBACKUP FORMAT</b> コマンドを使用して場所およびファイル名の形式を変更できます。</p> <p>制御ファイルの自動バックアップを多重化する（つまり、自動バックアップを複数の場所へ書き込む）ように Recovery Manager を構成することはできません。制御ファイルのバックアップを複数作成するには、バックアップ・ジョブの最後のコマンドを <b>BACKUP CURRENT CONTROLFILE FORMAT</b> にします。これにより、<b>FORMAT</b> で指定した場所に制御ファイルがバックアップされてから、自動バックアップが実行されます。</p> <p><b>注意:</b> <b>RUN</b> ブロック内または Recovery Manager プロンプトのいずれかで <b>SET CONTROLFILE AUTOBACKUP FORMAT</b> コマンドを指定すると、そのセッション内のみで構成済の自動バックアップ形式がオーバーライドされます。優先順位は次のとおりです。</p> <ol style="list-style-type: none"> <li>1. <b>RUN</b> ブロック内の <b>SET</b></li> <li>2. <b>RMAN</b> プロンプトの <b>SET</b></li> <li>3. <b>CONFIGURE CONTROLFILE AUTOBACKUP FORMAT</b></li> </ol> <p>自動バックアップ形式は、<b>CONFIGURE CONTROLFILE AUTOBACKUP</b> が <b>OFF</b> に設定されていても構成できますが、この場合、Recovery Manager では自動バックアップが生成されません。Recovery Manager で自動バックアップを作成するには、<b>CONFIGURE CONTROLFILE AUTOBACKUP</b> を <b>ON</b> に設定する必要があります。</p> </li> </ul>

構文の要素	説明
OFF	自動バックアップ機能を無効化します（デフォルト）。 このコマンドが OFF の場合、データファイル 1 を含むすべての BACKUP コマンドは、現行の制御ファイルおよびサーバー・パラメータ・ファイルをバックアップ・セットに自動的に組み込みます。それ以外の場合、Recovery Manager はこれらのファイルを含めません。
CLEAR	この構成をデフォルト設定の OFF に戻します。
FORMAT FOR DEVICE TYPE <i>deviceSpecifier</i> TO <i>formatSpec</i>	指定したデバイス・タイプへの制御ファイルの自動バックアップについて、デフォルトの場所とファイル名の形式を構成します (2-86 ページの例 2-45 を参照)。 デフォルトのフォーマットは、どのデバイスの場合も %F です。CONFIGURE で指定するデフォルトのフォーマット文字列には、%F 置換変数を含める必要があります。他の置換変数を使用すると、エラーが発生します。CLEAR を指定すると、フォーマットがデフォルトの %F に戻ります。 フラッシュ・リカバリ領域が有効で、フォーマットがデフォルトの '%F' の場合、Recovery Manager は autobackup というディレクトリのリカバリ領域に自動バックアップを作成します。有効になっていない場合は、オペレーティング・システム固有の場所 (UNIX、Linux および Windows では ?/dbs) が自動バックアップのデフォルトの場所になります。 SHOW コマンドの出力にある文字列 # default は、Recovery Manager がデフォルトのフォーマットを使用しているタイミングを示します。ディスク・フォーマットを手動で '%F' に構成すると、リカバリ領域が有効な場合でも、Recovery Manager はオペレーティング・システム固有のデフォルトの場所に自動バックアップを作成します。自動バックアップがリカバリ領域に作成されるようにフォーマットをデフォルトに戻すには、CONFIGURE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK CLEAR を実行します。 <i>formatSpec</i> で、自動ストレージ管理ディスク・グループを指定できます。次の例では、ASM ディスク・グループのチャンネルを構成します。 CONFIGURE CONTROLFILE AUTOBACKUP FOR DEVICE TYPE DISK TO '+dgroup1'; <b>関連項目</b> : %F 置換変数の意味については、3-21 ページの「 <i>formatSpec</i> 」を参照してください。

## deviceConf

この副次句は、チャンネルおよびデバイスに関する永続構成を作成します。

### 構成済チャンネルの名前

Recovery Manager は構成済チャンネルの名前を決定することに注意してください。Recovery Manager で使用される規則は、ORA\_*devicetype*\_n です。*devicetype* は DISK または sbt\_tape などのユーザー・デバイスのタイプ、n はチャンネル番号です。接頭辞 ORA\_ で始まるチャンネル名は、Recovery Manager 専用に予約されています。名前が ORA\_ で始まるチャンネルを手動で割り当てることはできません。

---

**注意:** デバイス・タイプの sbt と sbt\_tape はシノニムですが、Recovery Manager への入力が sbt と sbt\_tape のいずれの場合でも、Recovery Manager の出力は、常に sbt\_tape が表示されます。

---

Recovery Manager では、最初の DISK チャンネルに ORA\_DISK\_1、2 番目のチャンネルに ORA\_DISK\_2 という名前が付けられ、以降も同様に名前が付けられます。sbt チャンネルの場合は、最初のチャンネルに ORA\_SBT\_TAPE\_1、2 番目のチャンネルに ORA\_SBT\_TAPE\_2 という名前が付けられ、以降も同様に名前が付けられます。チャンネルをパラレル化すると、Recovery Manager によってチャンネルは常に番号順に割り当てられます。番号は、1 から始まり、パラレル化の設定値 (CONFIGURE DEVICE TYPE ... PARALLELISM n) で終わります。

特定の構成済チャンネルで **BACKUP** または **RESTORE** ジョブを実行するには、システムで生成されたチャンネル名を使用します。CONFIGURE CHANNEL コマンド (*deviceConf* 句を参照) でチャンネル番号を指定すると、システム生成のチャンネル名に同じ番号が使用されます。

チャンネルの自動割当ては、メンテナンス・コマンドにも適用されます。Recovery Manager で自動メンテナンス・チャンネルを割り当てる場合、他の自動割当てチャンネルと同じネーミング規則が使用されます。

### Oracle RAC 環境の構成済チャンネル

すべての Oracle RAC インスタンスの SYSDBA パスワードが同じであるかぎり、**ALLOCATE CHANNEL** または CONFIGURE コマンドの CONNECT オプションでパスワードを指定する必要はありません。user@database 形式の接続文字列を使用すると、Recovery Manager セッションの開始時に TARGET 接続で使用されたものと同じパスワードが自動的に使用されます (2-85 ページの例 2-43 を参照)。

構文の要素	説明
[AUXILIARY] CHANNEL [integer] DEVICE TYPE <i>deviceSpecifier</i>	<p>構成または消去する標準または AUXILIARY チャンネルと、そのチャンネルのデバイス・タイプを指定します。</p> <p><b>注意:</b> RUN コマンド内で <b>ALLOCATE CHANNEL</b> を指定して割り当てたチャンネルは、構成済の自動チャンネルをオーバーライドします。</p> <p>汎用チャンネルを構成するか、チャンネルを番号で指定できます。この場合、integer は 254 以下の値です。番号付けされたチャンネルの実例は、2-85 ページの例 2-43 を参照してください。</p> <p>AUXILIARY を指定すると、この構成は補助インスタンスで割り当てられたチャンネルにのみ使用されます。ターゲット・インスタンスで割り当てられたチャンネルとは異なるパラメータが補助チャンネルに必要な場合は、補助チャンネルの構成情報を指定します。補助デバイス構成を指定しない場合、Recovery Manager はターゲット・データベースのデバイス構成を使用して補助チャンネルを構成します。</p> <p>チャンネル・オプションは、1 つ以上指定する必要があります。たとえば、CONFIGURE CHANNEL 2 DEVICE TYPE DISK のようなコマンドは発行できませんが、CONFIGURE CHANNEL 2 DEVICE TYPE DISK MAXPIECESIZE 2500K というコマンドは発行できます。</p> <p>指定したデバイス・タイプの汎用チャンネルについて、新規コマンドにより、そのデバイス・タイプの以前の設定が消去されます。次のコマンドを実行するとします。</p> <pre>CONFIGURE CHANNEL DEVICE TYPE sbt MAXPIECESIZE 1G; CONFIGURE CHANNEL DEVICE TYPE sbt FORMAT 'bkup_%U';</pre> <p>2 番目のコマンドでは、最初のコマンドの MAXPIECESIZE 設定が消去されます。</p> <p><b>注意:</b> Recovery Manager では、<b>BACKUP</b> コマンドで同時に複数のデバイス・タイプに対して自動チャンネルが割り当てられることはありません。</p> <p><b>関連項目:</b> チャンネル番号で指定した自動チャンネルを構成する方法は、『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』を参照してください。</p>
<i>allocOperandList</i>	<p>構成済チャンネルの制御オプションを指定します。</p> <p>デフォルト以外の CONNECT または PARMS オプションを使用してチャンネルを構成し、バックアップまたはコピーを作成する場合は、同じ構成済チャンネルを使用するか、同じオプションを使用して手動でチャンネルを割り当てて、これらのバックアップをリストアまたはクロスチェックする必要があります。</p> <p>FORMAT パラメータで、自動ストレージ管理ディスク・グループを指定できます。次の例では、ASM ディスク・グループのチャンネルを構成します。</p> <pre>CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '+dgroup1';</pre> <p><b>関連項目:</b> 3-2 ページの「<i>allocOperandList</i>」を参照してください。</p>

構文の要素	説明
CLEAR	<p>指定したチャンネルを消去します。たとえば、CONFIGURE CHANNEL 1 DEVICE TYPE DISK CLEAR はチャンネル 1 のみをデフォルトに戻しますが、CONFIGURE CHANNEL DEVICE TYPE DISK CLEAR は汎用ディスク・チャンネルをデフォルトに戻します。CLEAR を指定する場合は、他のチャンネル・オプション (PARMS など) を指定できないため注意してください。</p>
DEFAULT DEVICE TYPE TO <i>deviceSpecifier</i>	<p>自動チャンネルのデフォルトのデバイス・タイプを指定します。デフォルトのデバイス・タイプは DISK です。CLEAR を指定すると、デフォルトのデバイス・タイプが DISK に戻ります。</p> <p>デフォルトでは、BACKUP コマンドで割り当てることができるのは、デフォルトのデバイス・タイプのチャンネルのみです。たとえば、自動チャンネルを DISK および sbt に対して構成し、デフォルトのデバイス・タイプを sbt に設定した場合、BACKUP DATABASE コマンドを実行すると、Recovery Manager ではテープ・チャンネルのみが割り当てられます。この動作をオーバーライドするには、RUN コマンドでチャンネルを手動で割り当てる方法と、BACKUP コマンド自体で DEVICE TYPE を指定する方法があります (2-84 ページの例 2-41 を参照)。</p> <p>RESTORE コマンドでは、デフォルトのデバイス・タイプに関係なく、すべての構成済デバイス・タイプの自動チャンネルが割り当てられます。RESTORE コマンドは、構成済の各デバイス・タイプの PARALLELISM 設定に従います。</p>
DEVICE TYPE <i>deviceSpecifier</i>	<p>この CONFIGURE コマンドで指定された設定の適用対象としてデバイス・タイプ (ディスクまたは sbt) を指定します。CLEAR オプションを使用すると、このデバイスのバックアップ・タイプおよび並列度の設定がデフォルトにリセットされます。</p> <p>CONFIGURE DEVICE TYPE コマンドを実行して、デバイス・タイプのデフォルト設定を構成し、このデバイス・タイプに対して CONFIGURE CHANNEL を実行しなかった場合、Recovery Manager は、他のチャンネル制御オプションを使用せずに、すべてのチャンネルを割り当てます。sbt デバイスを構成し、バックアップを次のように実行するとします。</p> <pre data-bbox="600 1029 1055 1092">CONFIGURE DEVICE TYPE sbt PARALLELISM 1; BACKUP DEVICE TYPE sbt DATABASE;</pre> <p>実際には、Recovery Manager では、このバックアップを次のように実行します。</p> <pre data-bbox="600 1155 1153 1291">RUN {   ALLOCATE CHANNEL ORA_SBT_TAPE_1 DEVICE TYPE sbt;   BACKUP DATABASE; }</pre>
BACKUP TYPE TO [[COMPRESSED] BACKUPSET   COPY]	<p>ディスク・バックアップまたはテープ・バックアップのデフォルトのバックアップ・タイプを構成します。SBT デバイスの場合、COPY オプションはサポートされていません。ディスクの場合のデフォルトは、BACKUPSET です。</p> <p>BACKUP TYPE を BACKUPSET に設定した場合に BACKUP コマンドを使用すると、バックアップが作成されるメディアに関係なく、常に、バックアップ・セットが作成されます。COMPRESSED オプションを指定すると、作成されるバックアップ・セットにバイナリ圧縮が使用されます。</p> <p>ディスク・バックアップのデフォルトの場所は、フラッシュ・リカバリ領域が構成されている場合はフラッシュ・リカバリ領域です。構成されていない場合は、Recovery Manager によってプラットフォーム固有の場所にバックアップが格納されます。バックアップ・ファイル名のデフォルトの形式は %U です。</p>

構文の要素	説明
PARALLELISM <i>integer</i>	<p>Recovery Manager のジョブに割り当てられるデバイス・タイプに指定された自動チャンネルの数を構成します。デフォルトでは、PARALLELISM は 1 に設定されています。</p> <p><b>注意:</b> CONFIGURE ... PARALLELISM パラメータは、チャンネルの並列度（バックアップおよびリストアの操作時に Recovery Manager が割り当てるチャンネル数）を指定します。RECOVERY_PARALLELISM 初期化パラメータは、インスタンス・リカバリで使用されるプロセス数を指定します。</p> <p>ディスク・バックアップの PARALLELISM を 2 に設定するとします (2-85 ページの例 2-42 を参照)。デフォルトのデバイス・タイプをディスクに設定し、Recovery Manager プロンプトで BACKUP DATABASE を実行すると、2 つのディスク・チャンネルが割り当てられます。Recovery Manager は、常に PARALLELISM で設定された数のチャンネルを割り当てますが、実際にはこれらのチャンネルのサブセットしか使用されない場合があります。</p> <p><b>注意:</b> 手動で番号を取得した <i>n</i> 個のチャンネルを構成する場合は、PARALLELISM 設定が <i>n</i> より大きくても小さくてもかまいません。たとえば、10 個の自動チャンネルの番号を手動で取得し、PARALLELISM を 2 や 12 に設定できます。</p> <p>デバイス・タイプの並列度を <i>n</i> に変更するには、新規の CONFIGURE DEVICE TYPE ... PARALLELISM <i>n</i> コマンドを実行します。たとえば、次のように、sbt の PARALLELISM を 3 に構成してから、2 に変更できます。</p> <pre>CONFIGURE DEVICE TYPE sbt PARALLELISM 3; CONFIGURE DEVICE TYPE sbt PARALLELISM 2;</pre>

## 例

### 例 2-40 デバイスおよびバックアップ・オプションの構成

この例では、デバイス・タイプ DISK および sbt のチャンネルを構成し、デフォルトのデバイス・タイプを sbt に設定します。また、バックアップの最適化を有効にし、リカバリ期間を 2 週間に構成します。

```
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '/disk1/backups/%U';
CONFIGURE CHANNEL DEVICE TYPE sbt PARMS 'ENV=(OB_DEVICE_1=tape1)';
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE BACKUP OPTIMIZATION ON;
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 14 DAYS;
```

### 例 2-41 デフォルト・デバイス・タイプのオーバーライド

この例では、データファイルと制御ファイルの DISK バックアップについて多重化を 2 に構成します (ただし、ディスクへの制御ファイルの自動バックアップは特例で、多重化されません)。次に、sbt をデフォルト・デバイスに構成します。

```
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 2;
CONFIGURE CHANNEL DEVICE TYPE sbt PARMS 'ENV=(OB_DEVICE_1=tape1)';
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
```

最初の BACKUP コマンドは、アーカイブ REDO ログをデフォルトの sbt チャンネルにバックアップします。2 番目の BACKUP コマンドは、データベースをディスクの場所にバックアップします。ディスク・バックアップの多重化が有効であるため、各出力バックアップ・セットのコピーが 2 つ作成されます。

```
BACKUP ARCHIVELOG ALL;
BACKUP DEVICE TYPE DISK
  DATABASE
  FORMAT '/disk1/db_backup_%U', '/disk2/db_backup_%U';
```



**例 2-42 ファイル・システムにまたがる自動チャンネルの構成**

この例では、2つのファイル・システムにまたがる自動ディスク・チャンネルを構成します。

```
CONFIGURE DEVICE TYPE DISK PARALLELISM 2;
CONFIGURE CHANNEL 1 DEVICE TYPE DISK FORMAT '/disk1/%U';
CONFIGURE CHANNEL 2 DEVICE TYPE DISK FORMAT '/disk2/%U';
```

PARALLELISM が 2 に設定されているため、次のコマンドは出力データを 2つのファイル・システム間で分割します。

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

次の LIST コマンドは、データファイルのバックアップがどのようにパラレル化されたかを示します。

```
RMAN> LIST BACKUPSET 2031, 2032;
```

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

BS Key	Type	LV Size	Device Type	Elapsed Time	Completion Time
2031	Full	401.99M	DISK	00:00:57	19-JAN-07
BP Key: 2038 Status: AVAILABLE Compressed: NO Tag: TAG20070119T100532					
Piece Name: /disk1/24i7ssnc_1_1					
List of Datafiles in backup set 2031					
File	LV	Type	Ckp SCN	Ckp Time	Name
1		Full	973497	19-JAN-07	/disk3/oracle/dbs/t_db1.f
5		Full	973497	19-JAN-07	/disk3/oracle/dbs/tbs_112.f

BS Key	Type	LV Size	Device Type	Elapsed Time	Completion Time
2032	Full	133.29M	DISK	00:00:57	19-JAN-07
BP Key: 2039 Status: AVAILABLE Compressed: NO Tag: TAG20070119T100532					
Piece Name: /disk2/25i7ssnc_1_1					
List of Datafiles in backup set 2032					
File	LV	Type	Ckp SCN	Ckp Time	Name
2		Full	973501	19-JAN-07	/disk3/oracle/dbs/t_ax1.f
3		Full	973501	19-JAN-07	/disk3/oracle/dbs/t_undo1.f
4		Full	973501	19-JAN-07	/disk3/oracle/dbs/tbs_111.f

**例 2-43 Oracle Real Application Clusters (Oracle RAC) 構成での自動チャンネルの構成**

この例では、Oracle RAC データベースに 2つのノードがあるとします。メディア・マネージャは Oracle Secure Backup です。テープ・ドライブ tape1 は node1 に直接接続され、テープ・ドライブ tape2 は node2 に直接接続されています。この例では、各クラスタ・ノードに SBT 自動チャンネルを構成します。

この例では、Oracle RAC インスタンス node1 および node2 へのチャンネル接続を示します。両方のチャンネル接続で、Recovery Manager は、ターゲット・データベース接続で入力したユーザー名とパスワードと同じものを使用します。

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 2;
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE CHANNEL 1 DEVICE TYPE sbt CONNECT '@node1'
PARMS 'ENV=(OB_DEVICE=tape1)';
CONFIGURE CHANNEL 2 DEVICE TYPE sbt CONNECT '@node2'
PARMS 'ENV=(OB_DEVICE=tape2)';
```

**例 2-44 補助ファイル名の構成**

この例では、CONFIGURE AUXNAME を使用して、データファイルの新しいファイル名を指定します。DUPLICATE コマンドによって、ディレクトリ構造が異なるリモート・ホストにデータベースが多重化されます。

```
# set auxiliary names for the datafiles
CONFIGURE AUXNAME FOR DATAFILE 1 TO '/oracle/auxfiles/aux_1.f';
CONFIGURE AUXNAME FOR DATAFILE 2 TO '/oracle/auxfiles/aux_2.f';
CONFIGURE AUXNAME FOR DATAFILE 3 TO '/oracle/auxfiles/aux_3.f';
CONFIGURE AUXNAME FOR DATAFILE 4 TO '/oracle/auxfiles/aux_4.f';

RUN
{
  ALLOCATE AUXILIARY CHANNEL dupdb1 TYPE DISK;
  DUPLICATE TARGET DATABASE TO dupdb
  LOGFILE
    GROUP 1 ('?/dbs/dupdb_log_1_1.f',
             '?/dbs/dupdb_log_1_2.f') SIZE 4M,
    GROUP 2 ('?/dbs/dupdb_log_2_1.f',
             '?/dbs/dupdb_log_2_2.f') SIZE 4M REUSE;
}
# Un-specify the auxiliary names for the datafiles so that they are not overwritten
# by mistake:
CONFIGURE AUXNAME FOR DATAFILE 1 CLEAR;
CONFIGURE AUXNAME FOR DATAFILE 2 CLEAR;
CONFIGURE AUXNAME FOR DATAFILE 3 CLEAR;
CONFIGURE AUXNAME FOR DATAFILE 4 CLEAR;
```

**例 2-45 制御ファイルの自動バックアップに使用するデフォルトの形式の指定**

次の例では、自動バックアップ機能を有効にし、DISK および sbt デバイスに対してデフォルトの自動バックアップ形式を構成します。

```
CONFIGURE CONTROLFILE AUTOBACKUP ON;
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '/disk2/%F';
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE sbt TO 'cf_auto_%F';
```

**例 2-46 スタンバイ・データベースの構成の作成**

プライマリ・データベース prod が、dgprod3 および dgprod4 という DB\_UNIQUE\_NAME 名の 2 つのスタンバイ・データベースに関連付けられているとします。Recovery Manager を起動し、TARGET として prod に接続して、リカバリ・カタログに接続するとします。次のコマンドでは、データベース dgprod3 および dgprod4 のデフォルトのデバイス・タイプを構成します。

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt
  FOR DB_UNIQUE_NAME dgprod3;
CONFIGURE DEVICE TYPE sbt PARALLELISM 2
  FOR DB_UNIQUE_NAME dgprod3;
CONFIGURE DEFAULT DEVICE TYPE TO DISK
  FOR DB_UNIQUE_NAME dgprod4;
```

この構成で 2 つのスタンバイ・データベースの制御ファイルが更新されるのは、リカバリ・カタログから制御ファイルへの逆方向の再同期が行われた後のみです。この再同期は、ユーザーが dgprod3 および dgprod4 に初めて接続するときに行われます。

次の SHOW コマンドは、dgprod3 という一意の名前を持つデータベースのデバイス・タイプの永続構成を表示します。

```
RMAN> SHOW DEVICE TYPE FOR DB_UNIQUE_NAME dgprod3;
RMAN configuration parameters for database with db_unique_name DGPROD3 are:

CONFIGURE DEVICE TYPE 'SBT_TAPE' PARALLELISM 2 BACKUP TYPE TO BACKUPSET;
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
```

次の SHOW コマンドは、リカバリ・カタログが認識しているデータベースで、DBID が 3257174182 のすべてのデータベースの永続構成を表示します (DBID の値は、先行する SET DBID コマンドで指定されます)。

```
SHOW ALL FOR DB_UNIQUE_NAME ALL;
```

#### 例 2-47 バックアップの最適化

次の例では、2-77 ページの表 2-3 で示したバックアップの最適化について説明します。バックアップの最適化が無効になっているとします。午前 9 時に、既存のすべてのアーカイブ・ログの 3 つのコピーをテープにバックアップします。バックアップをテープに多重化する場合は、BACKUP\_TAPE\_IO\_SLAVES 初期化パラメータを true にする必要があることに注意してください。

```
BACKUP DEVICE TYPE sbt COPIES 3 ARCHIVELOG ALL;
```

午前 11 時に、バックアップの最適化を有効にします。

```
CONFIGURE BACKUP OPTIMIZATION ON;
```

正午に、次のアーカイブ REDO ログのバックアップを実行します。

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

```
Starting backup at 19-JAN-07
current log archived
using channel ORA_SBT_TAPE_1
skipping archived log file /d1/db1r_605ab325_1_34_612112605.arc; already backed up 3 time(s)
skipping archived log file /d1/db1r_605ab325_1_35_612112605.arc; already backed up 3 time(s)
skipping archived log file /d1/db1r_605ab325_1_36_612112605.arc; already backed up 3 time(s)
skipping archived log file /d1/db1r_605ab325_1_37_612112605.arc; already backed up 3 time(s)
skipping archived log file /d1/db1r_605ab325_1_38_612112605.arc; already backed up 3 time(s)
skipping archived log file /d1/db1r_605ab325_1_39_612112605.arc; already backed up 3 time(s)
channel ORA_SBT_TAPE_1: starting archived log backup set
channel ORA_SBT_TAPE_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=40 RECID=170 STAMP=612270506
channel ORA_SBT_TAPE_1: starting piece 1 at 19-JAN-07
channel ORA_SBT_TAPE_1: finished piece 1 at 19-JAN-07 with 2 copies and tag
TAG20070119T110827
piece handle=2hi7t0db_1_1 comment=API Version 2.0,MMS Version 10.1.0.0
piece handle=2hi7t0db_1_2 comment=API Version 2.0,MMS Version 10.1.0.0
```

この場合、BACKUP ...COPIES の設定が CONFIGURE ...COPIES の設定をオーバーライドするため、 $n=2$  が設定されます。Recovery Manager は、sbt デバイスにログのコピーが 2 つ以上ある場合にのみ、ログのバックアップをスキップします。午前 9 時まで生成されたすべてのログについて、各ログの 3 つのコピーが sbt 上に存在するため、Recovery Manager はこれらのログのバックアップをスキップします。ただし、午前 9 時より後に生成されたすべてのログについては、2 つのコピーがバックアップされます。これは、そのログがまだテープにバックアップされていないためです。

## CONNECT

### 用途

CONNECT コマンドを使用すると、Recovery Manager とターゲット・データベース、補助データベースまたはリカバリ・カタログ・データベースとの接続を確立できます。

データベースへの SQL\*Plus 接続と同様に、データベースへの Recovery Manager 接続が指定され、認証されます。唯一異なるのは、ターゲット・データベースまたは補助データベースへの Recovery Manager 接続では SYSDBA 権限が必要なことです。AS SYSDBA キーワードは暗黙的に指定されており、明示的に指定することはできません。SQL\*Plus 使用時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。

---

**注意：** セキュリティ上、パスワードはコマンドラインにプレーン・テキストで入力しないでください。Recovery Manager プロンプトで要求された場合のみ、Recovery Manager に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。

---

**関連項目：** コマンドラインの接続オプションについては、2-244 ページの「RMAN」を参照してください。

### 前提条件

CONNECT TARGET、CONNECT CATALOG および CONNECT AUXILIARY コマンドは、そのコマンドで指定するデータベースに Recovery Manager がまだ接続されていない場合にかぎり、Recovery Manager プロンプトからのみ実行できます。別のターゲット、カタログまたは補助データベースに接続するには、新規 Recovery Manager セッションを開始する必要があります。

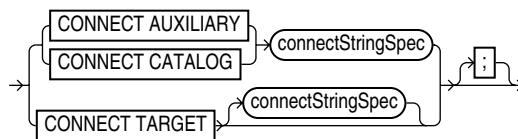
### 使用上の注意

次のすべての条件が満たされている場合、Recovery Manager セッションは NOCATALOG モードで実行されます。

- RMAN の起動時に、CATALOG または NOCATALOG を指定しなかったこと。
- Recovery Manager セッションで、CONNECT CATALOG をまだ実行していないこと。
- Recovery Manager リポジトリ接続が必要な BACKUP などのコマンドを実行すること (2-89 ページの例 2-49 を参照)。

### 構文

**connect::=**



(connectStringSpec::= (3-12 ページ))

## セマンティクス

構文の要素	説明
CONNECT AUXILIARY	Recovery Manager と補助データベース・インスタンスとの接続を確立します。 補助インスタンスは、 <a href="#">TRANSPORT TABLESPACE</a> および <a href="#">DUPLICATE</a> コマンドで使用されます。また、Recovery Manager の TSPITR 中にも使用されます。
CONNECT CATALOG	Recovery Manager とリカバリ・カタログ・データベースとの接続を確立します。 リカバリ・カタログが仮想プライベート・カタログ（「 <a href="#">CREATE CATALOG</a> 」を参照）の場合、このカタログに接続する Recovery Manager クライアントのパッチ・レベルは、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。このバージョン制限は、Oracle Database 11g の基本リカバリ・カタログへの Recovery Manager クライアント接続には影響しません。基本カタログに仮想プライベート・カタログのユーザーがいる場合も同様です。 Recovery Manager がすでにデフォルトの NOCATALOG モードである場合に、Recovery Manager セッションで CONNECT CATALOG コマンドを使用しようとする、RMAN-06445 エラーが発行されます（2-88 ページの「 <a href="#">使用上の注意</a> 」を参照）。 <b>注意</b> ：Data Guard 環境で Recovery Manager 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。
CONNECT TARGET	Recovery Manager とターゲット・データベースとの接続を確立します。 <b>注意</b> ：Data Guard 環境では、Recovery Manager はフィジカル・スタンバイ・データベースに TARGET として接続できます。リカバリ・カタログで DB_UNIQUE_NAME が認識されていないものの、DBID が登録済データベースと同じデータベースに対して CONNECT TARGET を実行すると、そのデータベースは、自動的かつ暗黙的にリカバリ・カタログに登録されます。  <i>connectStringSpec</i> データベースの接続情報を指定します。

## 例

### 例 2-48 リカバリ・カタログを使用しないターゲット・データベースへの接続

この例では、Recovery Manager を NOCATALOG モードで起動してから、Oracle Net サービス名が prod1 のターゲット・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るプロンプトが表示されます。

```
% rman NOCATALOG
RMAN> CONNECT TARGET SYS@prod1;

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

### 例 2-49 デフォルトの NOCATALOG モードでのターゲット・データベースへの接続

この例では、CATALOG または NOCATALOG のいずれも指定せずに、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。CONNECT CATALOG コマンドが実行されていないため、BACKUP コマンドを実行すると、Recovery Manager はデフォルトの NOCATALOG モードになります。

```
% rman
RMAN> CONNECT TARGET /
RMAN> BACKUP DATABASE;
```

Recovery Manager セッションのこの時点では、セッションがデフォルトの NOCATALOG モードになっているため、CONNECT CATALOG を実行できません。このセッションでカタログに接続しようすると、次のエラーが発生します。

```
RMAN> CONNECT CATALOG rman@catdb

RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-06445: cannot connect to recovery catalog after NOCATALOG has been used
```

#### 例 2-50 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補助データベースへの接続

この例では、ターゲット・データベースにはオペレーティング・システム認証機能を使用して接続し、リカバリ・カタログおよび補助データベースにはパスワード・ファイルを使用して接続します。Recovery Manager によって、パスワードの入力を求めるプロンプトが表示されます。

```
% rman
RMAN> CONNECT TARGET;

connected to target database: PROD (DBID=39525561)

RMAN> CONNECT CATALOG rman@catdb;

recovery catalog database Password: password
connected to recovery catalog database

RMAN> CONNECT AUXILIARY SYS@dupdb;

auxiliary database Password: password
connected to auxiliary database: DUPDB (not mounted)
```

# CONVERT

## 用途

CONVERT コマンドを使用すると、異なるプラットフォーム間でのトランスポートの準備として、表領域、データファイルまたはデータベースをトランスポート先プラットフォームの形式に変換できます。

Oracle Database 10g 以上のリリースでは、次の場合に CONVERT DATAFILE または CONVERT TABLESPACE が必要です。

- V\$TRANSPORTABLE\_PLATFORM.ENDIAN\_FORMAT の値が異なるプラットフォーム間でのデータファイルをトランスポートする場合。
- ENDIAN\_FORMAT が同じかどうかにかかわらず、プラットフォーム間で UNDO セグメントを持つ表領域（通常は、SYSTEM 表領域と UNDO 表領域、およびロールバック・セグメントを持つ表領域）をトランスポートする場合。通常、SYSTEM 表領域と UNDO 表領域が変換されるのは、データベース全体を変換するときのみです。

CONVERT を使用するのには、ASM に格納されているデータベースに表領域をトランスポートするような場合です。Linux の cp、Windows の COPY などのオペレーティング・システムのネイティブ・コマンドでは、ASM ディスク・グループの読み書きができません。

**関連項目：** CONVERT DATAFILE、CONVERT TABLESPACE および  
CONVERT DATABASE の使用方法の詳細は、『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』を参照してください。

## 前提条件

プラットフォームが、CONVERT コマンドでサポートされている必要があります。サポートされているプラットフォームは、V\$TRANSPORTABLE\_PLATFORM を問い合わせ確認します。プラットフォーム間の表領域トランスポートは、ソースとトランスポート先プラットフォームの両方がこのビューに含まれている場合にのみサポートされます。

ソース・データベースおよびトランスポート先データベースの両方も、初期化パラメータ COMPATIBLE が 10.0 以上に設定されて実行されている必要があります。次の互換性の前提条件に注意してください。

- COMPATIBLE の設定が 11.0.0 未満の場合、読取り専用の表領域またはトランスポートされた既存の表領域は、1 回以上読み書き両用にされていなければ、別のプラットフォームにトランスポートできません。表領域は、読み書き両用でオープンした後、すぐに読取り専用に戻すことができます。
- COMPATIBLE が 11.0.0 以上の場合、読み書き両用の表領域に関する前述の制限は適用されません。ただし、トランスポートされた既存の表領域の形式はすでに 10.0 に設定されている必要があります。つまり、それらの表領域は、トランスポートされる前に、表領域の COMPATIBLE を 10.0 に設定して読み書き両用にされている必要があります。

### CONVERT TABLESPACE の前提条件

CONVERT TABLESPACE は、ソース・データベースに TARGET として接続し、ソース・プラットフォームで表領域を変換する場合にのみ使用できます。

ソース・データベースはマウントまたはオープン状態である必要があります。変換する表領域は、変換時に読取り専用である必要があります。ソース・データベースで表領域を変換する場合は、トランスポート先データベースの状態は関係ありません。

### CONVERT DATAFILE の前提条件

CONVERT DATAFILE は、トランスポート先データベースに TARGET として接続し、トランスポート先プラットフォームでデータファイルを変換する場合にのみ使用できます。

CONVERT DATABASE ON DESTINATION で生成された CONVERT DATAFILE スクリプトを実行している場合は、NOMOUNT オプションを指定してトランスポート先データベースのインスタンスを起動する必要があります。CONVERT DATABASE ON DESTINATION で生成された CONVERT DATAFILE スクリプトを実行していない場合は、トランスポート先データベースを起動、マウントまたはオープンすることができます。

トランスポート先データベースでデータファイルのコピーを変換する場合は、ソース・データベースの状態は関係ありません。ただし、トランスポート先データベースで、CONVERT DATAFILE スクリプトをデータベース変換の一部として実行しており、そのスクリプトが (NFS マウントなどにより) ソース・データベースのデータファイルに直接アクセスしている場合は、ソース・データベースは読取り専用でオープンする必要があります。

トランスポート先のホストで表領域を変換する場合は、CONVERT DATAFILE を使用する必要があります。CONVERT TABLESPACE では、トランスポート先データベースが変換時にデータファイルを表領域に関連付けることができません。表領域に必要なデータファイルを変換した後は、そのファイルをトランスポート先データベースにトランスポートできます。

### CONVERT DATABASE の前提条件

CONVERT DATABASE は、ソース・データベースに TARGET として接続されている場合にのみ使用できます。また、ソース・データベースは読取り専用でオープンされている必要があります。CONVERT DATABASE ON DESTINATION を実行しても、CONVERT DATABASE の実行時、トランスポート先データベースの状態には関係ありません。

CONVERT DATABASE は、CONVERT TABLESPACE および CONVERT DATAFILE と同じメカニズムでデータファイルを変換するため、表領域とデータファイルの使用上の注意と制約事項も適用されます。

CONVERT DATABASE に追加される重要な前提条件として、ソースとターゲットの両方のプラットフォームで、同じエンディアン形式を共有する必要があるということがあります。たとえば、データベースを Microsoft Windows から Linux for x86 (ともにリトル・エンディアン) にトランスポートしたり、HP-UX から AIX (ともにビッグ・エンディアン) にトランスポートすることはできますが、Solaris から Linux x86 へはトランスポートできません。ただし、ターゲット・プラットフォームに新しいデータベースを手動で作成すると、CONVERT TABLESPACE または CONVERT DATAFILE を使用して、ソース・データベースの表領域を個々にトランスポートできます。

ソース・プラットフォームとトランスポート先プラットフォームのエンディアン形式は同じでも、トランスポート先プラットフォームのデータベースのデータファイルは、ソース・ホストまたはトランスポート先ホストのいずれかで変換する必要があります。エンディアン形式が同じ場合は変換を行う必要がないプラットフォーム間での表領域のトランスポートとは異なり、データベース全体のトランスポートでは、UNDO セグメント内のブロックなどの特定のタイプのブロックを再フォーマットして、トランスポート先プラットフォームとの互換性を確保する必要があります。

## 使用上の注意

変換処理はインプレースで実行されないため、入力ファイルが CONVERT によって変更されることはありません。かわりに、Recovery Manager によって指定された出力先に変換済ファイルが書き込まれます。

### データ型の制約事項

CONVERT では、エンディアン変換が必要なユーザー・データ型は処理されません。プラットフォーム固有のフォーマットでデータを格納する、基礎となる型に基づいて作成されたデータベースの間でオブジェクトをトランスポートするには、データ・ポンプ・インポートおよびエクスポート・ユーティリティを使用してください。

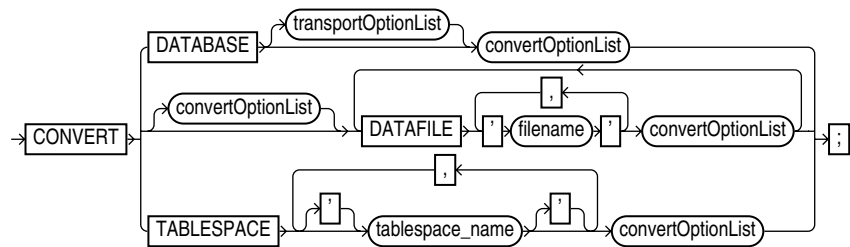


Oracle Database 10g より前のリリースでは、UTF8 などの可変幅キャラクタ・セットで作成された CLOB は、エンディアンに依存する固定幅フォーマットで格納されていました。CONVERT コマンドでは、これらの CLOB に対する変換は実行されません。かわりに、Recovery Manager によって、各 LOB 列のエンディアン形式が取得され、ターゲット・データベースに伝播されます。その後、SQL レイヤーでこのデータを読み取ると、いずれかのエンディアン形式に正確に基づいたデータが解析され、表領域が書き込み可能な場合はエンディアンに依存しない方法で書き込まれます。Oracle Database 10g 以上のリリースで作成された CLOB は、プラットフォームに依存しないキャラクタ・セット AL16UTF16 で格納されます。

**関連項目：** 表領域のトランスポート方法については、『Oracle Database 管理者ガイド』を参照してください。

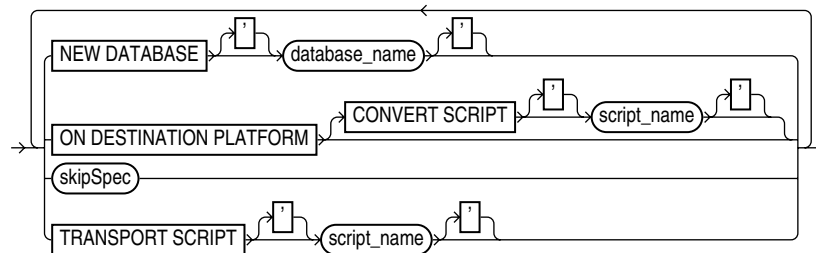
## 構文

### convert::=



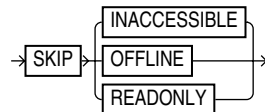
(transportOptionList::= (2-93 ページ)、convertOptionList::= (2-93 ページ))

### transportOptionList::=

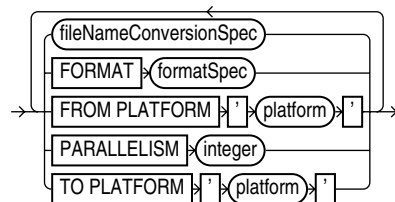


(skipSpec::= (2-93 ページ))

### skipSpec::=



### convertOptionList::=



(file\_name\_conversion\_spec::= (3-16 ページ)、formatSpec::= (2-94 ページ))

**formatSpec::=**

→ 'format\_string' →

## セマンティクス

### convert

この句は、変換するオブジェクト（データファイル、表領域またはデータベース）を指定します。

構文の要素	説明
DATABASE	<p>必要な他のデータベース・ファイルを確実に作成できるように、データファイルをトランスポート先プラットフォームの形式に変換します。</p> <p>CONVERT DATABASE は、データベース全体をソース・プラットフォームからトランスポート先プラットフォームにトランスポートするために使用します。ソース・プラットフォームとトランスポート先プラットフォームのエンディアン形式は同じである必要があります。</p> <p>状況に応じて、ソース・プラットフォームまたはトランスポート先プラットフォームのいずれかで CONVERT DATABASE を使用できます (2-101 ページの例 2-54 を参照)。データベースの次の要素は、直接トランスポートされません。</p> <ul style="list-style-type: none"> <li>■ ソース・データベースの REDO ログ・ファイルおよび制御ファイルはトランスポートされません。Recovery Manager では、トランスポートの処理中に、ターゲット・データベースの制御ファイルと REDO ログ・ファイルが新規に作成され、新規データベースの作成後に OPEN RESETLOGS が実行されます。変換されたデータベースの制御ファイルには、ソース・データベースの Recovery Manager リポジトリは含まれていません。ソース・データベースのバックアップは、変換されたデータベースでは使用できません。</li> <li>■ BFILE はトランスポートされません。CONVERT DATABASE では、BFILE データ型を使用したオブジェクトのリストが出力されるため、ユーザーが、BFILE を手動でコピーし、ターゲット・プラットフォームでの格納場所を修正する必要があります。</li> <li>■ ローカル管理の一時表領域のデータファイルはトランスポートされません。一時表領域は、変換スクリプトを実行すると、ターゲット・プラットフォームで再作成されます。</li> <li>■ 外部表およびディレクトリはトランスポートされません。CONVERT DATABASE の出力には、該当するオブジェクトのリストが表示されますが、ユーザーは、そのオブジェクトをターゲット・プラットフォーム上で再定義する必要があります。外部表およびディレクトリの管理の詳細は、『Oracle Database 管理者ガイド』を参照してください。</li> <li>■ パスワード・ファイルはトランスポートされません。ソース・データベースでパスワード・ファイルが使用されていた場合は、CONVERT DATABASE によって、すべてのユーザー名および関連する権限のリストが出力されます。この情報を使用して、ターゲット・データベースで新しいパスワード・ファイルを作成してください。パスワード・ファイルの管理の詳細は、『Oracle Database 管理者ガイド』を参照してください。</li> </ul> <p>CONVERT DATABASE を使用している場合は、Recovery Manager で次の問題が検出されると、問題が修正されるまで処理を続行することはできなくなります。</p> <ul style="list-style-type: none"> <li>■ データベースにアクティブまたは不正確なトランザクションが存在する。</li> <li>■ データベースに保存されている UNDO セグメントが存在する。</li> <li>■ データベースの COMPATIBILITY が 10 未満に設定されている。</li> <li>■ データベースの COMPATIBILITY が 10 以上に設定されている場合に、読み書き両方でオープンされていない表領域がある。</li> </ul>

構文の要素	説明
<p><i>transportOptionList</i></p>	<p>トランスポートを制御するオプションを指定します。</p>
<p>[<i>convertOptionList</i>]</p>	<p><b>関連項目</b> : 2-96 ページの「<i>transportOptionList</i>」を参照してください。</p>
<p>DATAFILE '<i>filename</i>'</p>	<p>トランスポート先データベースにトランスポートするデータファイルの名前を指定します (2-99 ページの例 2-52 を参照)。</p>
<p><i>convertOptionList</i></p>	<p>CONVERT DATAFILE コマンドは、プラットフォーム間でデータファイルをトランスポートする複数の手順のうちの手順です。データファイルは、『Oracle Database 管理者ガイド』で説明されている手順に従って、ライブ・データファイルを使用してトランスポートできます。また、『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』で説明されている手順を使用して、バックアップからもトランスポートできます。プラットフォーム間で表領域をトランスポートする前に、そのマニュアルを参照してください。</p>
	<p><i>convertOptionList</i> の FROM PLATFORM を使用して、変換するデータファイルのソース・プラットフォームを指定します。FROM PLATFORM を指定しなかった場合のデフォルト値は、トランスポート先のデータベース (Recovery Manager が TARGET として接続されているデータベース) のプラットフォームです。トランスポート先ホストのプラットフォームが、暗黙的にトランスポート先のプラットフォームになります。</p>
	<p>FROM PLATFORM や TO PLATFORM を指定せずに CONVERT DATAFILE を使用して、データファイルを ASM に移動したり、ASM から移動することもできます (2-100 ページの例 2-53 を参照)。その場合、CONVERT DATAFILE で作成されるデータファイル・コピーは、ターゲット・データベースに属しません。そのため、LIST DATAFILECOPY コマンドでは、コピーが表示されません。次の SQL 問合せを実行すると、データベースに属さない変換済データファイルがすべて表示されます。</p>
	<pre>SELECT NAME FROM V\$DATAFILE_COPY WHERE CONVERTED_FILE='YES';</pre>
	<p>CONVERT DATAFILE 構文では、複数のフォーマット名を使用できるため、各データファイルには別々のフォーマットを使用できます。DATAFILE 構文では、CONVERT キーワードの直後と各 DATAFILE '<i>filename</i>' 句の後の両方で <i>convertOptionList</i> を使用できます。ただし、次の状況では、Recovery Manager によってエラーが生成されます。</p>
	<ul style="list-style-type: none"> <li>■ FORMAT 以外の <i>convertOptionList</i> オプションが複数回指定されている。</li> <li>■ 複数の DATAFILE 句が指定されている場合に、FORMAT 以外の <i>convertOptionList</i> オプションが DATAFILE オプション・リストに指定されている。</li> </ul>

構文の要素	説明
<p>TABLESPACE  <i>tablespace_name</i>  <a href="#">convertOptionList</a></p>	<p>別のプラットフォーム上のトランスポート先データベースにトランスポートする予定の、ソース・データベース内の表領域の名前を指定します (2-98 ページの例 2-51 を参照)。</p> <p>このオプションを指定すると、指定した表領域のデータファイルが、別のトランスポート先プラットフォームのフォーマットで生成されます。変換したファイルは、トランスポート先プラットフォームにトランスポートできます。</p> <p>CONVERT TABLESPACE は、ソース・データベースに TARGET として接続し、ソース・プラットフォームで変換する場合にのみ使用できます。変換する表領域は、変換時に読取り専用である必要があります。CONVERT TABLESPACE は、変換予定のデータファイルがデータベースで認識されている場合に使用します。</p> <p>変換するデータファイルのトランスポート先プラットフォームを指定するには TO PLATFORM を使用します。TO PLATFORM を指定しなかった場合のデフォルト値は、Recovery Manager が TARGET として接続されているデータベースのプラットフォームです。ソース・ホストのプラットフォームが、暗黙的にソース・プラットフォームになります。</p> <p>CONVERT TABLESPACE コマンドは、プラットフォーム間で表領域をトランスポートする複数の手順のうちの手順にすぎません。表領域は、『Oracle Database 管理者ガイド』で説明されている手順に従って、ライブ・データファイルを使用してトランスポートできます。また、『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』で説明されている手順を使用して、バックアップからもトランスポートできます。プラットフォーム間で表領域をトランスポートする前に、そのマニュアルを参照してください。</p> <p><b>注意:</b> ソース・ホスト上の表領域のデータファイルを変換するには、CONVERT TABLESPACE ...TO を使用して、変換する表領域とトランスポート先のプラットフォームを指定します。Recovery Manager は、データファイルが読取り専用の表領域に属しているかどうかを検証しないため、CONVERT DATAFILE を使用してソース・プラットフォーム上の個々のデータファイルを変換しないでください (アクティブなデータファイルを変換してしまう可能性があります)。</p> <p><a href="#">convertOptionList</a></p> <p>変換を制御するオプションを指定します。</p> <p><b>関連項目:</b> 2-97 ページの「<a href="#">convertOptionList</a>」を参照してください。</p>

**transportOptionList**

この句は、トランスポートするデータファイル、表領域またはデータベースのオプションを指定します。

構文の要素	説明
<p>NEW DATABASE  <i>database_name</i></p> <p>ON DESTINATION PLATFORM</p>	<p>CONVERT DATABASE コマンドで生成された新しいデータベースに DB_NAME を指定します。</p> <p>トランスポート先のホストで実行してデータベースを作成できる CONVERT DATAFILE コマンド (CONVERT SCRIPT パラメータを参照) の変換スクリプトを生成します。</p> <p><b>注意:</b> このオプションが指定されている場合、CONVERT によってスクリプトは生成されますが、変換されたデータファイルのコピーは生成されません。</p> <p>このオプションは、変換処理に伴うソース・プラットフォームでのオーバーヘッドを回避する場合や、トランスポート先プラットフォームがわからない場合に有効です。たとえば、トランスポート可能な表領域を、複数の様々なターゲット・プラットフォームで使用できるようにパブリッシュすることが必要な場合もあります。</p> <p>ON DESTINATION PLATFORM オプションを指定して CONVERT を実行する場合は、ソース・データベースが読取り専用でオープンされている必要があります。ただし、CONVERT ON DESTINATION PLATFORM で生成されたスクリプトは、NOMOUNT で起動されたデータベース・インスタンス上で実行する必要があります。CONVERT DATAFILE コマンドの実行中に、変換スクリプトがソース・データベースからデータファイルを読み込む場合は、その実行中にソース・データベースを読み書き両方でオープンしないでください。</p>

構文の要素	説明
CONVERT SCRIPT <i>script_name</i>	CONVERT DATABASE ... ON TARGET PLATFORM によって生成された変換スクリプトを含むファイルの場所を指定します。 指定しない場合、この変換スクリプトは生成されません。
<i>skipSpec</i>	CONVERT DATABASE による変換処理時に、アクセス不能なデータファイル、オフラインのデータファイルまたは読取り専用のデータファイルがスキップされるように指定します。
TRANSPORT SCRIPT <i>script_name</i>	CONVERT DATABASE によって生成された変換スクリプトを含むファイルの場所を指定します。省略した場合、この変換スクリプトは生成されません。

**skipSpec**

この副次句は、変換処理から除外するファイルを指定します。

構文の要素	説明
SKIP	次のキーワードで指定された基準に従って、データファイルを変換対象から除外します。
INACCESSIBLE	I/O エラーのために読み取ることができないデータファイルを変換対象から除外するように指定します。 データファイルは、読取りが不可能な場合にのみアクセス不能とみなされます。一部のオフライン・データファイルは、ディスク上に残っているために読取りが可能です。他のデータファイルは削除または移動されたためにアクセス不可となり、読取り不可となります。
OFFLINE	オフライン・データファイルを変換対象から除外するように指定します。
READONLY	読取り専用データファイルを変換対象から除外するように指定します。

**convertOptionList**

この副次句は、変換の入出力オプションを指定します。

FORMAT または *fileNameConversionSpec* 引数を使用して、CONVERT コマンドで生成される出力ファイルの名前を制御できます。いずれも指定しない場合、出力ファイルの場所を管理する規則は、BACKUP AS COPY 操作で生成される出力ファイルを管理する規則と同じになります。この規則については、2-43 ページの「*backupTypeSpec*」を参照してください。

構文の要素	説明
<i>fileNameConversionSpec</i>	文字列のペアです。ペアの最初の文字列がいずれかの入力ファイル名に含まれている場合は、含まれている場所に関係なく、常に、同じペアの 2 番目の文字列と置換されます。必要な数の置換文字列のペアを使用できます。一重引用符または二重引用符を使用できます。 <b>関連項目：</b> ASM および Oracle Managed Files に関連する制約事項については、2-129 ページの「Oracle Managed Files での複製」を参照してください。

構文の要素	説明
FORMAT <i>formatSpec</i>	<p>出力ファイルのネーム・テンプレートを指定します。ここで有効なフォーマット値については、<b>BACKUP AS COPY</b> コマンドを参照してください。</p> <p>Recovery Manager が TARGET として接続されているデータベースでリカバリ領域が使用されている場合は、FORMAT 句を指定する必要があります。</p> <p>FROM PLATFORM または TO PLATFORM を指定せずに、CONVERT ...FORMAT を使用できます。プラットフォームを指定しない場合は、ソース・データベース上で CONVERT TABLESPACE を実行すると、カタログ化されていないデータファイル・コピーが生成されます。CONVERT DATAFILE をトランスポート先データベース上で実行した場合に、データファイル・コピーですでに同じエンディアンが使用されていると、このコマンドでは別のデータファイル・コピーが生成されます。</p> <p>2-100 ページの例 2-53 で示すとおり、CONVERT DATAFILE ...FORMAT を使用して、データファイルを ASM フォーマットに変換できます。非常に大きなデータファイルをホスト間でコピーすると、大量の領域が消費されます。その場合は、NFS の使用またはディスクの共有を検討してください。バックアップをソース・ホスト上に作成し、そのバックアップが含まれるディスクをトランスポート先ホストにマウントしてから、データファイルを ASM に変換できます。</p>
FROM PLATFORM ' <i>platform</i> '	<p>ソース・プラットフォームの名前を指定します。指定しなかった場合のデフォルトは、Recovery Manager が TARGET として接続されているデータベースのプラットフォームです。</p> <p>指定するプラットフォームは、V\$TRANSPORTABLE_PLATFORM の PLATFORM_NAME 列に表示されるプラットフォームの 1 つである必要があります。ソース・プラットフォームまたはターゲット・プラットフォームの正確な名前を CONVERT コマンドのパラメータとして指定する必要があります。次の SQL 文は、サポートされている Linux プラットフォームを問い合わせます。</p> <pre>SELECT PLATFORM_NAME, ENDIAN_FORMAT FROM   V\$TRANSPORTABLE_PLATFORM WHERE  UPPER(PLATFORM_NAME) LIKE 'LINUX%';</pre>
PARALLELISM <i>integer</i>	<p>この操作で使用するチャンネルの数を指定します。指定しない場合は、ディスクに対して割り当てられたチャンネルまたは構成されたチャンネルによってチャンネルの数が決定されます。</p>
TO PLATFORM ' <i>platform</i> '	<p>トランスポート先プラットフォームの名前を指定します。指定しなかった場合のデフォルトは、Recovery Manager が TARGET として接続されているデータベースのプラットフォームです。</p> <p>指定するプラットフォームは、V\$TRANSPORTABLE_PLATFORM の PLATFORM_NAME 列に表示されるプラットフォームの 1 つである必要があります。ソース・プラットフォームまたはターゲット・プラットフォームの正確な名前を CONVERT コマンドのパラメータとして指定する必要があります。次の SQL 文は、サポートされている Linux プラットフォームを問い合わせます。</p> <pre>SELECT PLATFORM_NAME, ENDIAN_FORMAT FROM   V\$TRANSPORTABLE_PLATFORM WHERE  UPPER(PLATFORM_NAME) LIKE 'LINUX%';</pre>

## 例

### 例 2-51 ソース・プラットフォームでの表領域の変換

ソース・データベース prodlin の表領域 finance および hr を、トランスポート先データベース prodsun のプラットフォーム形式に変換するとします。finance 表領域には、データファイル /disk2/orahome/fin/fin01.dbf および /disk2/orahome/fin/fin02.dbf が含まれています。hr 表領域には、データファイル /disk2/orahome/fin/hr01.dbf および /disk2/orahome/fin/hr02.dbf が含まれています。

prodlin データベースは、Linux ホスト lin01 上で実行されます。V\$DATABASE を問い合せて、プラットフォーム名が Linux IA (32-bit) で、リトル・エンディアン形式が使用されていることを確認します。prodsun データベースは、Solaris ホスト sun01 上で実行されます。V\$TRANSPORTABLE\_PLATFORM を問い合せて、Solaris ホストの PLATFORM\_NAME が Solaris[tm] OE (64-bit) で、ビッグ・エンディアン形式が使用されていることを確認します。

表領域の変換はソース・ホスト上で実行し、変換したデータファイルはホスト lin01 の /tmp/transport\_to\_solaris/ に格納するとします。この例は、ソース・データベースの COMPATIBLE が 10.0 以上に設定されていることを前提としています。

ソース・ホスト lin01 で、Recovery Manager クライアントを起動して次のコマンドを実行します。

```
CONNECT TARGET SYS@prodlin

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

SQL 'ALTER TABLESPACE finance READ ONLY';
SQL 'ALTER TABLESPACE hr READ ONLY';
CONVERT TABLESPACE finance, hr
  TO PLATFORM 'Solaris[tm] OE (64-bit)'
  FORMAT '/tmp/transport_to_solaris/%U';
```

この結果、変換されたデータファイルのセットが、Solaris (64-bit) プラットフォーム用の正しいエンディアン順序のデータとともに /tmp/transport\_to\_solaris/ ディレクトリに出力されます。

ここからは、表領域トランスポートの一般的な手順に従います。構造情報のファイルが未作成の場合はデータ・ポンプ・エクスポート・ユーティリティを使用して作成し、構造情報ファイルおよび /tmp/transport\_to\_solaris/ のデータファイルを、トランスポート先ホストの目的のディレクトリに移動し、データ・ポンプ・インポート・ユーティリティを使用して新しいデータベースに表領域を接続します。

### 例 2-52 トランスポート先プラットフォームでのデータファイルの変換

この例では、表領域 finance および hr をホスト sun01 のデータベース prodsun からトランスポート先ホスト lin01 のデータベース prodlin で使用可能な形式に変換します。変換前のデータファイルをトランスポート先ホスト lin01 のディレクトリ /tmp/transport\_from\_solaris/ に一時的に格納し、CONVERT DATAFILE を使用して変換を実行します。トランスポート先データベースにデータファイルをトランスポートすると、そのファイルは /disk2/orahome/dbs に格納されます。

この例は、表領域トランスポートの準備として次の手順を実行していることを前提としています。

- データ・ポンプ・エクスポート・ユーティリティを使用して、構造情報ファイル（この例では expdat.dmp）を作成していること。
- ソース・データベースで表領域 finance および hr を読み取り専用を設定していること。
- 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、expdat.dmp およびトランスポートする変換前のデータファイルを、トランスポート先ホスト lin01 の /tmp/transport\_from\_solaris ディレクトリにコピーしていること。データファイルは次のように格納されています。

```
- /tmp/transport_from_solaris/fin/fin01.dbf
- /tmp/transport_from_solaris/fin/fin02.dbf
- /tmp/transport_from_solaris/hr/hr01.dbf
- /tmp/transport_from_solaris/hr/hr02.dbf
```

- ソース・プラットフォームの名前を V\$TRANSPORTABLE\_PLATFORM を問い合せて、PLATFORM\_NAME が Solaris[tm] OE (64-bit) であることを確認していること。

変換を実行するときは、次の点に注意してください。

- データファイルは、表領域名ではなく、ファイル名で指定します。ローカル・インスタンスでは、データファイルが接続されるまで目的の表領域名を特定できません。
- FORMAT 引数で、変換されたデータファイルの名前および場所を制御します。
- トランスポート先ホストで変換する場合、FROM 引数を使用してソース・プラットフォームを指定する必要があります。指定しない場合、Recovery Manager は、ソース・プラットフォームが、変換が実行されるホストのプラットフォームと同じであると想定します。

Recovery Manager クライアントを起動し、TARGET としてトランスポート先データベース prodlin に接続します。次の CONVERT コマンドを使用して、トランスポートするデータファイルをトランスポート先ホストの形式に変換し、/disk2/orahome/dbs にその結果を保存します。

```
CONNECT TARGET SYS@prodlin

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

CONVERT DATAFILE
  '/tmp/transport_from_solaris/fin/fin01.dbf',
  '/tmp/transport_from_solaris/fin/fin02.dbf',
  '/tmp/transport_from_solaris/hr/hr01.dbf',
  '/tmp/transport_from_solaris/hr/hr02.dbf'
DB_FILE_NAME_CONVERT
  '/tmp/transport_from_solaris/fin','/disk2/orahome/dbs/fin',
  '/tmp/transport_from_solaris/hr','/disk2/orahome/dbs/hr'
FROM PLATFORM 'Solaris[tm] OE (64-bit)';
```

この結果、次のデータファイルが Linux 形式に変換されています。

- /disk2/orahome/dbs/fin/fin01.dbf
- /disk2/orahome/dbs/fin/fin02.dbf
- /disk2/orahome/dbs/hr/hr01.dbf
- /disk2/orahome/dbs/hr/hr02.dbf

ここからは、表領域トランスポートの概要の残りの説明に従います。データ・ポンプ・インポート・ユーティリティを使用して、変換済の表領域を新しいデータベースに接続し、可能な場合は、表領域を読み書き両用に設定します。

### 例 2-53 CONVERT DATAFILE を使用した ASM に対するコピー操作

この例では、通常のストレージから ASM にデータファイルをコピーする方法を示します。生成されるファイルは、ターゲット・データベースに属するデータファイル・コピーとはみなされないため、LIST DATAFILECOPY では表示されないことに注意してください。

ソースおよびトランスポート先のプラットフォームを指定せずに、CONVERT DATAFILE を使用します。出力場所は、ASM ディスク・グループ +DATAFILE を次に示すように指定します。

```
RMAN> CONVERT DATAFILE '/disk1/oracle/dbs/my_tbs_f1.df', '/disk1/oracle/dbs/t_ax1.f'
      FORMAT '+DATAFILE';
```

```
Starting conversion at 29-MAY-05
using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile conversion
input filename=/disk1/oracle/dbs/t_ax1.f
converted datafile=+DATAFILE/asmv/datafile/sysaux.280.559534477
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:16
channel ORA_DISK_1: starting datafile conversion
input filename=/disk1/oracle/dbs/my_tbs_f1.df
converted datafile=+DATAFILE/asmv/datafile/my_tbs.281.559534493
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:04
Finished conversion at 29-MAY-05
```



次の例では、表領域のデータファイルを一意に生成されたファイル名で ASM ストレージから /tmp ディレクトリにコピーする方法を示します。

```

RMAN> CONVERT TABLESPACE tbs_2 FORMAT '/tmp/tbs_2_%U.df';

Starting conversion at 03-JUN-05
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=20 devtype=DISK
channel ORA_DISK_1: starting datafile conversion
input datafile fno=00006 name=+DATAFILE/tbs_21.f
converted datafile=/tmp/tbs_2_data_D-L2_I-2786301554_TS-TBS_2_FNO-6_11gm2fq9.df
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile conversion
input datafile fno=00007 name=+DATAFILE/tbs_22.f
converted datafile=/tmp/tbs_2_data_D-L2_I-2786301554_TS-TBS_2_FNO-7_12gm2fqa.df
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile conversion
input datafile fno=00019 name=+DATAFILE/tbs_25.f
converted datafile=/tmp/tbs_2_data_D-L2_I-2786301554_TS-TBS_2_FNO-19_13gm2fqb.df
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile conversion
input datafile fno=00009 name=+DATAFILE/tbs_23.f
converted datafile=/tmp/tbs_2_data_D-L2_I-2786301554_TS-TBS_2_FNO-9_14gm2fqc.df
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile conversion
input datafile fno=00010 name=+DATAFILE/tbs_24.f
converted datafile=/tmp/tbs_2_data_D-L2_I-2786301554_TS-TBS_2_FNO-10_15gm2fqd.df
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:01
Finished conversion at 03-JUN-05

```

#### 例 2-54 異なるプラットフォームへのデータベースのトランスポート

CONVERT DATABASE の引数は、データファイルの変換をソース・プラットフォームで実行するか、トランスポート先のプラットフォームで実行するかによって異なります。ソースおよびトランスポート先のプラットフォームでの変換処理の詳細および例は、『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』を参照してください。データベースの変換は、その説明を最後まで読んでから実行してください。

Linux ホスト上のデータベース prod を Windows ホストにトランスポートするとします。データファイルは、トランスポート先のホストではなく、ソース・ホストで変換することにします。次の例では、Recovery Manager を Linux ホストの prod データベースに接続し、CONVERT DATABASE NEW DATABASE を実行してデータファイルを変換し、変換スクリプトを生成します。

```

CONNECT TARGET SYS@lin01

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

CONVERT DATABASE
  NEW DATABASE 'prodwin'
  TRANSPORT SCRIPT '/tmp/convertedb/transportscript'
  TO PLATFORM 'Microsoft Windows IA (32-bit)'
  DB_FILE_NAME_CONVERT '/disk1/oracle/dbs' '/tmp/convertedb';

```

次の例は部分的に変更し、Linux ホストで実行されているデータベースを Windows ホストにトランスポートしますが、データファイルの変換は、ソース・ホストでなく、トランスポート先のホストで実行します。次の例では、Recovery Manager を Linux ホストの prod データベースに接続し、CONVERT DATABASE ON DESTINATION PLATFORM を実行します。

```

CONNECT TARGET SYS@lin01

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

```

```

CONVERT DATABASE
  ON DESTINATION PLATFORM
  CONVERT SCRIPT '/tmp/convertdb/convertscript.rman'
  TRANSPORT SCRIPT '/tmp/convertdb/transportscript.sql'
  NEW DATABASE 'prodwin'
  FORMAT '/tmp/convertdb/%U';

```

Linux データベース上で実行される CONVERT DATABASE ON DESTINATION PLATFORM コマンドにより、データファイルを Windows 形式に変換するために Windows ホスト上で実行可能な変換スクリプトが生成されます。また、この CONVERT DATABASE コマンドでは、変換スクリプトも生成されます。

### 例 2-55 プラットフォームとストレージ・タイプが異なる場合のデータベースのトランスポート

この使用例では、sun01 という Solaris ホスト上のデータベース prod を、aix01 という AIX ホストに移動することにします。Solaris データファイルは、ASM ではないファイル・システムに保存されていますが、AIX ホストでは、データファイルを ASM に保存することにします。

次の例では、sun01 に接続し、CONVERT DATABASE を実行して必要なスクリプトを生成します。

```

CONNECT TARGET SYS@sun01

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

CONVERT DATABASE
  ON DESTINATION PLATFORM
  CONVERT SCRIPT '/tmp/convert_newdb.rman'
  TRANSPORT SCRIPT '/tmp/transport_newdb.sql'
  NEW DATABASE 'prodaix'
  DB_FILE_NAME_CONVERT '/u01/oradata/DBUA/datafile','+DATA';

```

変換スクリプトには、次の書式の文が記述されます。ここで、*your\_source\_platform* はソース・プラットフォームを表します。

```

CONVERT DATAFILE '/u01/oradata/DBUA/datafile/o1_mf_system_21g3905p_.dbf'
  FROM PLATFORM 'your_source_platform'
  FORMAT '+DATA/o1_mf_system_21g3905p_.dbf';

```

変換時の停止時間を短縮するには、ネットワーク経由でデータファイルをコピーしたりバックアップをリストアするのではなく、NFS を使用できます。たとえば、Solaris ファイル・システムは、AIX ホストに /net/solaris/oradata としてマウントできます。この場合、変換するソース・データファイルの場所として NFS マウントのディレクトリを参照するように変換スクリプトを編集することが必要になります。そのとき、コマンドは次のようになります。

```

CONVERT DATAFILE '/net/solaris/oradata/DBUA/datafile/o1_mf_system_21g3905p_.dbf'
  FROM PLATFORM 'your_source_platform'
  FORMAT '+DATA/o1_mf_system_21g3905p_.dbf';

```

次に、Recovery Manager をトランスポート先のデータベース・インスタンス（この例では、ホスト aix01 のインスタンス）に接続し、変換スクリプトを実行して、データファイルを変換します。その後、SQL\*Plus を aix01 のデータベース・インスタンスに接続し、変換スクリプトを実行してデータベースを作成します。

## CREATE CATALOG

### 用途

CREATE CATALOG コマンドを使用すると、リカバリ・カタログを作成できます。

リカバリ・カタログは、**基本リカバリ・カタログ**（一連のターゲット・データベースの Recovery Manager メタデータを含むデータベース・スキーマ）にすることができます。**仮想リカバリ・カタログ**は、基本カタログのサブセットにユーザーがアクセスするための一連のシノニムとビューです。

#### 関連項目：

- リカバリ・カタログの作成方法は、『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』を参照してください。
- リカバリ・カタログと、Recovery Manager 環境の他のコンポーネントの互換性の要件については、付録 B「Recovery Manager の互換性」を参照してください。

### 前提条件

このコマンドは、Recovery Manager プロンプトでのみ実行してください。Recovery Manager を CATALOG コマンドライン・オプションまたは CONNECT CATALOG コマンドを通じてリカバリ・カタログ・データベースに接続し、カタログ・データベースをオープンする必要があります。ターゲット・データベースへの接続は不要です。

リカバリ・カタログが基本カタログと仮想カタログのいずれの場合でも、リカバリ・カタログの所有者には、RECOVERY\_CATALOG\_OWNER ロールが付与されている必要があります。また、このユーザーには、リカバリ・カタログ表が格納される表領域に対する権限も付与されている必要があります。リカバリ・カタログは、リカバリ・カタログ所有者のデフォルト表領域に作成されます。

仮想リカバリ・カタログを作成している場合は、基本リカバリ・カタログの所有者が GRANT コマンドを使用して、CATALOG または REGISTER のいずれかの権限をこのユーザーに付与している必要があります (2-105 ページの例 2-57 を参照)。

Oracle Database 10g リリース以前の Recovery Manager クライアントの場合は、Recovery Manager クライアントが仮想カタログに接続する際の制約について、CONNECT CATALOG の説明を参照してください。

### 使用上の注意

リカバリ・カタログは、通常リカバリ・カタログ専用で作成されたデータベースに作成します。リカバリ・カタログを SYS スキーマに作成することはお薦めしません。

リカバリ・カタログは、1 つ作成して、中央管理されている Recovery Manager リポジトリとして、複数のデータベースで使用するようにしてください。そのため、このカタログは、**基本リカバリ・カタログ**と呼ばれます。

基本リカバリ・カタログ所有者は、GRANT や REVOKE を使用して、他のデータベース・ユーザーからのカタログへのアクセスを制限できます。制限を受けるユーザーでも、自分専用のメタデータ（**仮想プライベート・カタログ**と呼ばれます）への完全な読み / 書き両用アクセス権を持っています。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。基本リカバリ・カタログ所有者は、それぞれの仮想カタログ・ユーザーがアクセスできる内容を制御します。

10.2 以前のリリースの Recovery Manager で仮想カタログを使用する場合は、追加手順の実行も必要です。仮想プライベート・カタログを使用する前に、リカバリ・カタログ・データベースに仮想カタログ所有者として接続し、次の PL/SQL プロシージャを実行してください (*base\_catalog\_owner* は、基本リカバリ・カタログを所有するデータベース・ユーザーです)。

```
base_catalog_owner.DBMS_RCVCAT.CREATE_VIRTUAL_CATALOG
```

**関連項目:** RECOVERY\_CATALOG\_OWNER ロールの詳細は、『Oracle Database 管理者ガイド』を参照してください。

## 構文

**createCatalog::=**



## セマンティクス

構文の要素	説明
VIRTUAL	仮想プライベート・カタログを既存のリカバリ・カタログ内に作成します。 このコマンドは、Recovery Manager をリカバリ・カタログ・データベースに仮想カタログ・ユーザーとして接続した後で実行してください。 <b>注意:</b> 仮想プライベート・カタログのメカニズムはすべて、リカバリ・カタログ・スキーマ自体にあります。セキュリティは、Recovery Manager クライアントではなく、カタログ・データベースから提供されます。

## 例

### 例 2-56 リカバリ・カタログの作成とデータベースの登録

SQL\*Plus を起動し、管理者権限でリカバリ・カタログ *catdb* に接続するとします。次のように CREATE USER 文を実行して、パスワードをユーザー指定のパスワードで置き換えます (安全なパスワードの作成の詳細は、『Oracle Database セキュリティ・ガイド』を参照)。SQL 文では、データベース *catdb* にユーザー *catowner* が作成され、その *catowner* ユーザーに RECOVERY\_CATALOG\_OWNER ロールが付与されます。

```
SQL> CREATE USER catowner IDENTIFIED BY password
  2  DEFAULT TABLESPACE cattbs
  3  QUOTA UNLIMITED ON cattbs;
SQL> GRANT recovery_catalog_owner TO catowner;
SQL> EXIT
```

次に、Recovery Manager を起動し、次の Recovery Manager コマンドを実行して、*catowner* としてリカバリ・カタログ・データベースに接続し、リカバリ・カタログを作成します。

```
RMAN> CONNECT CATALOG catowner@catdb

recovery catalog database Password: password
connected to recovery catalog database

RMAN> CREATE CATALOG;
```

同じ Recovery Manager セッションで、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、`REGISTER DATABASE` コマンドを使用して、カタログにこのデータベースを登録します。

```

RMAN> CONNECT TARGET /
RMAN> REGISTER DATABASE;
RMAN> EXIT

```

## 例 2-57 仮想プライベート・カタログの作成

例 2-56 に示すように、リカバリ・カタログを作成し、データベースが登録してあるとします。ここで、データベース・ユーザー `vpc1` に仮想プライベート・カタログを作成することになります。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 `catdb` に接続します。vpc1 ユーザーを作成し、次のようにリカバリ・カタログの所有権を付与して、パスワードをユーザー指定のパスワードで置き換えます（安全なパスワードの作成の詳細は、『Oracle Database セキュリティ・ガイド』を参照）。

```

SQL> CREATE USER vpc1 IDENTIFIED BY password
  2  DEFAULT TABLESPACE vpcusers
  3  QUOTA UNLIMITED ON vpcusers;
SQL> GRANT recovery_catalog_owner TO vpc1;
SQL> EXIT

```

次に、Recovery Manager を起動し、カタログ所有者 `catowner` としてリカバリ・カタログ・データベースに接続します。デフォルトでは、仮想カタログ所有者は基本リカバリ・カタログにアクセスできません。`GRANT` コマンドを使用して、仮想プライベート・カタログに `vpc1` へのアクセス権を付与し、（データベース `prod2` ではなく）データベース `prod1` に対して Recovery Manager が操作できるようにします。

```

RMAN> CONNECT CATALOG catowner@catdb

recovery catalog database Password: password
connected to recovery catalog database

RMAN> GRANT CATALOG FOR DATABASE prod1 TO vpc1;
RMAN> EXIT;

```

この時点で、仮想プライベート・カタログ `vpc1` を使用するバックアップ・オペレータは、仮想カタログの作成準備ができています。次の例では、バックアップ・オペレータがリカバリ・カタログ・データベースに `vpc1` として接続し、`vpc1` の仮想プライベート・カタログを作成します。

```

RMAN> CONNECT CATALOG vpc1@catdb

recovery catalog database Password: password
connected to recovery catalog database

RMAN> CREATE VIRTUAL CATALOG;
RMAN> EXIT;

```

仮想カタログは最終的に Oracle Database 10g ターゲット・データベースで使用する予定なので、このオペレータは、PL/SQL プロシージャ `CREATE_VIRTUAL_CATALOG` を実行してから、仮想カタログを使用する必要があります（2-103 ページの「[使用上の注意](#)」を参照）。次の例では、バックアップ・オペレータが次のようにリカバリ・カタログ・データベースに `vpc1` として接続し、PL/SQL プロシージャを実行します。

```

SQL> CONNECT vpc1@catdb
Enter password: password
Connected.
SQL> BEGIN
  2  catowner.DBMS_RCVCAT.CREATE_VIRTUAL_CATALOG;
  3  END;
  4  /

```

## CREATE SCRIPT

### 用途

CREATE SCRIPT コマンドを使用すると、リカバリ・カタログ内にストアド・スクリプトを作成できます。ストアド・スクリプトは、名前が付けられ、後で実行するためにリカバリ・カタログに格納されている一連の Recovery Manager コマンドです。

#### 関連項目：

- ストアド・スクリプトの使用方法は、『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』を参照してください。
- ストアド・スクリプトの更新方法は、2-208 ページの「[REPLACE SCRIPT](#)」を参照してください。

### 前提条件

CREATE SCRIPT は、Recovery Manager プロンプトでのみ実行できます。Recovery Manager が、ターゲット・データベースとリカバリ・カタログに接続されている必要があります。リカバリ・カタログ・データベースはオープン状態である必要があります。

GLOBAL を指定する場合は、その名前のグローバル・スクリプトがリカバリ・カタログ内にまだ存在していないことが必要です。GLOBAL を指定しない場合は、同じ名前のローカル・スクリプトが、同じターゲット・データベースに対してまだ存在していないことが必要です。この前提条件が満たされない場合は、エラー RMAN-20401 が戻されます。

### 使用上の注意

ストアド・スクリプトには、ローカルとグローバルがあります。ローカル・スクリプトは、現在のターゲット・データベース用にのみ作成されますが、グローバル・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で使用できます。

グローバル・スクリプトをローカル・スクリプトと同じ名前で、またローカル・スクリプトをグローバル・スクリプトと同じ名前で作成することもできます。

#### ストアド・スクリプト内の置換変数

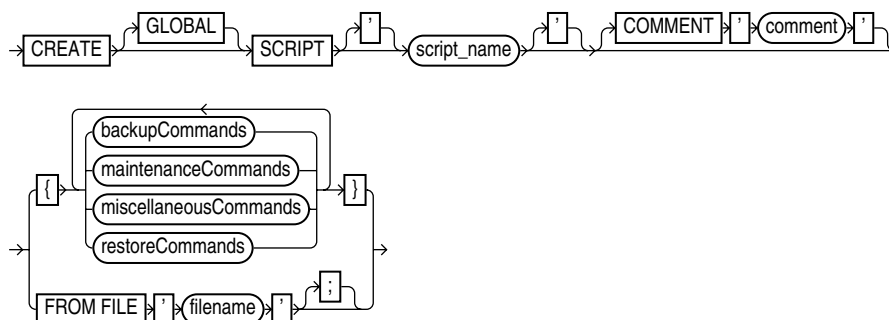
Recovery Manager では、ストアド・スクリプトで置換変数を使用できます。&1 は最初の値を配置する位置を示し、&2 は 2 番目の値を配置する位置を示します。その後も同様に続きます。特殊文字は、引用符で囲む必要があります。

置換変数の構文は、&integer の後にオプションでピリオドが続きます (&1.3 など)。オプションのピリオドは変数の一部であり、値と置換されますので、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、置換変数 &1.3 が含まれているコマンド・ファイルに値 mybackup を渡すと、その置換結果は mybackup3 になります。mybackup.3 という結果を得るには、構文 &1..3 を使用します。

置換変数を使用するストアド・スクリプトを作成する場合は、作成時に例となる値を指定する必要があります。これらの値は、Recovery Manager の起動時に USING 句で指定するか (2-244 ページの「[RMAN](#)」を参照)、またはプロンプトが表示されたときに入力します (例 2-60 を参照)。

## 構文

**createScript::=**



([backupCommands::=](#) (2-250 ページ)、[maintenanceCommands::=](#) (2-250 ページ)、[miscellaneousCommands::=](#) (2-251 ページ)、[restoreCommands::=](#) (2-251 ページ))

## セマンティクス

構文の要素	説明
GLOBAL	スクリプトをグローバルとして指定します。 <b>注意:</b> 仮想プライベート・カタログは、グローバル・スクリプトに対して読取り専用のアクセスが可能です。グローバル・スクリプトの作成と更新は、基本リカバリ・カタログへの接続中に行う必要があります。
SCRIPT <i>script_name</i>	スクリプトの名前を指定します。スクリプト名に空白または予約語が含まれている場合は、引用符で囲む必要があります。
COMMENT ' <i>comment</i> '	説明のコメントをリカバリ・カタログ内のストアド・スクリプトと関連付けます。このコメントは、 <code>LIST SCRIPT NAMES</code> の出力で使用されます。
<i>backupCommands</i>	ストアド・スクリプトに含めるコマンドを指定します。CREATE SCRIPT ' <i>script_name</i> ' {...} コマンドのカッコ内で使用できるコマンドは、RUN コマンド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、いずれもストアド・スクリプトで使用できます。コマンド <code>RUN</code> 、 <code>@</code> および <code>@@</code> は、ストアド・スクリプト内では無効です。
<i>maintenanceCommands</i>	
<i>miscellaneousCommands</i>	
<i>restoreCommands</i>	
FROM FILE ' <i>filename</i> '	指定したファイルからスクリプトを定義する一連のコマンドを読み取ります。このファイルは、有効なストアド・スクリプトの本体と同様である必要があります。このファイルの最初の行は、左カッコ ( { ) である必要があります。また、最終行には、右カッコ ( } ) が含まれている必要があります。このファイルの Recovery Manager コマンドは、ストアド・スクリプトで有効である必要があります。

## 例

### 例 2-58 ローカル・ストアド・スクリプトの作成

データベース `prod` をバックアップするローカル・ストアド・スクリプトを作成するとします。Recovery Manager を起動し、TARGET として `prod` に接続して、リカバリ・カタログに接続します。backup\_whole というストアド・スクリプトを作成し、EXECUTE SCRIPT を使用して、次のようにそのスクリプトを実行します。

```

CREATE SCRIPT backup_whole
COMMENT "backup whole database and archived redo logs"
{
    BACKUP
    INCREMENTAL LEVEL 0 TAG backup_whole
    FORMAT "/disk2/backup/%U"
  
```

```
        DATABASE PLUS ARCHIVELOG;
    }
    RUN { EXECUTE SCRIPT backup_whole; }
```

### 例 2-59 グローバル・ストアド・スクリプトの作成

この例では、Recovery Manager をターゲット・データベース prod に接続し、カタログ・ユーザー rco としてリカバリ・カタログ・データベース catdb に接続します。この例では、データベースおよびアーカイブ REDO ログをバックアップする global\_backup\_db というグローバル・スクリプトを作成します。

```
RMAN> CONNECT TARGET SYS@prod

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

```
RMAN> CONNECT CATALOG rco@catdb

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

```
RMAN> CREATE GLOBAL SCRIPT global_backup_db { BACKUP DATABASE PLUS ARCHIVELOG; }
RMAN> EXIT;
```

これで、Recovery Manager を prod2 などの別のターゲット・データベースに接続し、グローバル・ストアド・スクリプトを実行できます。

```
RMAN> CONNECT TARGET SYS@prod2

target database Password: password
connected to target database: PROD2 (DBID=36508508)
```

```
RMAN> CONNECT CATALOG rco@catdb

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

```
RMAN> RUN { EXECUTE SCRIPT global_backup_db; }
```

### 例 2-60 置換変数を使用するストアド・スクリプトの作成

次の例では、Recovery Manager をターゲット・データベースおよびリカバリ・カタログに接続し、**CREATE SCRIPT** を使用して、3つの置換変数が含まれるバックアップ・スクリプトを作成します。Recovery Manager によって、変数の初期値を入力するプロンプトが表示されます(太字がユーザー入力です)。

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

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

```
RMAN> CREATE SCRIPT backup_df { BACKUP DATAFILE &1 TAG &2.1 FORMAT '/disk1/&3_%U'; }
```

```
Enter value for 1: 1
```

```
Enter value for 2: df1_backup
```

```
Enter value for 3: df1
```

```
starting full resync of recovery catalog
full resync complete
created script backup_df
```



`EXECUTE SCRIPT` の実行時に、スクリプトに別の値を渡すことができます。次の例では、値 3、`test_backup` および `test` をストアド・スクリプトの置換変数に渡しています。

```
RMAN> RUN { EXECUTE SCRIPT backup_df USING 3 test_backup df3; }
```

値が置換され、スクリプトは次のように実行されます。

```
BACKUP DATAFILE 3 TAG test_backup1 FORMAT '/disk1/df3_%U';
```

## CROSSCHECK

### 用途

CROSSCHECK コマンドを使用すると、実際の物理的なバックアップおよびコピーを、Recovery Manager リポジトリ内の論理レコードと同期させることができます。

**関連項目：** リカバリ・カタログ内のデータベース・レコードを管理する方法は、『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』を参照してください。

### 前提条件

Recovery Manager はターゲット・データベース・インスタンスに接続され、そのインスタンスが起動されている必要があります。

ディスクのクロスチェックを行うためのメンテナンス・チャンネルは不要です。メディア・マネージャを使用する場合、それに対する自動チャンネルが構成されていないときは、[ALLOCATE CHANNEL FOR MAINTENANCE](#) を実行してから CROSSCHECK を実行する必要があります。たとえば、SBT デバイスの自動 SBT チャンネルを構成せずに SBT チャンネルでバックアップを作成した場合は、このチャンネルを手動で割り当ててからでなければ、CROSSCHECK によるバックアップのチェックは実行できません。さらに、各種メディア・マネージャ・オプション（プール、サーバー、ライブラリなど）でバックアップを実行した場合は、それぞれの組合せに対してメンテナンス・チャンネルを割り当てる必要があります。

クロスチェックでは、指定されたすべてのバックアップおよびコピーが検証されます。これは、以前のデータベース・インカネーションで作成されたものも対象となります。

### 使用上の注意

Recovery Manager では、操作対象のすべてのターゲット・データベースの制御ファイル内にバックアップに関する論理メタデータを常に保持します。Recovery Manager をリカバリ・カタログとともに使用する場合は、リカバリ・カタログに登録されたすべてのデータベースのメタデータも保持されます。

バックアップがディスク上に存在する場合、CROSSCHECK を実行すると、ファイルのヘッダーが有効かどうかを判断できます。バックアップがテープ上に存在する場合は、Recovery Manager が、チェック対象のバックアップ・ピースの名前と場所を Recovery Manager リポジトリに問い合わせます。Recovery Manager は、このメタデータをターゲット・データベースのサーバーに送信し、メディア管理ソフトウェアにバックアップの情報を問い合わせます。次に、メディア管理ソフトウェアがメディア・カタログをチェックして、バックアップのステータスをサーバーにレポートします。

#### EXPIRED および AVAILABLE ステータス

Recovery Manager リポジトリに記録されているバックアップ・セットおよびコピーのステータスは、[LIST](#)、[V\\$](#) ビュー、またはリカバリ・カタログ・ビュー（Recovery Manager をカタログとともに使用する場合）で確認できます。[表 2-4](#) に、各ステータスの意味を示します。

CROSSCHECK コマンドで処理されるのは、クロスチェックに使用されているチャンネルと同じデバイス・タイプで作成されたファイルのみです。CROSSCHECK コマンドは、リポジトリで AVAILABLE または EXPIRED のマークが付いているオブジェクトのみをチェックします。この処理は、DISK チャンネルの場合はディスク上のファイルを検査し、sbt チャンネルの場合はメディア・マネージャに問い合わせで行われます。

表 2-4 クロスチェックのステータスの意味

ステータス	説明
EXPIRED	<p>オブジェクトがファイル・システム内 (DISK の場合) またはメディア・マネージャ (sbt の場合) で見つかりません。バックアップ・セットは、そのセット内のすべてのバックアップ・ピースが EXPIRED の場合に、EXPIRED になります。</p> <p>CROSSCHECK コマンドは、見つからなかったファイルを削除せずに、そのファイルのリポジトリ・レコードを EXPIRED に更新します。DELETE EXPIRED を実行すると、期限切れのファイルのリポジトリ・レコードと、ステータスが EXPIRED の既存の物理ファイルを削除できます。</p> <p>バックアップが EXPIRED の場合は、後でクロスチェックを再実行して、期限切れになっていたバックアップが使用可能かどうかを確認できます。このような慎重な処理は、Recovery Manager をメディア・マネージャとともに使用している場合に効果があります。たとえば、一部のバックアップ・ピースまたはコピーが、PARMS チャネル設定ミスのため、誤って EXPIRED でマークされている場合は、ファイルが実際にメディア・マネージャに存在していることを確認した後、CROSSCHECK BACKUP を再実行して、これらのファイルを AVAILABLE ステータスにリストアします。</p>
AVAILABLE	<p>オブジェクトは Recovery Manager で使用可能です。バックアップ・セットを AVAILABLE にするには、そのセット内のすべてのバックアップ・ピースのステータスが AVAILABLE になる必要があります。</p>

### Data Guard 環境でのクロスチェック

バックアップの関連付けとアクセス可能性の違いについては、2-21 ページの「Data Guard 環境での Recovery Manager のバックアップ」を参照してください。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ます。Data Guard 環境のどのデータベースに接続されていても、バックアップに接続可能であれば、CHANGE、DELETE、CROSSCHECK などのメンテナンス・コマンドをバックアップに使用できます。通常、Recovery Manager では、データベース上で作成されたテープ・バックアップはその環境のすべてのデータベースにアクセス可能であるとみなされますが、ディスク・バックアップは作成元のデータベースのみにアクセス可能であるとみなされます。

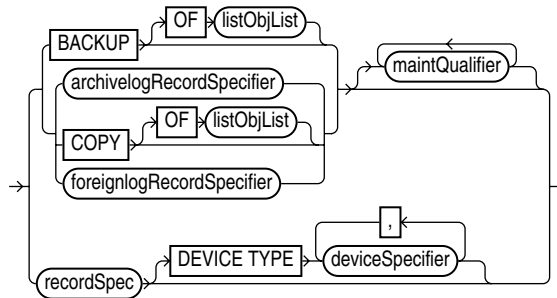
Recovery Manager は、バックアップと関連付けられたデータベースに TARGET として接続されている場合のみ、バックアップのステータスを AVAILABLE から EXPIRED または DELETED に更新できます。バックアップがターゲット・データベースに関連付けられていないために Recovery Manager が削除を実行できない場合は、そのバックアップが関連付けられているデータベースで、同じ CROSSCHECK 操作をそのファイルに実行するように求めるプロンプトが表示されます。このように、Recovery Manager では、不適切な SBT 構成による意図しないステータス変更を防止しています。

たとえば、Recovery Manager をスタンバイ・データベース standby1 に TARGET として接続し、そのデータベースをテープにバックアップするとします。バックアップを手動でテープから削除し、standby2 でバックアップのクロスチェックを実行すると、standby1 でクロスチェックを実行するように求めるプロンプトが表示されます。バックアップが削除されたことがメディア・マネージャからレポートされた場合は、standby1 のクロスチェックにより、テープ・バックアップのステータスが EXPIRED に更新されます。

## 構文

**crosscheck** ::=

```
→ CROSSCHECK {maintSpec} ;
```

**maintSpec::=**

(listObjList::= (3-27 ページ)、 archivelogRecordSpecifier::= (3-6 ページ)、  
 foreignlogRecordSpecifier::= (3-19 ページ)、 maintQualifier::= (3-30 ページ)、  
 recordSpec::= (3-35 ページ)、 deviceSpecifier::= (3-15 ページ))

## セマンティクス

構文の要素	説明
<code>maintSpec</code>	バックアップおよびコピーをクロスチェックします。 <code>maintSpec</code> オプションについては、3-32 ページの「 <code>maintSpec</code> 」のパラメータの説明を参照してください。

## 例

### 例 2-61 すべてのバックアップとコピーのクロスチェック

この例では、デフォルトの構成済チャンネルが DEVICE TYPE sbt であると仮定して、テープ上とディスク上のすべてのバックアップとコピーのステータスをクロスチェックします（出力の一部を示します）。Recovery Manager ではディスク・チャンネルが事前に構成されるため、手動で割り当てる必要はありません。

```

RMAN> CROSSCHECK BACKUP;

allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=84 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: Oracle Secure Backup
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=86 device type=DISK
backup piece handle=/disk2/backup/08i9umon_1_1 RECID=7 STAMP=614423319
crosschecked backup piece: found to be 'EXPIRED'
backup piece handle=/disk2/backup/09i9umso_1_1 RECID=8 STAMP=614423448
crosschecked backup piece: found to be 'EXPIRED'
backup piece handle=/disk1/cfauto/c-26213402-20070213-00 RECID=9 STAMP=614423452
crosschecked backup piece: found to be 'AVAILABLE'
backup piece handle=0bi9uo81_1_1 RECID=10 STAMP=614424833
crosschecked backup piece: found to be 'AVAILABLE'
backup piece handle=c-26213402-20070213-01 RECID=11 STAMP=614424851
crosschecked backup piece: found to be 'AVAILABLE'
.
.
.

```

**例 2-62 日付範囲内でのクロスチェック**

この例では、指定した 6 週間のバックアップ・セットの状態をメディア・マネージャに問い合わせます。Recovery Manager では、NLS\_DATE\_FORMAT パラメータで指定された日付書式が使用されることに注意してください。この例では、'DD-MON-YY' が使用されています。最初のコマンドは、テープ上のバックアップのみをクロスチェックします。

```
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt;  
CROSSCHECK BACKUP  
  COMPLETED BETWEEN '01-JAN-07' AND '14-FEB-07';  
RELEASE CHANNEL;
```

次のコマンドは、ディスクのみをクロスチェックするように DEVICE TYPE DISK を指定します。

```
CROSSCHECK BACKUP DEVICE TYPE DISK  
  COMPLETED BETWEEN '01-JAN-07' AND '14-FEB-07';
```

SBT がデフォルトのチャネルの場合は、デフォルトのチャネルで CROSSCHECK を実行して、ディスクと SBT バックアップの両方をクロスチェックできます。

```
CROSSCHECK BACKUP COMPLETED BETWEEN '01-JAN-07' AND '14-FEB-07';
```

---

## DELETE

### 用途

DELETE コマンドを使用すると、次の操作を実行できます。

- 物理バックアップおよびコピーを削除します。
- ターゲット制御ファイル内のリポジトリ・レコードを更新して、ファイルが削除されていることを示します。たとえば、V\$BACKUP\_PIECE.STATUS のバックアップ・ピースのレコードに、値 D が表示されます。
- 削除されたファイルのリポジトリ・レコードをリカバリ・カタログから削除します（カタログを使用している場合）。たとえば、RC\_BACKUP\_PIECE には、削除されたバックアップ・ピースの行が含まれなくなります。

**関連項目：** BACKUP ... DELETE INPUT コマンドについては、2-19 ページの「[BACKUP](#)」を参照してください。

### 前提条件

Recovery Manager がターゲット・データベースに接続していて、そのデータベースがマウントまたはオープン状態である必要があります。

Recovery Manager は、構成済のすべてのチャンネルを使用して削除を実行します。自動チャンネルが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 チャンネルでバックアップを作成したが、構成されているのがディスク・チャンネルのみであれば、DELETE 用に SBT チャンネルを手動で割り当てる必要があります。ディスク専用のファイルに対して DELETE を使用する場合は、自動または手動で割り当てられたメンテナンス・チャンネルが必要です。

### 使用上の注意

[CROSSCHECK](#) を実行してバックアップおよびコピーのリポジトリでのステータスを更新してから、DELETE を実行して目的のファイルを削除することを推奨します。Recovery Manager を対話方式で実行している場合に DELETE を実行すると、ファイルのリストが表示され、そのリスト内のファイルを削除する前に確認のプロンプトが表示されます。確認すると、Recovery Manager では削除されるたびに各項目が表示されます。コマンド・ファイルからコマンドを読み取る場合は、Recovery Manager によって確認のプロンプトは表示されません。

Recovery Manager リポジトリに記録されているバックアップおよびコピーのステータスは、[LIST](#)、V\$ ビュー、またはリカバリ・カタログ・ビュー（カタログを使用している場合）で確認できます。バックアップのリポジトリ・レコードに、バックアップの物理的な状態が反映されていないことがあります。たとえば、ディスク・バックアップを、Linux の rm コマンドで削除する場合です。バックアップ・レコードは rm では更新できないため、Recovery Manager リポジトリには、ファイルが存在していなくても使用可能と表示されます。

## ステータス値が異なるファイルに対する DELETE コマンドの動作

表 2-5 に、FORCE オプションが指定されていない場合の DELETE の動作を示します。

表 2-5 FORCE オプションが指定されていない場合の DELETE コマンドの動作

リポジトリでの状態	物理的な状態	DELETE コマンドの動作
AVAILABLE	メディア上に見つからない	オブジェクトは削除されず、ジョブの終了時に一致しないオブジェクトのリストがレポートされます。リポジトリの状態は更新されません。
EXPIRED	メディア上で見つかる	オブジェクトは削除されず、ジョブの終了時に一致しないオブジェクトのリストがレポートされます。リポジトリの状態は更新されません。
UNAVAILABLE	すべて	リポジトリ・レコードが削除され、存在する場合はオブジェクトが削除されます。I/O エラーはすべて無視されます。

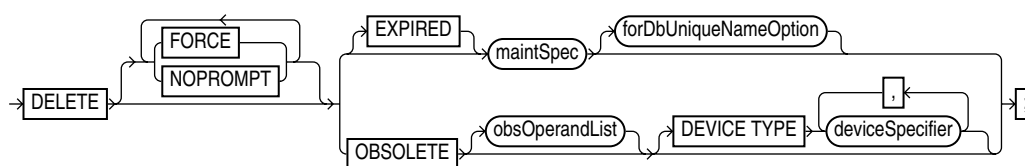
## Data Guard 環境でのバックアップの削除

バックアップの関連付けとアクセス可能性の違いについては、2-21 ページの「Data Guard 環境での Recovery Manager のバックアップ」を参照してください。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ます。Data Guard 環境のどのデータベースに接続されていても、バックアップに接続可能であれば、CHANGE、DELETE、CROSSCHECK などのメンテナンス・コマンドをバックアップに使用できます。通常、Recovery Manager では、データベース上で作成されたテープ・バックアップはその環境のすべてのデータベースにアクセス可能であるとみなされますが、ディスク・バックアップは作成元のデータベースのみにアクセス可能であるとみなされます。

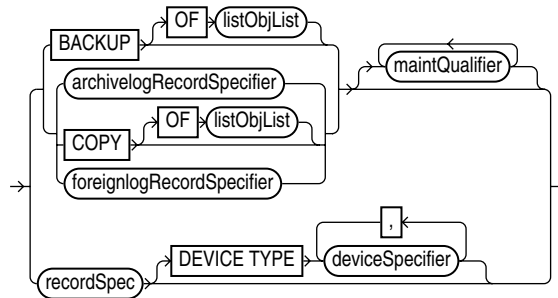
削除が正常に行われると、Recovery Manager はそのファイルのメタデータを削除します。ファイルが別のデータベースに関連付けられている場合も同様です。削除が正常に行われず、そのファイルが Data Guard 環境の別のデータベースに関連付けられている場合は、ファイルに関連付けられたデータベースに TARGET として接続した状態で、同じ DELETE コマンドを実行するように求めるプロンプトが表示されます。ファイルのメタデータを削除するには、DELETE ... FORCE を使用する必要があります。

## 構文

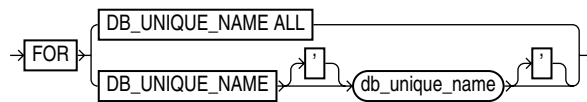
**delete::=**



(**maintSpec::=** (3-32 ページ)、**obsOperandList::=** (3-34 ページ)、**deviceSpecifier::=** (3-15 ページ))

**maintSpec::=**

(listObjList::= (3-27 ページ)、archivelogRecordSpecifier::= (3-6 ページ)、  
maintQualifier::= (3-30 ページ)、deviceSpecifier::= (3-15 ページ)、  
recordSpec::= (3-35 ページ))

**forDbUniqueNameOption::=**

## セマンティクス

構文の要素	説明
FORCE	指定したファイルを（メディア上に存在するかどうかに関係なく）削除し、リポジトリ・レコードを削除します（2-119 ページの例 2-66 を参照）。  削除されたオブジェクトに関する I/O エラーは無視されます。また、 <b>CONFIGURE ARCHIVELOG DELETION POLICY</b> の設定も無視されます。ジョブが終了すると、削除されたオブジェクトの数が表示されます。
NOPROMPT	先にファイル・リストを表示したり確認を求めるプロンプトを表示せずに、指定したファイルを削除します。DELETE NOPROMPT コマンドでは、削除される各項目が表示されます。
EXPIRED	リポジトリ内のステータスが EXPIRED になっているファイルのみを削除します（2-118 ページの例 2-63 を参照）。CROSSCHECK コマンドの実行時にファイルが存在しないか、アクセスできなければ、 <b>Recovery Manager</b> ではバックアップとコピーが期限切れとしてマークされます。期限切れのファイルを判断するには、LIST EXPIRED コマンドを実行します。  DELETE EXPIRED コマンドを実行したときに、なんらかの理由で EXPIRED のマークが付いたバックアップが存在している場合、 <b>Recovery Manager</b> はその物理ファイルを削除しません。
<i>maintSpec</i>	バックアップおよびコピーを削除します。  <i>maintQualifier</i> 句で削除ルールを設定できます。たとえば、テープにバックアップされたアーカイブ・ログを削除できます（2-118 ページの例 2-65 を参照）。  <b>注意:</b> DELETE ARCHIVELOG ALL では、アーカイブ・ログの削除方針のみが考慮され、構成済の保存方針は考慮されません。  <b>関連項目:</b> 3-32 ページの「 <i>maintSpec</i> 」および 3-30 ページの「 <i>maintQualifier</i> 」を参照してください。



構文の要素	説明
<i>forDbUniqueNameOption</i>	<p>Data Guard 環境の指定した DB_UNIQUE_NAME のみに関連付けられた <i>maintSpec</i> のバックアップおよびコピーを削除します。</p> <p><b>注意:</b> DELETE OBSOLETE コマンドで FOR DB_UNIQUE_NAME オプションを使用することはできません。</p> <p>Recovery Manager は、指定した DB_UNIQUE_NAME に関連付けられたテープ・バックアップを正常に削除すると、これらのファイルのメタデータをリカバリ・カタログから削除します。これらのファイルが Data Guard 環境の別のデータベースに関連付けられているために Recovery Manager が削除を実行できない場合は、これらのファイルが関連付けられているデータベースで、同じ DELETE 操作をファイルに実行するように求めるプロンプトが表示されます。</p> <p><b>注意:</b> FORCE を使用して、デフォルトの動作をオーバーライドしたり、Recovery Manager が別のデータベースに関連付けられたファイルの削除するように指定することはできません。このように、Recovery Manager では、SBT の不適切な Recovery Manager 構成による誤った削除操作を防止しています。削除が許可されていないファイルのメタデータを削除するには、CHANGE RESET DB_UNIQUE_NAME コマンドを使用します。</p> <p><b>関連項目:</b> この句のオプションについては、3-18 ページの「<i>forDbUniqueNameOption</i>」を参照してください。</p>
OBSOLETE	<p>Recovery Manager リポジトリに記録されているデータファイルのバックアップとコピーのうち、廃止、つまり不要になったものを削除します (2-118 ページの例 2-64 を参照)。また、不要なアーカイブ REDO ログおよびログ・バックアップも Recovery Manager によって削除されます。</p> <p>Recovery Manager では、データファイルのうち不要になったバックアップとコピーが判別されてから、ログ (およびそのバックアップ) が不要になる時期が判断されます。データファイルの作成は、保存するログの決定時にバックアップとみなされます。</p> <p>Recovery Manager では、まず <i>obsOperandList</i> で指定したオプションを使用して、不要になったファイルが判断されます。<i>obsOperandList</i> でオプションを指定しなければ、CONFIGURE RETENTION POLICY で指定したオプションが使用されます。</p> <p><b>注意:</b> DELETE OBSOLETE では、バックアップの保存方針のみが考慮されます。不要なログの決定に、構成済のアーカイブ・ログの削除方針が使用されることはありません。これとは対照的に、DELETE ARCHIVELOG ALL では、構成済のアーカイブ・ログの削除方針のみが考慮されます。</p> <p><b>注意:</b> KEEP UNTIL TIME 句を使用して作成されたバックアップは、指定した KEEP 時間が過ぎると不要となり、DELETE OBSOLETE によって削除されます。KEEP 時間が期限切れになったアーカイブ・バックアップのバックアップ保存方針は考慮されません。</p>
<i>obsOperandList</i>	<p>不要になるバックアップとコピーの判断基準を指定します。</p> <p><b>関連項目:</b> 3-34 ページの「<i>obsOperandList</i>」を参照してください。</p>
DEVICE TYPE <i>deviceSpecifier</i>	<p>削除の対象を、指定したデバイス・タイプで作成された不要なバックアップとコピーのみに制限します。</p> <p><b>関連項目:</b> 3-15 ページの「<i>deviceSpecifier</i>」を参照してください。</p>

## 例

**例 2-63 期限切れのバックアップの削除**

この例では、構成済の sbt チャンネルを使用して、1 か月以上経過している表領域 users の期限切れバックアップの有無についてメディア・マネージャをチェックし、該当するリカバリ・カログ・レコードを削除します。

```
CROSSCHECK BACKUPSET OF TABLESPACE users
  DEVICE TYPE sbt COMPLETED BEFORE 'SYSDATE-31';
DELETE NOPROMPT EXPIRED BACKUPSET OF TABLESPACE users
  DEVICE TYPE sbt COMPLETED BEFORE 'SYSDATE-31';
```

**例 2-64 不要なバックアップの削除**

この例では、データベースのリカバリに不要になったバックアップとコピーを、先週の任意の SCN まで削除します。また、不要になったアーカイブ REDO ログも削除されます。

```
DELETE NOPROMPT OBSOLETE RECOVERY WINDOW OF 7 DAYS;
```

**例 2-65 バックアップ済のアーカイブ REDO ログの削除**

Recovery Manager の設定を次のように構成しているとします。

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE ARCHIVELOG DELETION POLICY TO
  BACKED UP 2 TIMES
  TO DEVICE TYPE sbt;
```

次の DELETE コマンドは、構成済の削除方針（ログはテープに 2 回バックアップする必要がある）を満たす必要がない、ディスク上のすべてのアーカイブ REDO ログを削除します（次に出力を示します）。

```
RMAN> DELETE ARCHIVELOG ALL;
```

```
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=84 device type=DISK
```

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

```
=====
```

Key	Thrd	Seq	S	Low	Time
107	1	4	A	12-FEB-07	
				Name:	/orcva/PROD/archivelog/2007_02_12/o1_mf_1_4_2x28bpcm_.arc
108	1	5	A	12-FEB-07	
				Name:	/orcva/PROD/archivelog/2007_02_12/o1_mf_1_5_2x28g7s9_.arc
109	1	6	A	12-FEB-07	
				Name:	/orcva/PROD/archivelog/2007_02_13/o1_mf_1_6_2x3bbqym_.arc
157	1	7	A	13-FEB-07	
				Name:	/orcva/PROD/archivelog/2007_02_13/o1_mf_1_7_2x3w2cvs_.arc
164	1	8	A	13-FEB-07	
				Name:	/orcva/PROD/archivelog/2007_02_13/o1_mf_1_8_2x3w40vr_.arc
171	1	9	A	13-FEB-07	
				Name:	/orcva/PROD/archivelog/2007_02_13/o1_mf_1_9_2x3w8pf7_.arc
318	1	10	A	13-FEB-07	
				Name:	/orcva/PROD/archivelog/2007_02_13/o1_mf_1_10_2x3zx6d9_.arc
330	1	11	A	13-FEB-07	
				Name:	/orcva/PROD/archivelog/2007_02_13/o1_mf_1_11_2x403wco_.arc
448	1	12	A	13-FEB-07	
				Name:	/orcva/PROD/archivelog/2007_02_13/o1_mf_1_12_2x40wn6x_.arc
455	1	13	A	13-FEB-07	
				Name:	/orcva/PROD/archivelog/2007_02_13/o1_mf_1_13_2x412s3m_.arc
583	1	14	A	13-FEB-07	
				Name:	/orcva/PROD/archivelog/2007_02_13/o1_mf_1_14_2x428p9d_.ar
638	1	15	A	13-FEB-07	

```
Name: /orcva/PROD/archive/2007_02_13/ol_mf_1_15_2x42f0gj_.arc
```

Do you really want to delete the above objects (enter YES or NO)?

### 例 2-66 バックアップ・セットの強制削除

次の例では、タグ weekly\_bkup を持つバックアップ・セットのコピーを削除します。

```
RMAN> DELETE NOPROMPT BACKUPSET TAG weekly_bkup;
```

リポジトリにはバックアップ・セットが AVAILABLE として表示されても、オブジェクトは実際にはメディア上で使用できないため、Recovery Manager では警告が表示されます。

```
List of Backup Pieces
BP Key  BS Key  Pc#  Cp#  Status      Device Type Piece Name
-----
809     806     1    1    AVAILABLE   SBT_TAPE   Ori9uu08_1_1
```

```
RMAN-06207: WARNING: 1 objects could not be deleted for SBT_TAPE channel(s) due
RMAN-06208:           to mismatched status. Use CROSSCHECK command to fix status
```

```
RMAN-06210: List of Mismatched objects
```

```
RMAN-06211: =====
```

```
RMAN-06212: Object Type  Filename/Handle
```

```
RMAN-06213: -----
```

```
RMAN-06214: Backup Piece  Ori9uu08_1_1
```

次のコマンドを実行すると、Recovery Manager によってバックアップ・セットが強制的に削除されます（例には出力例も含まれます）。

```
RMAN> DELETE FORCE NOPROMPT BACKUPSET TAG weekly_bkup;
```

```
using channel ORA_SBT_TAPE_1
```

```
using channel ORA_DISK_1
```

```
List of Backup Pieces
BP Key  BS Key  Pc#  Cp#  Status      Device Type Piece Name
-----
809     806     1    1    AVAILABLE   SBT_TAPE   Ori9uu08_1_1
```

```
deleted backup piece
```

```
backup piece handle=Ori9uu08_1_1 RECID=26 STAMP=614430728
```

```
Deleted 1 objects
```

## DELETE SCRIPT

### 用途

DELETE SCRIPT コマンドを使用すると、ローカルまたはグローバルのストアド・スクリプトをリカバリ・カタログから削除できます。

### 前提条件

DELETE SCRIPT は、Recovery Manager プロンプトでのみ実行できます。Recovery Manager が、リカバリ・カタログとターゲット・データベースに接続している必要があります。リカバリ・カタログ・データベースはオープン状態である必要があります。

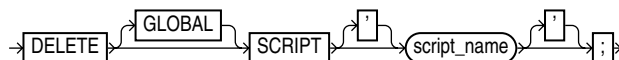
### 使用上の注意

ストアド・スクリプトには、ローカルとグローバルがあります。ローカル・スクリプトは、現行のターゲット・データベース用にのみ作成されますが、グローバル・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で使用できます。

GLOBAL を指定する場合は、その名前のグローバル・スクリプトが、リカバリ・カタログ内にすでに存在している必要があります。存在していない場合は、エラー RMAN-06710 が戻されます。GLOBAL を指定しなかった場合は、Recovery Manager によって、指定した名前のローカル・ストアド・スクリプトが現行のターゲット・データベースに定義されていないかどうか確認されます。そのようなスクリプトがターゲット・データベースに定義されていない場合は、同じ名前のグローバル・ストアド・スクリプトが検索され、存在した場合は削除されます。

### 構文

**deleteScript::=**



### セマンティクス

構文の要素	説明
GLOBAL	<p>スクリプトをグローバルとして指定します。</p> <p>グローバル・スクリプトを削除する場合は、Recovery Manager が仮想プライベート・カタログに接続していないことが必要です。仮想カタログのユーザーは、グローバル・スクリプトを実行できますが、変更することはできません。</p> <p><b>関連項目：</b> <a href="#">グローバル・スクリプトとローカル・スクリプトの違いについては、2-106 ページの「使用上の注意」を参照してください。</a></p>
SCRIPT <i>script_name</i>	<p>削除するスクリプトの名前を指定します。スクリプト名に空白または予約語が含まれている場合は、引用符で囲む必要があります。</p>

## 例

### 例 2-67 グローバル・スクリプトの削除

この例では、リカバリ・カタログからグローバル・スクリプト backup\_db を削除します（例には出力例も含まれます）。

```
RMAN> LIST SCRIPT NAMES;
```

```
List of Stored Scripts in Recovery Catalog
```

```
Scripts of Target Database PROD
```

```
Script Name  
Description
```

```
-----  
backup_whole  
backup whole database and archived redo logs
```

```
Global Scripts
```

```
Script Name  
Description
```

```
-----  
global_backup_db  
back up any database from the recovery catalog, with logs
```

```
RMAN> DELETE GLOBAL SCRIPT global_backup_db;
```

```
deleted global script: global_backup_db
```

```
RMAN> LIST SCRIPT NAMES;
```

```
List of Stored Scripts in Recovery Catalog
```

```
Scripts of Target Database PROD
```

```
Script Name  
Description
```

```
-----  
backup_whole  
backup whole database and archived redo logs
```

---

## DROP CATALOG

### 用途

DROP CATALOG コマンドを使用すると、リカバリ・カタログまたは仮想プライベート・カタログを削除できます。

**関連項目：** リカバリ・カタログを削除する方法は、『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』を参照してください。

### 前提条件

このコマンドは、Recovery Manager プロンプトでのみ実行してください。

CATALOG コマンドライン・オプションまたは **CONNECT CATALOG** コマンドを使用して、リカバリ・カタログ・スキーマまたは仮想プライベート・カタログに接続する必要があります。リカバリ・カタログ・データベースはオープンの状態である必要があります。

ターゲット・データベースへの接続は不要です。

### 使用上の注意

DROP CATALOG コマンドを実行すると、この操作を実行することを確認するためにコマンドを再度入力するよう求められます。

基本リカバリ・カタログは **CREATE CATALOG** で作成されますが、仮想プライベート・カタログは **CREATE VIRTUAL CATALOG** で作成されます。基本リカバリ・カタログを削除するには、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、DROP CATALOG を実行します。

---

**注意：** 基本リカバリ・カタログを削除すると、すべての Recovery Manager メタデータがリカバリ・カタログから削除されます。リカバリ・カタログに記録されていても、ターゲット・データベースの制御ファイルに記録されていないバックアップは、Recovery Manager で使用できません。

---

仮想プライベート・カタログを削除するには、その仮想プライベート・カタログに接続しているときに DROP CATALOG コマンドを実行します。仮想プライベート・カタログに接続しているときに DROP CATALOG コマンドを実行しても、基本リカバリ・カタログ自体は削除されません。削除されるのは、基本カタログを参照するシノニムおよびビューのみです。

リリースが 10.2 以前の Recovery Manager クライアントを使用している場合に仮想カタログを削除するには、別の方法を使用する必要があります。仮想プライベート・カタログを削除する前に、リカバリ・カタログ・データベースに仮想プライベート・カタログの所有者として接続し、次の PL/SQL プロシージャを実行してください (`base_catalog_owner` は、基本リカバリ・カタログを所有するデータベース・ユーザーです)。

```
base_catalog_owner.DBMS_RCVCAT.DROP_VIRTUAL_CATALOG
```

基本リカバリ・カタログを削除すると、仮想プライベート・カタログは削除していなくても使用できなくなります。ただし、専任データベース・ユーザーが仮想プライベート・カタログを所有している場合は、DROP USER ... CASCADE を実行して、その仮想カタログを削除できます。

## 構文

**dropCatalog::=**



## 例

### 例 2-68 仮想プライベート・カタログの削除

データベース・ユーザー `vpu1` に属する仮想プライベート・カタログを削除するとします。ただし、その基本リカバリ・カタログは削除しません。この例では、リカバリ・カタログ・データベースに `vpu1` として接続し、このユーザーの仮想プライベート・カタログを削除します。仮想プライベート・カタログを削除しても、その基本リカバリ・カタログは影響を受けません。

```
RMAN> CONNECT CATALOG vpu1@catdb
```

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

```
RMAN> DROP CATALOG;
```

```
recovery catalog owner is VPU1
enter DROP CATALOG command again to confirm catalog removal
```

```
RMAN> DROP CATALOG;
```

```
virtual catalog dropped
```

## DROP DATABASE

### 用途

DROP DATABASE コマンドを使用すると、ターゲット・データベースを削除できます。Recovery Manager がリカバリ・カタログに接続している場合、そのデータベースの登録も同時に解除できます。Recovery Manager は、ターゲット・データベースに属す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。デフォルトでは、Recovery Manager によって確認のプロンプトが表示されます。

### 前提条件

このコマンドは、Recovery Manager プロンプトでのみ実行してください。ターゲット・データベースに接続している必要があります。ターゲット・データベースは、排他的にマウントされ、オープンされていない状態で、RESTRICT モードで起動されている必要があります。

### 構文

**dropDatabase::=**



### セマンティクス

構文の要素	説明
INCLUDING BACKUPS	ターゲット・データベースに関連付けられたバックアップ・セット、プロキシ・コピー、イメージ・コピーおよびアーカイブ REDO ログをすべての構成済デバイス・タイプから削除します。  <b>注意:</b> リカバリ・カタログを使用している場合、データベースの削除時に NOCATALOG モードで Recovery Manager を実行すると、Recovery Manager は、リカバリ・カタログでは認識されるがターゲット・データベースの制御ファイルには存在しないバックアップは削除しません。
NOPROMPT	確認のプロンプトを表示せずに、データベースを削除します。

### 例

#### 例 2-69 データベースの削除

この例では、リカバリ・カタログに登録されている test1 というテスト・データベースを削除します。Recovery Manager クライアントを起動し、TARGET としてデータベース test1 に接続して、リカバリ・カタログに接続します。その後、次のコマンドを実行して、ターゲット・データベース・ファイルと、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、コピーおよびアーカイブ・ログを削除します。

```

RMAN> CONNECT TARGET SYS@test1

target database Password: password
connected to target database: TEST1 (DBID=39525561)

RMAN> STARTUP FORCE MOUNT
RMAN> SQL 'ALTER SYSTEM ENABLE RESTRICTED SESSION';
RMAN> DROP DATABASE INCLUDING BACKUPS NOPROMPT;
  
```



## DUPLICATE

### 用途

DUPLICATE コマンドを使用すると、ソース・データベースのコピーを作成できます。Recovery Manager で作成できるのは、次のいずれかのタイプのデータベースです。

- **複製データベース**。これは、一意の DBID を持つソース・データベース（またはソース・データベースのサブセット）のコピーです。複製データベースは一意の DBID を持っているため、ソース・データベースから独立しており、同じリカバリ・カタログに登録できます。通常、複製データベースはテストに使用します。
- **スタンバイ・データベース**。これは、ソース・データベース（Data Guard 環境では**プライマリ・データベース**と呼ばれます）の特別なコピーであり、プライマリ・データベースのアーカイブ REDO ログを適用して更新されます。スタンバイ・データベースは新規 DBID を取得しません。

Recovery Manager では、複製処理をオープンまたはマウントされているデータベースから直接実行することも、既存の Recovery Manager バックアップおよびコピーから実行することもできます。

#### 関連項目：

- DUPLICATE コマンドで複製データベースを作成する方法は、
- 『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』を参照してください。
- スタンバイ・データベースを作成、管理およびバックアップする方法は、『Oracle Data Guard 概要および管理』を参照してください。

### 追加トピック

- [前提条件](#)
- [使用上の注意](#)
- [構文](#)
- [セマンティクス](#)
- [例](#)

### 前提条件

Recovery Manager は、ソース・データベースに TARGET として接続されている必要があります。ソース・データベースは、コピーされるデータベースです。ソース・データベースはマウントまたはオープン状態である必要があります。スタンバイ・データベースは、ソース・データベースにはできません。

Recovery Manager は、複製データベースのインスタンスに AUXILIARY として接続されている必要があります。複製データベースのインスタンスは、**補助インスタンス**と呼ばれます。補助インスタンスは、NOMOUNT オプションで起動されている必要があります。

ソース・データベースと複製データベースのプラットフォームは、同じである必要があります。DUPLICATE の処理では、プラットフォームが同じ 32 ビット・バージョンと 64 ビット・バージョンは、同じプラットフォームとみなされます。たとえば、Linux IA (32-bit) Little と Linux IA (64-bit) Little は、同じプラットフォームとみなされます。ただし、32 ビットと 64 ビットのプラットフォーム間でデータベースの複製が終了したら、utlirp.sql スクリプトを実行して、PL/SQL コードを新しい形式に変換する必要があります。このスクリプトは、Linux および UNIX プラットフォームの ORACLE\_HOME/rdbms/admin にあります。

DUPLICATE コマンドを実行するには、構成済または割当て済の補助チャンネルが1つ以上必要です。これらのチャンネルは、補助データベース・インスタンスで複製処理を実行します。

Recovery Manager は、ソース・データベースに TARGET として接続されている必要があります。次のような場合、Recovery Manager はソース・データベースのチャンネル構成を補助チャンネルに使用します。

- 補助チャンネルの手動割当てに `ALLOCATE CHANNEL` を使用しなかった場合
- 補助チャンネルの構成に `CONFIGURE` を使用しなかった場合

CONNECT 文字列を使用するように自動ターゲット・チャンネルを構成した場合は、Recovery Manager が補助インスタンスのチャンネルに同じチャンネル構成を使用しようとします。かわりに、補助チャンネルを手動で割り当てることをお勧めします。

ソース・ホストは、ソース・データベースが格納されているデータベースです。トランスポート先ホストは、複製データベースが作成されるデータベースです。複製データベースをソース・ホストに作成する場合は、DUPLICATE コマンドの実行時に、ソース制御ファイルが使用中であることが原因でエラーが発生しないように、CONTROL\_FILES 初期化パラメータを適切に設定してください。また、ソース・データベース・ファイルが複製データベース・ファイルで上書きされないように、すべての \*\_DEST 初期化パラメータも適切に設定してください。

COMPATIBLE 初期化パラメータが 11.0.0 以上に設定されている場合、デフォルトでは、トランスポート後に読み書き両用に設定されていなかったトランスポート表領域が複製されます。それ以外の場合、Recovery Manager では、トランスポート後に読み書き両用に設定されていないかぎり、トランスポート表領域は複製できません。

### 表領域と列の暗号化

列レベルで機能する透過的データ暗号化と表領域暗号化のデータベース暗号化機能のいずれにも、ウォレットが使用されています。次の制限事項に注意してください。

- 複製する表領域が暗号化されている場合は、複製データベースにウォレットを手動でコピーする必要があります。
- 複製データベースにウォレットがすでに存在する場合は、ソース・データベースのウォレットを複製データベースにコピーすることはできません。したがって、ウォレットがすでに存在するデータベースには、暗号化データをトランスポートできません。データの透過的暗号化機能で列を暗号化する場合は、パスワードで保護されたエクスポート・ファイルに列をエクスポートし、そのデータを複製データベースにインポートできます。
- 暗号化された表領域は、エンディアン形式が異なるプラットフォーム間で複製することはできません。

**関連項目：** 透過的データ暗号化については、『Oracle Database Advanced Security 管理者ガイド』を参照してください。

### バックアップベースの複製に固有の前提条件

FROM ACTIVE DATABASE を指定せずに DUPLICATE を実行する場合、補助チャンネルは1つ以上必要ですが、ソース・データベースに通常のチャンネルは必要ありません。

バックアップからデータベースを複製する場合は、複製データベースの作成とリカバリに使用するすべてのバックアップおよびアーカイブ REDO ログに、トランスポート先ホストのサーバー・セッションからアクセスする必要があります。トランスポート先ホストとソース・ホストが同じでない場合は、ソース・ホストのディスク上のバックアップを、ソース・データベースでのフルパス名と同じフルパス名で、トランスポート先ホストから使用できるようにする必要があります。

### アクティブなデータベースの複製に固有の前提条件

FROM ACTIVE DATABASE を指定して DUPLICATE を実行する場合、1 つ以上の通常のターゲット・チャンネルおよび 1 つ以上の AUXILIARY チャンネルが必要です。

補助データベース・インスタンスに接続する際は、ネット・サービス名を指定する必要があります。この要件は、補助インスタンスがローカル・ホストに存在する場合にも適用されます。ソース・データベースと補助インスタンスは、同じ SYSDBA パスワードを使用する必要があります。したがって、両方のインスタンスにパスワード・ファイルが存在している必要があります。補助インスタンスを起動してソース・データベースから接続できるように、パスワードを 1 つにしてパスワード・ファイルを作成できます。

パスワード・ファイルに対する DUPLICATE の動作は、複製データベースがスタンバイ・データベースとして動作するかどうかで異なります。作成する複製データベースがスタンバイ・データベースでない場合、デフォルトでは、Recovery Manager はパスワード・ファイルをコピーしません。PASSWORD FILE オプションを使用すると、補助インスタンス上の既存のパスワード・ファイルを上書きするように Recovery Manager に指定できます。

スタンバイ・データベースを作成すると、デフォルトでは、Recovery Manager がパスワード・ファイルをスタンバイ・ホストにコピーするため、既存のパスワード・ファイルは上書きされます。その場合は、PASSWORD FILE 句は必要ありません。

ソース・データベースはマウントまたはオープン状態である必要があります。ソース・データベースがオープン状態の場合は、アーカイブ処理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されておらず、それがスタンバイ・データベースでもない場合は、一貫性を保った状態で停止されていたことが必要です。

アクティブなデータベースの複製を実行する場合は、UNTIL 句を使用できません。Recovery Manager は、データファイルを一貫性のとれた時点にリカバリできるように、オンライン・データファイルが完全にコピーされた時間を選択します。

**関連項目：** 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。

## 使用上の注意

アクティブなデータベース複製では、補助サービス名を使用して、ソース・データベースがネットワーク経由でトランスポート先ホストの補助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、すでに存在する Recovery Manager バックアップおよびコピーが使用されます。表 2-6 に、ソース・データベースから複製されるファイルを示します。

表 2-6 複製されるファイル

ソース・データベースのファイル	アクティブなデータベース	バックアップベース
制御ファイル	FOR STANDBY が指定されると、ソース・データベースからコピーされ、指定されない場合は再作成されます。	FOR STANDBY が指定されると、バックアップからリストアされ、指定されない場合は再作成されます。
データファイル	ソース・データベースからコピーされます (SKIP オプションで除外されている場合を除く)。	バックアップからリストアされます (SKIP オプションで除外されている場合を除く)。
一時ファイル	再作成されます (2-130 ページの「一時ファイルの再作成」を参照)。	再作成されます (2-130 ページの「一時ファイルの再作成」を参照)。
オンライン REDO ログ・ファイル	再作成されます。	再作成されます。
スタンバイ REDO ログ・ファイル	プライマリ・データベース上で FOR STANDBY が指定および定義されている場合は、再作成されます。	プライマリ・データベース上で FOR STANDBY が指定および定義されている場合は、再作成されます。
アーカイブ REDO ログ・ファイル	ソース・データベースからコピーされます (ただし、複製に必要な場合のみ)。	バックアップまたはカタログ化済のコピーから取得されます (ただし、複製に必要な場合のみ)。

表 2-6 複製されるファイル (続き)

ソース・データベースのファイル	アクティブなデータベース	バックアップベース
サーバー・パラメータ・ファイル	ソース・データベースからコピーされず (2-134 ページの「dupOptionList」の SPFILE 句を参照)。	SPFILE 句が指定されていれば、バックアップからリストアされます (2-134 ページの「dupOptionList」を参照)。
フラッシュバック・ログ・ファイル	再作成されません。	再作成されません。
ブロック・チェンジ・トラッキング・ファイル	再作成されません。	再作成されません。
パスワード・ファイル	スタンバイ・データベースの場合はデフォルトでコピーされ、非スタンバイのデータベースでは、PASSWORD FILE オプションが指定されている場合にのみコピーされます。	再作成されません。
フラッシュ・リカバリ領域のバックアップおよび他のファイル	コピーされません。	コピーされません。

データファイルがオフラインである場合または除外されている場合を除き、複製データベースにはすべてのデータファイルが格納されます。表領域を除外するには、SKIP 句を使用するか、または DUPLICATE ... TABLESPACE を使用して表領域のサブセットのみを含めます。

**注意：** リカバリ・カタログで認識できるのは、現在の読取り専用表領域のみです。現在、表領域が読み書き両用に設定されていても、UNTIL を使用して、データベースを表領域が読取り専用であった過去の SCN に複製する場合、この表領域は複製データベースには含まれません。過去に読取り専用であった表領域は、オフラインの表領域とみなされるため、複製には含まれません。

フラッシュ・リカバリ領域を明示的に定義している場合は、複製データベースまたはスタンバイ・データベース上でこの領域が定義されます。また、ソース・データベース上でフラッシュ・リカバリが定義されており、DUPLICATE コマンドを使用してコピーまたはリストアされたサーバー・パラメータ・ファイルを補助インスタンスで使用しているときは、複製データベースまたはスタンバイ・データベース上でフラッシュ・リカバリが定義されます。

アクティブなデータベース複製を使用する場合は、2-133 ページの「dupsbyOptionList」にある FROM ACTIVE DATABASE の説明を参照してください。使用上の注意は、2-134 ページの「dupOptionList」を参照してください。

### バックアップベースの複製

NOARCHIVELOG モードのデータベースをバックアップベースで複製する場合は、メディア・リカバリに NOREDO オプションが使用されます。したがって、増分バックアップが存在していると、Recovery Manager は、リカバリ時には、その増分バックアップのみをリストアされたファイルに適用します。ARCHIVELOG モードのデータベースをバックアップベースで複製する場合、デフォルトでは、Recovery Manager によってそのコマンドが実行された時点で最後に作成されたアーカイブ REDO ログまで、または SET UNTIL 句で指定した時点までリカバリされます。

バックアップベースの複製を使用している場合で、ソース・データベースと補助インスタンスが別々のホストに存在する場合は、ソース・データベースのバックアップが補助インスタンスから使用できるようにする方法を決定する必要があります。

ターゲット・データベースで、ASM ストレージのリカバリ領域が使用されていない場合は、DUPLICATE コマンドを実行する前に、次のいずれかのタスクを実行します。

- SBT バックアップを使用している場合は、バックアップを作成するテープがトランスポート先ホストからアクセスできるようにします。
- ディスク・バックアップを使用している場合で、トランスポート先ホスト上でソース・ホストと同じバックアップ・ディレクトリ名を使用できる場合は、次のいずれかを実行します。
  - ソース・ホストのバックアップおよびコピーをトランスポート先ホストの同じパスに手動で転送します。
  - NFS または共有ディスクを使用し、トランスポート先ホストの同じパスにアクセスできることを確認します。
- ディスク・バックアップを使用しているときに、トランスポート先ホスト上でソース・ホストと同じバックアップ・ディレクトリ名を使用できない場合は、『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』で説明されている方法を使用します。

ソース・データベースで ASM ストレージのリカバリ領域が使用されている場合は、DUPLICATE コマンドを実行する前に、次のいずれかのタスクを実行します。

- データベースのバックアップを、フラッシュ・リカバリ領域の外に作成します。このバックアップは、次のようにしてアクセス可能にすることができます。
  - NFS を使用して、バックアップを同じ名前前でトランスポート先ホストにマウントします。
  - NFS を使用して、バックアップを別の名前前でトランスポート先ホストにマウントしてから、Recovery Manager がソース・データベースに TARGET として接続されている状態で、バックアップに対して CATALOG を実行します。
- フラッシュ・リカバリ領域をテープにバックアップし、それを複製処理に使用します。

### Oracle Managed Files での複製

ソース・データベース・ファイルが Oracle Managed Files (OMF) 形式の場合は、初期化パラメータの DB\_FILE\_NAME\_CONVERT および LOG\_FILE\_NAME\_CONVERT、または [fileNameConversionSpec](#) 句を使用して、複製データベースの新しい OMF ファイル名を生成することはできません。OMF ファイル名は一意であり、Oracle データベースによって生成されます。

このルールの唯一の例外は、ASM ディスク・グループ名のみを変更する場合です。ソース・データファイルおよびオンライン REDO ログ・ファイルが、ASM ディスク・グループ +SOURCEDSK に格納されているとします。複製データベース・ファイルは、ASM ディスク・グループ +DUPDSK に格納することにします。その場合、初期化パラメータを次のように設定できます。

```
DB_FILE_NAME_CONVERT = ("+SOURCEDSK", "+DUPDSK")
LOG_FILE_NAME_CONVERT = ("+SOURCEDSK", "+DUPDSK")
```

Recovery Manager は、DB\_FILE\_NAME\_CONVERT または LOG\_FILE\_NAME\_CONVERT を使用してディスク・グループ名を変換し、その名前に基づいた有効なファイル名を新しく生成します。

ソース・ファイルが Oracle Managed Files 形式の場合にデータファイルの名前を付けるオプションとしては、次のサポート・オプションがあります。

- SET NEWNAME を使用すると、個々のデータファイルの名前を指定できます。
- DB\_FILE\_CREATE\_DEST を設定すると、新しいデータベースのすべてのデータファイルを Oracle Managed Files にできます (ただし、SET NEWNAME が使用されるファイルは除きます)。DB\_FILE\_CREATE\_DEST を設定する場合は、DB\_FILE\_NAME\_CONVERT を設定しないでください。

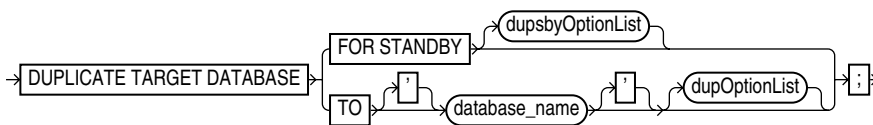
Oracle Managed Files から複製されるオンライン REDO ログに名前を付けるサポート・オプションには、DB\_CREATE\_FILE\_DEST、DB\_RECOVERY\_FILE\_DEST および DB\_CREATE\_ONLINE\_LOG\_DEST\_n があります。

**一時ファイルの再作成**

Oracle Managed Files に DUPLICATE を使用する場合は、データベースをプライマリ・データベースとしてオープンするか、読取り専用でオープンするときのいずれかの時点で、Recovery Manager が現行の DB\_CREATE\_FILE\_DEST で一時ファイルを再作成します。Oracle Managed Files を使用しない場合は、DB\_FILE\_NAME\_CONVERT を使用して、新しいデータベースの一時ファイル名を変換します。スタンバイ・データベースまたは複製データベースを読取り専用または読み書き両用モードでオープンする場合、Oracle では、必要に応じて DB\_FILE\_NAME\_CONVERT に基づいた変換名で一時ファイルが自動作成されます。一時ファイルごとに異なる名前を指定するには、2-273 ページの「SWITCH TEMPFILE」を参照してください。

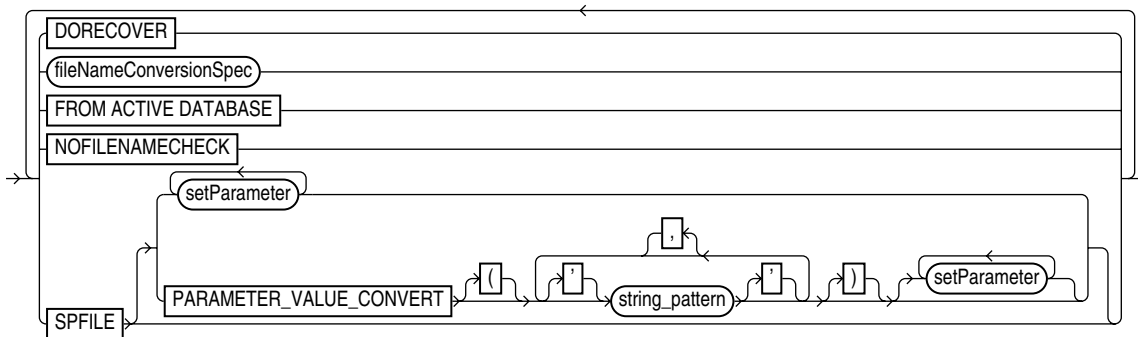
**構文**

**duplicate::=**



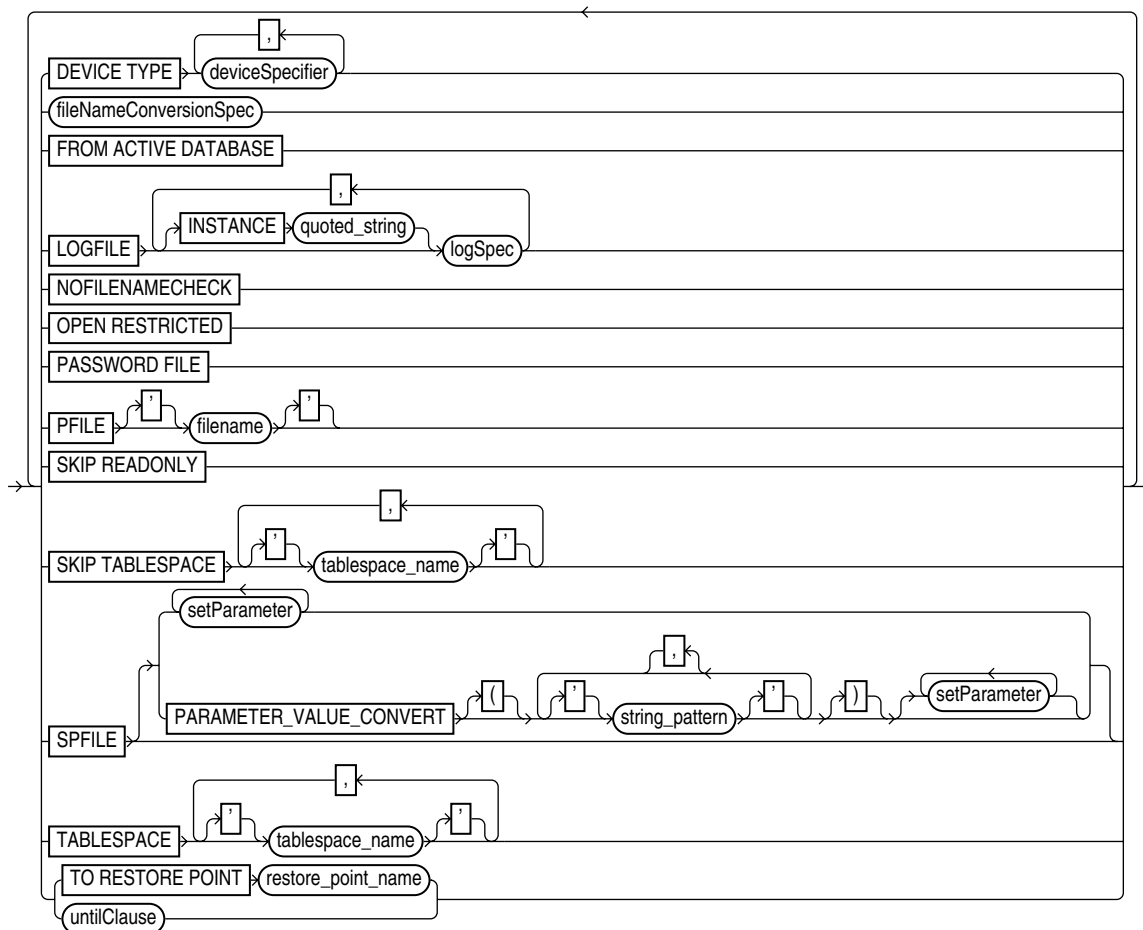
(dupsbyOptionList::= (2-130 ページ)、dupOptionList::= (2-131 ページ))

**dupsbyOptionList::=**



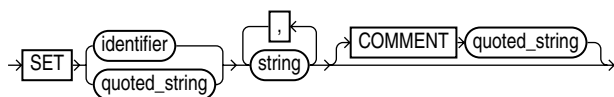
(fileNameConversionSpec::= (3-16 ページ)、setParameter::= (2-131 ページ))

**dupOptionList::=**

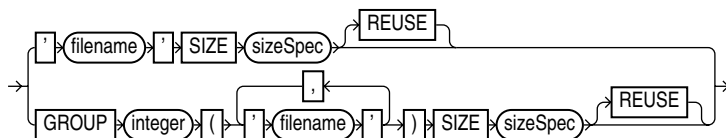


(deviceSpecifier::= (3-15 ページ)、 fileNameConversionSpec::= (3-16 ページ)、  
logSpec::= (2-131 ページ)、 setParameter::= (2-131 ページ)、 untilClause::= (3-38 ページ))

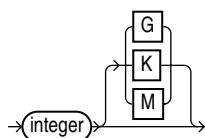
**setParameter::=**



**logSpec::=**



**sizeSpec::=**



## セマンティクス

### duplicate

この句を使用すると、データベースまたは表領域を複製できます。構文は、2-130 ページの [duplicate::=](#) 図を参照してください。

構文の要素	説明
FOR STANDBY	複製対象のデータベースをスタンバイ・データベースとして使用するよう指定します (2-142 ページの <a href="#">例 2-75</a> を参照)。  DUPLICATE コマンドでスタンバイ・データベースを作成するには、FOR STANDBY オプションを指定する必要があります。DUPLICATE ... FOR STANDBY コマンドを使用すると、スタンバイ制御ファイルをリストアしてマウントすることによって、スタンバイ・データベースが作成されます。FROM ACTIVE DATABASE を指定すると、Recovery Manager はプライマリ・データベースからスタンバイ・データベースにデータファイルをコピーします。指定しない場合、Recovery Manager はソース・データベースのデータファイルのバックアップをスタンバイ・データベースにリストアします。SET UNTIL を指定しないかぎり、Recovery Manager は、最新ファイルをリストアします。  DORECOVER を指定すると、Recovery Manager ではデータベースもリカバリされません。スタンバイ・データベースは、複製の完了後もマウントされたままです。  スタンバイ・データベース上のオンライン REDO ログのファイル名の変換には、SET NEWNAME または <a href="#">CONFIGURE AUXNAME</a> は使用できません。  Recovery Manager をスタンバイ・データベースに接続してから、DUPLICATE ... FOR STANDBY を使用してスタンバイ・データベースを追加作成することはできません。スタンバイ・データベースを追加作成するには、Recovery Manager をオリジナルのプライマリ・データベースに接続して DUPLICATE ... FOR STANDBY を実行します。  <b>注意:</b> DUPLICATE コマンドを使用してスタンバイ・データベースを作成することはできますが、アクティブ化することはできません。  Recovery Manager をスタンバイ・データベースに接続し、そのプライマリ・データベースがすでに登録されているリカバリ・カタログに接続すると、スタンバイ・データベースは、Recovery Manager によって認識され、暗黙的に登録されます。スタンバイ・データベースには、REGISTER コマンドを使用しないでください。
<a href="#">dupsbyOptionList</a>	スタンバイ・データベースの作成時にのみ適用するオプションを指定します。2-133 ページの「 <a href="#">dupsbyOptionList</a> 」を参照してください。
TO <i>database_name</i>	複製データベースの名前を指定します。この複製データベースは、スタンバイ・データベースにできません。  SPFILE 句を指定しない場合は、指定するデータベース名が、複製データベース・インスタンス (Recovery Manager が AUXILIARY として接続されているインスタンス) の初期化パラメータ・ファイルに記述されている名前に一致する必要があります。一致しない場合、データベースはエラーを発行します。  複製データベースがソース・データベースと同じ Oracle ホームにある場合、ソース・データベースと複製データベースに同じデータベース名は使用できません。ただし、複製データベースがソース・データベースとは別の Oracle ホームにある場合は、その Oracle ホームにある他のデータベース名と異なっていれば、同じ名前でも使用できます。複製データベースの管理を容易にするために、ソース・データベースと複製データベースを別の名前をすることをお勧めします。
<a href="#">dupOptionList</a>	スタンバイ・データベースではない複製データベースの作成時にのみ適用するオプションを指定します。2-134 ページの「 <a href="#">dupOptionList</a> 」を参照してください。



**dupsbyOptionList**

この副次句は、スタンバイ・データベースの作成時にのみ適用するオプションを指定します。構文は、2-130 ページの **dupsbyOptionList::=** 図を参照してください。

構文の要素	説明
DORECOVER	<p>スタンバイ・データベースの作成後に、<b>Recovery Manager</b> がそのスタンバイ・データベースをリカバリするように指定します。 <b>untilClause</b> を指定すると、<b>Recovery Manager</b> は、指定された SCN または時点までリカバリし、データベースをマウント状態のままにします。</p> <p><b>Recovery Manager</b> は、メディア・リカバリの完了後にスタンバイ・データベースをマウント状態のままにしますが、スタンバイ・データベースを手動リカバリ・モードおよび管理リカバリ・モードには設定しません。<b>Recovery Manager</b> がスタンバイ・データベースを作成した後、スタンバイ・データベースを手動リカバリ・モードまたは管理リカバリ・モードに設定したり、読取り専用モードでオープンする前に、すべてのギャップ・シーケンスの問題を解決する必要があります。</p> <p>制御ファイルのチェックポイント SCN は、スタンバイ・サイトから使用できるか <b>Recovery Manager</b> バックアップに含まれるアーカイブ REDO ログに含める必要があります。たとえば、スタンバイ制御ファイルを作成し、その直後に順序が 100 の現行ログをアーカイブするとします。この場合、スタンバイ制御ファイルのバックアップはその時点以後にとられるため、少なくともログ順序 100 までスタンバイ・データベースをリカバリしないと、データベースから ORA-1152 エラー・メッセージが発行されます。</p>
<b>fileNameConversionSpec</b>	<p>スタンバイ・データベースで元のデータファイル名を新しいデータファイル名に変換する方法を指定します。</p> <p><b>関連項目</b> : 3-16 ページの「<b>fileNameConversionSpec</b>」を参照してください。</p>
FROM ACTIVE DATABASE	<p>スタンバイ・データベースのファイルは、ソース・データベースのバックアップからではなく、ソース・データベースから直接提供されるように指定します (2-140 ページの例 2-71 を参照)。</p> <p><b>関連項目</b> : コマンドの前提条件については、2-127 ページの「<b>アクティブなデータベースの複製に固有の前提条件</b>」を参照してください。</p>
NOFILENAMECHECK	<p>ソース・データベースのデータファイルが、使用中のスタンバイ・データベース・ファイルと同じ名前を共有しているかどうかを、<b>Recovery Manager</b> でチェックしないようにします。</p> <p>スタンバイとプライマリのデータファイルおよびオンライン REDO ログのファイル名が同じ場合は、<b>NOFILENAMECHECK</b> オプションが必須です (2-141 ページの例 2-74 を参照)。したがって、複製データベース・ファイル名をソース・データベース・ファイル名と同じにする場合に、データベースのホストが異なるときは、<b>NOFILENAMECHECK</b> を指定する必要があります。</p> <p><b>関連項目</b> : 2-134 ページの「<b>dupOptionList</b>」にある <b>NOFILENAMECHECK</b> の説明を参照してください。</p>

構文の要素	説明
SPFILE	<p>ソース・データベースのサーバー・パラメータ・ファイルを、スタンバイ・データベースにコピーします。コピー先の場所は、このファイルに対するオペレーティング・システム固有のデフォルトの場所です。</p> <p>Recovery Manager は、サーバー・パラメータ・ファイルを使用してスタンバイ・データベース作成用の補助インスタンスを起動します。DUPLICATE コマンドの残りのオプションは、サーバー・パラメータ・ファイルによってデータベース・インスタンスが起動された後で処理されます。</p> <p>SPFILE 句を指定して DUPLICATE を実行する場合は、テキストベースの初期化パラメータ・ファイルを使用して補助インスタンスがすでに起動されている必要があります。その場合に、一時初期化パラメータ・ファイルのパラメータが必要なのは、DB_NAME のみです（このパラメータは、任意の値に設定できます）。</p> <p>Recovery Manager は、バイナリのサーバー・パラメータ・ファイルをコピーし、SPFILE 句の設定に基づいてパラメータを変更してから、そのサーバー・パラメータ・ファイルでスタンバイ・インスタンスを再起動します。SPFILE を指定した場合は、テキストベースの一時初期化パラメータ・ファイルを使用してインスタンスが起動されることはありません。</p> <p>DUPLICATE で FROM ACTIVE DATABASE を指定した場合は、サーバー・パラメータ・ファイルがソース・データベース・インスタンスで使用されている必要があります。DUPLICATE で FROM ACTIVE DATABASE を指定しなかった場合は、サーバー・パラメータ・ファイルのバックアップがスタンバイ・データベースにリストアされます。</p> <p><b>関連項目：</b> SPFILE の使用例は、2-139 ページの例 2-70 および 2-140 ページの例 2-71 を参照してください。</p>
<code>setParameter</code>	<p>指定された初期化パラメータを指定された値に設定します。2-138 ページの「<code>setParameter</code>」を参照してください。</p>
PARAMETER_VALUE_CONVERT string_pattern [setParameter]	<p>最初の文字列に一致するすべての初期化パラメータ値を、2 番目の文字列で置換します。DB_FILE_NAME_CONVERT および LOG_FILE_NAME_CONVERT は、このルールから除外され、影響を受けません。</p> <p>PARAMETER_VALUE_CONVERT を使用すれば、初期化パラメータの値をまとめて設定し、それぞれを明示的に設定せずに済みます。たとえば、ソース・データベースではディスク・グループ +ALPHA を使用していても、スタンバイ・データベースで +BETA を使用する場合、SPFILE PARAMETER_VALUE_CONVERT ('+ALHPA', '+BETA') と指定することで、これらのディスク・グループを参照するすべてのパラメータを変更できます。</p> <p><b>注意：</b> PARAMETER_VALUE_CONVERT のパラメータ値は、大 / 小文字が区別されます（同じ値を、初期化パラメータ・ファイルやサーバー・パラメータ・ファイルに直接設定する場合は、大 / 小文字が区別されないこともあります）。</p>

### dupOptionList

この副次句には、複製時の各種機能（ファイルの名前付け、複製のエンド・ポイントの決定など）を制御するオプションが含まれます。構文は、2-131 ページの `dupOptionList::=` 図を参照してください。

---

**注意：** この句のオプションのいくつかは、2-133 ページの「`dupsbyOptionList`」のオプションと同じです。それらのオプションの説明は、ここでは省略します。

---

複製データベースのファイル名がソース・データベースのファイル名と異なる必要がある場合（トランスポート先ホストとソース・ホストが同じである場合など）は、データファイルとオンライン REDO ログに新しいファイル名を指定するか、ソース・データベースのファイル名を変換します。複製データベースのオンライン REDO ログおよびデータファイルにファイル名を指定しないと、Recovery Manager ではソース・データベースのデータファイル名が使用されません。

構文の要素	説明
DEVICE TYPE <i>deviceSpecifier</i>	<p>指定したデバイス (DISK または sbt など) のみに自動チャネルを割り当てます。</p> <p>このオプションが有効なのは、構成済の自動チャネルがあり、チャネルを手動で割り当てていない場合のみです。たとえば、自動ディスクおよびテープ・チャネルを CONFIGURE して DUPLICATE...DEVICE TYPE DISK を実行すると、Recovery Manager ではディスク・チャネルのみが割り当てられます。</p> <p><b>関連項目:</b> 3-15 ページの「<i>deviceSpecifier</i>」を参照してください。</p>
<i>fileNameConversionSpec</i>	<p>ソース・データベースのファイル名を複製データベースのファイル名にマップする 1 つ以上のパターンを指定します (2-140 ページの例 2-72 を参照)。</p> <p>DUPLICATE コマンドで設定される DB_FILE_NAME_CONVERT は、初期化パラメータ DB_FILE_NAME_CONVERT (設定されている場合) をオーバーライドします。たとえば、初期化パラメータ・ファイルの設定が DB_FILE_NAME_CONVERT=('disk1','disk2') であっても、DUPLICATE...DB_FILE_NAME_CONVERT ('disk1','disk3') を実行した場合、disk1 サブストリングは disk2 に変換されません。かわりに、Recovery Manager によって disk1 サブストリングが disk3 に変換されます。</p> <p>指定リストのファイルが DUPLICATE の変換パラメータに影響されない場合は、他の方法 (SET NEWNAME など) でそのファイルの名前を変更する必要があります。</p> <p><b>注意:</b> SPFILE 句を指定した場合に、DUPLICATE...DB_FILE_NAME_CONVERT を実行すると、SPFILE 構文で指定されている変換パラメータはすべてオーバーライドされます。たとえば、DUPLICATE コマンドで、SPFILE 句の内側と外側の両方で DB_FILE_NAME_CONVERT を指定した場合、SPFILE 句の外側の設定が優先されます。</p> <p><b>関連項目:</b> 3-16 ページの「<i>fileNameConversionSpec</i>」を参照してください。</p>
FROM ACTIVE DATABASE	<p>複製データベースのファイルは、ソース・データベースのバックアップからではなく、ソース・データベースから直接提供されるように指定します (2-139 ページの例 2-70 を参照)。UNTIL の時刻を指定しなかった場合は、オンライン・データファイルがコピーされる時刻に基づいて、Recovery Manager は複製の終了時刻を決定します。</p> <p><b>関連項目:</b> コマンドの前提条件については、2-127 ページの「<b>アクティブなデータベースの複製に固有の前提条件</b>」を参照してください。</p>
LOGFILE	<p>スタンバイ・データベースではない複製データベースを作成するときのオンライン REDO ログ作成のオプションを指定します (例 2-72 を参照)。</p>
INSTANCE ' <i>inst_name</i> '	<p>指定されたインスタンスのオンライン REDO ログを Real Applications Cluster (Oracle RAC) データベースに作成します。インスタンス名は、最大 80 文字の文字列です。</p> <p>Recovery Manager は、指定されたインスタンスにマップされたスレッドを自動的に使用します。INSTANCE 名を指定しなかった場合は、デフォルトのインスタンスにログ・ファイルが作成されます。</p> <p>この句は、DUPLICATE TARGET DATABASE を使用して、Oracle RAC データベースをシングル・インスタンスのデータベースに複製する場合に役に立ちます。それ以外の場合、INSTANCE を使用する必要はありません。LOGFILE 句を使用する場合は、INSTANCE を使用して、データベースのバックアップ時 (バックアップベースで複製する場合) または UNTIL TIME の実行時 (アクティブなデータベースを複製する場合) にオープンしていた各スレッドの RAC インスタンスの名前を指定します。</p>
<i>logSpec</i>	<p>オンライン REDO ログ・ファイルのファイル名およびグループを指定します。</p> <p><b>関連項目:</b> 有効なオプションについては、2-138 ページの「<i>logSpec</i>」を参照してください。</p>

構文の要素	説明
NOFILENAMECHECK	<p>ソース・データベース・ファイルが複製データベース・ファイルと同じ名前を共有するときに、ソース・データベースのデータファイルおよびオンライン REDO ログ・ファイルが使用中であるかどうかのチェックを、Recovery Manager でチェックしないようにします (2-141 ページの例 2-73 を参照)。複製操作によって有用なデータが上書きされないかどうかを確認するのは、ユーザーの責任です。</p> <p>このオプションが必要になるのは、ディスク構成、ディレクトリ構造およびファイル名がソース・データベースのホストと同じであるが、それとは別のホストに複製データベースを作成する場合です。たとえば、host1 の /dbs ディレクトリに小規模なデータベースがあるとします。</p> <pre data-bbox="548 495 857 573">/oracle/dbs/system_prod1.dbf /oracle/dbs/users_prod1.dbf /oracle/dbs/rbs_prod1.dbf</pre> <p>このデータベースを host2 に複製する必要があるとします。また、host2 には host1 と同じファイル・システム /oracle/dbs/* があり、複製データベースとソース・データベースで同じファイル名を使用する必要があるとします。この場合は、NOFILENAMECHECK を指定して、エラー・メッセージを回避します。Recovery Manager は別のホストを認識していないため、ファイル名のチェックが不要であることを自動的に判断できません。</p> <p>ソース・データベースと同じホストに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。設定されていると、次のエラーが発行される場合があります。</p> <pre data-bbox="548 852 1305 1031">RMAN-10035: exception raised in RPC: ORA-19504: failed to create file "/oracle/dbs/tbs_01.f" ORA-27086: skgfglk: unable to lock file - already in use SVR4 Error: 11: Resource temporarily unavailable Additional information: 8 RMAN-10031: ORA-19624 occurred during call to DBMS_BACKUP_RESTORE.RESTOREBACKUPPIECE</pre>
OPEN RESTRICTED	<p>SQL 文 ALTER SYSTEM ENABLE RESTRICTED SESSION を発行して、制限されたセッションを複製データベースで有効にします。Recovery Manager は、複製データベースがオープン状態になる直前にこの文を発行します。</p>
PASSWORD FILE	<p>補助インスタンスで現在使用されているパスワード・ファイルを、ソース・データベースのパスワード・ファイルで上書きするように指定します (2-139 ページの例 2-70 を参照)。このオプションは、FROM ACTIVE DATABASE が指定されている場合にのみ有効です。指定されていない場合は、Recovery Manager によりエラーが発行されます。</p> <p>FOR STANDBY を指定すると、Recovery Manager がパスワード・ファイルをデフォルトでコピーします。指定しない場合は、デフォルトではコピーしません。PASSWORD FILE を使用すると、ソース・データベースのパスワード・ファイルで、既存のパスワード・ファイルを上書きするように指定できます。本番データベースで使用可能なすべてのパスワードを複製データベースに組み込む場合は、PASSWORD FILE オプションを使用します。</p>
PFILE <i>filename</i>	<p>補助インスタンスで使用するテキストベースの初期化パラメータ・ファイルを指定します (2-140 ページの例 2-72 を参照)。Recovery Manager では、複製中に補助インスタンスが自動的に停止され、再起動されます。デフォルトの場所にあるサーバー・パラメータ・ファイルを補助インスタンスで使用しない場合は、補助インスタンスの起動時に Recovery Manager で使用す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。初期化パラメータ・ファイルは、複製の実行に使用する Recovery Manager クライアントと同じホストに存在する必要があります。</p> <p>デフォルトの場所にあるサーバー・パラメータ・ファイルを補助インスタンスで使用する場合は、PFILE を指定する必要はありません。</p>

構文の要素	説明
SKIP READONLY	<p>現行の読取り専用表領域にあるデータファイルを複製データベースから除外します (2-141 ページの例 2-73 を参照)。デフォルトでは、Recovery Manager は現在の読取り専用表領域を複製します。</p> <p>表領域は現在読み書き両用に設定されていますが、<i>untilClause</i> を使用して表領域が読取り専用であった SCN にデータベースを複製する場合、Recovery Manager はこの表領域を複製データベースに含めません。過去に読取り専用であった表領域は、オフラインの表領域とみなされるため、複製には含まれません。</p> <p><b>注意：</b>スキップした読取り専用表領域のレコードは、DBA_TABLESPACES にまだ存在しています。この機能によって、後で読取り専用表領域をアクティブ化できます。たとえば、読取り専用表領域のデータを書込み可能な DVD に格納し、後でその DVD をマウントしてデータを参照できます。</p>
SKIP TABLESPACE <i>tbs_name</i>	<p>指定した表領域を複製データベースから除外します (2-140 ページの例 2-72 を参照)。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。</p> <p>ソース・データベースのバックアップの一部が存在しない場合にそのデータベースを複製するには、SKIP TABLESPACE を指定する必要があります。SKIP TABLESPACE を指定しない場合、Recovery Manager は次の複製を試みます。</p> <ul style="list-style-type: none"> <li>■ データファイルがオンラインであるかどうかに関係なく、オンライン表領域内のすべてのデータファイル。</li> <li>■ NORMAL 以外のオプションでオフライン化されたすべての表領域。たとえば、Recovery Manager は IMMEDIATE オプションでオフライン化された表領域の複製を試みます。OFFLINE NORMAL 表領域は複製できませんが、この種の表領域は複製後に手動で追加できます。</li> <li>■ 表領域またはデータファイルの有効なバックアップが存在しない場合、DUPLICATE コマンドは失敗します。</li> </ul> <p>Recovery Manager では、完全かどうかはチェックされません。たとえば、データ表領域は複製できますが、データの索引を含む表領域や、パーティション表の 1 パーティションのみを含む表領域は複製できません。</p>
SPFILE	<p>サーバー・パラメータ・ファイルをソース・データベースから複製データベースにコピーします。2-133 ページにある「<i>dupsbyOptionList</i>」の SPFILE の説明を参照してください。</p>
<i>setParameter</i>	<p>指定された初期化パラメータを指定された値に設定します。2-138 ページの「<i>setParameter</i>」を参照してください。</p>
PARAMETER_VALUE_CONVERT <i>string_pattern</i> [ <i>setParameter</i> ]	<p>最初の文字列に一致するすべての初期化パラメータ値を、2 番目の文字列で置換します。2-133 ページにある「<i>dupsbyOptionList</i>」の PARAMETER_VALUE_CONVERT の説明を参照してください。</p>
TABLESPACE <i>tablespace_name</i>	<p>指定したデータベースに入れる表領域を指定します。複製データベースから除外する表領域を指定する SKIP TABLESPACE とは異なり、このオプションは、組み込む表領域を指定し、残りの表領域をスキップします。</p> <p><b>注意：</b>SYSTEM 表領域、SYSAUX 表領域および UNDO 表領域は、Recovery Manager によって複製データベースに自動的に含まれます。これらの表領域はスキップできません。</p>
TO RESTORE POINT <i>restore_point_name</i>	<p>リストア・ポイントを作成した時点の SCN を上限として、バックアップベースの複製のリストア・ポイントを指定します。指定した値は含まれます。上限値が含まれるため、Recovery Manager は、対応する SCN 以前のデータベースの複製に使用できるファイルのみを選択します。</p> <p><b>注意：</b><i>untilClause</i> に適用される制約と同じものが、TO RESTORE POINT にも適用されます。</p>

構文の要素	説明
<code>untilClause</code>	<p>バックアップベースの複製の Point-in-Time リカバリの終了時刻、SCN またはログ順序番号を設定します。(2-140 ページの例 2-72 を参照)。UNTIL 句は、アクティブなデータベース複製ではサポートされません。</p> <p>DUPLICATE コマンドの前に SET UNTIL を実行しても、同じ結果が得られます。複製に UNTIL 句を指定する場合は、次の制約が適用されます。</p> <ul style="list-style-type: none"> <li>■ NOREDO を使用するかどうかは、データベースの現在の状態に基づいて決定されます。指定した UNTIL の時刻または SCN の時点でアーカイブ・モードのデータベースが、現在のアーカイブ・モードと異なる場合、NOREDO は使用されません。</li> <li>■ 表領域が複製の時点で読取り専用だった場合は、SKIP READONLY が指定されていなくても、その表領域は含まれません。</li> <li>■ DUPLICATE コマンドのエンド・ポイントを、最新の ALTER DATABASE OPEN RESETLOGS の SCN より前にすることはできません。以前のデータベース・インカネーションへの複製はサポートされていません。</li> <li>■ 複製データベースを現時点、つまり最新の SCN までリカバリすることはできません。Recovery Manager は、使用可能な最新のアーカイブ・ログまたはその前まで複製データベースをリカバリしますが、オンライン・ログまではリカバリできません。</li> </ul> <p><b>関連項目:</b> 3-38 ページの「<code>untilClause</code>」を参照してください。</p>

### setParameter

この副次句は、サーバー・パラメータ・ファイルの値を指定します。

構文の要素	説明
<code>SET identifier string</code>	<p>指定された初期化パラメータを指定された値に設定します (2-140 ページの例 2-71 を参照)。SET を使用すると、メモリー内の差異を調整し、レプリケーション・オプションを無効化し、複製データベースの他のオプションを設定できます。</p> <p>この SET 機能は、サーバー・パラメータ・ファイルをリストアした後で複製を一時停止し、ALTER SYSTEM SET 文を発行して初期化パラメータ・ファイルを変更する操作と同じです。</p> <p>SET は、PARAMETER_VALUE_CONVERT の後に処理されます。あるパラメータで指定されたファイル名が PARAMETER_VALUE_CONVERT によって設定され、同じパラメータで指定されたファイル名が SET によって設定される場合、SET の値が PARAMETER_VALUE_CONVERT による設定をオーバーライドします。</p> <p><b>注意:</b> DUPLICATE コマンドで DB_FILE_NAME_CONVERT が指定された場合は、そのファイル名の設定は、SPFILE SET で指定された競合する設定をオーバーライドします。</p>
<code>COMMENT 'string'</code>	<p>パラメータ設定のオプションのコメントを指定します。</p>

### logSpec

この副次句では、スタンバイ・データベースではない複製データベースを作成するときのオンライン REDO ログを指定します。構文図は、2-131 ページの `logSpec::=` 図を参照してください。

LOGFILE 句を指定しない場合、Recovery Manager は LOG\_FILE\_NAME\_CONVERT が設定されていれば、それを使用します。LOGFILE も LOG\_FILE\_NAME\_CONVERT も設定しなければ、Recovery Manager は複製データベースの REDO ログ・ファイルにソース・データベースの REDO ログ・ファイル名を使用します。この場合は、NOFILENAMECHECK オプションを指定する必要があります。

構文の要素	説明
'filename' SIZE sizeSpec	オンライン REDO ログ・メンバーのファイル名と、KB 単位 (k) または MB 単位 (M) によるファイル・サイズを指定します。デフォルトはバイト単位です。
REUSE	データベースで既存のファイルを再利用できます。ファイルがすでに存在している場合は、そのサイズが SIZE パラメータと一致しているかどうかデータベースで検証されます。ファイルが存在しない場合は作成されます。
GROUP integer ('filename', ... ) SIZE sizeSpec	オンライン REDO ログ・メンバーを含むグループ、オンライン REDO ログ・メンバーのファイル名、および KB 単位 (k) または MB 単位 (M) のファイル・サイズを指定します。デフォルトはバイト単位です。
REUSE	データベースで既存のログを再利用できます。

## 例

### 例 2-70 アクティブなデータベースからの複製

データベース prod1 を基にしてテスト・データベースを新しいホストに作成するとします。新しいホストのディレクトリ構造はソース・ホストと同じであるため、複製データベースのファイルには、ソース・データベースのファイルと同じ名前を使用できます。データベースは、Recovery Manager バックアップを使用せずに作成でき、複製中も prod1 をオープン状態のままにしておくことができます。

prod1 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、任意の値に設定された DB\_NAME パラメータのみを含む初期化パラメータ・ファイルをトランスポート先ホストに作成できます。補助インスタンスを起動する前に、ソース・データベースと同じ SYSDBA パスワードを使用するパスワード・ファイルを作成する必要があります。補助インスタンスは、その後で起動します。

デフォルトでは、Recovery Manager はスタンバイ・データベースではない複製データベースを作成するときにパスワード・ファイルを複製しません。PASSWORD FILE オプションを指定すると、Recovery Manager はパスワード・ファイルをトランスポート先ホストにコピーします。ソース・データベースで使用可能なすべてのパスワードを複製データベースに組み込む場合は、PASSWORD FILE オプションを使用します。

補助チャンネルを構成する必要はありません。Recovery Manager は、ソース・データベースに構成された通常のチャンネルを使用してデータベース・ファイルをコピーします。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・インスタンスに次のように接続して、データベースを複製します。

```
% rman
RMAN> CONNECT TARGET SYS@prod1

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

RMAN> CONNECT AUXILIARY SYS@dup1

auxiliary database Password: password
connected to auxiliary database: DUP1 (not mounted)

RMAN> DUPLICATE TARGET DATABASE TO dup1
2> FROM ACTIVE DATABASE
3> PASSWORD FILE
4> SPFILE;
```

### 例 2-71 アクティブなデータベース複製でのサーバー・パラメータ・ファイルのコピー

データベース prod1 を基にしてスタンバイ・データベースを新しいホストに作成するとします。トランスポート先ホストとソース・ホストはディレクトリ構造が異なるため、スタンバイ・データベース・ファイルは、/disk1 ではなく /disk2 に保存されます。スタンバイ・データベースは、Recovery Manager バックアップを使用せずに作成でき、複製中も prod1 をオープン状態のままにしておくことができます。

最初に行うのは、スタンバイ・データベース用に必要最小限の初期化パラメータ・ファイルを作成することです。次に、スタンバイ・インスタンスを起動します。このパラメータ・ファイルは、必要最小限のものとし、それは、SPFILE オプションを使用すると、サーバー・パラメータ・ファイルが新規ホストにコピーされ、各種パラメータが新しい指定値に設定されるためです。

Recovery Manager クライアントを起動し、CONNECT を実行して TARGET としてソース・データベースに接続して、補助インスタンスに接続します。補助チャンネルを構成する必要はありません。Recovery Manager は、ソース・ホストの通常のチャンネルを使用してデータベース・ファイルをコピーします。次のコマンドを入力できます。

```
DUPLICATE TARGET DATABASE TO dup1
FOR STANDBY
FROM ACTIVE DATABASE
PASSWORD FILE
SPFILE
  PARAMETER_VALUE_CONVERT '/disk1', '/disk2'
  SET DB_FILE_NAME_CONVERT '/disk1','/disk2'
  SET LOG_FILE_NAME_CONVERT '/disk1','/disk2'
  SET SGA_MAX_SIZE 200M
  SET SGA_TARGET 125M;
```

### 例 2-72 複製用の新しいファイル名の手動設定

host1 上のソース・データベースを、host2 上の newdb に複製するとします。

この使用例では、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ません。テキストベースの初期化パラメータ・ファイルを host2 上に作成し、インスタンスを起動します。

host2 で DUPLICATE を実行する場合は、PFILE パラメータを使用して、初期化パラメータ・ファイルの場所を指定する必要があります。Recovery Manager クライアントは、複製データベースの初期化パラメータ・ファイルと同じホストで使用が必要があることに注意してください。

表領域の example と history は、複製データベースに含めないことにします。したがって、これらの表領域については、DUPLICATE ...SKIP TABLESPACE を指定します。また、複製データベースの状態は、24 時間前の本番データベースと同じにします。したがって、DUPLICATE ...UNTIL TIME を使用します。

この例は、ソース・データベースのデータファイルが host1 のディレクトリ /h1/oracle/dbs/trgt に存在していることを前提としています。データファイルは、ディレクトリ /h2/oracle/oradata/newdb に複製するため、DUPLICATE ...DB\_FILE\_NAME\_CONVERT で複製データファイルの名前を生成するように指定します。複製データベース内のオンライン REDO ログ・ファイルの名前は、DUPLICATE ...LOGFILE を使用して指定します。

host2 上で Recovery Manager クライアントを起動し、CONNECT を実行して TARGET としてソース・データベースに接続して、補助インスタンスに接続します。次の RUN コマンドを入力できます。

```
RUN
{
  ALLOCATE AUXILIARY CHANNEL newdb DEVICE TYPE sbt;
  DUPLICATE TARGET DATABASE TO newdb
  PFILE ?/dbs/initNEWDB.ora
  UNTIL TIME 'SYSDATE-1' # specifies incomplete recovery
  SKIP TABLESPACE example, history # skip desired tablespaces
  DB_FILE_NAME_CONVERT ('/h1/oracle/dbs/trgt/', '/h2/oracle/oradata/newdb/')
}
```



```

LOGFILE
GROUP 1 ('?/oradata/newdb/redo01_1.f',
        '?/oradata/newdb/redo01_2.f') SIZE 4M,
GROUP 2 ('?/oradata/newdb/redo02_1.f',
        '?/oradata/newdb/redo02_2.f') SIZE 4M,
GROUP 3 ('?/oradata/newdb/redo03_1.f',
        '?/oradata/newdb/redo03_2.f') SIZE 4M REUSE;
}

```

### 例 2-73 複製データベースでのソース・データベースのファイル名の使用

Recovery Manager バックアップを使用して、テスト用の複製データベースを作成するとします。条件は、次のとおりです。

- ソース・ホストとは異なるトランスポート先ホストにリストアを行います。
- Recovery Manager は、リカバリ・カタログに接続していません。
- 自動チャネルを構成済です。
- ソース・ホストとトランスポート先ホストのファイル構造は同じです。
- 複製データベース・ファイルをソース・データベース・ファイルと同じ名前にします。
- 読取り専用表領域は複製しません。
- ソース・データベース上のファイルの名前が複製データベース・ファイルと同じ場合に、ソース・データベース上のファイルが使用中かどうかを、Recovery Manager ではチェックしません。

Recovery Manager クライアントを起動し、**CONNECT** を実行して TARGET としてソース・データベースに接続して、補助インスタンスに接続します。次のコマンドを入力できます。

```

DUPLICATE TARGET DATABASE TO ndbnewh
LOGFILE
    '?/dbs/log_1.f' SIZE 4M,
    '?/dbs/log_2.f' SIZE 4M
SKIP READONLY
NOFILENAMECHECK;

```

### 例 2-74 同じディレクトリ構造を持つスタンバイ・データベースの作成

Recovery Manager バックアップを使用して、ソース・ホストと同じディレクトリ構造を持つリモート・ホストにスタンバイ・データベースを作成するとします。ソース・データベースは prod1 と呼ばれ、Data Guard 環境のプライマリ・データベースになります。

最初に、Recovery Manager クライアントを起動し、**CONNECT** を実行して TARGET としてソース・データベース prod1 に接続して、補助インスタンスに接続します。次に、**CONFIGURE** を実行して、DB\_UNIQUE\_NAME が standby1 のスタンバイ・データベースのデフォルトのデバイス・タイプを sbt に構成できます。

```

CONFIGURE DEFAULT DEVICE TYPE sbt FOR DB_UNIQUE_NAME standby1;
CONFIGURE DEVICE TYPE sbt PARALLELISM 2 FOR DB_UNIQUE_NAME standby1;

```

スタンバイ・データベースの作成に必要なすべてのバックアップがテープにあるとします。スタンバイ・データベースの初期化パラメータ・ファイルで、DB\_UNIQUE\_NAME を standby1 に設定します。

スタンバイ・データベースでは、初期化パラメータ・ファイルのデフォルトの場所が使用されています。スタンバイ・インスタンス NOMOUNT が起動したら、Recovery Manager クライアントを起動し、**CONNECT** を実行して TARGET としてソース・データベースに接続して、補助インスタンスとリカバリ・カタログに接続します。次の DUPLICATE コマンドを実行して、**NOFILENAMECHECK** オプションを指定します (スタンバイとプライマリのデータファイルおよびオンライン REDO ログ・ファイルの名前は同じです)。

```

DUPLICATE TARGET DATABASE FOR STANDBY
NOFILENAMECHECK;

```

### 例 2-75 OMF および ASM でのスタンバイ・データベースの作成

Recovery Manager バックアップを使用して、OMF および ASM を使用するホストにスタンバイ・データベースを作成するとします。ソース・データベースは prod1 と呼ばれ、Data Guard 環境のプライマリ・データベースになります。

最初に、Recovery Manager クライアントを起動し、**CONNECT** を実行して TARGET としてデータベース prod1 に接続して、リカバリ・カタログに接続します。次の **CONFIGURE** コマンドを実行して、DB\_UNIQUE\_NAME が standby1、ネット・サービス名が sby1 となるスタンバイ・データベースのデフォルト・デバイス・タイプを sbt に構成します。

```
CONFIGURE CONNECT IDENTIFIER "sby1" FOR DB_UNIQUE_NAME standby1;  
CONFIGURE DEFAULT DEVICE TYPE TO sbt FOR DB_UNIQUE_NAME standby1;  
CONFIGURE DEVICE TYPE sbt PARALLELISM 2 FOR DB_UNIQUE_NAME standby1;
```

スタンバイ・データベースの作成に必要なバックアップは、すべてテープに保存されているとします。データベース standby1 の初期化パラメータ・ファイルで、次のパラメータを設定します。

- DB\_UNIQUE\_NAME の値を standby1 に設定します。
- DB\_CREATE\_FILE\_DEST および DB\_RECOVERY\_FILE\_DEST を、スタンバイ・ホスト上の目的の ASM ディスク・グループに設定します。たとえば、DB\_CREATE\_FILE\_DEST を +DATAFILE に、DB\_RECOVERY\_FILE\_DEST を +FLASH\_REC\_AREA に設定します。

スタンバイ・インスタンスが NOMOUNT モードであることを確認します。Recovery Manager クライアントを起動し、**CONNECT** を実行して TARGET としてデータベース prod1 に接続して、AUXILIARY として standby1 インスタンスに接続し、リカバリ・カタログに接続します。次のコマンドを入力して、スタンバイ・データベースを作成します。

```
DUPLICATE TARGET DATABASE FOR STANDBY TO standby1;
```

リストアされるデータファイルの新しい OMF/ASM データファイル名は、Recovery Manager で自動的に生成されます。新しいデータベース名とファイル名は、リカバリ・カタログと自動的に再同期されます。

## EXECUTE SCRIPT

### 用途

EXECUTE SCRIPT コマンドを使用すると、リカバリ・カタログに格納されているローカルまたはグローバルの Recovery Manager スクリプトを実行できます。

#### 関連項目：

- ストアド・スクリプトの使用方法は、『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』を参照してください。
- 2-106 ページの「[CREATE SCRIPT](#)」および 2-208 ページの「[REPLACE SCRIPT](#)」を参照してください。

### 前提条件

EXECUTE SCRIPT は、[RUN](#) コマンドのカッコ内でのみ使用してください。CATALOG コマンドライン・オプションまたは [CONNECT CATALOG](#) コマンドを使用して、Recovery Manager をリカバリ・カタログに接続する必要があります。リカバリ・カタログ・データベースはオープン状態である必要があります。

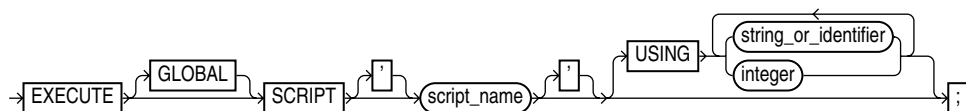
### 使用上の注意

[RUN](#) ブロック内で EXECUTE SCRIPT コマンドを実行すると、Recovery Manager によって、スクリプトの内容がその RUN ブロックのコンテキストに挿入されます。そのため、スクリプト内でチャンネルを割り当てている場合は、RUN ブロック内でチャンネルを割り当てないでください。

GLOBAL が指定されている場合は、同じ名前のグローバル・スクリプトがすでにリカバリ・カタログに存在している必要があります。存在していない場合は、エラー RMAN-06004 が戻されます。GLOBAL を指定しなかった場合は、Recovery Manager によって、現行のターゲット・データベースに対して定義されているローカル・ストアド・スクリプトが検索されます。その名前のローカル・スクリプトが見つからなかった場合は、同じ名前のグローバル・スクリプトが検索され、見つかった場合はそのスクリプトが実行されます。

### 構文

**executeScript ::=**



## セマンティクス

構文の要素	説明
GLOBAL	ローカル・ストアド・スクリプトのかわりに、グローバル・ストアド・スクリプトを実行するように指定します。 <b>関連項目:</b> グローバル・スクリプトとローカル・スクリプトの違いについては、2-106 ページの「 <a href="#">使用上の注意</a> 」を参照してください。
SCRIPT <i>script_name</i>	実行するストアド・スクリプトの名前を指定します。スクリプト名に空白または予約語が含まれている場合は、引用符で囲む必要があります。
USING [ <i>string_or_identifier</i>   <i>integer</i> ]	ストアド・スクリプト内の置換変数で使用する値を 1 つ以上指定します (例 2-77 を参照)。 <b>関連項目:</b> 置換変数を使用するストアド・スクリプトの作成方法については、「 <a href="#">CREATE SCRIPT</a> 」を参照してください。また、Recovery Manager で置換変数を使用する方法については、2-244 ページの「 <a href="#">RMAN</a> 」および 2-2 ページの「 <a href="#">@</a> 」を参照してください。

## 例

### 例 2-76 ストアド・スクリプトの実行

この例では、[LIST](#) を使用して、リカバリ・カタログに格納されているスクリプトを表示し、[PRINT SCRIPT](#) を使用して、例 2-59 「[グローバル・ストアド・スクリプトの作成](#)」で作成された `global_backup_db` の内容を表示します。最後に、`global_backup_db` を実行して、データベースをバックアップします。

```
RMAN> LIST SCRIPT NAMES;

List of Stored Scripts in Recovery Catalog

Global Scripts

Script Name
Description
-----
global_backup_db
back up any database from the recovery catalog, with logs

RMAN> PRINT SCRIPT global_backup_db;

printing stored global script: global_backup_db
{
  BACKUP DATABASE PLUS ARCHIVELOG;
}

RMAN> RUN { EXECUTE GLOBAL SCRIPT global_backup_db; }
```

executing global script: global\_backup\_db

```
Starting backup at 07-JUN-07
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=120 device type=DISK
.
.
.
```

### 例 2-77 置換変数を使用するストアド・スクリプトの作成と実行

Recovery Manager を起動してターゲット・データベースおよびリカバリ・カタログに接続したら、**REPLACE SCRIPT** を使用して、3つの置換変数が含まれるバックアップ・スクリプトを作成します。Recovery Manager によって、変数の初期値を入力するプロンプトが表示されます(太字がユーザー入力です)。

```
RMAN> REPLACE SCRIPT backup_df { BACKUP DATAFILE &1 TAG &2.1 FORMAT '/disk1/&3_%U'; }
```

```
Enter value for 1: 1
```

```
Enter value for 2: df1_backup
```

```
Enter value for 3: df1
```

```
starting full resync of recovery catalog  
full resync complete  
created script backup_df
```

後で、別の値で backup\_df スクリプトを実行できます。次の例では、値 3、test\_backup および test をストアド・スクリプトの置換変数に渡しています。

```
RMAN> RUN { EXECUTE SCRIPT backup_df USING 3 test_backup df3; }
```

値が置換された後、Recovery Manager によって BACKUP コマンドが次のように実行されます。

```
BACKUP DATAFILE 3 TAG test_backup1 FORMAT '/disk1/df3_%U';
```

## EXIT

---

### 用途

EXIT コマンドを使用すると、Recovery Manager ユーティリティを停止できます。このコマンドの機能は、QUIT コマンドと同じです。

### 前提条件

このコマンドは Recovery Manager プロンプトでのみ実行できます。

### 構文

**exit::=**

→EXIT→

### 例

#### **例 2-78 Recovery Manager の終了**

この例は、Recovery Manager を終了します。

```
RMAN> EXIT
```

## FLASHBACK DATABASE

### 用途

FLASHBACK DATABASE コマンドを使用すると、データベースをターゲットの時刻、SCN またはログ順序番号に戻すことができます。

このコマンドは、その実行時に存在するデータファイルに対して Oracle Database が加えていた変更を UNDO することで機能します。フラッシュバックを実行すると、論理的な障害は修正できますが、物理的な障害は修正できません。したがって、このコマンドを使用して、ディスクの障害や誤って削除したデータファイルをリカバリすることはできません。

通常、FLASHBACK DATABASE は、Point-in-Time リカバリを伴う RESTORE 操作よりも非常に短時間で実行できます。これは、指定したフラッシュバック時刻以降にデータベースに対して行われた変更の数によって、FLASHBACK DATABASE の実行に必要な時間が決まるためです。これに対して、リストアされたバックアップに基づく従来の Point-in-Time リカバリでは、データベースのサイズによって実行に必要な時間が決まります。

Data Guard 環境では、フラッシュバック・データベースの様々な用途が他にもあります。

#### 関連項目：

- コマンドの前提条件の完全なリストおよび FLASHBACK DATABASE の使用上の注意については、『Oracle Database SQL 言語リファレンス』を参照してください。
- Data Guard 環境でのフラッシュバック・データベースの用途については、『Oracle Data Guard 概要および管理』を参照してください。

### 前提条件

このコマンドは、Recovery Manager プロンプトまたは RUN コマンドから実行できます。

Recovery Manager は TARGET としてデータベースに接続されている必要があります。また、データベースは、Oracle Database 10g 以降である必要があります。ターゲット・データベースは、現行の制御ファイルを使用してマウント済である必要があります。つまり、バックアップを制御ファイルとすることはできません。また、制御ファイルの再作成もできません。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。

FLASHBACK DATABASE を使用して、制御ファイルのリストアまたは再作成が行われたときより前に戻すことはできません。データベースの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、既存のすべてのフラッシュバック・ログ情報が破棄されます。

フラッシュ・リカバリ領域は、フラッシュバック・ロギングが有効になるように構成され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 Oracle Managed Files として格納され、フラッシュ・リカバリ領域が構成されていない場合は作成されません。SQL 文 ALTER DATABASE ... FLASHBACK ON を使用して、フラッシュバックのターゲット時刻より前にフラッシュバック・ロギング機能を使用可能にしておく必要があります。フラッシュバック・ロギングが有効になっているかどうかは、V\$DATABASE.FLASHBACK\_ON を問い合わせることで確認できます。

データベースには、SQL 文 ALTER TABLESPACE ... FLASHBACK OFF でフラッシュバック機能が無効にされていたオンライン表領域が含まれていない必要があります。

## 使用上の注意

フラッシュバック・データベースの操作は、データベース全体に適用されます。個々の表領域をフラッシュ・バックすることはできません。フラッシュバック・データベース操作は、RECOVER によって実行される、データベースの Point-in-Time リカバリ (DBPITR) に似ていますが、Recovery Manager では、変更を UNDO して、ターゲットの時刻または SCN の前の時点に戻すために、フラッシュバック・ログが使用されます。Recovery Manager では、必要なアーカイブ REDO ログはバックアップから自動的にリストアされ、一貫性を保つようにデータベースがリカバリされます。一時表領域のデータはフラッシュ・バックされないことに注意してください。

フラッシュバック・データベース操作に使用可能な最も古い SCN は、DB\_FLASHBACK\_RETENTION\_TARGET 初期化パラメータの設定と、使用可能なディスク領域の制限の下で実際に保存されているフラッシュバック・ログとによって決まります。V\$DATABASE.CURRENT\_SCN の現行のデータベース SCN を表示します。

### フラッシュバック・データベースに対する NOLOGGING 操作の影響

NOLOGGING 操作が行われていたターゲット時刻を指定して FLASHBACK DATABASE を使用すると、NOLOGGING 操作の影響を受けたデータベース・オブジェクトとデータファイルのブロックが破損している可能性があります。たとえば、NOLOGGING モードでダイレクト・パス INSERT 操作を行い、その操作が 4 月 3 日の 9:00 ~ 9:15 に実行されるとします。後でフラッシュバック・データベースを使用して、その日の 09:07 に戻す場合、ダイレクト・パス INSERT で更新されたオブジェクトとデータファイルは、フラッシュバック・データベースが終了した後で、破損ブロックが存在するままになる可能性があります。

NOLOGGING 操作と重なるターゲット時刻または SCN を指定して FLASHBACK DATABASE を使用することは、可能なかぎり避けてください。また、NOLOGGING 操作を実行したら、その直後に、影響を受けた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を実行して、操作後の時点へのリカバリが可能になるようにしてください。FLASHBACK DATABASE を使用して、ダイレクト・パス INSERT などの操作の実行していた時点に戻ることが予想される場合は、その操作を LOGGING モードで実行することを検討してください。

**関連項目：** NOLOGGING モードをサポートしている操作の詳細は、『Oracle Database SQL 言語リファレンス』の logging\_clause の説明を参照してください。

### データファイルの状態の変更がフラッシュバック・データベースに与える影響

FLASHBACK DATABASE コマンドを実行しても、すべての必要なファイルおよびリソースがあることが確認されるまで、データベースの変更は開始されません。フラッシュバック・データベース操作は、データファイル、REDO ログ・ファイルまたはフラッシュバック・ログの欠落が原因では失敗しません。

データファイルの状態が、現行の SCN とフラッシュバックのターゲット SCN との間で変化している場合、FLASHBACK DATABASE コマンドの動作は、状態変化の内容によって異なります。詳細は、表 2-7 を参照してください。



表 2-7 データファイルの状態の変化に対する FLASHBACK DATABASE の応答

フラッシュバック中に実行するデータファイル操作	FLASHBACK DATABASE コマンドの応答
追加	データファイルのレコードを制御ファイルから削除します。
削除	データファイルを制御ファイルに追加し、オフラインとしてマークしてフラッシュバックしません。データファイルを同じ時刻または SCN までリストアおよびリカバリできます。
名前の変更	名前の変更は無視されます。データファイルは、現在の名前のままです。
サイズ変更	失敗します。データファイルをオフラインにしてから、FLASHBACK DATABASE コマンドを再実行できます。データファイルは、フラッシュバックされません。データファイルを同じ時刻または SCN までリストアおよびリカバリできます。
オフラインにする	この操作は無視されます。データファイルは、現在のオンライン状態のままです。
オンラインにする	この操作は無視されます。データファイルは、現在のオフライン状態のままです。
読取り専用または読み書き両用にする	制御ファイル内のデータファイルの状態を変更します。

### フラッシュバック・ロギングが無効になっている表領域

ALTER TABLESPACE ...FLASHBACK OFF 文が、複数の表領域に対して実行されていることがあります。FLASHBACK DATABASE の実行時に、表領域をターゲット SCN に戻せるだけのフラッシュバック・データがない場合は、Recovery Manager によってエラーが発行され、データベースは変更されません。FLASHBACK DATABASE が失敗するかまたは中断された場合、データベースは必ずマウントされたままになります。

この使用例では、V\$TABLESPACE を問い合せて、フラッシュバック・ロギングが無効になっている表領域を特定します。次の選択肢があります。

- 該当する表領域にあるデータファイルをオフラインにします。その後、RESTORE を実行してから、RECOVER を実行し、それらのデータファイルを、データベースの残りデータファイルと同じ時点で調整します。
- ALTER DATABASE DATAFILE ...OFFLINE FOR DROP 文を使用して、該当するデータファイルを削除します。これで、RESETLOGS オプションでデータベースをオープンできます。データベースがオープン状態になったら、削除したデータファイルが含まれていた表領域に対して DROP TABLESPACE 文を実行します。

### フラッシュバック・データベース後のデータベースの状態

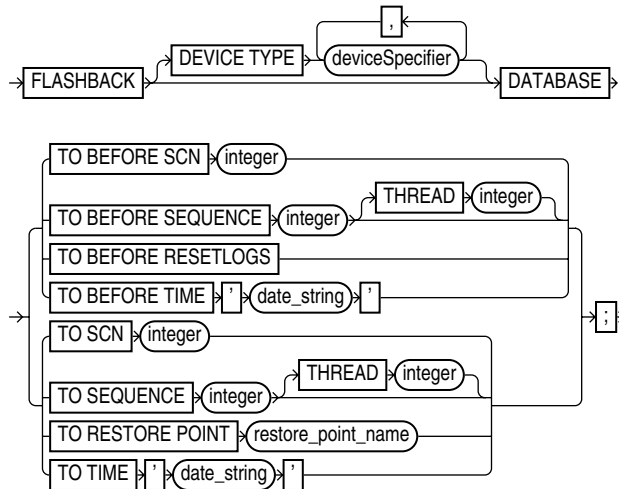
FLASHBACK DATABASE の実行後に、データベースが、ターゲット時刻の直前の SCN の状態ではない場合があります。データベースの SCN は、トランザクション以外のイベントによって更新される場合があります。FLASHBACK DATABASE TO のコマンド形式を使用する場合に、ターゲット SCN に対応するトランザクションが存在すると、フラッシュバック後のデータベースは、そのトランザクションまでのすべての変更がデータベースに含まれます。そうでない場合は、使用するコマンド形式が FLASHBACK DATABASE TO と FLASHBACK DATABASE TO BEFORE のどちらであるかには関係なく、そのトランザクションの直前までのすべての変更がデータファイルに含まれます。FLASHBACK DATABASE の処理として、指定したターゲット SCN より後の変更が適用されることはありません。

FLASHBACK DATABASE が終了したら、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、意図した結果が得られていることを確認してください。結果が期待どおりでなかった場合は、RECOVER DATABASE を実行して、フラッシュバック操作を開始した時点の状態にデータベースをリカバリすることができます。それから、FLASHBACK DATABASE を再度実行できます。

フラッシュバックの結果が期待どおりだった場合、OPEN RESETLOGS を実行して、ターゲット時刻より後の変更をすべて破棄できます。そのかわりに、失ったデータをデータ・ポンプでエクスポートし、RECOVER DATABASE を使用してデータベースをフラッシュ操作の前の状態に戻してから、失ったデータをデータ・ポンプで再インポートすることもできます。

## 構文

**flashback::=**



(deviceSpecifier::= (3-15 ページ))

## セマンティクス

構文の要素	説明
DEVICE TYPE <i>deviceSpecifier</i>	指定したデバイス・タイプ専用の自動チャンネルを割り当てます。たとえば、自動ディスクおよびテープ・チャンネルを構成して FLASHBACK...DEVICE TYPE DISK を発行すると、Recovery Manager ではディスク・チャンネルのみが割り当てられます。Recovery Manager では、フラッシュバック・データベース操作中のバックアップから REDO ログをリストアする必要があります。最新のフラッシュバック・ログとターゲット時刻の間の変更は、アーカイブ REDO ログに基づいて再作成される必要があります。自動チャンネルがテープに割り当てられていない場合に、必要な REDO ログがテープにあると、FLASHBACK DATABASE 操作は失敗します。 <b>関連項目</b> : 3-15 ページの「 <i>deviceSpecifier</i> 」を参照してください。
TO BEFORE SCN <i>integer</i>	指定した SCN の直前の状態へデータベースを戻します。指定した時点より以前の SCN までのすべての変更が適用されますが、指定した SCN に対応する変更が存在する場合、その変更は適用されません。デフォルトでは、指定された SCN で現在のインカネーションまたは祖先のインカネーションが解決されます。デフォルトは、RESET DATABASE INCARNATION コマンドを使用してオーバーライドできます。 フラッシュバックの可能な最も小さな SCN は、V\$FLASHBACK_DATABASE_LOG の OLDEST_FLASHBACK_SCN を問い合せて概略値を確認できます。
TO BEFORE SEQUENCE <i>integer</i> [THREAD <i>integer</i> ]	REDO ログ順序番号とスレッドを上限として指定します。Recovery Manager は、指定した順序番号およびスレッド番号のログの最後の変更（は含まない）までの変更を適用します。
TO BEFORE RESETLOGS	データベースの状態を、最後に OPEN RESETLOGS を実行した SCN の時点までのすべての変更を含む状態に戻します。 <b>注意</b> : FLASHBACK DATABASE を実行して、データベースを最後の OPEN RESETLOGS 操作より前の時点に戻すことができるのは、データベースを Oracle Database 10g リリース 2 以降にアップグレードしている場合のみです。

構文の要素	説明
TO BEFORE TIME 'date_string'	指定した時刻より前までのすべての変更を含む状態にデータベースを戻します (指定した時刻の変更は含まれません)。  フラッシュバックが可能な最も古い時刻は、V\$FLASHBACK_DATABASE_LOG の OLDEST_FLASHBACK_TIME を問い合せて概略値を確認できます。
TO SCN integer	指定した SCN の時点 (を含む) までデータベースを戻します。デフォルトでは、指定された SCN で現在のインカネーションまたは祖先のインカネーションが解決されます。Recovery Manager の RESET DATABASE コマンドを使用してリカバリ・ターゲット・インカネーションの設定を行うと、デフォルトをオーバーライドできます。  フラッシュバックの可能な最も小さな SCN は、V\$FLASHBACK_DATABASE_LOG の OLDEST_FLASHBACK_SCN を問い合せて概略値を確認できます。
TO SEQUENCE integer THREAD integer	REDO ログ順序番号とスレッドを上限として指定します。Recovery Manager は、指定した順序番号およびスレッド番号のログの最後の変更 (を含む) までの変更を適用します。
TO RESTORE POINT restore_point_name	指定したリストア・ポイントに対応する SCN までデータベースを戻します。通常のリストア・ポイントまたは保証付きリストア・ポイントを指定できます。
TO TIME 'date_string'	指定した時刻の状態へデータベースを戻します。現行の形式への時刻の変換には、SQL のすべての DATE 式を使用できます。たとえば、FLASHBACK DATABASE TO TIME 'SYSDATE-7' を使用できます。  フラッシュバックが可能な最も古い時刻は、V\$FLASHBACK_DATABASE_LOG の OLDEST_FLASHBACK_TIME を問い合せて概略値を確認できます。

## 例

### 例 2-79 特定の SCN への FLASHBACK DATABASE

2月14日の午後5時に、破損したデータベースを多くの表に挿入したとします。SQL\*Plus をデータベースに接続して、フラッシュバック・ウィンドウの最も古い SCN を問い合せます。

```
SQL> SELECT OLDEST_FLASHBACK_SCN, OLDEST_FLASHBACK_TIME
       2 FROM   V$FLASHBACK_DATABASE_LOG;
```

```
OLDEST_FLASHBACK_SCN OLDEST_FLASHBACK
-----
411010 2007/02/14 16:49
```

次に、新しい端末を開いて、Recovery Manager クライアントを起動し、ターゲット・データベースとリカバリ・カタログに接続します。次のように Recovery Manager コマンドを入力します (例には、FLASHBACK DATABASE の出力例が含まれます)。

```
RMAN> SHUTDOWN IMMEDIATE
RMAN> STARTUP MOUNT
RMAN> FLASHBACK DATABASE TO SCN 411010;
```

```
Starting flashback at 15-FEB-07
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=104 device type=DISK
```

```
starting media recovery
media recovery complete, elapsed time: 00:00:07
```

```
Finished flashback at 15-FEB-07
```

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

**例 2-80 リストア・ポイントへの FLASHBACK DATABASE**

データベースに大量の更新をロードする準備をしているとします。更新を実行する前に、保証付きリストア・ポイントを作成します。

```
SQL> CREATE RESTORE POINT before_update GUARANTEE FLASHBACK DATABASE;
```

バルク更新が失敗し、データベースに大量の破損データが残りました。Recovery Manager セッションを起動し、ターゲット・データベースとリカバリ・カタログに接続して、保証付きリストア・ポイントをリストします。

```
RMAN> LIST RESTORE POINT ALL;
```

SCN	RSP Time	Type	Time	Name
412742		GUARANTEED	15-FEB-07	BEFORE_UPDATE

データベースをマウントして、リストア・ポイントにフラッシュバックし（例には出力例も含まれます）、RESETLOGS オプションを使用してデータベースをオープンします。

```
RMAN> SHUTDOWN IMMEDIATE
```

```
RMAN> STARTUP MOUNT
```

```
RMAN> FLASHBACK DATABASE TO RESTORE POINT 'BEFORE_UPDATE';
```

```
Starting flashback at 15-FEB-07
```

```
allocated channel: ORA_DISK_1
```

```
channel ORA_DISK_1: SID=104 device type=DISK
```

```
starting media recovery
```

```
archived log for thread 1 with sequence 34 is already on disk as file
```

```
/disk2/oracle/oradata/prod/arch/archive1_34_614598462.dbf
```

```
media recovery complete, elapsed time: 00:00:01
```

```
Finished flashback at 15-FEB-07
```

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

## GRANT

### 用途

GRANT コマンドを使用すると、仮想プライベート・カタログ・スキーマに対する権限をデータベース・ユーザーに割り当てることができます。デフォルトでは、仮想カタログのユーザーは基本リカバリ・カタログにアクセスできません。

### 前提条件

このコマンドは、Recovery Manager プロンプトで実行してください。

CREATE CATALOG を使用して基本リカバリ・カタログが作成されていない場合は、GRANT で仮想プライベート・カタログの権限を割り当てることができません。

### 使用上の注意

**基本リカバリ・カタログ**を作成し、そこにすべてのデータベースのメタデータを格納することをお勧めします。その後、仮想プライベート・カタログ・スキーマを所有する Oracle Database ユーザーを作成できます。仮想プライベート・カタログのユーザーは、RECOVERY\_CATALOG\_OWNER ロールを付与されている必要があります。

Recovery Manager を基本リカバリ・カタログに接続し、GRANT コマンドを使用して、リカバリ・カタログ権限を仮想カタログの所有者に割り当てます。その後、CREATE VIRTUAL CATALOG を実行して、このユーザーの仮想カタログ・スキーマを作成します。REVOKE を使用すると、カタログ権限を取り消すことができます。

#### 同じデータベース上で CATALOG 権限を持つユーザー間の関係

ここで、GRANT の使用方法を説明します。データベース prod1 および prod2 が、基本リカバリ・カタログに登録されているとします。基本リカバリ・カタログに SYS としてログインしている状態で、仮想プライベート・カタログの 2 人のユーザー (vpc1 および vpc2) を作成します。この両方のユーザーに、データベース prod1 に対する CATALOG FOR DATABASE アクセス権を付与します (prod2 については付与しません)。

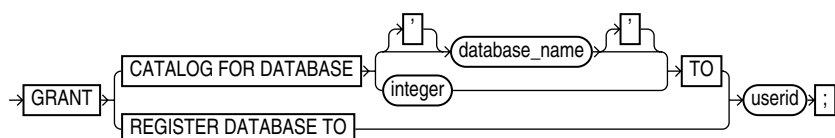
この例では、基本リカバリ・カタログの所有者によって作成された prod1 のバックアップのメタデータには、vpc1 と vpc2 の両方がアクセスできます。相互に作成した prod1 のバックアップのメタデータには、両方のユーザーがアクセスできます。データベース prod2 のバックアップ・メタデータには、vpc1 と vpc2 の両方ともアクセスできません。

#### GRANT REGISTER と GRANT CATALOG 間の関係

REGISTER DATABASE をユーザーに付与すると、このユーザーが登録したすべてのデータベースに対して暗黙的にリカバリの CATALOG FOR DATABASE 権限が付与されます。あるユーザー (virtcat など) の REGISTER DATABASE 権限のみに対して REVOKE を実行した場合、virtcat が登録したデータベース (prod など) の CATALOG FOR DATABASE 権限が暗黙的に取り消されることはありません。CATALOG FOR DATABASE 権限には prod の登録権限が含まれているため、virtcat は、prod の登録解除と登録を継続できます。virtcat が prod に対する操作 (prod の再登録を含む) を実行できないようにするには、virtcat から REVOKE ALL PRIVILEGES を実行します。

### 構文

**grant::=**



## セマンティクス

構文の要素	説明
CATALOG FOR DATABASE [database_name   integer] TO userid	<p>指定したデータベースのリカバリ・カタログへのアクセス権を指定したユーザーに付与します。</p> <p><b>注意:</b> 指定したデータベースで付与されるカタログ操作には、そのデータベースの登録と登録解除が含まれます。</p> <p>データベースは、データベース名または DBID で指定します。名前で指定した場合、その名前のデータベースがカタログに複数登録されていると、Recovery Manager によりエラーが戻されます。その場合は、DBID でデータベースを指定してください。</p> <p>リカバリ・カタログに登録済のデータベースへのアクセス権を付与するには、GRANT CATALOG コマンドを使用する必要があります。また、カタログに未登録のターゲット・データベースに対するアクセス権も付与できます。こうして、仮想プライベート・カタログのユーザーがデータベースを登録することが可能になります。アクセス権を付与する場合は、未登録のデータベースの DBID を使用する必要があります。</p>
REGISTER DATABASE TO userid	<p>指定したユーザーに、リカバリ・カタログで現在認識されていないデータベースを REGISTER DATABASE で登録する権限を付与します。</p> <p>REGISTER DATABASE をユーザーに付与すると、このユーザーが登録したすべてのデータベースに対して暗黙的にリカバリの CATALOG FOR DATABASE 権限が付与されます。データベースの CATALOG FOR DATABASE 権限には、そのデータベースの登録と登録解除が含まれます。</p> <p>たとえば、ユーザー virtcat に REGISTER DATABASE が付与され、このユーザーがカタログにデータベース prod を登録するとします。Recovery Manager は、暗黙的に prod に対するリカバリの CATALOG FOR DATABASE 権限を virtcat に付与します。</p>

## 例

### 例 2-81 仮想プライベート・カタログの権限の付与

データベース・ユーザー rco がデータベース catdb に基本リカバリ・カタログを所有しているとします。この基本リカバリ・カタログには、データ・センター内の数多くのデータベースの Recovery Manager メタデータが格納されています。目標は、データ・センターの 2 人のバックアップ・オペレータのために、仮想プライベート・カタログを作成することです。

SQL\*Plus を起動し、SYS として catdb データベースに接続します。次に、CREATE USER 文を使用して、catdb 上に bckop2 および bckop3 ユーザーを作成します。次のように、リカバリ・カタログの所有権をこれらのユーザーに付与できます。

```
SQL> GRANT recovery_catalog_owner TO bckop2, bckop3;
SQL> EXIT
```

次に、Recovery Manager クライアントを起動し、ユーザー rco としてリカバリ・カタログ・データベースに接続します。Recovery Manager の GRANT コマンドを使用して、bckop2 に、自分の仮想プライベート・カタログにすべてのデータベースを登録する権限を与えます。ただし、bckop3 には、データ・センター内のデータベースのサブセットのみへのアクセス権を与えます。

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

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

```
RMAN> GRANT REGISTER DATABASE TO bckop2;
RMAN> GRANT CATALOG FOR DATABASE prod TO bckop3;
RMAN> GRANT CATALOG FOR DATABASE prodb TO bckop3;
RMAN> EXIT;
```

新しい Recovery Manager セッションを開始し、bckop2 の仮想カタログを作成します（例には、CREATE VIRTUAL CATALOG の出力例も含まれます）。仮想カタログを作成するたびに、Recovery Manager を終了して再起動する必要があります。

```
RMAN> CONNECT CATALOG bckop2@catdb
```

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

```
RMAN> CREATE VIRTUAL CATALOG;
```

```
found eligible base catalog owned by RCO  
created virtual catalog against base catalog owned by RCO
```

```
RMAN> EXIT;
```

新しい Recovery Manager セッションを開始し、bckop3 の仮想カタログを作成します（例には、CREATE VIRTUAL CATALOG の出力例も含まれます）。

```
RMAN> CONNECT CATALOG bckop3@catdb
```

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

```
RMAN> CREATE VIRTUAL CATALOG;
```

```
found eligible base catalog owned by RCO  
created virtual catalog against base catalog owned by RCO
```

```
RMAN> EXIT;
```

次の例では、バックアップ・オペレータ dba1 が、catdb の bckop3 スキーマにある自分の仮想プライベート・カタログを使用して、ターゲット・データベースのバックアップのメタデータを格納します。

```
RMAN> CONNECT TARGET /  
RMAN> CONNECT CATALOG bckop3@catdb
```

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

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

## HOST

### 用途

HOST コマンドを使用すると、**Recovery Manager** 内からオペレーティング・システムのコマンドラインのサブ・シェルを呼び出すことができます。

### 構文

**host:=**



### 前提条件

このコマンドは、**RUN** コマンドのカッコ内または **Recovery Manager** プロンプトで実行してください。

### セマンティクス

構文の要素	説明
HOST	コマンド・プロンプトを表示し、ユーザーがサブシェルを終了すると、 <b>Recovery Manager</b> を再開します (2-156 ページの例 2-82 を参照)。
'command'	指定文字列内のコマンドを実行し、 <b>Recovery Manager</b> を継続します (2-157 ページの例 2-83 を参照)。

### 例

#### 例 2-82 バックアップ内でのオペレーティング・システムへの切替え

この例では、datafile 3 のイメージ・コピーを作成し、Linux のプロンプトに切り替えて、コピーがディレクトリにあるかどうかをチェックしてから、**Recovery Manager** セッションを再開します (Linux セッションの出力は太字のインデント付きで表示されています)。

```
RMAN> BACKUP DATAFILE 3 FORMAT '/disk2/df3.cpy';
```

```
Starting backup at 15-FEB-07
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00003 name=/disk1/oracle/oradata/prod/undotbs01.d bf
channel ORA_DISK_1: starting piece 1 at 15-FEB-07
channel ORA_DISK_1: finished piece 1 at 15-FEB-07
piece handle=/disk2/df3.cpy tag=TAG20070215T111326 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 15-FEB-07
```

```
RMAN> HOST;
```

```
% ls /disk2/df3.cpy
/disk2/df3.cpy
% exit
exit
host command complete
```

```
RMAN>
```



**例 2-83 Recovery Manager 内でのオペレーティング・システム・コピーの実行**

この例では、データファイル system01.dbf のバックアップを作成してから、Linux の ls コマンドを実行して、/disk2 ディレクトリのすべてのファイルを表示します。

```
BACKUP DATAFILE '?/oradata/prod/system01.dbf'  
    FORMAT '/disk2/system01.dbf';  
HOST 'ls -lt /disk2/*';
```

## IMPORT CATALOG

### 用途

IMPORT CATALOG コマンドを使用すると、リカバリ・カタログ・スキーマにあるメタデータを別のカタログ・スキーマにインポートできます。様々なバージョンのカタログ・スキーマを作成し、複数のターゲット・データベースのメタデータを保存する場合は、このコマンドを使用すると、すべてのデータベース用に1つのカタログ・スキーマを保持することができます。

**関連項目:** [CREATE CATALOG](#) (2-103 ページ)

### 前提条件

Recovery Manager が、インポート先のリカバリ・カタログ (カタログ・データをインポートする先のカタログ) に接続されている必要があります。このリカバリ・カタログは、仮想プライベート・カタログにはできません。

ターゲット・データベースに接続していなくても、カタログ・スキーマはマージできます。このコマンドは、[RUN](#) コマンドのカッコ内または Recovery Manager プロンプトで実行してください。

ソース・リカバリ・カタログ・スキーマのバージョンは、使用する Recovery Manager 実行可能ファイルのバージョンと一致している必要があります。ソース・スキーマのバージョンが低い場合は、カタログ・スキーマを現行バージョンにアップグレードします。ソース・スキーマのバージョンが高い場合は、バージョンの高い Recovery Manager 実行可能ファイルで、再度操作を行ってください。

同じ名前のデータベースが、ソース・リカバリ・カタログ・スキーマとインポート先カタログ・スキーマの両方に登録されていないことを確認してください。両方のスキーマに登録されているデータベースがある場合は、[UNREGISTER](#) を実行して、そのデータベースのソース・リカバリ・カタログでの登録を解除し、IMPORT コマンドを再実行します。

### 使用上の注意

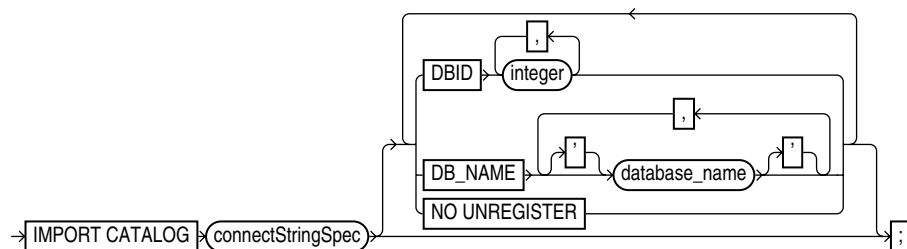
インポート操作が途中で失敗すると、インポートはロールバックされます。このため、一部のみのインポートすることはできません。登録解除の操作は、インポートとは別に行われます。デフォルトでは、インポートが正常に行われると、インポート済のデータベース ID が、ソース・リカバリ・カタログ・スキーマから登録解除されます。

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。スクリプトのインポート時に、インポート先のスキーマにすでにスクリプト名が含まれていることが原因で、グローバル・スクリプトの名前が競合することがあります (ローカル・スクリプトの場合は競合しません)。その場合は、Recovery Manager により、グローバル・スクリプト名が COPY OF *script\_name* に変更されます。たとえば、bp\_cmd という名前は COPY OF bp\_cmd に変更されます。

変更後のグローバル・スクリプト名がそれでも一意でない場合は、COPY (2) OF *script\_name* という名前に変更されます。そのスクリプト名も存在した場合は、スクリプト名が COPY (3) OF *script\_name* に変更されます。こうして、スクリプト名が一意になるまで、COPY (n) OF というパターンが継続されます。

## 構文

**import::=**



(connectStringSpec::= (3-12 ページ))

## セマンティクス

構文の要素	説明
<code>connectStringSpec</code>	ソース・リカバリ・カタログの接続文字列を指定します（このカタログは、メタデータがインポートされるカタログです）。
DBID <code>integer</code>	ソース・カタログ・スキーマからメタデータをインポートするデータベースの DBID のリストを指定します（例 2-85 を参照）。  指定しなかった場合は、Recovery Manager により、ソース・カタログ・スキーマのすべてのデータベース ID のメタデータがマージされ、インポート先のカタログ・スキーマに入れられます。メタデータがマージされたデータベースが、リカバリ・カタログ・スキーマにすでに登録されていた場合は、エラーが発行されます。
DB_NAME <code>database_name</code>	ソース・カタログ・スキーマからメタデータをインポートするデータベースのリストを指定します（例 2-85 を参照）。  指定しなかった場合は、Recovery Manager により、ソース・カタログ・スキーマのすべてのデータベースのメタデータがマージされ、インポート先のカタログ・スキーマに入れられます。同じ DBID が両方のリカバリ・カタログに登録されていた場合は、エラーが発行されます。
NO UNREGISTER	インポート済のデータベース ID を、ソース・カタログ・スキーマに強制的に保持します。デフォルトでは、インポート済のデータベース ID はソース・リカバリ・カタログ・スキーマから登録解除されます。

## 例

### 例 2-84 すべての登録済データベースのメタデータのインポート

この例では、ユーザー rcat が所有する 10.2 のカタログ・スキーマがデータベース inst1 に含まれ、ユーザー rco が所有する 11.1 のカタログ・スキーマがデータベース catdb に含まれています。Recovery Manager は、rcat に登録されているすべてのデータベース ID のメタデータを、rco が所有するリカバリ・カタログにインポートします。rcat に登録されたすべてのターゲット・データベースは登録解除されます。

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

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

```
RMAN> IMPORT CATALOG rcat@prod;
```

```
Starting import catalog at 15-FEB-07
source recovery catalog database Password: password
connected to source recovery catalog database
```

```
import validation complete
database unregistered from the source recovery catalog
Finished import catalog at 15-FEB-07
```

### 例 2-85 登録済データベースのサブセットのメタデータのインポート

この例は、例 2-84 を部分的に変更したものです。リカバリ・カタログ全体をインポートするかわりに、DBID が 1618984270 のデータベースのメタデータのみをインポートします。

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

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

```
RMAN> IMPORT CATALOG rcat@inst1 DBID=1618984270;
```

```
Starting import catalog at 15-FEB-07
source recovery catalog database Password: password
connected to source recovery catalog database
import validation complete
database unregistered from the source recovery catalog
Finished import catalog at 15-FEB-07
```

---

## LIST

### 用途

LIST コマンドを使用すると、**Recovery Manager** リポジトリに記録されているバックアップおよびその他のオブジェクトに関する情報を表示できます。

**関連項目：** リストとレポートの作成方法は、『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』および 2-244 ページの「**RMAN**」を参照してください。

### 追加トピック

- [前提条件](#)
- [使用上の注意](#)
- [構文](#)
- [セマンティクス](#)
- [LIST コマンドの出力](#)
- [例](#)

### 前提条件

LIST は、**Recovery Manager** プロンプトでのみ実行できます。次のいずれかの条件が満たされている必要があります。

- **Recovery Manager** がターゲット・データベースに接続されている必要があります。**Recovery Manager** がリカバリ・カタログに接続されておらず、LIST FAILURE コマンドを実行していない場合は、ターゲット・データベースがマウントまたはオープン状態である必要があります。**Recovery Manager** がリカバリ・カタログに接続されていない場合は、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。
- **Recovery Manager** がリカバリ・カタログに接続され、**SET DBID** が実行されている必要があります。

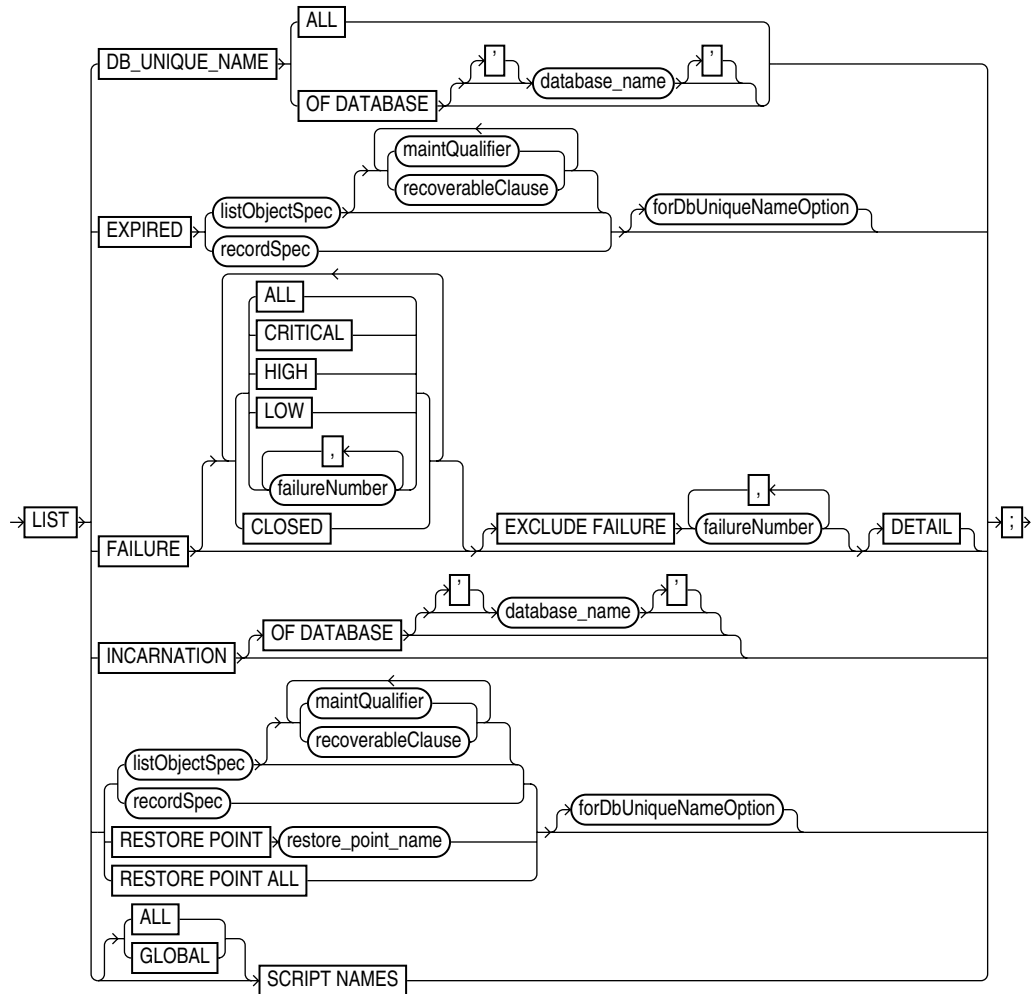
### 使用上の注意

LIST FAILURE を除く LIST コマンドは、**CROSSCHECK** および **DELETE** コマンドの実行対象のバックアップおよびコピーを表示します。LIST FAILURE コマンドは、**ADVISE FAILURE** および **REPAIR FAILURE** コマンドの実行対象となる障害を表示します。

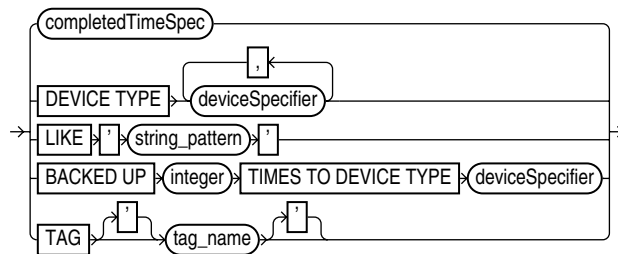
2-21 ページの「[Data Guard 環境での Recovery Manager のバックアップ](#)」では、Data Guard 環境での **Recovery Manager** のバックアップ処理方法について説明しています。通常、**Recovery Manager** では、環境内の 1 つのデータベース上で作成されたテープ・バックアップはその環境のすべてのデータベースにアクセス可能であるとみなされますが、ディスク・バックアップは作成元のデータベースのみにアクセス可能であるとみなされます。Data Guard 環境では、LIST を実行すると、接続されているターゲット・データベースにアクセス可能なファイルが表示されます。

LIST の出力は、標準出力またはメッセージ・ログのいずれかに対して行われます。同時に両方に出力されることはありません。

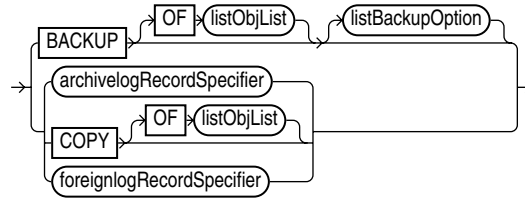
## 構文

**list::=**

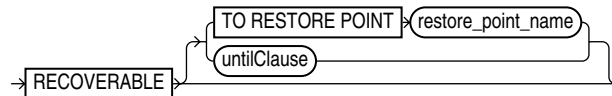
(listObjectSpec::= (2-163 ページ)、recordSpec::= (3-35 ページ)、  
 maintQualifier::= (3-30 ページ)、forDbUniqueNameOption::= (3-18 ページ)、  
 untilClause::= (3-38 ページ))

**maintQualifier::=**

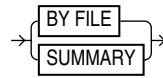
(completedTimeSpec::= (3-10 ページ)、deviceSpecifier::= (3-15 ページ))

**listObjectSpec::=**

(listObjList::= (3-27 ページ)、listBackupOption::= (2-163 ページ)、  
archivelogRecordSpecifier::= (3-6 ページ)、foreignlogRecordSpecifier::= (3-19 ページ))

**recoverableClause::=**

(untilClause::= (3-38 ページ))

**listBackupOption::=**

## セマンティクス

**list**

構文の要素	説明
DB_UNIQUE_NAME	リカバリ・カタログに登録されている 1 つ以上のデータベースの DB_UNIQUE_NAME をリストします。  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。また、Recovery Manager が、マウントされているかオープンされているターゲット・データベースに接続しているか、または SET DBID コマンドでターゲット・データベースを指定する必要もあります。プライマリ・データベースの DBID と、それに関連付けられているスタンバイ・データベースの DBID は同じであることに注意してください。これらのデータベースは、DB_UNIQUE_NAME によって区別されます。 <b>関連項目</b> : 出力の説明については、表 2-27 を参照してください。
ALL	Recovery Manager リポジトリに登録されているすべてのデータベースの DB_UNIQUE_NAME をリストします。
OF DATABASE database_name	指定された DB_NAME のすべてのデータベースの DB_UNIQUE_NAME をリストします。
EXPIRED	リポジトリ内で EXPIRED (見つからなかったことを示します) とマークされているバックアップ・セット、プロキシ・コピーおよびイメージ・コピーを表示します。出力の説明については、表 2-8 「バックアップ・セットのリスト (データファイルのバックアップ・セットの場合)」を参照してください。  LIST EXPIRED に最新出力が表示されるようにするには、CROSSCHECK コマンドを定期的に行ってください。CROSSCHECK コマンドを使用すると、ディスクおよびテープで、リポジトリに記録されているバックアップおよびコピーが検索されます。見つからない場合は、リポジトリ・レコードが状態 EXPIRED に更新されます。
listObjectSpec	リストする期限切れオブジェクトのタイプを指定します。 <b>関連項目</b> : 2-166 ページの「listObjectSpec」を参照してください。

構文の要素	説明
<code>maintQualifier</code>	リストの範囲を制限します。 <b>関連項目</b> : 3-30 ページの「 <code>maintQualifier</code> 」を参照してください。
<code>recordSpec</code>	リストする期限切れオブジェクトを指定します。 <b>関連項目</b> : 3-35 ページの「 <code>recordSpec</code> 」を参照してください。
<code>forDbUniqueNameOption</code>	Data Guard 環境の指定した DB_UNIQUE_NAME にのみ関連付けられた、 <code>listObjectSpec</code> または <code>recordSpec</code> の期限切れファイルをリストします。 データベースは <code>db_unique_name</code> で指定できますが、ALL を使用すると、特定の DBID についてカタログに記録されている一意の名前のデータベースをすべて指定できます。リカバリ・カタログ内のデータベースは、DBID および DB_UNIQUE_NAME 初期化パラメータの値によって一意に識別されます。 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。また、Recovery Manager は、マウント済またはオープン状態のターゲット・データベースに接続しているか、 <code>SET DBID</code> コマンドを実行している必要があります。 <b>関連項目</b> : この句のオプションについては、3-18 ページの「 <code>forDbUniqueNameOption</code> 」を参照してください。
FAILURE	データ・リカバリ・アドバイザによって記録された障害をリストします。Recovery Manager が接続しているデータベースは、単一インスタンス・データベースである必要があります。また、フィジカル・スタンバイ・データベース以外である必要があります。 <b>関連項目</b> : 2-7 ページの「Oracle RAC およびデータ・リカバリ・アドバイザ」を参照してください。 データ・リカバリ・アドバイザは、データの損失や破損の原因となる幅広い物理的な問題を検出して修復できます。物理的な破損は、通常、I/O サブシステムのエラーや人為的エラーが原因で発生します。データ・リカバリ・アドバイザは、一部のタイプの論理的な破損については、検出して処理することができません。このタイプの破損については、Oracle サポート・サービスに問い合わせる必要があります。 データ・リカバリ・アドバイザでいう <b>障害</b> とは、一連の修復アクションに対応付けられるデータの永続的な破損のことです。データ障害は、各種チェック（データベースまたはその構成要素の状態を評価する診断手順）で検出されます。それぞれのチェックで、1つ以上の障害が診断され、その障害は一連の修復に対応付けられます。 一般的な例としては、LIST FAILURE を実行して障害をリストしてから、ADVISE FAILURE を使用して修復オプションを表示し、REPAIR FAILURE を実行して障害を修復します。これらのコマンドを同じ Recovery Manager セッションで実行します。 LIST FAILURE でオプションを指定しない場合は、ステータスが OPEN で、優先順位が最も高い障害のみがリストされます。したがって、CRITICAL と HIGH の障害が存在する場合は、コマンド出力に必ずリストされます。優先順位が LOW の障害は、優先順位が CRITICAL の障害と HIGH の障害が両方とも存在しない場合にのみリストされます。障害は発生した逆順で保存されます（最新の障害が最初にリストされます）。 LIST FAILURE コマンドを実行しても、新しい障害を診断するチェックは開始されません。このコマンドは、それまでに実行済の評価結果をリストします。したがって、LIST FAILURE を繰り返して実行した場合は、あるコマンドと別のコマンドを実行する間に発生したエラーに対して、データベースで自動的に障害が診断されたときにのみ、新しい障害が表示されます。ただし、LIST FAILURE は、その実行時に既存のすべての障害を再評価します。ユーザーが障害を手動で修正した場合や、障害が一過性のもので存在しなくなっていた場合は、その障害が LIST FAILURE の出力から除かれます。 <b>関連項目</b> : LIST FAILURE の出力表の列ヘッダーの説明は、表 2-26 を参照してください。実例については、例 2-91 を参照してください。
ALL	ステータスが OPEN の障害を、優先順位とは無関係にすべてリストします。
CRITICAL	ステータスが OPEN で CRITICAL な障害のみをリストします。



構文の要素	説明
HIGH	優先順位が HIGH でステータスが OPEN の障害のみをリストします。
LOW	優先順位が LOW でステータスが OPEN の障害のみをリストします。
<i>failureNumber</i>	障害を障害番号で指定します。
CLOSED	クローズされている障害のみをリストします。
EXCLUDE FAILURE	指定された障害をリストから除外します。
<i>failureNumber</i>	
DETAIL	まとめられた障害を展開してリストします。たとえば、あるファイル内の複数箇所でブロックが破損している場合に DETAIL オプションを指定すると、それぞれの破損ブロックがリストされます。
INCARNATION	データベースのインカネーションに関する情報を表示します。  RESETLOGS オプションでデータベースをオープンするたびに、データベースの新規インカネーションが作成されます。LIST INCARNATION を実行したときにデータベースのインカネーションが <i>n</i> 個表示された場合は、そのデータベースのオンライン REDO ログが <i>n</i> -1 回リセットされています。  LIST 出力には、指定したデータベース名に該当するすべてのデータベース・インカネーション・レコードの主キーが（インカネーション・キーが含まれる Inc Key 列に）含まれます。Recovery Manager が現行とみなしているインカネーションを前のインカネーションに変更するには、RESET DATABASE コマンドでこのキーを使用します。  <b>関連項目</b> ：LIST INCARNATION の出力表の列ヘッダーの説明は、表 2-23 を参照してください。実例については、例 2-90 を参照してください。
OF DATABASE	データベースの名前を指定します。OF DATABASE オプションを指定しない場合、LIST 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を表示します。
<i>database_name</i>	
<i>listObjectSpec</i>	リストする期限切れオブジェクトのタイプを指定します。  <b>関連項目</b> ：2-166 ページの「 <i>listObjectSpec</i> 」を参照してください。
<i>maintQualifier</i>	リストの範囲を制限します。  <b>関連項目</b> ：3-30 ページの「 <i>maintQualifier</i> 」を参照してください。
<i>recoverableClause</i>	表示するリストを、リポジトリ内のステータスが AVAILABLE で、ターゲット・データベースの現行のインカネーションでリストアとリカバリに使用できるデータファイル・バックアップまたはコピーに限定します。  <b>関連項目</b> ：2-167 ページの「 <i>recoverableClause</i> 」を参照してください。
<i>recordSpec</i>	リストするオブジェクトを指定します。  <b>関連項目</b> ：3-35 ページの「 <i>recordSpec</i> 」を参照してください。
<i>untilClause</i>	終了時刻、SCN またはログ順序番号を指定します。  <b>関連項目</b> ：3-38 ページの「 <i>untilClause</i> 」を参照してください。
RESTORE POINT	Recovery Manager リポジトリで認識されているリストア・ポイントを表示します。  <b>関連項目</b> ：LIST RESTORE POINT の出力表の列ヘッダーの説明は、表 2-25 を参照してください。
<i>restore_point_name</i>	指定されたリストア・ポイントを表示します。
ALL	Recovery Manager リポジトリで認識されているすべてのリストア・ポイントを表示します。

構文の要素	説明
<code>forDbUniqueNameOption</code>	<p>Data Guard 環境の指定した DB_UNIQUE_NAME にのみ関連付けられたバックアップおよびリストア・ポイントをリストします。</p> <p>データベースは <code>db_unique_name</code> で指定できますが、ALL を使用すると、特定の DBID についてカタログに記録されている一意の名前のデータベースをすべて指定できます。リカバリ・カタログ内のデータベースは、DBID および DB_UNIQUE_NAME 初期化パラメータの値によって一意に識別されます。</p> <p>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。また、マウントされているかオープンされているターゲット・データベースに接続していることも必要です。</p> <p><b>関連項目:</b> この句のオプションについては、3-18 ページの「<code>forDbUniqueNameOption</code>」を参照してください。</p>
ALL SCRIPT NAMES	<p>接続されているリカバリ・カタログのすべてのデータベースに定義されているすべてのグローバル・スクリプトおよびローカル・スクリプトが、コメントとともにリストされます。</p> <p>リカバリ・カタログに接続している必要がありますが、ターゲット・データベースに接続している必要はありません。</p>
GLOBAL SCRIPT NAMES	<p>接続されているリカバリ・カタログで定義されているグローバル・スクリプトのみが、コメントとともにリストされます。</p> <p>リカバリ・カタログに接続している必要がありますが、ターゲット・データベースに接続している必要はありません。</p>
SCRIPT NAMES	<p>現在のターゲット・データベースで実行できるローカル・スクリプトおよびグローバル・スクリプトがリストされます。</p> <p>この形式でコマンドを使用するには、ターゲット・データベースとリカバリ・カタログに接続している必要があります。</p> <p><b>関連項目:</b> 出力の説明は、表 2-24 を参照してください。実例については、例 2-107 を参照してください。</p>

### *listObjectSpec*

この副次句は、リストするオブジェクトのタイプを指定します。

構文の要素	説明
BACKUP	<p>バックアップ・セット（バックアップ・ピースの詳細も含む）およびプロキシ・コピーに関する情報を表示します。</p> <p><b>関連項目:</b> LIST BACKUP 出力の説明は、表 2-8 を参照してください。実例については、例 2-86 を参照してください。</p>
OF <i>listObjList</i>	<p>操作するオブジェクトのリストを <i>listObjList</i> 句で指定したオブジェクト型に限定します。オブジェクトを指定しなければ、LIST ではデフォルトで OF DATABASE CONTROLFILE ARCHIVELOG ALL が使用されます。</p> <p><b>注意:</b> LIST BACKUP ... LIKE コマンドは有効ではありません。ただし、LIST BACKUP OF ARCHIVELOG LIKE のみは有効です。</p> <p><b>関連項目:</b> 3-27 ページの「<i>listObjList</i>」を参照してください。</p>
<i>listBackupOption</i>	<p>バックアップのサマリーまたは詳細情報をリストするかどうかを指定します。</p>
<i>archivelogRecordSpecifier</i>	<p>アーカイブ REDO ログの範囲情報を表示します。</p>
COPY	<p>データファイルのコピー、アーカイブ REDO ログおよびアーカイブ REDO ログのイメージ・コピーに関する情報のみを表示します。デフォルトでは、LIST COPY によりすべてのデータベース・ファイルとアーカイブ REDO ログのコピーが表示されます。使用可能および使用不可のイメージ・コピーのみでなく、リストアできないコピー、期限切れのコピーまたは使用不可能なコピーも出力に含まれます。</p> <p><b>関連項目:</b> LIST COPY の出力表の列ヘッダーの説明は、2-175 ページの表 2-20 および 2-176 ページの表 2-22 を参照してください。</p>

構文の要素	説明
OF <i>listObjList</i>	操作するオブジェクトのリストを <i>listObjList</i> 句で指定したオブジェクト型に限定します。 <b>関連項目</b> : 3-27 ページの「 <i>listObjList</i> 」を参照してください。
<i>foreignlogRecordSpecifier</i>	外部アーカイブ REDO ログの範囲情報を表示します。

### *recoverableClause*

この副次句は、リカバリ可能なバックアップを指定します。

構文の要素	説明
RECOVERABLE	表示するリストを、リポジトリ内のステータスが AVAILABLE で、ターゲット・データベースの現行のインカネーションでリストアとリカバリに使用できる、期限切れのデータファイル・バックアップまたはコピーに限定します。このリストには、増分を適用できる有効な親を持たない増分バックアップ以外のすべてのバックアップおよびコピーが含まれます。
TO RESTORE POINT <i>restore_point_name</i>	リストア・ポイントを作成した時点の SCN を上限として、リストア・ポイントを指定します。指定した値は含まれます。上限値が含まれるため、 <b>Recovery Manager</b> は、リストア・ポイントに対応する SCN までリカバリできるファイルのみをリストします。
<i>untilClause</i>	終了時刻、SCN またはログ順序番号を指定します。 <b>関連項目</b> : 3-38 ページの「 <i>untilClause</i> 」を参照してください。

### *listBackupOption*

バックアップのサマリーを出力するか、特定のデータファイルのバックアップをリストするかを指定します。

構文の要素	説明
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リストします。 <b>関連項目</b> : LIST BACKUP ... BY FILE の出力については、表 2-17、表 2-18 または表 2-19 を参照してください。
SUMMARY	バックアップごとに 1 行のサマリーを表示します。 <b>関連項目</b> : LIST BACKUP ... SUMMARY の出力については、表 2-15 を参照してください。出力例については、例 2-87 を参照してください。

## LIST コマンドの出力

出力で表示される情報については、次の表を参照してください。

- 表 2-8 「バックアップ・セットのリスト (データファイルのバックアップ・セットの場合)」
- 表 2-9 「バックアップ・ピースのリスト (ピースが 1 つのみのセットの場合)」
- 表 2-10 「バックアップ・セット内のデータファイルのリスト」
- 表 2-11 「バックアップ・セット内のアーカイブ・ログのリスト」
- 表 2-12 「バックアップ・セット ... のバックアップ・セット・コピー ... (複数ピースの場合のみ)」
- 表 2-13 「バックアップ・セット ... のコピー ... のバックアップ・ピースのリスト (複数ピースの場合)」

- 表 2-14 「プロキシ・コピーのリスト」
- 表 2-15 「バックアップ・セットのリスト (LIST BACKUP ... SUMMARY)」
- 表 2-16 「バックアップ・ピースのリスト (LIST BACKUPPIECE ...)」
- 表 2-17 「データファイル・バックアップのリスト (LIST BACKUP ... BY FILE)」
- 表 2-18 「アーカイブ・ログ・バックアップのリスト (LIST BACKUP ... BY FILE)」
- 表 2-19 「制御ファイル・バックアップのリスト (LIST BACKUP ... BY FILE)」
- 表 2-20 「データファイルのコピーのリスト」
- 表 2-21 「制御ファイルのコピーのリスト」
- 表 2-22 「アーカイブ・ログ・コピーのリスト」
- 表 2-23 「データベースのインカネーションのリスト」
- 表 2-24 「リカバリ・カタログのストアド・スクリプトのリスト (LIST SCRIPT NAMES)」
- 表 2-25 「リストア・ポイントのリスト (LIST RESTORE POINT)」
- 表 2-26 「障害のリスト」
- 表 2-27 「データベースのリスト (LIST DB\_UNIQUE\_NAME)」

**表 2-8 バックアップ・セットのリスト (データファイルのバックアップ・セットの場合)**

列	指定対象
BS Key	このバックアップ・セットを識別する一意のキー。 Recovery Manager がリカバリ・カタログに接続している場合、BS Key はカタログ内のバックアップ・セットの主キーです。これは、RC_BACKUP_SET ビューの BS_KEY に対応します。Recovery Manager がデフォルトの NOCATALOG モードで接続している場合、BS Key では V\$BACKUP_SET からの RECID が表示されます。
Type	バックアップのタイプ Full または Incr (増分)。 <b>注意:</b> データファイルのバックアップ・セットにのみ含まれる列。
LV	バックアップのレベル。非増分の場合は NULL、増分の場合はレベル 0 またはレベル 1。 <b>注意:</b> データファイルのバックアップ・セットにのみ含まれる列。
Size	バックアップのサイズ (バイト)。 <b>注意:</b> データファイルのバックアップ・セットにのみ含まれる列。
Device Type	バックアップが行われたデバイスのタイプ (DISK または sbt など)。
Elapsed Time	バックアップの期間。
Completion Time	バックアップ・セットをとった日付と時刻。このフィールドの書式は、NLS_LANG および NLS_DATE_FORMAT の環境設定によって変わります。
List of Datafiles in backup set ...	表 2-10 を参照してください。

表 2-9 バックアップ・ピースのリスト（ピースが1つのみのセットの場合）

列	指定対象
BP Key	リカバリ・カタログまたはターゲット・データベース制御ファイルにあるバックアップ・ピースの一意の識別子。  Recovery Manager がリカバリ・カタログに接続している場合、BS Key はカタログ内のバックアップ・ピースの主キーです。これは、RC_BACKUP_PIECE ビューの BP_KEY に対応します。Recovery Manager が NOCATALOG モードで接続している場合、BP Key では V\$BACKUP_PIECE からの RECID が表示されます。  <b>注意：</b> リカバリ・カタログおよび制御ファイルにある KEY の値は異なります。
Status	バックアップ・ピースの状態。AVAILABLE、UNAVAILABLE または EXPIRED（各状態の説明は、CHANGE、CROSSCHECK および DELETE コマンドを参照）。
Compressed	バックアップ・ピースが圧縮されているかどうか（YES または NO）。
Tag	バックアップ・セットに適用されるタグ。タグ名には大 / 小文字区別がなく、すべて大文字で表示されることに注意してください。
Piece Name/Handle	バックアップ・ピースのファイル名またはハンドル。バックアップ・ピースが SBT にある場合は、メディア ID が名前とともに表示されます。
SPFILE Included	サーバー・パラメータ・ファイルがバックアップに含まれます。
Control File Included	制御ファイルがバックアップに含まれます。  <b>注意：</b> この行は、現行の制御ファイルがバックアップに含まれる場合にのみ表示されます。
Ckp SCN	バックアップ制御ファイルのチェックポイント SCN。指定した SCN より前に REDO レコードに記録されたデータベース変更はすべて、制御ファイルに反映されます。  <b>注意：</b> この行は、現行の制御ファイルがバックアップに含まれる場合にのみ表示されます。
Ckp time	バックアップ制御ファイルのチェックポイントの時刻。指定した時刻より前に REDO レコードに記録されたデータベース変更はすべて、制御ファイルに反映されます。  <b>注意：</b> この行は、現行の制御ファイルがバックアップに含まれる場合にのみ表示されます。

表 2-10 バックアップ・セット内のデータファイルのリスト

列	指定対象
File	バックアップされたファイルの数。
LV	バックアップのレベル。非増分の場合は NULL、増分の場合はレベル 0 またはレベル 1。
Type	バックアップのタイプ Full または Incr（増分）。
Ckp SCN	データファイルをバックアップした時点のデータファイルのチェックポイント。この SCN より前のデータベース変更はすべて、ファイルに書き込まれます。指定した SCN 以降の変更はファイルに書き込まれません。

表 2-10 バックアップ・セット内のデータファイルのリスト (続き)

列	指定対象
Ckp Time	データファイルをバックアップした時点のデータファイルのチェックポイント。時刻より前のデータベース変更はすべて、ファイルに書き込まれます。指定した時刻以降の変更はファイルに書き込まれません。
Name	このバックアップ・セットからリストアされており、 <code>SET NEWNAME</code> コマンドが入力されていない場合に、このファイルがリストアされる場所。 <b>関連項目:</b> 2-254 ページの「 <code>SET</code> 」を参照してください。

表 2-11 バックアップ・セット内のアーカイブ・ログのリスト

列	指定対象
Thrd	REDO ログのスレッド番号。
Seq	アーカイブ・ログのログ順序番号。
Low SCN	アーカイブ・ログ内の最小 SCN。
Low Time	データベースが、この順序番号を持つ REDO ログに切り替わった時刻。
Next SCN	次のアーカイブ・ログ順序の下位 SCN
Next Time	次のアーカイブ・ログ順序の下位の時刻

表 2-12 バックアップ・セット ... のバックアップ・セット・コピー ... (複数ピースの場合のみ)

列	指定対象
Device Type	バックアップが行われたデバイスのタイプ (DISK または sbt など)。
Elapsed Time	バックアップの期間。
Completion Time	バックアップ・セットをとった日付と時刻。このフィールドの書式は、 <code>NLS_LANG</code> および <code>NLS_DATE_FORMAT</code> の環境設定によって変わります。
Tag	バックアップ・セットに適用されるタグ。タグ名には大 / 小文字区別がなく、すべて大文字で表示されることに注意してください。

表 2-13 バックアップ・セット ... のコピー ... のバックアップ・ピースのリスト (複数ピースの場合)

列	指定対象
BP Key	リカバリ・カタログまたはターゲット・データベース制御ファイルにあるバックアップ・ピースの一意の識別子。  Recovery Manager がリカバリ・カタログに接続している場合、BP Key はカタログ内のバックアップ・ピースの主キーです。これは、 <code>RC_BACKUP_PIECE</code> ビューの <code>BP_KEY</code> に対応します。Recovery Manager が <code>NOCATALOG</code> モードで接続している場合、BP Key では <code>V\$BACKUP_PIECE</code> からの <code>RECID</code> が表示されます。  <b>注意:</b> リカバリ・カタログおよび制御ファイルにある KEY の値は異なります。
Pc#	バックアップ・セット内のバックアップ・ピースの数。

表 2-13 バックアップ・セット ... のコピー ... のバックアップ・ピースのリスト（複数ピースの場合）（続き）

列	指定対象
Status	バックアップ・ピースの状態。AVAILABLE、UNAVAILABLE または EXPIRED（各状態の説明は、CHANGE、CROSSCHECK および DELETE コマンドを参照）。
Piece Name	バックアップ・ピースのファイル名またはハンドル。バックアップ・ピースが SBT に格納されている場合は、メディア ID も表示されます。

表 2-14 プロキシ・コピーのリスト

列	指定対象
PC Key	このプロキシ・コピーを識別する一意のキー。 Recovery Manager がリカバリ・カタログに接続している場合、PC Key はカタログ内のプロキシ・コピーの主キーです。これは、RC_PROXY_DATAFILE ビューの XDF_KEY または RC_PROXY_CONTROLFILE ビューの XCF_KEY に対応します。Recovery Manager が NOCATALOG モードで接続している場合、PC Key では V\$PROXY_DATAFILE からの RECID が表示されます。
File	コピーされたデータファイルの絶対ファイル番号。
Status	プロキシ・コピーのステータス（AVAILABLE、UNAVAILABLE または EXPIRED）。各ステータスの説明は、CHANGE、CROSSCHECK および DELETE コマンドを参照。
Completion Time	バックアップ・セットをとった日付と時刻。このフィールドの書式は、NLS_LANG および NLS_DATE_FORMAT の環境設定によって変わります。
Ckp SCN	プロキシ・コピー制御ファイルのチェックポイント SCN。指定した SCN より前に REDO レコードに記録されたデータベース変更はすべて、制御ファイルに反映されます。
Ckp time	プロキシ・コピー制御ファイルのチェックポイント時刻。指定した時刻より前に REDO レコードに記録されたデータベース変更はすべて、制御ファイルに反映されます。
Datafile name	このバックアップ・セットからリストアップされており、SET NEWNAME コマンドが入力されていない場合に、このファイルがリストアップされる場所。 <b>関連項目：</b> 2-254 ページの SET コマンドを参照してください。
Handle	このプロキシ・コピー用のメディア・マネージャのハンドル。オブジェクトが sbt にある場合は、メディア ID も表示されます。
Tag	プロキシ・コピーに適用されるタグ。タグ名には大 / 小文字区別がなく、すべて大文字で表示されることに注意してください。

表 2-15 バックアップ・セットのリスト (LIST BACKUP ... SUMMARY)

列	指定対象
Key	このバックアップ・セットを識別する一意のキー。 Recovery Manager がリカバリ・カタログに接続している場合、BS Key はカタログ内のバックアップ・セットの主キーです。これは、RC_BACKUP_SET ビューの BS_KEY に対応します。Recovery Manager が NOCATALOG モードで接続している場合、BS Key では V\$BACKUP_SET からの RECID が表示されます。
TY	バックアップのタイプ。バックアップ・セット (B) またはプロキシ・コピー (P)。
LV	増分バックアップの場合は、増分バックアップのレベル (0 または 1)。 データファイルの全体バックアップを含むバックアップ・セットの場合は F。 アーカイブ REDO ログを含むバックアップ・セットの場合は A。
S	バックアップの状態。A (使用可能)、U (使用不可能) または X (セットのすべてのバックアップ・ピースが期限切れ)。それぞれの状態の詳細は、CHANGE、CROSSCHECK および DELETE コマンドを参照してください。
Device Type	バックアップが行われたデバイスのタイプ (DISK または sbt など)。
Completion Time	バックアップ・セットをとった日付と時刻。このフィールドの書式は、NLS_LANG および NLS_DATE_FORMAT の環境設定によって変わります。
#Pieces	バックアップ・セット内のバックアップ・ピースの数。
#Copies	セット内のバックアップ・ピースごとに作成されたコピー数。多重化が実行されていない場合、この数値は 1 です。それ以外の場合、値の範囲は 2 ~ 4 です。
Compressed	バックアップ・セットが Recovery Manager により圧縮された場合は YES。Recovery Manager により圧縮されていない場合は NO。
Tag	バックアップ・セットに適用されるタグ。アスタリスク (*) は、バックアップ・ピースのタグが同じバックアップ・セット内で異なることを示します。この状態は、CATALOG または BACKUP BACKUPSET を使用しているときにユーザーがタグを変更した場合に発生します。タグ名には大 / 小文字区別がなく、すべて大文字で表示されることに注意してください。

表 2-16 バックアップ・ピースのリスト (LIST BACKUPPIECE ...)

列	指定対象
BP Key	リカバリ・カタログまたはターゲット・データベース制御ファイルにあるバックアップ・ピースの一意の識別子。 Recovery Manager がカタログに接続している場合、BP Key はカタログ内のバックアップ・ピースの主キーです。これは、RC_BACKUP_PIECE ビューの BP_KEY に対応します。Recovery Manager が NOCATALOG モードで接続している場合、BP Key では V\$BACKUP_PIECE からの RECID が表示されます。 <b>注意:</b> リカバリ・カタログおよび制御ファイルにある KEY の値は異なります。



表 2-16 バックアップ・ピースのリスト (LIST BACKUPPIECE ...) (続き)

列	指定対象
BS Key	このバックアップ・セットを識別する一意のキー。 Recovery Manager がリカバリ・カタログに接続している場合、BS Key はカタログ内のバックアップ・セットの主キーです。これは、RC_BACKUP_SET ビューの BS_KEY に対応します。Recovery Manager が NOCATALOG モードで接続している場合、BS Key では V\$BACKUP_SET からの RECID が表示されます。
Pc#	バックアップ・セット内のバックアップ・ピースの数。
Cp#	バックアップ・セット内のこのバックアップ・ピースのコピー数。多重化が実行されていない場合、この数値は 1 です。それ以外の場合、値の範囲は 2 ~ 4 です。
Status	バックアップ・ピースの状態。AVAILABLE、UNAVAILABLE または EXPIRED (各状態の説明は、CHANGE、CROSSCHECK および DELETE コマンドを参照)。
Device Type	バックアップが行われたデバイスのタイプ (DISK または sbt など)。
Piece Name	バックアップ・ピースのファイル名またはハンドル。ピースが SBT に格納されている場合は、ハンドルおよびメディア ID が表示されます。

表 2-17 データファイル・バックアップのリスト (LIST BACKUP ... BY FILE)

列	指定対象
File	絶対データファイル番号。
Key	このバックアップ・セットを識別する一意のキー。 Recovery Manager がリカバリ・カタログに接続している場合、Key はカタログ内のバックアップ・セットの主キーです。これは、RC_BACKUP_SET ビューの BS_KEY に対応します。Recovery Manager が NOCATALOG モードでターゲット・データベースに接続している場合、Key では V\$BACKUP_SET からの RECID が表示されます。
TY	バックアップのタイプ。バックアップ・セット (B) またはプロキシ・コピー (P)。
LV	バックアップ・レベル。非増分の場合は F、増分の場合はレベル 0 またはレベル 1。
S	バックアップの状態。A (使用可能)、U (使用不可能) または X (セットのすべてのバックアップ・ピースが期限切れ)。それぞれの状態の詳細は、CHANGE、CROSSCHECK および DELETE コマンドを参照してください。
Ckp SCN	データファイルをバックアップした時点のデータファイルのチェックポイント。この SCN より前のデータベース変更はすべて、ファイルに書き込まれます。指定した SCN 以降の変更はファイルに書き込まれません。
Ckp Time	データファイルをバックアップした時点のデータファイルのチェックポイント。時刻より前のデータベース変更はすべて、ファイルに書き込まれます。指定した時刻以降の変更はファイルに書き込まれません。

表 2-17 データファイル・バックアップのリスト (LIST BACKUP ... BY FILE) (続き)

列	指定対象
#Pieces	バックアップ・セット内のバックアップ・ピースの数。
#Copies	セット内のバックアップ・ピースごとに作成されたコピー数。多重化が実行されていない場合、この数値は 1 です。それ以外の場合、値の範囲は 2 ~ 4 です。
Compressed	バックアップが Recovery Manager により圧縮された場合は YES。Recovery Manager により圧縮されていない場合は NO。
Tag	バックアップ・セットに適用されるタグ。アスタリスク (*) は、バックアップ・ピースのタグが同じバックアップ・セット内で異なることを示します。この状態は、CATALOG または BACKUP BACKUPSET を使用しているときにユーザーがタグを変更した場合に発生します。タグ名には大 / 小文字区別がなく、すべて大文字で表示されることに注意してください。

表 2-18 アーカイブ・ログ・バックアップのリスト (LIST BACKUP ... BY FILE)

列	指定対象
Thrd	REDO ログのスレッド番号。
Seq	アーカイブ・ログのログ順序番号。
Low SCN	アーカイブ・ログ内の最小 SCN。
Low Time	データベースが、この順序番号を持つ REDO ログに切り替わった時刻。
BS Key	このバックアップ・セットを識別する一意のキー。 Recovery Manager がリカバリ・カタログに接続している場合、BS Key はカタログ内のバックアップ・セットの主キーです。これは、RC_BACKUP_SET ビューの BS_KEY に対応します。Recovery Manager が NOCATALOG モードで接続している場合、BS Key では V\$BACKUP_SET からの RECID が表示されます。
S	バックアップの状態。A (使用可能)、U (使用不可能) または X (セットのすべてのバックアップ・ピースが期限切れ)。それぞれの状態の詳細は、CHANGE、CROSSCHECK および DELETE コマンドを参照してください。
#Pieces	バックアップ・セット内のバックアップ・ピースの数。
#Copies	セット内のバックアップ・ピースごとに作成されたコピー数。多重化が実行されていない場合、この数値は 1 です。それ以外の場合、値の範囲は 2 ~ 4 です。
Compressed	バックアップが Recovery Manager により圧縮された場合は YES。Recovery Manager により圧縮されていない場合は NO。
Tag	バックアップ・セットに適用されるタグ。タグ名には大 / 小文字区別がなく、すべて大文字で表示されることに注意してください。

表 2-19 制御ファイル・バックアップのリスト (LIST BACKUP ... BY FILE)

列	指定対象
CF Ckp SCN	制御ファイルのチェックポイント SCN。
Ckp Time	アーカイブ・ログのログ順序番号。
BS Key	このバックアップ・セットを識別する一意のキー。 Recovery Manager がリカバリ・カタログに接続している場合、BS Key はカタログ内のバックアップ・セットの主キーです。これは、RC_BACKUP_SET ビューの BS_KEY に対応します。Recovery Manager が NOCATALOG モードで接続している場合、BS Key では V\$BACKUP_SET からの RECID が表示されます。
S	バックアップの状態。A (使用可能)、U (使用不可能) または X (セットのすべてのバックアップ・ピースが期限切れ)。それぞれの状態の詳細は、CHANGE、CROSSCHECK および DELETE コマンドを参照してください。
#Pieces	バックアップ・セット内のバックアップ・ピースの数。
#Copies	セット内のバックアップ・ピースごとに作成されたコピー数。多重化が実行されていない場合、この数値は 1 です。それ以外の場合、値の範囲は 2 ~ 4 です。
Compressed	バックアップが Recovery Manager により圧縮された場合は YES。Recovery Manager により圧縮されていない場合は NO。
Tag	バックアップ・セットに適用されるタグ。タグ名には大 / 小文字区別がなく、すべて大文字で表示されることに注意してください。

表 2-20 データファイルのコピーのリスト

列	指定対象
Key	データファイルのコピーを指す一意の識別子。CHANGE コマンドでこの値を使用して、データファイルのコピーの状態を変更します。 Recovery Manager がリカバリ・カタログに接続している場合、Key はカタログ内のデータファイル・コピーの主キーです。これは、RC_DATAFILE_COPY ビューの CDF_KEY に対応します。Recovery Manager が NOCATALOG モードで接続している場合、Key では V\$DATAFILE_COPY からの RECID が表示されます。 <b>注意:</b> リカバリ・カタログおよび制御ファイルにある KEY の値は異なります。
File	コピー元のデータファイルのファイル番号。
S	コピーの状態。A (使用可能)、U (使用不可能)、X (期限切れ)。それぞれの状態の詳細は、CHANGE、CROSSCHECK および DELETE コマンドを参照してください。
Completion Time	コピーを取った日付と時刻。このフィールドの書式は、NLS_LANG および NLS_DATE_FORMAT の環境変数によって変わります。
Ckp SCN	データファイルのコピーを取った時点のデータファイルのチェックポイント。SCN より前のデータベース変更は、すべてこのデータファイルに書き込まれます。
Ckp TIME	データファイルのコピーを取った時点のデータファイルのチェックポイント。この時点より前のデータベース変更は、すべてこのデータファイルに書き込まれます。
Name	データファイルのコピーのファイル名。
Tag	プロキシ・コピーに適用されるタグ。タグ名には大 / 小文字区別がなく、すべて大文字で表示されることに注意してください。

表 2-21 制御ファイルのコピーのリスト

列	指定対象
Key	<p>制御ファイルのコピーを指す一意の識別子。CHANGE コマンドでこの値を使用して、コピーの状態を変更します。</p> <p>Recovery Manager がリカバリ・カタログに接続している場合、Key はカタログ内の制御ファイル・コピーの主キーです。これは、RC_CONTROLFILE_COPY ビューの CCF_KEY に対応します。</p> <p>Recovery Manager が NOCATALOG モードで接続している場合、Key では V\$DATAFILE_COPY からの RECID が表示されます。</p> <p><b>注意:</b> リカバリ・カタログおよび制御ファイルにある Key の値は異なります。</p>
S	<p>コピーの状態。A (使用可能)、U (使用不可能)、X (期限切れ)。それぞれの状態の詳細は、CHANGE、CROSSCHECK および DELETE コマンドを参照してください。</p>
Completion Time	<p>コピーを取った日付と時刻。このフィールドの書式は、NLS_LANG および NLS_DATE_FORMAT の環境変数によって変わります。</p>
Ckp SCN	<p>この制御ファイルのコピーを取った時点の制御ファイルのチェックポイント。</p>
Ckp TIME	<p>この制御ファイルのコピーを取った時点の制御ファイルのチェックポイント。</p>
Name	<p>制御ファイルのコピーのファイル名。</p>
Tag	<p>制御ファイル・コピーに適用されるタグ。タグ名には大 / 小文字区別がなく、すべて大文字で表示されることに注意してください。</p>

表 2-22 アーカイブ・ログ・コピーのリスト

列	指定対象
Key	<p>アーカイブ REDO ログ・コピーの一意の識別子。CHANGE コマンドでこの値を使用して、コピーの状態を変更します。</p> <p>Recovery Manager がリカバリ・カタログに接続している場合、Key はカタログ内のバックアップ・セットの主キーです。これは、RC_ARCHIVED_LOG ビューの AL_KEY に対応します。Recovery Manager が NOCATALOG モードで接続している場合、Key では V\$ARCHIVED_LOG からの RECID が表示されます。</p> <p><b>注意:</b> リカバリ・カタログおよび制御ファイルにある Key の値は異なります。</p>
Thrd	<p>REDO ログのスレッド番号。</p>
Seq	<p>ログ順序番号。</p>
S	<p>コピーの状態。A (使用可能)、U (使用不可能)、X (期限切れ)。それぞれの状態の詳細は、CHANGE、CROSSCHECK および DELETE コマンドを参照してください。</p>
Low Time	<p>データベースが、この順序番号を持つ REDO ログに切り替わった時刻。</p>
Name	<p>アーカイブ REDO ログ・コピーのファイル名。</p>

表 2-23 データベースのインカーネーションのリスト

列	指定対象
DB Key	Inc Key と結合して、Recovery Manager がリカバリ・カタログのデータベース・インカーネーションを識別する一意のキー。このキーを使用して、リカバリ・カタログでデータベースの登録を解除します。つまり、そのデータベースに関連するすべての行をリカバリ・カタログから削除します。
Inc Key	DB Key と結合して、Recovery Manager がリカバリ・カタログのデータベース・インカーネーションを識別する一意のキー。このキーは、最新の RESETLOGS より前の時点までデータベースをリカバリするときに、 <code>RESET DATABASE TO INCARNATION</code> で使用します。
DB Name	DB_NAME パラメータにリストされたデータベース名。
DB ID	データベース作成時にデータベースが自動的に生成するデータベース識別番号。
STATUS	現行インカーネーションの場合は CURRENT、現行インカーネーションの親インカーネーションの場合は PARENT、孤立したインカーネーションの場合は ORPHAN。
Reset SCN	インカーネーションが作成されたときの SCN。
Reset Time	インカーネーションが作成された時刻。

表 2-24 リカバリ・カタログのストア・スクリプトのリスト (LIST SCRIPT NAMES)

列	指定対象
Script Name	ストア・スクリプトの名前。
Description	スクリプトの作成時に指定したコメント。

表 2-25 リストア・ポイントのリスト (LIST RESTORE POINT)

列	指定対象
SCN	リストア・ポイントの SCN。
RSP Time	CREATE RESTORE POINT 文で指定した時間。それ以外の場合は NULL です。
Type	保証付きリストア・ポイントの場合は GUARANTEED。通常のリストア・ポイントの場合は NULL。
Time	リストア・ポイントが作成された時刻。
Name	リストア・ポイントの名前。

表 2-26 障害のリスト

列	指定対象
Failure ID	障害を示す一意の識別子。
Priority	<p>障害の優先順位: CRITICAL、HIGH または LOW。</p> <p>障害の優先順位が CRITICAL の場合は、データベース全体が使用不可能になるので、すぐに対処する必要があります。通常、CRITICAL な障害が発生するとインスタンスが停止し、その後のインスタンスの起動時に障害が診断されます。データベースは CRITICAL な障害がすべて修正されるまで使用できません (「ADVISE FAILURE」を参照)。</p> <p>障害の優先順位が HIGH の場合は、データベースの一部が使用できなくなるか、リカバリできなくなるため、通常は合理的に短時間で修復する必要があります。このような障害の例には、データ・ブロックの物理的な破損、致命的でない I/O エラー、アーカイブ REDO ログ・ファイルまたはバックアップ・ファイルの欠落などがあります。</p> <p>優先順位が LOW の障害は、それより重要な障害が修正されるまで無視できます。たとえば、ブロックの破損には、優先順位の HIGH が初期的に割り当てられますが、このブロックがデータベースの可用性に影響しない場合は、CHANGE FAILURE を使用して優先順位を LOW に変更できます。</p>
Status	障害の修復ステータス。適切な修復処理が開始されるまで、障害のステータスは OPEN (未修復) になります。修復が完了すると、障害ステータスが CLOSED に変更されます。
Time Detected	障害が診断されたときの日付。
Summary	障害の概要。

表 2-27 データベースのリスト (LIST DB\_UNIQUE\_NAME)

列	指定対象
DB Key	Inc Key と結合して、Recovery Manager がリカバリ・カタログのデータベース・インカネーションを識別する一意のキー。プライマリ・データベースおよびスタンバイ・データベースは、同じ DB Key 値を共有します。
DB Name	データベースの DB_NAME。プライマリ・データベースおよびスタンバイ・データベースは、同じ DB Name 値を共有します。
DB ID	データベースの DBID。プライマリ・データベースおよびスタンバイ・データベースは、同じ DBID を共有します。
Database Role	データベースがプライマリ・データベースである場合は PRIMARY、データベースがスタンバイ・データベースである場合は STANDBY。
Db_unique_name	データベースの DB_UNIQUE_NAME。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 DB_UNIQUE_NAME 値は異なります。

## 例

## 例 2-86 バックアップのリスト

この例では、すべてのバックアップをリストします。出力には、ディスク上の2つのバックアップ・セットが表示されます。1つにはデータファイルが含まれ、もう1つには自動バックアップが含まれます。また、アーカイブ REDO ログ・ファイルを含む1つの SBT バックアップも表示されます。

```
RMAN> LIST BACKUP;
```

```
List of Backup Sets
```

```
=====
```

```
BS Key   Type LV Size      Device Type Elapsed Time Completion Time
-----
200      Full   509.78M    DISK          00:01:03      15-FEB-07
        BP Key: 202   Status: AVAILABLE Compressed: NO Tag: TAG20070215T171219
        Piece Name:
```

```
/disk2/PROD/backupset/2007_02_15/o1_mf_nnndf_TAG20070215T171219_2xb17nbb_.bkp
```

```
List of Datafiles in backup set 200
```

```
File LV Type Ckp SCN    Ckp Time Name
-----
1      Full 421946    15-FEB-07 /disk1/oradata/prod/system01.dbf
2      Full 421946    15-FEB-07 /disk1/oradata/prod/sysaux01.dbf
3      Full 421946    15-FEB-07 /disk1/oradata/prod/undotbs01.dbf
4      Full 421946    15-FEB-07 /disk1/oradata/prod/cwmlite01.dbf
5      Full 421946    15-FEB-07 /disk1/oradata/prod/drsys01.dbf
6      Full 421946    15-FEB-07 /disk1/oradata/prod/example01.dbf
7      Full 421946    15-FEB-07 /disk1/oradata/prod/indx01.dbf
8      Full 421946    15-FEB-07 /disk1/oradata/prod/tools01.dbf
9      Full 421946    15-FEB-07 /disk1/oradata/prod/users01.dbf
```

```
BS Key   Type LV Size      Device Type Elapsed Time Completion Time
-----
201      Full   7.98M     DISK          00:00:03      15-FEB-07
        BP Key: 203   Status: AVAILABLE Compressed: NO Tag: TAG20070215T171219
        Piece Name:
```

```
/disk2/PROD/backupset/2007_02_15/o1_mf_ncsnf_TAG20070215T171219_2xb19prg_.bkp
```

```
SPFILE Included: Modification time: 15-FEB-07
```

```
SPFILE db_unique_name: PROD
```

```
Control File Included: Ckp SCN: 421968      Ckp time: 15-FEB-07
```

```
BS Key   Size      Device Type Elapsed Time Completion Time
-----
227      30.50M    SBT_TAPE    00:00:11      15-FEB-07
        BP Key: 230   Status: AVAILABLE Compressed: NO Tag: TAG20070215T171334
        Handle: 0bia4rtv_1_1 Media:
```

```
List of Archived Logs in backup set 227
```

```
Thrd Seq    Low SCN    Low Time Next SCN    Next Time
-----
1      5          389156    15-FEB-07 411006    15-FEB-07
1      6          411006    15-FEB-07 412972    15-FEB-07
1      7          412972    15-FEB-07 417086    15-FEB-07
1      8          417086    15-FEB-07 417114    15-FEB-07
1      9          417114    15-FEB-07 417853    15-FEB-07
1      10         417853    15-FEB-07 421698    15-FEB-07
1      11         421698    15-FEB-07 421988    15-FEB-07
```

**例 2-87 バックアップのサマリー・リストの表示**

この例では、2-179 ページの例 2-86 でリストされている Recovery Manager のバックアップのサマリーを表示します。

```
RMAN> LIST BACKUP SUMMARY;
```

```
List of Backups
```

```
=====
Key      TY LV S Device Type Completion Time #Pieces #Copies Compressed Tag
-----
200      B F A DISK      15-FEB-07      1      1      NO      TAG20070215T171219
201      B F A DISK      15-FEB-07      1      1      NO      TAG20070215T171219
227      B A A SBT_TAPE   15-FEB-07      1      1      NO      TAG20070215T171334
```

**例 2-88 ファイルによるバックアップのリスト**

この例では、すべてのバックアップをファイル別にグループ化します。データファイルのバックアップのタグ列は、1つのバックアップ・セットにタグの異なるバックアップ・ピースが含まれていることを示しています。アーカイブ・ログ・バックアップのステータス列は、バックアップ・セットが使用できないことを示しています。

```
RMAN> LIST BACKUP BY FILE;
```

```
List of Datafile Backups
```

```
=====
File Key      TY LV S Ckp SCN      Ckp Time #Pieces #Copies Compressed Tag
-----
1      329      B F A 454959      16-FEB-07 1      1      NO      DF1
2      349      B F A 454997      16-FEB-07 1      1      NO      DF2
3      527      B F A 455218      16-FEB-07 1      1      NO      FRI_BKP
      368      B F A 455022      16-FEB-07 1      1      NO      DF3
4      387      B F X 455042      16-FEB-07 1      1      NO      DF4
5      407      B F A 455063      16-FEB-07 1      1      NO      DF5
6      428      B F A 455083      16-FEB-07 1      2      NO      *
7      450      B F X 455103      16-FEB-07 1      1      NO      DF7
8      473      B F A 455123      16-FEB-07 1      1      NO      DF8
9      497      B F A 455143      16-FEB-07 1      1      NO      DF9
```

```
List of Archived Log Backups
```

```
=====
Thrd Seq      Low SCN      Low Time BS Key S #Pieces #Copies Compressed Tag
-----
1      5      389156      15-FEB-07 227 U 1      1      NO      TAG20070215T171334
1      6      411006      15-FEB-07 227 U 1      1      NO      TAG20070215T171334
1      7      412972      15-FEB-07 227 U 1      1      NO      TAG20070215T171334
1      8      417086      15-FEB-07 227 U 1      1      NO      TAG20070215T171334
1      9      417114      15-FEB-07 227 U 1      1      NO      TAG20070215T171334
1      10     417853      15-FEB-07 227 U 1      1      NO      TAG20070215T171334
1      11     421698      15-FEB-07 227 U 1      1      NO      TAG20070215T171334
```

```
List of Control File Backups
```

```
=====
CF Ckp SCN Ckp Time BS Key S #Pieces #Copies Compressed Tag
-----
454974      16-FEB-07 330 A 1      1      NO      DF1
421968      15-FEB-07 201 A 1      1      NO      TAG20070215T171219
```

```
List of SPFILE Backups
```

```
=====
Modification Time BS Key S #Pieces #Copies Compressed Tag
-----
15-FEB-07      330 A 1      1      NO      DF1
15-FEB-07      201 A 1      1      NO      TAG20070215T171219
```



**例 2-89 イメージ・コピーのリスト**

次の例では、Recovery Manager で認識されるすべてのデータファイル、制御ファイルおよびアーカイブ REDO ログのコピーのリストを表示します。

```
RMAN> LIST COPY;
```

```
List of Datafile Copies
```

```
=====
```

Key	File S	Completion Time	Ckp SCN	Ckp Time
618	1 A	16-FEB-07	461057	16-FEB-07
	Name: /disk2/PROD/datafile/o1_mf_system_2xdbrg13_.dbf			
	Tag: TAG20070216T140706			
631	2 A	16-FEB-07	461163	16-FEB-07
	Name: /disk2/PROD/datafile/o1_mf_sysaux_2xdbzybx_.dbf			
	Tag: TAG20070216T141109			

```
List of Control File Copies
```

```
=====
```

Key	S	Completion Time	Ckp SCN	Ckp Time
619	A	16-FEB-07	461133	16-FEB-07
	Name: /disk2/PROD/controlfile/o1_mf_TAG20070216T140706_2xdbz5tb_.ctl			
	Tag: TAG20070216T140706			
594	A	16-FEB-07	460650	16-FEB-07
	Name: /disk2/PROD/controlfile/o1_mf_TAG20070216T135954_2xdbbz99_.ctl			
	Tag: TAG20070216T135954			

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

```
=====
```

Key	Thrd	Seq	S	Low Time
105	1	5	A	15-FEB-07
	Name: /disk1/oradata/prod/arch/archive1_5_614616887.dbf			
122	1	6	A	15-FEB-07
	Name: /disk1/oradata/prod/arch/archive1_6_614616887.dbf			
123	1	7	A	15-FEB-07
	Name: /disk1/oradata/prod/arch/archive1_7_614616887.dbf			
124	1	8	A	15-FEB-07
	Name: /disk1/oradata/prod/arch/archive1_8_614616887.dbf			
125	1	9	A	15-FEB-07
	Name: /disk1/oradata/prod/arch/archive1_9_614616887.dbf			
185	1	10	A	15-FEB-07
	Name: /disk1/oradata/prod/arch/archive1_10_614616887.dbf			
221	1	11	A	15-FEB-07
	Name: /disk1/oradata/prod/arch/archive1_11_614616887.dbf			
262	1	12	A	15-FEB-07
	Name: /disk1/oradata/prod/arch/archive1_12_614616887.dbf			
263	1	13	A	15-FEB-07
	Name: /disk1/oradata/prod/arch/archive1_13_614616887.dbf			

**例 2-90 データベース・インカーネーションのリスト**

この例では、リカバリ・カタログに記録されているすべてのデータベース・インカーネーションをリストします。

```
RMAN> LIST INCARNATION;
```

```
List of Database Incarnations
DB Key  Inc Key DB Name  DB ID          STATUS  Reset SCN  Reset Time
-----
78      94      PROD      1619073740    PARENT  1          14-FEB-07
78      79      PROD      1619073740    CURRENT 388003     15-FEB-07
```

**例 2-91 障害のリスト**

この例では、優先順位に関係なくすべての障害をリストします。ALL を指定しない場合、LIST FAILURE の出力には優先順位が LOW の障害は含まれません。

```
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      HIGH    OPEN    23-APR-07  Datafile 1: '/disk1/oradata/prod/system01.dbf'
contains one or more corrupt blocks
```

## PRINT SCRIPT

### 用途

PRINT SCRIPT コマンドを使用すると、ローカルまたはグローバル・ストアド・スクリプトを標準出力またはファイルに出力できます。

### 前提条件

PRINT SCRIPT は、Recovery Manager プロンプトでのみ実行できます。Recovery Manager が、ターゲット・データベースとリカバリ・カタログに接続されている必要があります。リカバリ・カタログ・データベースはオープンの状態である必要があります。

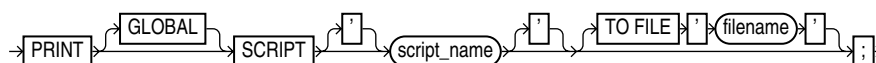
指定したスクリプトがローカル・スクリプトである場合は、スクリプトを作成または置換したときに接続していたターゲット・データベースに Recovery Manager を接続する必要があります。

### 使用上の注意

GLOBAL を指定しない場合、ローカル・スクリプトまたはグローバル・スクリプト `script_name` が検索され、出力されます。ローカル・スクリプトが検出された場合は、そのスクリプトが出力されます。ローカル・スクリプトが検出されず、グローバル・スクリプト `script_name` が検出された場合は、グローバル・スクリプトが出力されます。

### 構文

**printScript:=**



### セマンティクス

構文の要素	説明
GLOBAL	スクリプトをグローバルとして指定します。  GLOBAL を指定する場合は、指定した名前のグローバル・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は、Recovery Manager によりエラー RMAN-06004 が戻されます。  <b>関連項目：</b> グローバル・スクリプトとローカル・スクリプトの違いについては、2-106 ページの「 <a href="#">使用上の注意</a> 」を参照してください。
<code>script_name</code>	出力するスクリプトの名前を指定します。スクリプト名に空白または予約語が含まれている場合は、引用符で囲む必要があります。
TO FILE ' <code>filename</code> '	出力を標準出力ではなく指定したファイルに送信します。

### 例

#### 例 2-92 スクリプトのファイルへの出力

この例では、スクリプトをファイル `/tmp/global_backup_db.rman` に出力します。

```
RMAN> PRINT GLOBAL SCRIPT global_backup_db TO FILE "/tmp/global_backup_db.rman";
```

**例 2-93 スクリプトの画面表示**

この例では、ストアド・スクリプトを標準出力に出力します（例には出力例も含まれます）。

```
RMAN> PRINT SCRIPT backup_whole;
```

```
printing stored script: backup_whole
{
  BACKUP
    INCREMENTAL LEVEL 0 TAG backup_whole
    FORMAT "/disk2/backup/%U"
    DATABASE PLUS ARCHIVELOG;
}
```

---

## QUIT

### 用途

QUIT コマンドを使用すると、Recovery Manager ユーティリティを停止できます。このコマンドの機能は、EXIT コマンドと同じです。

### 前提条件

このコマンドは Recovery Manager プロンプトでのみ実行できます。

### 構文

**quit:=**

→QUIT→

### 例

#### 例 2-94 Recovery Manager の終了

この例では、Recovery Manager を終了します（例には出力例も含まれます）。

```
RMAN> QUIT
```

```
Recovery Manager complete.
```

## RECOVER

### 用途

RECOVER コマンドを使用すると、次の個々のタスクのいずれかを実行できます。

- データベース全体、またはリストアされた 1 つ以上のデータファイルの完全リカバリを実行します。
- データベースの Point-in-Time リカバリ (DBPITR) または表領域の Point-in-Time リカバリ (TSPITR) を実行します。
- 適宜ロールフォワードするように、増分バックアップをデータファイルのイメージ・コピー (リストアされたデータファイルではなく) に適用します。
- データファイル内の破損したデータ・ブロックまたはその集合をリカバリします。

**関連項目：** データファイルのリカバリ方法は、『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』を参照してください。

### 前提条件

リカバリに必要なすべての REDO または増分の変更は、ディスクまたは SBT に存在している必要があります。Recovery Manager でリカバリ中に増分バックアップまたはアーカイブ REDO ログをリストアする必要がある場合は、構成済の自動チャンネルを使用するか、またはそのバックアップを作成したものと同一タイプのチャンネルを手動で割り当てる必要があります。

暗号化されている表領域に対してメディア・リカバリを実行する場合は、この表領域のメディア・リカバリを実行するときに Oracle ウォレットがオープンしている必要があります。暗号化されている表領域については、『Oracle Database 管理者ガイド』を参照してください。

RECOVER BLOCK には次の前提条件が適用されます。

- ターゲット・データベースは、ARCHIVELOG モードで実行され、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。
- スタンバイ・データベースは、ターゲット・データベースにはできません。
- リカバリの実行対象は、メディア破損マークが付いているブロックのみです。V\$DATABASE\_BLOCK\_CORRUPTION ビューは、ファイルに対して最新の BACKUP または BACKUP ... VALIDATE コマンドの実行以降にファイル内で破損としてマークされたブロックを示します。
- 破損ブロックを含むデータファイルのバックアップは、プロキシ・バックアップではなく、全体バックアップである必要があります。プロキシ・バックアップしか存在しない場合に、それをディスク上のデフォルト以外の場所にリストアすると、Recovery Manager ではリストアされたファイルがデータファイルのコピーとして認識されます。このデータファイルのコピーを、ブロック・メディア・リカバリに使用できます。
- Recovery Manager でリカバリに使用できるのはアーカイブ REDO ログのみです。ブロック・メディア・リカバリでは、欠落しているかアクセス不能なログは残りませんが、欠落しているかアクセス不能なレコードは残る場合があります (『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』を参照)。

Recovery Manager でフラッシュバック・ログを検索して破損ブロックの正常なコピーを見つげられるようにする場合は、ターゲット・データベースに対してフラッシュバック・データベースを有効にする必要があります。

## 使用上の注意

デフォルトでは、完全リカバリが実行されます。Point-in-Time リカバリの場合は、**RUN** コマンドの **RESTORE** と **RECOVER** の両方のコマンドの前に **SET UNTIL** コマンドを入力して、両方のコマンドに **UNTIL** の時刻が適用されるようにすることをお勧めします。データベースのリストア後に **SET UNTIL** を実行すると、ターゲット時刻にデータベースのリカバリを実行できない場合があります。これは、リストア済ファイルのタイムスタンプがターゲット時刻より遅れるためです。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、**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るので注意してください。

### 増分バックアップおよびアーカイブ REDO ログ・ファイル

**RECOVER BLOCK** の場合を除き、**Recovery Manager** ではリカバリに増分バックアップとアーカイブ REDO ログの両方を使用できます。次の検索順序を使用します。

1. ディスクまたはテープへの増分バックアップ・セット
2. ディスク上のアーカイブ REDO ログ
3. ディスク上のアーカイブ REDO ログのバックアップ
4. テープ上のアーカイブ REDO ログのバックアップ・セット

**Recovery Manager** でアーカイブ REDO ログのリストア先を選択するときは、次の優先順位を使用します。

1. **SET ARCHIVELOG DESTINATION**
2. 値が **LOCATION=USE\_DB\_RECOVERY\_FILE\_DEST** に設定されている **LOG\_ARCHIVE\_DEST\_n** パラメータ
3. **LOG\_ARCHIVE\_DEST\_1**

**Recovery Manager** では、増分バックアップからリストアされなかったデータファイルに増分バックアップを適用できます。増分バックアップのオーバーラップしているレベルが存在する場合は、**Recovery Manager** によって、最も長い期間をカバーしているレベルが自動的に選択されます。

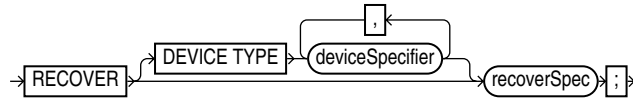
### RESETLOGS を介したリカバリ

データファイルのリカバリを行う前に、**RESTORE** を使用してこれらをリストアする必要があります。リカバリするデータファイルが親インカネーション由来のものである場合、**Recovery Manager** では、**RESETLOGS** 操作を介して透過的にリカバリできます。必要に応じて、**RECOVER** コマンドを使用しても、以前のデータベース・インカネーションからアーカイブ・ログおよび増分バックアップをリストアおよび適用できます。これは、これらのログが以前のリリースの Oracle データベースで生成されている場合も同様です。

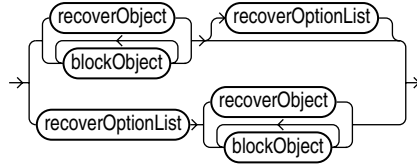
**OPEN RESETLOGS** を使用してリカバリを行う場合は、リカバリに必要なすべてのログが揃っていることを確認します。以前のデータベース・インカネーションでは、バックアップの時点から **RESETLOGS SCN** より番号が 1 少ない SCN までのログが含まれている必要があります。また、このインカネーションの表には、データベース・バックアップの作成時からの **RESETLOGS** 操作の完全な履歴が含まれている必要があります。**V\$DATABASE\_INCARNATION** に完全なメタデータが見つからない場合は、欠落しているインカネーションからアーカイブ REDO ログに **CATALOG** を使用して、このメタデータを再作成できます。

**関連項目：** アーカイブ REDO ログをリストアするデフォルトの位置については、2-221 ページの **RESTORE** コマンドを参照してください。フラッシュ・リカバリ領域にログをステージングする場合、**Recovery Manager** は自動的に **MAXSIZE** オプションを指定します。

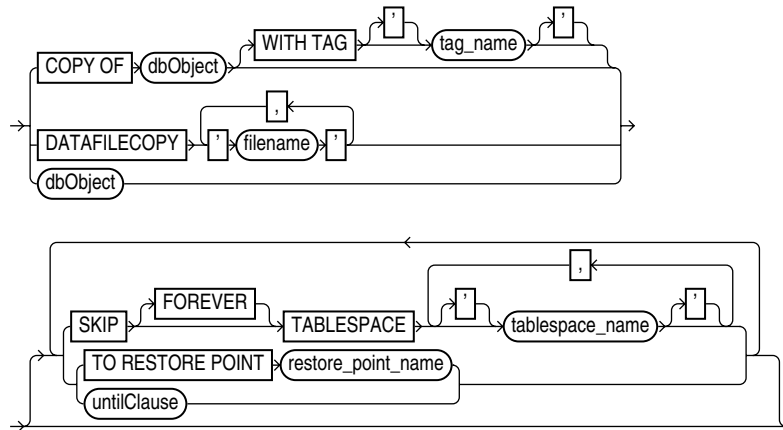
## 構文

**recover::=**

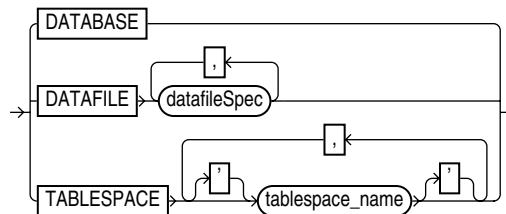
(deviceSpecifier::= (3-15 ページ)、 recoverObject::= (2-188 ページ)、  
recoverOptionList::= (2-189 ページ))

**recoverSpec::=**

(recoverObject::= (2-188 ページ)、 blockObject::= (2-189 ページ)、  
recoverOptionList::= (2-189 ページ))

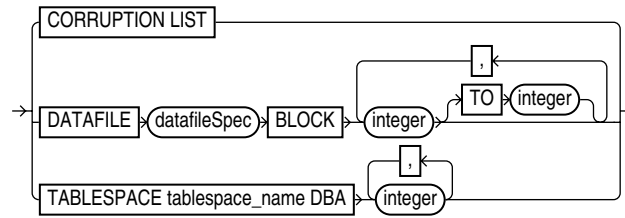
**recoverObject::=**

(dbObject::= (2-188 ページ)、 blockObject::= (2-189 ページ)、 untilClause::= (3-38 ページ))

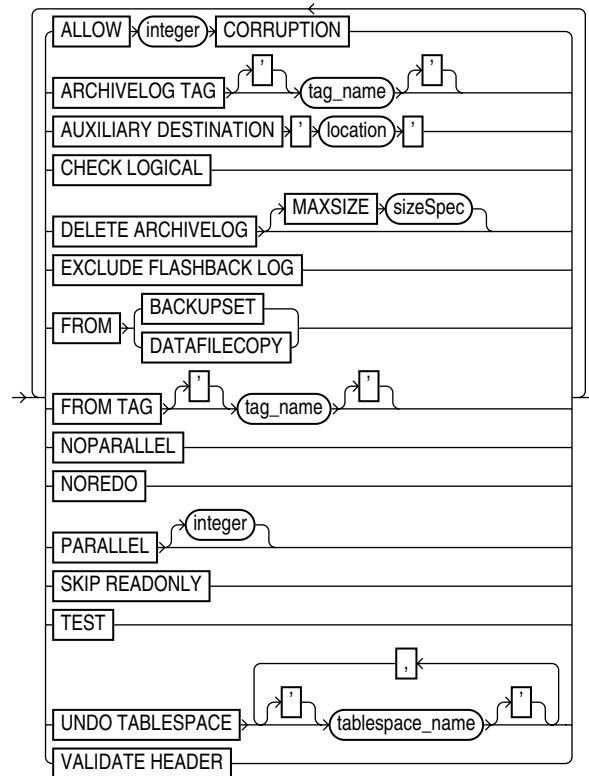
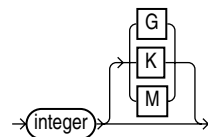
**dbObject::=**

(datafileSpec::= (3-14 ページ))



**blockObject::=**

(datafileSpec::= (3-14 ページ))

**recoverOptionList::=****sizeSpec::=**

## セマンティクス

### recover

構文の要素	説明
DEVICE TYPE <a href="#">deviceSpecifier</a>	指定したデバイス・タイプ専用の自動チャンネルを割り当てます。たとえば、自動ディスクおよびテープ・チャンネルを構成して RECOVER DEVICE TYPE DISK を発行すると、Recovery Manager ではディスク・チャンネルのみが割り当てられます。  DEVICE TYPE オプションを指定する前に、 <a href="#">CONFIGURE DEVICE TYPE</a> コマンドを使用してデバイス・タイプを構成する必要があります（事前構成される DISK 以外）。  <b>注意：</b> チャンネルを手動で割り当ててから、RECOVER DEVICE TYPE を実行することはできません。  <b>関連項目：</b> 3-15 ページの「 <a href="#">deviceSpecifier</a> 」を参照してください。
<a href="#">recoverSpec</a>	リカバリされるオブジェクトのタイプを指定します。

### recoverSpec

構文の要素	説明
<a href="#">recoverObject</a>	リカバリされるオブジェクトのタイプを指定します。
<a href="#">blockObject</a>	ブロック・メディア・リカバリを使用してリカバリするブロックを指定します。
<a href="#">recoverOptionList</a>	リカバリ・オプションを指定します。

### recoverObject

この副次句は、リカバリするファイルを指定します。構文図は、2-188 ページの「[recoverObject::=](#)」を参照してください。

構文の要素	説明
COPY OF <a href="#">dbObject</a>	ファイルの最近の増分バックアップと同じ時刻またはそれ以前の時刻にロールフォワードするために、増分バックアップを指定したイメージ・コピーに適用します。既存のイメージ・コピーは上書きされ、リカバリ中はファジー状態になります。Recovery Manager では、イメージ・コピーのリカバリ後に自動バックアップを実行します。  このコマンドを使用すると、データファイル・コピーが更新されますが、このコマンドは、現行のデータファイルのメディア・リカバリではありません。このコマンドを BACKUP... FOR RECOVER OF COPY 構文と組み合わせて使用すると、増分更新バックアップで計画が実装されます。  次の要件が満たされている必要があります。 <ul style="list-style-type: none"> <li>リカバリする各データファイルのコピーが 1 つ以上存在する必要がある。</li> <li>リカバリするイメージ・コピーより後に取られた増分バックアップが存在する必要がある。</li> </ul> 操作を適用できる増分バックアップに対象となるコピーが複数ある場合は、Recovery Manager によって、適切なコピーが 1 つ選択される。  <b>注意：</b> Recovery Manager では、指定された時刻にリカバリできない場合、エラーではなく警告が発行されます。これは、増分バックアップを使用できないためです。
WITH TAG <a href="#">tag_name</a>	ロールフォワードするイメージ・コピーを識別するタグ名を指定します。
DATAFILECOPY ' <a href="#">filename</a> '	増分バックアップを指定したデータファイルのイメージ・コピーに適用します（例 2-98 を参照）。RECOVER COPY OF の説明を参照してください。

構文の要素	説明
<code>dbObject</code>	リカバリを必要とするデータ・ブロックを指定します。 <b>関連項目</b> : 2-191 ページの「 <code>dbObject</code> 」を参照してください。
SKIP	メディア・リカバリ開始前に、指定された表領域にあるデータファイルをオフラインにします。これらのファイルは、メディア・リカバリが完了した後もオフラインのままです。 このオプションは、一時データのみが含まれている表領域のリカバリを行わないようにしたり、いくつかの表領域のリカバリを延期する場合に有効です。
FOREVER	DROP オプションを使用してデータファイルをオフラインにします (2-197 ページの例 2-97 を参照)。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に、指定された表領域を削除する場合は、SKIP FOREVER TABLESPACE を使用します。 <b>注意</b> : 不完全リカバリを実行する場合は、SKIP に FOREVER オプションが必要になります。
TABLESPACE <code>tablespace_name</code>	オフラインにする表領域の名前を指定します。
TO RESTORE POINT <code>restore_point_name</code>	リストア・ポイントを作成した時点の SCN を上限として、RECOVER コマンドを終了するリストア・ポイントを指定します。指定した値は含まれます。上限値が含まれるため、Recovery Manager は、リストア・ポイントに対応する SCN までリカバリできるファイルのみを選択します。
<code>untilClause</code>	RECOVER コマンドを終了する過去の時刻、SCN、またはログ順序番号を指定します。 1 つ以上の表領域で使用する場合、この句は指定された表領域の Point-in-Time リカバリ (TSPITR) 操作を示します。また、RECOVER DATAFILE とともに使用できません。RECOVER DATABASE には使用できません (詳細は 2-187 ページの「 <b>使用上の注意</b> 」を参照)。データベースの Point-in-Time リカバリ (DBPITR) の後は、RESETLOGS オプションを指定してデータベースをオープンする必要があります。 <b>関連項目</b> : 3-38 ページの「 <code>untilClause</code> 」を参照してください。

**dbObject**

この副次句は、データベースまたはデータベースのサブセットのいずれのリカバリを行うのかを指定します。構文図は、2-188 ページの「`dbObject::=`」を参照してください。

構文の要素	説明
DATABASE	データベース全体のリカバリを指定します (2-197 ページの例 2-97 を参照)。データベースはマウントする必要がありますが、オープンはしないでください。 デフォルトでは、RECOVER DATABASE コマンドは、リカバリされる時点で NORMAL モードでオフラインにされているファイルのリカバリは行いません。Recovery Manager は、NORMAL モードでオフラインにされたファイルをそれ以上のチェックはしないで、除外します。 制御ファイルを失った後にリカバリを行う場合は、ディスク上のデータファイルの実際の位置を指すように制御ファイルを自動的に更新します (2-198 ページの例 2-99 を参照)。 <b>注意</b> : Recovery Manager は、データファイルの追加用の REDO を検出すると、追加されるデータファイルを含む表領域がリカバリ中にスキップされないかぎり、新しいデータファイルを自動的に作成します。この自動作成は、バックアップの制御ファイルがリカバリされる前にリストアされ、バックアップの制御ファイルに最近追加されたデータファイルのレコードが存在しない場合に実行されます。

構文の要素	説明
DATAFILE <i>datafileSpec</i>	<p>リカバリする 1 つ以上のデータファイルのリストをファイル名または絶対データファイル番号で指定します。ターゲット・データベースはマウントまたはオープン状態である必要があります。データベースがオープン状態の場合は、リカバリするデータファイルがオフラインである必要があります。</p> <p>リカバリ・カタログを使用していない場合、ファイル名は制御ファイルに記録されているデータファイルの名前にする必要があります。リカバリ・カタログを使用している場合、制御ファイル内の名前が最近更新されたとしても、データファイルのファイル名はカタログに記録された最新の名前にする必要があります。たとえば、制御ファイルでデータファイルの名前が変更されたが、カタログを再同期化する前にインスタンスで障害が発生したとします。この場合、RECOVER コマンドでは、データファイルの古い名前を指定してください。この名前がカタログに記録されているためです。</p> <p><b>注意：</b>データベース全体をある 1 つの時点にリカバリしたり、表領域をデータベースの残りの表領域とは異なるある時点にリカバリ (TSPITR) することはできませんが、個々のデータファイルを異なる時点へ任意にリカバリすることはできません。TSPITR の詳細は、『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』で説明する手順を参照してください。</p> <p><b>関連項目：</b>3-14 ページの「<i>datafileSpec</i>」を参照してください。</p>
TABLESPACE <i>tablespace_name</i>	<p>リカバリする 1 つ以上の表領域のリストを指定します (2-196 ページの例 2-95 と 2-196 ページの例 2-96 を参照)。ターゲット・データベースはマウントまたはオープン状態である必要があります。データベースがオープン状態の場合は、リカバリする表領域がオフラインである必要があります。</p> <p><b>注意：</b>Recovery Manager は、データファイルの追加に対する REDO を検出すると、新しいデータファイルを自動的に作成します。この自動作成は、バックアップの制御ファイルがリカバリされる前にリストアップされ、バックアップの制御ファイルに最近追加されたデータファイルのレコードが存在しない場合に実行されます。</p>

### blockObject

この副次句は、リカバリを必要とするデータ・ブロックを指定します。構文図は、2-189 ページの「**blockObject:=**」を参照してください。ブロック・メディア・リカバリに固有の前提条件については、2-186 ページの「**前提条件**」を参照してください。

RECOVER CORRUPTION LIST を使用して V\$DATABASE\_BLOCK\_CORRUPTION ビューに示されたすべてのブロックをリカバリするか、またはデータファイル番号とブロック番号または表領域とデータ・ブロック・アクセス (DBA) を指定できます。個々のブロックについて実行できるのは、完全リカバリのみです。

構文の要素	説明
CORRUPTION LIST	<p>V\$DATABASE_BLOCK_CORRUPTION ビューに一覧表示された物理破損ブロックをすべてリカバリします。論理的な破損は、ブロック・メディア・リカバリを使用して修復できません。論理的な破損を修復するには、バックアップからデータファイルをリストアし、メディア・リカバリを実行します。</p> <p>V\$DATABASE_BLOCK_CORRUPTION ビューには、<b>Recovery Manager</b> コマンド、<b>ANALYZE</b>、<b>dbv</b>、<b>SQL</b> 問合せなどの <b>Oracle</b> データベース・コンポーネントにより破損としてマークされているブロックが表示されます。つまり、ORA-1578 エラーが発生したプロセスはすべて、このビューにブロック破損が記録されます。このビューに行が追加されるのは、次の 2 種類の破損の場合です。</p> <ul style="list-style-type: none"> <li>■ 物理的な破損（メディア破損と呼ばれることもあります）。データベースでは、この種のブロックは認識されません。チェックサムは無効であり、ブロックには 0（ゼロ）のみが含まれるか、ブロックのヘッダーとフッターが一致しません。</li> <li>■ 論理的な破損。ブロックには有効なチェックサムがあり、ヘッダーとフッターは一致しますが、内容に論理的な一貫性がありません。</li> </ul> <p>このビューには、ブロックとセグメント間の関係を検証して検出できても、個々のブロックのチェックでは検出できない破損については記録されません。</p> <p><b>注意：</b>ブロックが修復されたことを確定または検出する <b>Recovery Manager</b> コマンドはいずれも、V\$DATABASE_BLOCK_CORRUPTION を更新します。たとえば、ブロック・メディア・リカバリが正常に終了すると、<b>Recovery Manager</b> でリポジトリが更新されます。<b>BACKUP</b>、<b>RESTORE</b> または <b>VALIDATE</b> コマンドは、ブロックが修復されていることを検出すると、修復されたブロックをビューから削除します。</p>
DATAFILE <i>datafileSpec</i> BLOCK <i>integer</i> TO <i>integer</i>	<p>データファイル内の個々のデータ・ブロックまたはその集合をリカバリします。TO の範囲は含まれるため、BLOCK 10 TO BLOCK 20 の場合ブロック 10 とブロック 20 の両方とも含まれます。</p> <p>ブロック・メディア・リカバリは、データ消失や破損がデータファイル全体ではなく少数のブロックに適用される場合に役立ちます。通常、ブロックの破損は <b>ADVISE FAILURE</b> コマンドまたはトレース・ファイル内のエラー・メッセージによってレポートされます。ブロック・レベルのデータ消失の原因は、次のとおりです。</p> <ul style="list-style-type: none"> <li>■ I/O エラーによる軽度のデータ消失</li> <li>■ ディスクに書き込まれるメモリーの破損</li> </ul> <p><i>recoverOptionList</i> からオプションを指定しない場合に、データベースでフラッシュバック・データベースが有効化されているときは、<b>RECOVER BLOCK</b> を使用して、最初にフラッシュバック・ログ、次にバックアップを検索し、リストアするブロックの正常なバージョンを見つけます。</p> <p>メディア破損マークが付いているブロックには、リカバリが完了するまでアクセスできません。</p> <p><b>注意：</b>個々のブロックについて実行できるのは、完全リカバリのみです。つまり、すべての REDO をブロックに適用する前に、リカバリを停止することはできません。</p>
TABLESPACE <i>tablespace_name</i> DBA <i>integer</i>	<p><b>関連項目：</b>3-14 ページの「<i>datafileSpec</i>」を参照してください。</p> <p>破損ブロックを含む表領域の名前または番号、および破損ブロックのデータ・ブロック・アドレス (DBA) を指定します。ブロック・メディア・リカバリの実行対象は、破損ブロックのみです。</p> <p><b>注意：</b>データファイルのヘッダー・ブロック（ブロック 1）はリカバリできません。</p>

**recoverOptionList**

この副次句は、各種のリカバリ・オプションを指定します。構文図は、2-189 ページの「`recoverOptionList::=`」を参照してください。

構文の要素	説明
ALLOW <i>integer</i> CORRUPTION	<p>リカバリ処理中に許容可能な破損ブロックの数を指定できます。REDO ログが破損した場合に備えて、このパラメータを設定できます。</p> <p>試行リカバリでこの句を使用する（TEST 句とともに使用する）場合は、<i>integer</i> は 1 より大きい値を指定できます。標準リカバリでこの句を使用すると、<i>integer</i> は 0 または 1 のみを指定できます。</p>
ARCHIVELOG TAG <i>tag_name</i>	<p>リカバリ中に使用するアーカイブ・ログ・バックアップ用のタグを指定します。タグ名には大 / 小文字区別がなく、すべて大文字で表示されます。リカバリに必要なすべてのアーカイブ REDO ログがタグ付きのバックアップに含まれていなければ、Recovery Manager は使用可能なバックアップから必要に応じてログまたは増分バックアップを使用します。</p>
AUXILIARY DESTINATION ' <i>location</i> '	<p>個々のファイルに対して別の位置が明示的に指定されていない場合は、TSPITR の実行中に作成される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 ログの位置を指定します。</p> <p>TSPITR で AUXILIARY DESTINATION を指定しない場合は、UNTIL 句を使用して RECOVER TABLESPACE を実行する前に、補助セット・データファイル、制御ファイルおよびオンライン REDO ログのそれぞれの名前を指定する必要があります。指定しない場合、TSPITR は失敗します。</p> <p><b>関連項目：</b>補助セットの宛先の詳細は、『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』の TSPITR に関する章を参照してください。</p>
CHECK LOGICAL	<p>物理的な破損チェックを通過したデータ・ブロックと索引ブロックについて、論理的な破損がないかどうかをテストします。たとえば、行ベースまたは索引エントリの破損がないかどうかを調べます。Recovery Manager は論理的な破損を発見すると、alert.log とサーバー・セッション・トレース・ファイルにそのブロックのログを書き込みます。</p> <p>SET MAXCORRUPT の設定値によって、ファイルに許容される物理的および論理的な破損の合計数が指定されます。デフォルトでは、MAXCORRUPT は 0 であるため、破損ブロックが存在する場合、メディア・リカバリは正常に実行されません。破損ブロックを含むリカバリを許容する場合は、MAXCORRUPT をメディア・リカバリが正常に実行されなくなる最小破損ブロック数に設定します。たとえば、1 つの破損ブロックを許容するには、MAXCORRUPT を 1 に設定します。</p> <p>あるファイルで検出された物理的な破損と論理的な破損の合計数が MAXCORRUPT の設定値以下の場合、Recovery Manager のコマンドは完了し、破損ブロック範囲が V\$DATABASE_BLOCK_CORRUPTION に移入されます。そうでない場合、このコマンドは、V\$DATABASE_BLOCK_CORRUPTION に移入を行わずに終了します。</p>
DELETE ARCHIVELOG	<p>不要になったバックアップまたはコピーからリストアされたアーカイブ・ログを削除します。Recovery Manager は、RESTORE コマンドの開始前にディスク上に存在していたアーカイブ・ログは削除しません。MAXSIZE を指定しないと、リストアされたアーカイブ・ログは適用後に削除されます。</p> <p><b>注意：</b>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、デフォルトで DELETE ARCHIVELOG オプションが使用可能になります。</p>
MAXSIZE <i>sizeSpec</i>	<p>リストアされたアーカイブ REDO ログには、<i>sizeSpec</i> 以内のディスク領域を使用してください。リカバリに MAXSIZE 値より大きいログのリストアが必要な場合は、MAXSIZE 値を増やす必要があることを示すエラーがレポートされます。MAXSIZE がログを含むバックアップ・セットより小さい場合、Recovery Manager はログを抽出するためにバックアップ・セットを複数回読み込む必要があります。この場合、Recovery Manager によって、MAXSIZE を増やす必要があることを示す警告が発行されます。</p>
EXCLUDE FLASHBACK LOG	<p>リストアするブロックを見つける場合にフラッシュバック・ログを検索しないように指定します。デフォルトでは、フラッシュバック・データベースが有効である場合に、Recovery Manager でフラッシュバック・ログを検索します。</p>

構文の要素	説明
FROM BACKUPSET	バックアップ・セットのみをリストアするように指定します。
FROM DATAFILECOPY	データファイル・イメージ・コピーのみをリストアするように指定します。
FROM TAG 'tag_name'	<p>リカバリ中に使用する増分バックアップ用のタグを指定します。リカバリに必要なすべての増分がタグ付きのバックアップに含まれていなければ、Recovery Manager は使用可能なバックアップから必要に応じてログまたは増分バックアップを使用します。タグ名には大 / 小文字区別がなく、すべて大文字で表示されます。</p> <p><b>関連項目:</b> 多重化バックアップ・セットの個別コピーにタグを適用する方法と、バックアップ・タグのデフォルトのファイル名形式については、2-19 ページの「<a href="#">BACKUP</a>」を参照してください。</p>
NOPARALLEL	メディア・リカバリをパラレルで実行しないように指定します。RECOVER のデフォルトはパラレル実行です (RECOVER ... PARALLEL オプションの説明を参照)。
NOREDO	<p>リカバリ中の REDO ログの適用を抑止します。増分バックアップのみを適用します。</p> <p>このオプションは、増分バックアップを使用して NOARCHIVELOG データベースの全体バックアップを更新する場合に使用できます (2-198 ページの例 2-100 を参照)。REDO ログが使用できない場合は、NOREDO オプションが必須です。NOARCHIVELOG モードで運用されているデータベースのリカバリ時に NOREDO を指定しない場合、Recovery Manager はリカバリを終了してエラー・メッセージを発行します。</p> <p><b>注意:</b> NOARCHIVELOG モードで運用されているデータベースの増分バックアップは、一貫性のある停止後のみ実行できます。</p> <p>また、スタンバイ・データベースまたは複製データベースを更新する場合も使用できます。BACKUP INCREMENTAL FROM SCN コマンドで作成した増分バックアップは、スタンバイ・データベースまたは複製データベースで適用できます。スタンバイ・データベースの手順については、『Oracle Data Guard 概要および管理』を参照してください。</p>
PARALLEL	<p>パラレル・リカバリ (デフォルト) を指定します。</p> <p>デフォルトでは、データベースでパラレルのメディア・リカバリを使用して、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させます。パラレル・リカバリのデフォルトの動作をオーバーライドするには、NOPARALLEL オプションを指定した RECOVER または RECOVER PARALLEL 0 を使用します。</p> <p>パラレル・メディア・リカバリでは、データベースは、ロールフォワード時に各種プロセスを様々なデータ・ブロックに割り当てる分業体制をとるため、操作がより効率的に行われます。プロセスの数は CPU_COUNT 初期化パラメータから導出されます。デフォルトでは、これはシステム上の CPU の数と等しくなります。たとえば、パラレル・リカバリを CPU_COUNT が 4 のシステム上で実行して、1 つのデータファイルのみがリカバリされた場合、生成された 4 つのプロセスがデータファイルからブロックを読み取り、REDO を適用します。</p> <p>通常、リカバリはデータ・ブロックに対する読取りと書込み時に I/O バウンドとなります。ブロック・レベルでパラレル化すると、リカバリにより I/O 合計が増加する場合に、非同期 I/O でのオペレーティング・システム制限をなくすことなどによって、リカバリのパフォーマンスが改善されます。効率的な非同期 I/O が行われるシステムでは、パラレル・メディア・リカバリを使用してもあまりメリットはありません。</p> <p><b>注意:</b> RECOVERY_PARALLELISM 初期化パラメータは、インスタンス・リカバリまたはクラッシュ・リカバリのみを制御します。メディア・リカバリは、RECOVERY_PARALLELISM に使用する値による影響は受けません。</p> <p><b>関連項目:</b> 『Oracle Database SQL 言語リファレンス』の CREATE TABLE の PARALLEL 句に関する説明を参照してください。</p>
<i>integer</i>	<p>並列度として <i>integer</i> を指定します。</p> <p>各パラレル・スレッドでは、1 つまたは 2 つのパラレル実行サーバーを使用します。通常、並列度は指定する必要がありません。</p>

構文の要素	説明
SKIP READONLY	リカバリから読取り専用ファイルを除外します。
TEST	<p>試行リカバリを開始します。</p> <p>試行リカバリは、標準リカバリの手順で問題が発生した場合に役立ちます。試行リカバリを使用すると、データベースで REDO の適用後を予測し、発生する可能性のある問題を検出できます。試行リカバリでは、標準リカバリと同じ方法で REDO を適用しますが、ディスクへの変更書込みは行わず、試行リカバリの終了時に変更をロールバックします。</p> <p><b>注意:</b>最後の RESETLOGS 操作以降に実行したバックアップをリストアする場合にのみ、この句を使用できます。それ以外の場合、データベースはエラーを戻します。</p>
UNDO TABLESPACE 'tablespace_name'	<p>ターゲット時刻に UNDO セグメントを持つ表領域のリストを指定します。RECOVER TABLESPACE でのみ使用可能です。</p> <p>TSPITR 中は、Recovery Manager では TSPITR ターゲット時刻に UNDO セグメントがあったのはどの表領域かという情報が必要です。通常、この情報は、リカバリ・カタログを使用している場合はそのリカバリ・カタログにあります。</p> <p>リカバリ・カタログがないか、リカバリ・カタログ内に情報がない場合、Recovery Manager では、ターゲット時刻に UNDO セグメントを持つ一連の表領域が現在 UNDO セグメントを持つ一連の表領域と同一であるとみなされます。この想定が正しくない場合、TSPITR は失敗してエラーが戻されます。この場合は、UNDO TABLESPACE を使用できます。</p>
VALIDATE HEADER	<p>Recovery Manager がリカバリに必要なファイルのリストアに使用できるバックアップをレポートして検証します（リストアは行いません）。</p> <p>RECOVER を VALIDATE HEADER とともに実行すると、Recovery Manager は、RESTORE ... PREVIEW オプションを指定した場合と同じ機能を実行します。ただし、Recovery Manager は、リストアおよびリカバリに必要なファイルのリストに加えて、バックアップ・ファイルのヘッダーを検証し、ファイルがディスク上にあるか、または Recovery Manager リポジトリのメタデータに対応するメディア管理カタログにあるかを判別します。</p> <p><b>関連項目:</b> RESTORE ... PREVIEW オプションの説明を参照してください。</p>

## 例

### 例 2-95 オープン状態のデータベースでの表領域のリカバリ

表領域 users のデータファイルを格納しているディスクが、ハードウェアのエラーが原因で使用できなくなり、数分後に修復されたとします。この例では、表領域 users をオフラインにし、自動チャネルを使用してデータファイルをデフォルト位置にリストアしてリカバリ（テープからリストアされたログを削除）してから、この表領域をオンラインに戻します。

```
SQL "ALTER TABLESPACE users OFFLINE IMMEDIATE";
RESTORE TABLESPACE users;
RECOVER TABLESPACE users DELETE ARCHIVELOG MAXSIZE 2M;
SQL "ALTER TABLESPACE users ONLINE";
```

### 例 2-96 リストアしたデータファイルの新しい位置へのリカバリ

この例では、事前構成済のディスク・チャネルを使用し、ディスク上のデータファイルのコピーおよびテープのバックアップを使用するために1つのメディア管理チャネルを手動で割り当て、表領域 users にあるデータファイルの1つを別の位置にリストアします。

```
RUN
{
  ALLOCATE CHANNEL ch1 DEVICE TYPE sbt;
  SQL "ALTER TABLESPACE users OFFLINE IMMEDIATE";
  SET NEWNAME FOR DATAFILE '/disk1/oradata/prod/users01.dbf'
  TO '/disk2/users01.dbf';
```



```

RESTORE TABLESPACE users;
SWITCH DATAFILE ALL;
RECOVER TABLESPACE users;
SQL "ALTER TABLESPACE users ONLINE";
}

```

### 例 2-97 バックアップ制御ファイルとリカバリ・カタログを使用した DBPITR の実行

すべてのデータファイル、制御ファイルおよびアーカイブ REDO ログ 58 が、ディスク障害により消失したと仮定します。また、増分バックアップを取っていないと仮定します。使用可能なアーカイブ REDO ログでデータベースをリカバリする必要があります。表領域 tools は、最後にバックアップが実行される前から読取り専用であったため、リストアする必要はありません。Recovery Manager をターゲット・データベースおよびリカバリ・カタログに接続した後は、次のコマンドを発行します。

```

STARTUP FORCE NOMOUNT;
RUN
{
  SET UNTIL SEQUENCE 40 THREAD 1; # Recover database until log sequence 40
  RESTORE CONTROLFILE;
  ALTER DATABASE MOUNT;
  RESTORE DATABASE SKIP TABLESPACE temp;
  RECOVER DATABASE SKIP TABLESPACE temp;
}
ALTER DATABASE OPEN RESETLOGS;

```

Recovery Manager は、データファイル 8（読取り専用表領域のデータファイル）のリストアおよびリカバリを自動的にスキップします。出力例の次の部分がスキップを示します。

```

using channel ORA_DISK_1
allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=104 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: Oracle Secure Backup

skipping datafile 8; already restored to file /disk1/oradata/prod/tools01.dbf
channel ORA_DISK_1: starting datafile backup set restore
.
.
.
Finished restore at 19-FEB-07

Starting recover at 19-FEB-07
using channel ORA_DISK_1
using channel ORA_SBT_TAPE_1
datafile 8 not processed because file is read-only

```

### 例 2-98 バックアップの増分更新

バックアップを増分更新することによって、データベースのイメージ・コピーの全体バックアップに伴うオーバーヘッドを避けると同時に、データベースのメディア・リカバリに必要な時間を最小限にすることもできます。この例では、先週中の任意の SCN までリカバリを行うことができますが、1 日分より多くの REDO を適用する必要がないようにできます。

次のスクリプトを毎日実行するとします。初回実行時には、スクリプトによって、指定したタグを使用してディスク上にデータベースのイメージ・コピーのバックアップが作成されます。2 回目から 7 回目の実行では、データベースのレベル 1 のバックアップが作成されます。8 回目以降の実行では、レベル 1 の増分が 7 日前に作成されたデータファイル・コピーに適用され、前日からの変更を含む新しいレベル 1 のバックアップが作成されます。

```

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;
}

```

### 例 2-99 スタンバイ・データベースで制御ファイルを失った場合のリカバリ

スタンバイ・データベース dgprod3 の制御ファイルが、メディア障害によって消失したとします。プライマリ・データベースおよびスタンバイ・データベースは、SBT ストレージを共有します。プライマリ・データベースの制御ファイルのバックアップはテープ上に存在します。

Recovery Manager クライアントを起動し、TARGET として dgprod3 に接続して、リカバリ・カタログに接続します。次の Recovery Manager コマンドは、スタンバイ・データベースで使用できる制御ファイルをリストアップし、ファイル名をディスク上の既存ファイルに更新して、スタンバイ・データベースのリカバリを行います。

```

RESTORE CONTROLFILE;
ALTER DATABASE MOUNT;
RECOVER DATABASE;

```

これで、スタンバイ・データベースへの REDO の適用を開始できます。

### 例 2-100 NOARCHIVELOG モードで運用されているデータベースのリカバリ

増分バックアップを適用することによって、NOARCHIVELOG モードで実行されているデータベースの変更に対して制限付きリカバリを実行できます。NOARCHIVELOG モードで実行されているデータベースのすべてのバックアップと同様、増分バックアップには一貫性が必要です。そのため、データベースがオープンされているときにバックアップすることはできません。

リカバリ・カタログを使用してデータベース prod を NOARCHIVELOG モードで実行するとします。日曜日の午後に、データベースを一貫した状態で停止し、テープにデータベース prod のレベル 0 のバックアップを作成します。水曜日と金曜日の午前 3 時に、データベースを一貫した状態で停止し、テープにレベル 1 の差分増分バックアップを作成します。

土曜日にデータベースでメディア障害が発生し、半分のデータファイルとオンライン REDO ログが破損します。オンライン・ログが消失しているため、RECOVER コマンドで NOREDO オプションを指定する必要があります。このオプションを指定しない場合、Recovery Manager は、金曜日の増分バックアップを適用した後で REDO ログを検索し、REDO ログが検出されない場合にはエラー・メッセージを発行します。

Recovery Manager を prod およびカタログ・データベースに接続した後、次のようにリカバリを実行します。

```

STARTUP FORCE NOMOUNT;
RESTORE CONTROLFILE; # restore control file from consistent backup
ALTER DATABASE MOUNT;
RESTORE DATABASE; # restore datafiles from consistent backup
RECOVER DATABASE NOREDO; # specify NOREDO because online redo logs are lost
ALTER DATABASE OPEN RESETLOGS;

```

リカバリしたデータベースには、金曜日の増分バックアップの時刻までの変更のみ反映されています。アーカイブ REDO ログが存在しないため、増分バックアップ後の変更をリカバリできません。

### 例 2-101 データベース内のすべての破損ブロックのリカバリ

この例では、バックアップの妥当性チェックを実行して V\$DATABASE\_BLOCK\_CORRUPTION ビューに移入し、このビューに記録された破損ブロックをリカバリします。両方のコマンドの出力例を示します。

```

RMAN> VALIDATE DATABASE;

Starting validate at 19-FEB-07
using channel ORA_DISK_1
channel ORA_DISK_1: starting validation of datafile
channel ORA_DISK_1: specifying datafile(s) for validation

```

```
.
.
.
List of Datafiles
=====
File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
1 FAILED 0 4070 57600 555975
File Name: /disk1/oradata/prod/system01.dbf
Block Type Blocks Failing Blocks Processed
-----
Data 1 41550
Index 0 7677
Other 0 4303
```

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

```
Starting recover at 19-FEB-07
using channel ORA_DISK_1
allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=104 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: Oracle Secure Backup
searching flashback logs for block images until SCN 547548
finished flashback log search, restored 1 blocks
```

```
starting media recovery
media recovery complete, elapsed time: 00:00:03
```

```
Finished recover at 19-FEB-07
```

---

## REGISTER DATABASE

---

### 用途

REGISTER DATABASE コマンドを使用すると、ターゲット・データベースのメタデータを Recovery Manager でメンテナンスできるように、ターゲット・データベースをリカバリ・カタログに登録できます。Recovery Manager によって、ターゲット・データベース自体からターゲット・データベースに登録するために必要なすべての情報が取得されます。

**関連項目：**『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』および 2-103 ページの「[CREATE CATALOG](#)」を参照してください。

### 前提条件

このコマンドは、Recovery Manager プロンプトでのみ実行してください。Recovery Manager は、リカバリ・カタログおよびマウント済またはオープン状態のターゲット・データベースに接続されている必要があります。現在リカバリ・カタログに登録されていないデータベースに登録する必要があります。

登録できるデータベースは、リカバリ・カタログ内で一意の DBID を持つターゲット・データベースのみです。データベースは、同じ名前でも DBID の値が異なれば登録できます。ただし、スタンバイ・データベースは登録できません。

### 使用上の注意

Recovery Manager では、スタンバイ・データベースのプライマリ・データベースがすでにリカバリ・カタログに登録されており、次のいずれかの条件が該当する場合、自動的に新しいスタンバイ・データベースがリカバリ・カタログに登録されます。

- Recovery Manager が、リカバリ・カタログで認識されない DB\_UNIQUE\_NAME を持つデータベース・インスタンスに接続されている場合。
- リカバリ・カタログで認識されない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。

重複する DBID が見つかり、REGISTER DATABASE コマンドは失敗します。この状況になるのは、データベースを作成するときに、[DUPLICATE](#) コマンドを使用せずに既存のデータベースからファイルをコピーしたときです。この障害が発生した場合は、DBNEWID ユーティリティを使用して、コピーしたデータベースの DBID を変更してから、REGISTER DATABASE コマンドを再試行できます。

RESETLOGS オプションを指定してデータベースをオープンし、その後にこのデータベースをリカバリ・カタログに登録すると、リカバリ・カタログでは、古いインカネーションの DB\_NAME が UNKNOWN として記録されます。これは、古いインカネーションが以前に登録されていないためです。これらのレコードは削除しないでください。

---

**注意：** 同じデータベース名と DBID を持つ異なるターゲット・データベースを Recovery Manager で使用する場合は、Recovery Manager の起動時に必ず正しいリカバリ・カタログのスキーマを指定するように注意してください。

---

**関連項目：** DBNEWID ユーティリティの使用方法は、『Oracle Database ユーティリティ』を参照してください。

## 構文

**register:=**

→ REGISTER DATABASE > ; →

## 例

### 例 2-102 データベースの登録

この例では、新しいターゲット・データベースをリカバリ・カタログに登録します。例には出力例も含まれます。

```
RMAN> CONNECT TARGET /

connected to target database: PROD (DBID=1619241818)

RMAN> CONNECT CATALOG rman@catdb

recovery catalog database Password: password
connected to recovery catalog database

RMAN> REGISTER DATABASE;

database registered in recovery catalog
starting full resync of recovery catalog
full resync complete
```

## RELEASE CHANNEL

### 用途

RELEASE CHANNEL コマンドを使用すると、ターゲット・データベース・インスタンスに接続した状態で、通常のチャンネルまたはメンテナンス・チャンネルを解放できます。通常のチャンネルは `ALLOCATE CHANNEL` を使用して割り当てられますが、メンテナンス・チャンネルは `ALLOCATE CHANNEL FOR MAINTENANCE` で割り当てられます。

### 前提条件

通常のチャンネルを解放するには、`release::=` 図で示される構文を使用します。この形式の RELEASE CHANNEL は RUN コマンド内でのみ実行し、`ALLOCATE CHANNEL` コマンドで使用した識別子と同じ識別子を付けてチャンネル名を指定してください。

メンテナンス・チャンネルを解放するには、`releaseForMaint::=` 図で示される構文を使用します。この形式の RELEASE CHANNEL は、Recovery Manager プロンプトでのみ実行し、RUN コマンド内で実行できません。

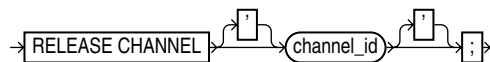
### 使用上の注意

メンテナンス・チャンネルは、RUN コマンド内で発行された `ALLOCATE CHANNEL` および RELEASE CHANNEL コマンドの影響を受けません。

RUN 内で RELEASE CHANNEL を使用してチャンネルを解放するのはオプションです。これは、Recovery Manager が RUN コマンドの終了時に通常のチャンネルをすべて自動的に解放するためです。

### 構文

**release::=**



**releaseForMaint::=**



### セマンティクス

構文の要素	説明
<code>channel_id</code>	<code>ALLOCATE CHANNEL</code> コマンド内で使用する、大 / 小文字の区別があるチャンネル ID を指定します (例 2-103 を参照)。

### 例

#### 例 2-103 RUN コマンドで割り当てられたチャンネルの解放

この例では、日次バックアップ用のテーブ・セットを示すパラメータを使用して `ch1` という SBT チャンネルを割り当て、データベースをバックアップしてから、このチャンネルを解放します。次に、週次バックアップ用のテーブ・セットのパラメータを使用して `ch1` という SBT チャンネルを割り当て、別のデータベース・バックアップを作成します。

```
RUN
{
  ALLOCATE CHANNEL ch1 DEVICE TYPE sbt
```

```

    PARMS='ENV=(OB_MEDIA_FAMILY=daily_bkp)';
BACKUP DATABASE;
RELEASE CHANNEL ch1;
ALLOCATE CHANNEL ch1 DEVICE TYPE sbt
    PARMS='ENV=(OB_MEDIA_FAMILY=weekly_bkp)';
BACKUP DATABASE;
}

```

RUN コマンドの最後に RELEASE CHANNEL コマンドは必要ありません。これは、Recovery Manager が自動的にチャンネル ch1 を解放するためです。

#### 例 2-104 メンテナンス・チャンネルの解放

この例では、Recovery Manager セッションの出力例が表示されます。SBT メンテナンス・チャンネルを割り当てた後に、テープ上のバックアップをクロスチェックし、削除します。SBT チャンネルを解放した後、Recovery Manager がデフォルトのディスク・チャンネルを使用してデータベースをバックアップします。

```

RMAN> ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt;

```

```

allocated channel: ORA_MAINT_SBT_TAPE_1
channel ORA_MAINT_SBT_TAPE_1: SID=105 device type=SBT_TAPE
channel ORA_MAINT_SBT_TAPE_1: Oracle Secure Backup

```

```

RMAN> CROSSCHECK BACKUP;

```

```

crosschecked backup piece: found to be 'AVAILABLE'
backup piece handle=ljiah8ln_1_1 RECID=25 STAMP=615031479
crosschecked backup piece: found to be 'AVAILABLE'
backup piece handle=lkiah8pk_1_1 RECID=26 STAMP=615031612
crosschecked backup piece: found to be 'AVAILABLE'
backup piece handle=lniah973_1_1 RECID=28 STAMP=615032036
Crosschecked 3 objects

```

```

RMAN> DELETE BACKUP;

```

```

List of Backup Pieces

```

BP Key	BS Key	Pc#	Cp#	Status	Device Type	Piece Name
1333	1331	1	1	AVAILABLE	SBT_TAPE	ljiah8ln_1_1
1334	1332	1	1	AVAILABLE	SBT_TAPE	lkiah8pk_1_1
1427	1423	1	1	AVAILABLE	SBT_TAPE	lniah973_1_1

```

Do you really want to delete the above objects (enter YES or NO)? YES

```

```

deleted backup piece
backup piece handle=ljiah8ln_1_1 RECID=25 STAMP=615031479
deleted backup piece
backup piece handle=lkiah8pk_1_1 RECID=26 STAMP=615031612
deleted backup piece
backup piece handle=lniah973_1_1 RECID=28 STAMP=615032036
Deleted 3 objects

```

```

RMAN> RELEASE CHANNEL;

```

```

released channel: ORA_MAINT_SBT_TAPE_1

```

```

RMAN> BACKUP DATABASE;

```

```

Starting backup at 20-FEB-07
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=105 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set

```

## REPAIR FAILURE

### 用途

REPAIR FAILURE コマンドを使用すると、データ・リカバリ・アドバイザーで識別されたデータベース障害を修復できます。

ワークフローは、LIST FAILURE を実行して障害を表示し、ADVISE FAILURE で修復オプションを表示して、REPAIR FAILURE で障害を修復することをお勧めします。

### 前提条件

ターゲット・データベースのインスタンスは起動されている必要があります。データベースは、単一インスタンス・データベースである必要があります。また、フィジカル・スタンバイ・データベースは指定できません。

最大で1つの Recovery Manager セッションで REPAIR FAILURE コマンドが実行されていることを確認します。REPAIR FAILURE ...PREVIEW だけは例外で、Recovery Manager の同時セッションで使用できます。

自動修復を実行するには、データ・リカバリ・アドバイザーで特定のバックアップおよびアーカイブ REDO ログが必要になる場合があります。リカバリに必要なファイルが使用できない場合は、リカバリを行うことはできません。

### 使用上の注意

修復は、単一修復で複数の障害が修復できる可能性がある場合に統合されます。REPAIR FAILURE は、現行のセッションで ADVISE FAILURE がまだ実行されていない場合にはこのコマンドを暗黙的に実行します。

Recovery Manager は、まだ障害に関連しているかどうかを常に検証し、修復済の障害を自動的にクローズします。また、すでに修復済の障害、および ADVISE FAILURE が実行されて新しい障害が報告がされたために不要となった障害を修復しようとすることはありません。

デフォルトでは、REPAIR FAILURE を使用すると、実行開始前に確認のプロンプトが表示されます。修復を実行した後で、既存の障害もすでに修復されている可能性があるため、Recovery Manager は既存のすべての障害を再評価します。

#### Oracle RAC およびデータ・リカバリ・アドバイザー

Oracle RAC データベースのすべてのインスタンスでデータ障害が発生した場合に、単一インスタンス・モードでデータベースをマウントしてデータ・リカバリ・アドバイザーを使用すると、制御ファイル、SYSTEM データファイルおよびディクショナリの障害を検出して修復できます。また、ヘルス・チェックを開始して、他のデータベース・コンポーネントにデータ障害がないかをテストすることもできます。この方法では、他のクラスタ・インスタンスに対してローカルなデータ障害（アクセス不能なデータファイルなど）は検出されません。

### 構文

**repair::=**





## セマンティクス

### repair

構文の要素	説明
REPAIR FAILURE	自動診断リポジトリに記録された障害を修復します。 REPAIR FAILURE を他のコマンド・オプションを指定しないで実行すると、Recovery Manager は現行のセッションで最後に実行した <b>ADVISE FAILURE</b> コマンドの最初の修復オプションを使用します。REPAIR FAILURE は、現行のセッションで ADVISE FAILURE がまだ実行されていない場合にはこのコマンドを暗黙的に実行します。
USING ADVISE OPTION <i>integer</i>	修復オプションは、オプション番号で指定します（障害番号では指定しません）。修復オプション番号は <b>ADVISE FAILURE</b> コマンドで取得できます。
NOPROMPT	確認のプロンプトを抑止します。 コマンド・ファイルの REPAIR FAILURE を実行する場合は、これがデフォルトのオプションです。
PREVIEW	修復は行わず、すべての修復アクションとコメントが記述されたスクリプトを生成します。デフォルトでは、スクリプトは標準出力に表示されます。 <b>SPOOL</b> 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（2-207 ページの <a href="#">例 2-106</a> を参照）。

## 例

### 例 2-105 障害の修復

この例では、データ・リカバリ・アドバイザーが認識できるすべての障害を修復します。この例では、2つの障害（データファイルの欠落、および破損ブロックのあるデータファイル）を修復します。リカバリ後、データベースをオープンするかどうかを確認するメッセージが表示されます（ユーザー入力のテキストは太字で示しています）。

```
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 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

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
```

### 例 2-106 修復のプレビュー

次の例では、現行のセッションで最後に実行した **ADVISE FAILURE** コマンドの最初の修復オプションによる修復をプレビューします。この例では、**LIST FAILURE** および **ADVISE FAILURE** の各コマンドの出力例は示されていません。

```
RMAN> LIST FAILURE;
.
.
.
RMAN> ADVISE FAILURE;
.
.
.
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_3200987003.hm
```

```
contents of repair script:
  # block media recovery
  recover datafile 1 block 56416;
```

**SPOOL** と **REPAIR FAILURE ...PREVIEW** を併用すると、修復スクリプトをファイルに書き込むことができます。その後、このスクリプトを編集して手動で実行できます。次の例では、修復プレビューのログを `/tmp/repaircmd.dat` にスプールします。

```
RMAN> SPOOL LOG TO '/tmp/repaircmd.dat';
RMAN> REPAIR FAILURE PREVIEW;
RMAN> SPOOL LOG OFF;
```

## REPLACE SCRIPT

### 用途

REPLACE SCRIPT コマンドを使用すると、リカバリ・カタログに格納されている既存のスクリプトを置換できます。既存のスクリプトがない場合は、REPLACE SCRIPT によりスクリプトが作成されます。

**関連項目：** ストアド・スクリプトの作成方法は、2-106 ページの「CREATE SCRIPT」を参照してください。

### 前提条件

REPLACE SCRIPT は、Recovery Manager プロンプトでのみ実行できます。Recovery Manager が、ターゲット・データベースとリカバリ・カタログに接続されている必要があります。リカバリ・カタログ・データベースはオープンの状態である必要があります。

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と同じターゲット・データベースに接続する必要があります。

#### ストアド・スクリプトでの置換変数

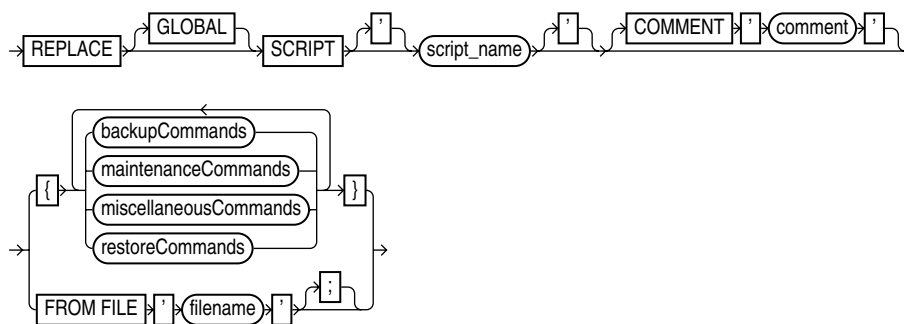
Recovery Manager では、ストアド・スクリプトで置換変数を使用できます。&1 は最初の値を配置する位置を示し、&2 は 2 番目の値を配置する位置を示します。その後も同様に続きます。特殊文字は、引用符で囲む必要があります。

置換変数の構文は、&integer の後にオプションでピリオドが続きます (&1.3 など)。オプションのピリオドは変数の一部であり、値と置換されますので、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、置換変数 &1.3 が含まれているコマンド・ファイルに値 mybackup を渡すと、その置換結果は mybackup3 になります。mybackup.3 という結果を得るには、構文 &1..3 を使用します。

置換変数を使用するストアド・スクリプトを作成する場合は、作成時に例となる値を指定する必要があります。これらの値は、Recovery Manager の起動時に USING 句で指定するか (2-244 ページの「RMAN」を参照)、またはプロンプトが表示されたときに入力します (2-108 ページの例 2-60 を参照)。

### 構文

**replaceScript::=**



(**backupCommands::=** (2-250 ページ)、**maintenanceCommands::=** (2-250 ページ)、**miscellaneousCommands::=** (2-251 ページ)、**restoreCommands::=** (2-251 ページ))

## セマンティクス

構文の要素	説明
GLOBAL	<p>スクリプトをグローバルとして指定します。</p> <p><b>注意:</b> 仮想プライベート・カタログは、グローバル・スクリプトに対して読取り専用のアクセスが可能です。グローバル・スクリプトの作成と更新は、Recovery Manager が基本リカバリ・カタログに接続している間に行う必要があります。</p> <p><b>関連項目:</b> グローバル・スクリプトとローカル・スクリプトの違いについては、2-106 ページの「<a href="#">使用上の注意</a>」を参照してください。</p>
SCRIPT <i>script_name</i>	置換するローカル・スクリプトまたはグローバル・スクリプトを指定します。
COMMENT ' <i>comment</i> '	説明のコメントをリカバリ・カタログ内のストアド・スクリプトと関連付けます。コメントは、LIST SCRIPT NAMES の出力で示されます。
<i>backupCommands</i>	<p>ストアド・スクリプトに含めるコマンドを指定します。REPLACE SCRIPT '<i>script_name</i>' {...} コマンドの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、いずれもストアド・スクリプトで使用できます。コマンド RUN、@ および @@ は、ストアド・スクリプト内では使用できません。</p>
<i>maintenanceCommands</i>	
<i>miscellaneousCommands</i>	
<i>restoreCommands</i>	
FROM FILE ' <i>filename</i> '	<p>指定したファイルからスクリプトを定義する一連のコマンドを読み取ります。</p> <p>このファイルは、有効なストアド・スクリプトの本体と同様である必要があります。このファイルの最初の行は、左カッコ ( { ) である必要があります。また、最終行には、右カッコ ( } ) が含まれている必要があります。このファイルの Recovery Manager コマンドは、ストアド・スクリプトで有効である必要があります。</p>

## 例

### 例 2-107 リカバリ・カタログ・スクリプトの置換

Recovery Manager クライアントを起動し、TARGET としてデータベース prod に接続して、リカバリ・カタログに接続します。CREATE SCRIPT を使用して、次のように backup\_db という名前のグローバル・スクリプトを作成します。

```
CREATE GLOBAL SCRIPT backup_db
COMMENT "back up any database from the recovery catalog, with logs"
{
    BACKUP DATABASE;
}
```

次に、LIST SCRIPT NAMES を使用して、リカバリ・カタログで認識されているすべてのスクリプトをリストします。

```
RMAN> LIST SCRIPT NAMES;
```

List of Stored Scripts in Recovery Catalog

Global Scripts

```
Script Name
Description
-----
backup_db
back up any database from the recovery catalog, with logs
```

次に、`backup_db` グローバル・スクリプトを編集するために、次の `REPLACE SCRIPT` コマンドを実行します。

```
RMAN> REPLACE SCRIPT backup_db { BACKUP DATABASE PLUS ARCHIVELOG; }
```

replaced script backup\_db

GLOBAL キーワードを指定していないため、Recovery Manager が `backup_db` という名前のグローバル・スクリプトに加えて、`backup_db` という名前のローカル・スクリプトを作成します。`LIST SCRIPT NAMES` は、リカバリ・カタログに記録されているグローバル・スクリプトとローカル・スクリプトを両方とも示します。

```
RMAN> LIST SCRIPT NAMES;
```

List of Stored Scripts in Recovery Catalog

Scripts of Target Database PROD

```
Script Name
Description
-----
```

```
backup_db
```

Global Scripts

```
Script Name
Description
-----
```

```
backup_db
back up any database from the recovery catalog, with logs
```

次に、`DELETE SCRIPT` を使用して `backup_db` という名前のローカル・スクリプトを削除し、次のようにグローバル・スクリプトを置換します。

```
RMAN> DELETE SCRIPT backup_db;
```

deleted script: backup\_db

```
RMAN> REPLACE GLOBAL SCRIPT backup_db { BACKUP DATABASE PLUS ARCHIVELOG; }
```

replaced global script backup\_db

ここでの `LIST SCRIPT NAMES` コマンドは、カタログに `backup_db` という名前のスクリプトのみが存在していることを示します。

```
RMAN> LIST SCRIPT NAMES;
```

List of Stored Scripts in Recovery Catalog

Global Scripts

```
Script Name
Description
-----
```

```
backup_db
```

`PRINT SCRIPT` コマンドを実行すると、グローバル・スクリプトの変更が確認されます。

```
RMAN> PRINT GLOBAL SCRIPT backup_db;
```

```
printing stored global script: backup_db
{ BACKUP DATABASE PLUS ARCHIVELOG; }
```

## REPORT

### 用途

REPORT コマンドを使用すると、Recovery Manager リポジトリの詳細分析を実行できます。Recovery Manager は、レポートを標準出力またはメッセージ・ログ・ファイルに書き出します。

**関連項目：** Recovery Manager レポートの作成方法は、『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』を参照してください。

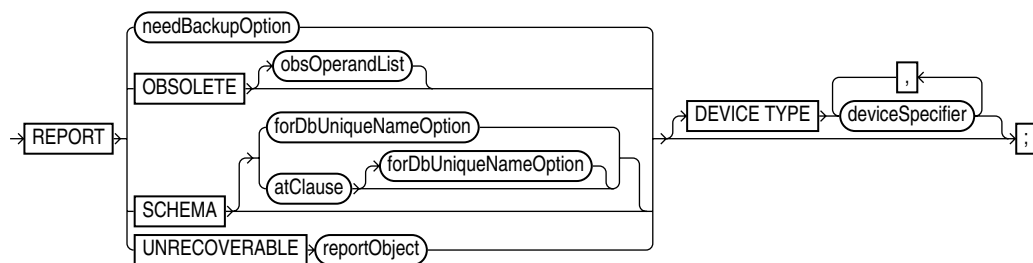
### 前提条件

このコマンドは、Recovery Manager プロンプトでのみ実行してください。次のいずれかの条件が満たされている必要があります。

- Recovery Manager は、ターゲット・データベースに接続されている必要があります。
- Recovery Manager はリカバリ・カタログに接続され、SET DBID が実行されている必要があります。

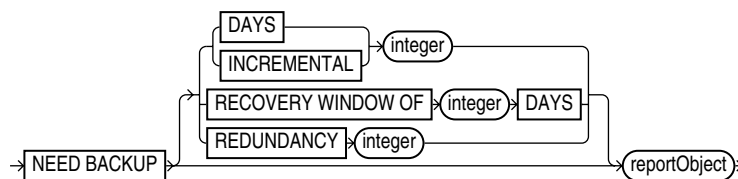
### 構文

report::=

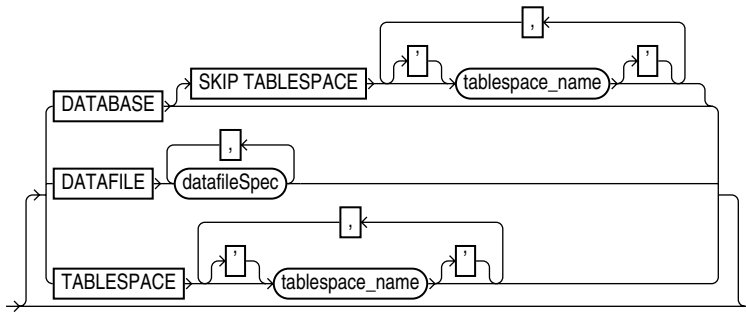


(needBackupOption::= (2-211 ページ)、atClause::= (2-212 ページ)、  
reportObject::= (2-212 ページ)、obsOperandList::= (3-34 ページ)、  
deviceSpecifier::= (3-15 ページ))

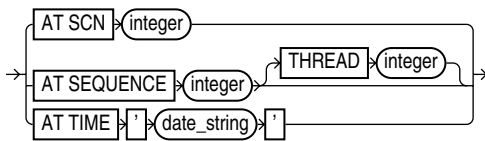
needBackupOption::=



(reportObject::=2-212 ページ)

**reportObject::=**

(datafileSpec::= (3-14 ページ))

**atClause::=****セマンティクス****report**

この句では、レポートのタイプを指定します。

構文の要素	説明
<i>needBackupOption</i>	バックアップの必要なファイルをリストします。 <b>関連項目</b> : 2-213 ページの「 <i>needBackupOption</i> 」を参照してください。
OBSOLETE <i>obsOperandList</i>	Recovery Manager リポジトリに記録され、不要になったために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 ログをリストします。出力の説明は、表 2-33 を参照してください。このコマンドは、次の 2 つの手順で使用します。 <ol style="list-style-type: none"> <li>バックアップされているデータファイルごとに、Recovery Manager では、保存方針に従って不要とされていない最も古い全体バックアップ、レベル 0 のバックアップまたはイメージ・コピーが識別されます。この手順で識別されたバックアップまたはイメージ・コピーよりも古いデータファイルのバックアップは、すべて不要とみなされます。</li> <li>不要ではない最も古い全体バックアップよりも古いアーカイブ REDO ログおよびレベル 1 の増分バックアップは、すべて不要とみなされます。これらのファイルは、適用可能な全体バックアップまたはレベル 0 のバックアップが存在しないため不要となります。レベル 1 の増分バックアップまたはアーカイブ REDO ログは、不要とされていないレベル 0 のバックアップや全体バックアップに適用可能であれば、不要とみなされません。</li> </ol> <p>副次句 <i>obsOperandList</i> には、Recovery Manager で不要と判断するために使用する条件を記述します。 <i>obsOperandList</i> でパラメータを指定しなければ、CONFIGURE RETENTION POLICY で指定したオプションが使用されます (2-217 ページの例 2-110 を参照)。このオプションを DEVICE TYPE と併用すると、Recovery Manager では指定したデバイスで作成されたバックアップとコピーのみが考慮されます。保存方針が無効化されている場合、Recovery Manager はどのバックアップも不要とみなしません。したがって、他のオプションを指定しないで REPORT OBSOLETE を実行し、保存方針が NONE に指定されている場合に、Recovery Manager はエラーを発行します。</p> <p><b>注意</b> : KEEP UNTIL TIME 句を使用して作成したバックアップは、構成済の保存方針の設定に関係なく、KEEP 時間を過ぎると不要とされます。</p>



構文の要素	説明
SCHEMA	<p>指定時点でのターゲット・データベースのすべてのデータファイル（永続および一時）と表領域の名前をリストします。出力の説明は、表 2-28 を参照してください。</p> <p><code>forDbUniqueNameOption</code> が指定されていない REPORT SCHEMA の場合、ターゲット・データベース接続は必須ですが、リカバリ・カタログ接続はオプションです。</p>
<code>forDbUniqueNameOption</code>	<p>DB_UNIQUE_NAME で指定したデータベースのすべてのデータファイルと表領域の名前をレポートします。</p> <p>データベースは <code>db_unique_name</code> で指定できますが、ALL を使用すると、特定の DBID についてカタログに記録されている一意の名前のデータベースをすべて指定できます。リカバリ・カタログ内のデータベースは、DBID および DB_UNIQUE_NAME 初期化パラメータの値によって一意に識別されます。</p> <p>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。Recovery Manager がターゲット・データベースに接続され、SET DBID が実行されている必要があります。</p> <p><b>関連項目：</b>この句のオプションについては、3-18 ページの「<code>forDbUniqueNameOption</code>」を参照してください。</p>
<code>atClause</code>	<p>SCN、ログ順序番号または時間を指定します。</p>
UNRECOVERABLE <code>reportObject</code>	<p>リカバリ不能なすべてのデータファイルをリストします。出力の説明は、2-216 ページの表 2-34 を参照してください。</p> <p>最後のバックアップ以降に、データファイル内のオブジェクトに対して UNRECOVERABLE 操作が実行されていると、そのデータファイルはリカバリ不能とみなされます。リカバリ不能操作では、REDO は生成されません。例は、表データのダイレクト・ロードと NOLOGGING オプションを使用した更新です。</p> <p><b>注意：</b>データファイルのバックアップのいずれかが存在しないということのみでは、リカバリ不能とみなされる理由にはなりません。このようなデータファイルは、ファイルの作成時以降の REDO ログが存在していれば、CREATE DATAFILE コマンドを使用してリカバリできます。</p>
DEVICE TYPE <code>deviceSpecifier</code>	<p>ストレージ・デバイスのタイプを指定します。Recovery Manager は、レポート用に指定したデバイスに存在しているバックアップとコピーのみを使用可能とみなします。</p>

### **needBackupOption**

この句は、バックアップの必要なファイルに関してのみレポートを行います。

構文の要素	説明
NEED BACKUP	<p>新しいバックアップが必要で、指定した <code>reportObject</code> にあるすべてのデータファイルをリストします。</p> <p>レポートは、最新のバックアップをリストアップすることを前提としています。オプションを指定しなければ、Recovery Manager は現行の保存方針の構成を使用します。保存方針が無効化されている場合（CONFIGURE RETENTION POLICY TO NONE）、Recovery Manager はエラーを生成します。</p>

構文の要素	説明
DAYS <i>integer</i>	<p>完全なリカバリのために、指定した日数より多くのアーカイブ REDO ログ・ファイルが必要とするすべてのデータファイルをリストします。たとえば、REPORT NEED BACKUP DAYS 7 DATABASE では、リカバリに 7 日間分より多くのアーカイブ REDO ログが必要なデータファイルが表示されます。出力の説明は、表 2-29 を参照してください。</p> <p>ターゲット・データベースの制御ファイルがマウントされている現行ファイルである場合、Recovery Manager はこのレポートに次のような最適化を行います。</p> <ul style="list-style-type: none"> <li>■ オフラインであって、最新のバックアップにすべての変更内容が保存されているファイルは含まれません。</li> <li>■ 以前はオフラインであったが、現在はオンラインであり、最新のバックアップにオフラインのときまでのすべての変更内容が保存されているファイルは、オンラインの期間が指定した日数を超過している場合にかぎり、レポートに含まれます。</li> </ul>
INCREMENTAL <i>integer</i>	<p>リカバリに必要な増分バックアップのしきい値を指定します (2-217 ページの例 2-109 を参照)。<i>integer</i> で指定した数より多くの増分バックアップがデータファイルの完全リカバリに必要な場合、データファイルには新規の全体バックアップが必要になります。出力の説明は、表 2-30 を参照してください。</p> <p><b>注意:</b> バックアップが存在しないファイルはこのリストに示されていません。それらのファイルを表示するには、REPORT NEED BACKUP REDUNDANCY コマンドを発行します。</p>
RECOVERY WINDOW OF <i>integer</i> DAYS	<p>指定した日数の期間、リカバリ期間ベースの保存方針を満たすための十分なバックアップがないデータファイルをレポートします。つまり、SYSDATE - <i>integer</i> の任意の時点まで Point-in-Time リカバリを行うための十分なバックアップがないデータファイルです。出力の説明は、表 2-31 を参照してください。</p>
REDUNDANCY <i>integer</i>	<p>データファイルをバックアップが必要ない範囲にあるとみなすために必要なバックアップまたはコピーの最小数を指定します。つまり、データファイルのバックアップまたはコピーが <i>integer</i> より少ない場合、このファイルにはバックアップが必要です。たとえば、REDUNDANCY 2 は、データファイルのバックアップまたはコピーが 2 つ未満の場合は、新しいバックアップが必要であることを意味します。出力の説明は、表 2-32 を参照してください。</p>
<i>reportObject</i>	<p>レポートを生成するオブジェクトを指定します。</p>

### **reportObject**

この副次句では、レポートに含めるデータファイルを指定します。レポートには、データベース全体（必要に応じて、特定の表領域をスキップ）または表領域のリスト、データファイルのリストを含めることができます。Recovery Manager には以前のインカネーションからのオブジェクトが含まれます。

構文の要素	説明
DATABASE	<p>データベースにある全ファイルのバックアップまたはデータファイルのコピーをリストします。</p> <p><b>注意:</b> DATABASE 指定から特定の表領域を除外できるように、SKIP TABLESPACE <i>tablespace_name</i> を指定します。</p>
DATAFILE <i>datafileSpec</i>	<p>指定したデータファイルをリストします。Recovery Manager は、指定したデータファイルを少なくとも 1 つ含むバックアップまたはデータファイルのコピーについてレポートを作成します。</p>
TABLESPACE <i>tablespace_name</i>	<p>指定した表領域にあるデータファイルをリストします。Recovery Manager は、指定した表領域にあるデータファイルを少なくとも 1 つ含むバックアップまたはデータファイルのコピーについてレポートを作成します。</p>

**atClause**

この副次句では、時刻、SCN またはログ順序番号で特定の時点を指定します。AT 句を使用して REPORT SCHEMA コマンドを発行するときは、リカバリ・カタログに接続している必要があります。

構文の要素	説明
AT SCN <i>integer</i>	SCN を指定します。
AT SEQUENCE <i>integer</i>	ログ順序番号を指定します。この整数は、指定したログが最初にオープンされた時刻を示します。
THREAD <i>integer</i>	REDO THREAD 番号を指定します。この整数は、スレッドが最初にオープンされた時刻を示します。
AT TIME ' <i>date_string</i> '	日付を指定します (2-217 ページの例 2-108 を参照)。NLS_LANG および NLS_DATE_FORMAT 環境変数で時刻の書式を指定します。

**レポート出力**

出力で表示される情報については、次の表を参照してください。

- 表 2-28 「データベース・スキーマのレポート」
- 表 2-29 「*n* 日分より多くのアーカイブ・ログがリカバリ用に必要なファイルのレポート」
- 表 2-30 「*n* より多くの増分がリカバリ中に必要なファイルのレポート」
- 表 2-31 「*n* 日のリカバリ期間を満たすようにバックアップを行う必要があるファイルのレポート」
- 表 2-32 「*n* 未満の冗長度のバックアップしかないファイルのレポート」
- 表 2-33 「不要なバックアップとコピーのレポート」
- 表 2-34 「リカバリ不能操作が原因でバックアップが必要となったファイルのレポート」

**表 2-28 データベース・スキーマのレポート**

列	指定対象
File	絶対データファイル番号。
Size (MB)	MB 単位のファイル・サイズ。
Tablespace	表領域名。
RB segs	データファイル専用。表領域にロールバック・セグメントがある場合は YES、ない場合は NO です (リカバリ・カタログに接続されている場合のみ)。Recovery Manager をカタログに接続していない場合、*** が表示されます。
Datafile Name	永続データファイル専用。データファイルのファイル名。
Maxsize (MB)	一時ファイル専用。一時ファイルの最大サイズ。
Tempfile Name	一時ファイル専用。一時ファイルのファイル名。

**表 2-29 *n* 日分より多くのアーカイブ・ログがリカバリ用に必要なファイルのレポート**

列	指定対象
File	<i>n</i> 日分より多くのアーカイブ REDO ログがリカバリ用に必要であるデータファイルの絶対ファイル番号。
Days	リカバリ用に必要なアーカイブ REDO データの日数。
Name	データファイルの名前。

表 2-30  $n$  より多くの増分がリカバリ中に必要なファイルのレポート

列	指定対象
File	$n$ より多くの増分が完全リカバリのために必要なデータファイルの絶対ファイル番号。
Incrementals	完全リカバリのために必要な増分バックアップの数。
Name	データファイルの名前。

表 2-31  $n$  日のリカバリ期間を満たすようにバックアップを行う必要があるファイルのレポート

列	指定対象
File	$n$ 日のリカバリ期間を満たすようにバックアップを行う必要があるデータファイルの絶対ファイル番号。
Days	完全リカバリのために必要な日数。
Name	バックアップが必要なデータファイルの名前。

表 2-32  $n$  未満の冗長さのバックアップしかないファイルのレポート

列	指定対象
File	冗長さ $n$ 未満のバックアップしかないデータファイルの絶対データファイル番号。
#bkps	このファイルに対して存在しているバックアップの数。
Name	ファイルの名前。

表 2-33 不要なバックアップとコピーのレポート

列	指定対象
Type	オブジェクトがバックアップ・セットか、バックアップ・ピースか、プロキシ・コピーかまたはデータファイルのコピーかの区分。
Key	ターゲット・データベースの制御ファイルでこのバックアップを識別する一意のキー。
Completion Time	バックアップまたはコピーを取った時刻。
Filename/handle	バックアップまたはデータファイルのコピーのファイル名またはメディア・ハンドル。

表 2-34 リカバリ不能操作が原因でバックアップが必要となったファイルのレポート

列	指定対象
File	リカバリ不能操作のために新しいバックアップが必要となったデータファイルの絶対番号。
Type Of Backup Required	FULL または INCREMENTAL。このファイルの全データのリカバリ能力を保証するために必要なバックアップのタイプによって決まります。FULL の場合は、全体バックアップ、レベル 0 のバックアップまたはデータファイルのコピーを作成します。INCREMENTAL の場合は、全体または増分バックアップです。
Name	データファイルの名前。

## 例

**例 2-108 データベース・スキーマのレポート**

リカバリ・カタログが必要なこの例では、20 分前のすべてのデータファイルの名前と表領域をレポートします。

```
RMAN> REPORT SCHEMA AT TIME 'sysdate-20/1440';
```

Report of database schema for database with db\_unique\_name PROD

## List of Permanent Datafiles

```
=====
```

File	Size(MB)	Tablespace	RB segs	Datafile Name
1	450	SYSTEM	YES	/disk1/oradata/prod/system01.dbf
2	197	SYSaux	YES	/disk1/oradata/prod/sysaux01.dbf
3	20	UNDOTBS	YES	/disk1/oradata/prod/undotbs01.dbf
4	10	CWMLITE	YES	/disk1/oradata/prod/cwmlite01.dbf
5	10	DRSYS	YES	/disk1/oradata/prod/drsys01.dbf
6	10	EXAMPLE	YES	/disk1/oradata/prod/example01.dbf
7	10	INDX	YES	/disk1/oradata/prod/indx01.dbf
8	10	TOOLS	YES	/disk1/oradata/prod/tools01.dbf
9	10	USERS	YES	/disk1/oradata/prod/users01.dbf

## List of Temporary Files

```
=====
```

File	Size(MB)	Tablespace	Maxsize (MB)	Tempfile Name
1	40	TEMP	32767	/disk1/oradata/prod/temp01.dbf

**例 2-109 増分バックアップが必要なデータファイルのレポート**

この例では、データベース内のデータファイルのうち、現行の状態にリカバリするために1つ以上の増分バックアップの適用が必要なすべてのデータファイルをレポートします。

```
RMAN> REPORT NEED BACKUP INCREMENTAL 1;
```

Report of files that need more than 1 incrementals during recovery

## File Incrementals Name

```
-----
```

1	2	/disk1/oradata/prod/system01.dbf
2	2	/disk1/oradata/prod/sysaux01.dbf
3	2	/disk1/oradata/prod/undotbs01.dbf
4	2	/disk1/oradata/prod/cwmlite01.dbf
5	2	/disk1/oradata/prod/drsys01.dbf
6	2	/disk1/oradata/prod/example01.dbf
7	2	/disk1/oradata/prod/indx01.dbf
9	2	/disk1/oradata/prod/users01.dbf

**例 2-110 不要なバックアップとコピーのレポート**

次の例では、現在の保存方針に従って冗長とみなされる不要なバックアップとコピーをレポートします。保存方針は冗長度 1 に設定されています。

```
RMAN> REPORT OBSOLETE;
```

RMAN retention policy will be applied to the command

RMAN retention policy is set to redundancy 1

Report of obsolete backups and copies

Type	Key	Completion Time	Filename/Handle
Archive Log	1022	19-FEB-07	/disk1/prod/arch/archive1_59_614712405.dbf
Archive Log	1023	19-FEB-07	/disk1/prod/arch/archive1_61_614712405.dbf
Archive Log	1024	19-FEB-07	/disk1/prod/arch/archive1_60_614712405.dbf
Archive Log	1025	19-FEB-07	/disk1/prod/arch/archive1_55_614712405.dbf

Backup Set	1032	19-FEB-07	
Backup Piece	1050	19-FEB-07	
			/disk2/PROD/backupset/2007_02_19/o1_mf_nnndf_TAG20070216T173839_2xnpmp01_.bkp
Datafile Copy	1073	19-FEB-07	
			/disk2/PROD/datafile/o1_mf_system_2xmz515m_.dbf
Backup Set	1035	19-FEB-07	
Backup Piece	1053	19-FEB-07	
			/disk2/PROD/backupset/2007_02_19/o1_mf_nnndf_TAG20070219T111434_2xmpozym_.bkp
Datafile Copy	1074	19-FEB-07	
			/disk2/PROD/datafile/o1_mf_sysaux_2xmz6zdg_.dbf
Datafile Copy	1075	19-FEB-07	
			/disk2/PROD/datafile/o1_mf_undotbs_2xmz7rof_.dbf
Datafile Copy	1076	19-FEB-07	
			/disk2/PROD/datafile/o1_mf_cwmlite_2xmz7vrg_.dbf
Datafile Copy	1077	19-FEB-07	/disk2/PROD/datafile/o1_mf_drsys_2xmz7wyc_.dbf
Datafile Copy	1078	19-FEB-07	
			/disk2/PROD/datafile/o1_mf_example_2xmz7y5s_.dbf
Datafile Copy	1079	19-FEB-07	/disk2/PROD/datafile/o1_mf_indx_2xmz81jg_.dbf
Datafile Copy	1081	19-FEB-07	/disk2/PROD/datafile/o1_mf_users_2xmz85vo_.dbf
Datafile Copy	1777	20-FEB-07	/disk2/users01.dbf

## RESET DATABASE

### 用途

RESET DATABASE TO INCARNATION コマンドを使用すると、Recovery Manager リポジトリのターゲット・データベースのインカネーションを以前のデータベース・インカネーションに再設定できます。このコマンドは次のような状況でのみ使用する必要があります。

- **RESTORE** または **RECOVER** を使用して、データベースを現行の RESETLOGS タイムスタンプより前の SCN に戻す場合。
- **FLASHBACK DATABASE** を使用して、データベースを孤立したデータベース・インカネーションに戻す場合。

**関連項目：** RESET DATABASE コマンドを使用する必要がある状況については、『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』を参照してください。

### 前提条件

RESET DATABASE TO INCARNATION は、Recovery Manager プロンプトでのみ実行できます。Recovery Manager は、ターゲット・データベースに接続している必要があります。

Recovery Manager が NOCATALOG モードで実行されている場合は、ターゲット・データベースをマウントする必要があります。マウントされる制御ファイルには、指定したデータベース・インカネーションのレコードが含まれている必要があります。

Recovery Manager が CATALOG モードで実行されている場合は、ターゲット・データベースのマウントまたはアンマウントが可能です。データベースをマウントする場合は、指定したデータベース・インカネーションのレコードが制御ファイルに含まれている必要があります。

### 使用上の注意

Recovery Manager を NOCATALOG モードで使用すると、RESET DATABASE TO INCARNATION コマンドは Recovery Manager セッション全体で永続的になります。

### 構文

**reset::=**

```
→ RESET DATABASE TO INCARNATION (primaryKey) ;
```

### セマンティクス

構文の要素	説明
<code>primary_key</code>	<p>現行のインカネーションを、データベース・インカネーションの DBINC レコードの主キーで指定した現行以外のインカネーションに変更します。インカネーション・キーは、REDO ストリームを一意にタグ付けしたり、識別するために使用します。</p> <p>可能なキー値を取得するには、<b>LIST INCARNATION OF DATABASE</b> を実行します。RESET DATABASE TO INCARNATION を発行した後は、<b>FLASHBACK DATABASE</b>、<b>RESTORE</b>、<b>RECOVER</b> などの Recovery Manager コマンドを実行できます。</p>

## 例

**例 2-111 NOCATALOG モードでの以前のインカネーションへの Recovery Manager の再設定**

NOCATALOG モードでは、リカバリするインカネーションに関する情報が含まれている制御ファイルをマウントする必要があります。次の使用例では、データベース trgt の破棄されたインカネーションにデータベースを再設定し、不完全リカバリを実行します。

```
CONNECT TARGET / NOCATALOG
```

```
# step 1: start and mount a control file that knows about the incarnation to which
# you want to return. if the current control file does not know about it, then
# you must restore an older control file
STARTUP NOMOUNT;
RESTORE CONTROLFILE UNTIL TIME 'SYSDATE-250';
ALTER DATABASE MOUNT;
```

```
# step 2: obtain the primary key of old incarnation
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    1334358386    PARENT  154381    OCT 30 2007 16:02:12
1       116     TRGT    1334358386    CURRENT 154877    OCT 30 2007 16:37:39
```

```
# step 3: in this example, reset database to incarnation key 2
RESET DATABASE TO INCARNATION 2;
```

```
# step 4: restore and recover the database to a point before the RESETLOGS
RESTORE DATABASE UNTIL SCN 154876;
RECOVER DATABASE UNTIL SCN 154876;
```

```
# step 5: make this incarnation the current incarnation and then list incarnations:
ALTER DATABASE OPEN RESETLOGS;
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    1334358386    PARENT  154381    OCT 30 2007 16:02:12
1       116     TRGT    1334358386    PARENT  154877    OCT 30 2007 16:37:39
1       311     TRGT    1334358386    CURRENT 154877    AUG 13 2007 17:17:03
```



## RESTORE

### 用途

RESTORE コマンドを使用すると、**Recovery Manager** バックアップのリストア、検証またはプレビューを実行できます。通常、バックアップのリストアは、メディア障害によって現行のデータファイル、制御ファイルまたはアーカイブ REDO ログが破損したとき、あるいは Point-in-Time リカバリの実行前に行います。

### 前提条件

データファイルを現行の位置にリストアするには、リストアする表領域またはデータファイルをオフラインにしてデータベースを起動、マウントまたはオープンする必要があります。

Data Guard 環境で **Recovery Manager** を使用する場合は、**Recovery Manager** がリカバリ・カタログに接続されている必要があります。

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のリストアを行う前に、次のいずれかの操作を実行します。

- 本番データベースが使用しているリカバリ領域とは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。
- 本番データベースが使用す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違う名前に設定します。

前述の操作をどちらも実行しないと、**Recovery Manager** では、本番データベースをリストアしていると判断し、フラッシュ・リカバリ領域のフラッシュバック・ログを使用不可能とみなして削除します。

### 使用上の注意

RESTORE コマンドは、全体バックアップ、レベル 0 の増分バックアップまたはイメージ・コピーをリストアします。ファイルのリストア先は、そのファイルのデフォルトの位置または別の位置です。

デフォルトでは、**Recovery Manager** が読取り専用データファイルをチェックし、それが存在していること、読取り可能であること、および適切なチェックポイントがあることを確認します。これらの条件が満たされない場合、**Recovery Manager** はファイルをリストアします。すべての条件が満たされている場合、**Recovery Manager** はファイルをリストアしません。

#### バックアップの選択

デフォルトでは、RESTORE は最新のバックアップ・セットまたはファイル・コピーを使用します。つまり、最小のメディア・リカバリで済むファイル・コピーまたはバックアップ・セットを使用します。**Recovery Manager** は、RESTORE コマンドで割り当てたチャンネルと同じタイプのチャンネルで作成したバックアップのみをリストアします。たとえば、あるデータファイルのバックアップを DISK チャンネルと sbt チャンネルに作成し、RESTORE コマンドには DISK チャンネルのみを割り当てた場合、**Recovery Manager** は sbt のバックアップをリストアしません。チャンネルを手動で割り当てない場合、**Recovery Manager** は、DEVICE TYPE オプションによる制限に従って、必要とする可能性があるすべての自動チャンネルを割り当てます。

Oracle RAC 構成では、バックアップ、制御ファイルのコピーおよびデータファイルのコピーは、テープ上またはローカル・ファイル・システム上でファイルを読み込めるチャンネルから自動的にリストアされます。たとえば、`inst1` に接続している `ch1` はテープ・ドライブからログ 1000 を読み取ることができても、`inst2` に接続しているチャンネル `ch2` がテープ・ドライブから同じログを読み取ることができない場合、`ch1` がログのリストアをできないため、`ch2` がこのログをリストアします。チャンネルが別の PARMS 設定または CONNECT 設定を使用している場合は、自動位置検索が自動的に使用可能になります。

データファイル名がシンボリック・リンクの場合、制御ファイルにはリンク・ファイルのファイル名が格納されますが、**Recovery Manager** は、リンク・ファイルが指すデータファイルで I/O を実行します。ただし、リンク・ファイルが消失し、最初にシンボリック・リンクを再作成せずにデータファイルをリストアすると、**Recovery Manager** は、リンク・ファイルが示す位置ではなく、リンク・ファイルの位置にデータファイルをリストアします。

**関連項目：** リストア・フェイルオーバーの詳細は、『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』を参照してください。

### 暗号化されたバックアップ・セットを使用したリストア操作

2-20 ページの「バックアップ・セットの暗号化」で説明したとおり、リストア操作中の **Recovery Manager** による暗号化されたバックアップ・セットの処理方法は、バックアップが作成された暗号化モードによって異なります。**CONFIGURE** および **SET** を使用すると、**Recovery Manager** によるデータベース・バックアップの暗号化設定を管理できます。次のリストアに関する考慮事項に注意してください。

- 透過モードで暗号化されたバックアップの場合、必要なパスワードは、データベースの **Wallet** で使用可能である必要があります。バックアップ作成時に使用したウォレットは、リストア時にオープンして使用可能にする必要があります。**SET DECRYPTION** は必要ありません。
- パスワード・モードで暗号化されたバックアップの場合、必要なパスワードは、**SET DECRYPTION** を使用して指定する必要があります。
- デュアル・モードで暗号化されたバックアップの場合、必要なパスワードは、データベースのウォレットで使用可能であるか、または **SET DECRYPTION** を使用して指定する必要があります。

---

**注意：** **Wallet** ベースの暗号化は、パスワードが含まれないため、パスワード・ベースの暗号化よりも安全です。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可能な必要があるため、必要な場合のみ使用してください。

---

### リストア・フェイルオーバー

バックアップ・ピース、イメージ・コピーまたはプロキシ・コピーにアクセスできないか、またはブロックが破損している場合、**Recovery Manager** によってリストア・フェイルオーバーが実行されます。**RESTORE** コマンドは、バックアップまたはイメージ・コピーの使用可能な別のコピーを同じデバイスと他のデバイスで自動的に検索します。使用可能なコピーが存在しない場合は、**Recovery Manager** によって以前のバックアップが検索されます。適切なコピーが見つかるまで、使用可能な以前のバックアップの検索は続行されます。**Recovery Manager** は、必要に応じて、以前のデータベース・インカネーションから適用可能なバックアップを使用します。

リストアしているデータファイルに使用できるバックアップがない場合は、**Recovery Manager** によって、作成 SCN としてチェックポイントの変更が指定されている空のデータファイルが作成されます。リカバリ時は、データファイルの作成時までさかのぼってすべてのアーカイブ **REDO** ログがリストアされます。また、データファイルの履歴内のすべての変更が再適用され、内容が再作成されます。

**関連項目：** 2-20 ページの「バックアップ・セットの暗号化」または『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』を参照してください。

### リストアされたデータファイルの場所

データファイルをデフォルトの場所にリストアすると、Recovery Manager によって同じファイル名を持つファイルが上書きされます。データファイルが適切な場所にあり、そのヘッダーに必要なデータが含まれている場合、デフォルトでは、Recovery Manager によってそのデータファイルがリストアされません。Recovery Manager では、データファイル本体の破損ブロックをスキャンしません。

デフォルトのファイル名を使用できないことが Recovery Manager で確認された場合（たとえば、ファイルが Oracle 管理ファイルであるか、または自動ストレージ管理ディスク・グループに存在する場合）には、Recovery Manager によって同じ場所またはディスク・グループに新しいファイルの作成が試行されます。

ファイルをデフォルト以外の場所にリストアするには、`SET NEWNAME` コマンドを使用して、リストア対象ファイルの名前を変更してから、`SWITCH` コマンドでそのファイルを現行のファイルにします（2-234 ページの例 2-113 を参照）。`SWITCH` コマンドを発行しなければ、Recovery Manager は、リストアされたファイルを有効なコピーとみなし、将来のリストア処理で使われます。表 2-35 に、`SET NEWNAME` および `SWITCH` コマンドと併用した `RESTORE` の動作について説明します。

表 2-35 SET NEWNAME、SWITCH および RESTORE

SET NEWNAME の実行	SWITCH の実行	RESTORE の動作
実行しない	N/A	Recovery Manager は、ファイルを現行のパス名にリストアします。
実行する	実行する	Recovery Manager は、 <code>SET NEWNAME</code> で指定されたパス名にファイルをリストアします。Recovery Manager は、制御ファイルの現行のデータファイルの名前をリストアされたファイルの名前で置き換えます。古い名前を持つデータファイルをデータファイルのコピーとして記録します。
実行する	実行しない	Recovery Manager は、 <code>SET NEWNAME</code> で指定されたパス名にファイルをリストアします。Recovery Manager では、制御ファイルの現行データファイルの名前が更新されません。リストアされたファイルは、Recovery Manager リポジトリにデータファイルのコピーとしてリストアされます。

一時ファイルがバックアップされず、一時ファイル用の REDO も生成されないため、Recovery Manager では一時ファイルのリストアまたはリカバリは行われません。Recovery Manager による一時ファイル名の追跡は、必要に応じて一時ファイルを自動的に再作成するためにのみ実行されています。

### 制御ファイルのリストア時における Recovery Manager の動作

制御ファイルのリストア時における Recovery Manager の動作は、様々な要素によって決まります。表 2-36 に、これらの要素を示します。自動バックアップのリストアに必要なコマンドおよびオプションについては、表 2-37 を参照してください。

表 2-36 RESTORE CONTROLFILE の使用例

Recovery Manager の接続	RESTORE CONTROLFILE;	RESTORE CONTROLFILE FROM AUTOBACKUP;	RESTORE CONTROLFILE ... TO 'filename';	RESTORE CONTROLFILE ... FROM 'media_handle' または TAG 'user_tag';
カタログなし、ターゲット・データベースは NOMOUNT 状態で起動済	エラー。FROM AUTOBACKUP を指定する必要があります。	CONTROL_FILES の場所にリストアします。必要なコマンドおよびオプションについては、表 2-37 を参照してください。	FROM AUTOBACKUP を指定する必要があります。filename にのみリストアします。	最初に SET DBID を実行します。指定したファイルからリストアします (TAG からはリストアできません)。TO 'filename' を使用しない場合は、すべての CONTROL_FILES の場所にリストアします。
カタログなし、ターゲット・データベースはマウント済またはオープン状態	エラー。TO 'filename' を使用する必要があります。この場合、filename は CONTROL_FILES リストにないファイルです。	エラー。TO 'filename' を使用する必要があります。この場合、filename は CONTROL_FILES リストにないファイルです。	filename にのみリストアします。この場合、filename は CONTROL_FILES リストにないファイルです。	指定したファイルからリストアします。TO 'filename' を使用しない場合は、すべての CONTROL_FILES の場所にリストアします。
カタログあり、ターゲット・データベースは NOMOUNT 状態で起動済	CONTROL_FILES の場所にリストアします。DB_NAME がカタログ内で一意でない場合のみ、SET DBID を実行します。	テスト用のリカバリ・カタログでのみ使用します。	filename にのみリストアします。この場合、filename は CONTROL_FILES リストにないファイルです。	指定したファイルからリストアします。TO 'filename' を使用しない場合は、すべての CONTROL_FILES の場所にリストアします。
カタログあり、ターゲット・データベースはマウント済またはオープン状態	エラー。TO 'filename' を使用する必要があります。この場合、filename は CONTROL_FILES リストにないファイルです。	リカバリ・カタログでは使用しません。	filename にのみリストアします。この場合、filename は CONTROL_FILES リストにないファイルです。	Recovery Manager によりエラー RMAN-06496 が発行されます。かわりに、TO 'filename' を使用します。

Data Guard 環境で Recovery Manager を使用すると、Recovery Manager は、プライマリ制御ファイルからスタンバイ制御ファイルへの変換およびスタンバイ制御ファイルからプライマリ制御ファイルへの変換を透過的に行います。また、RESTORE および RECOVER を発行すると、データファイル、オンライン REDO ログ、スタンバイ REDO ログおよび一時ファイルのファイル名を自動的に更新します。2-21 ページの「Data Guard 環境での Recovery Manager のバックアップ」で説明するとおり、リカバリ・カタログには、常に各データベースのバックアップ・ファイル名に関する正しい情報が含まれています。

#### 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・オプション

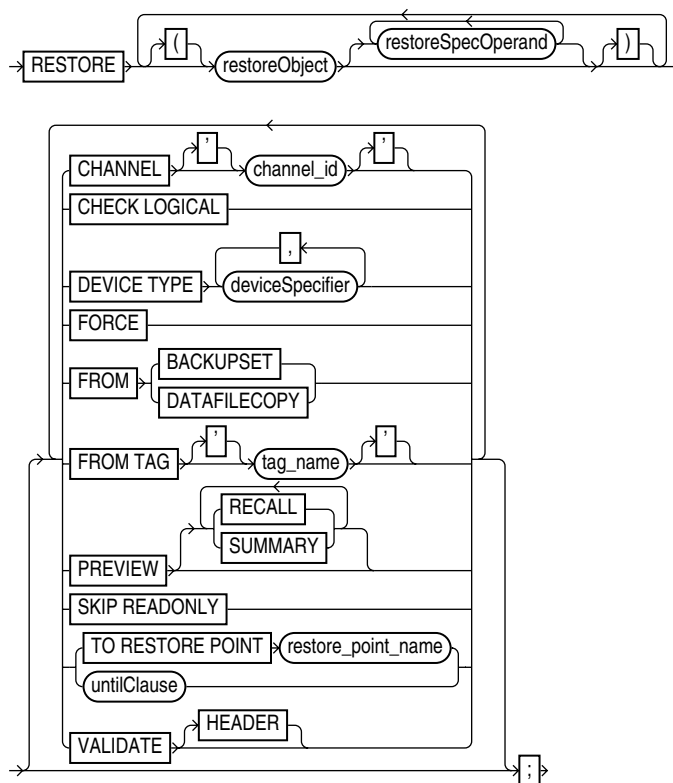
自動バックアップをリストアする場合、使用するコマンドおよびオプションは、自動バックアップのタイプ（制御ファイルまたはサーバー・パラメータ・ファイル）と場所（フラッシュ・リカバリ領域の内部または外部）によって決まります。表 2-37 に、これらのオプションを示します。

表 2-37 RESTORE ... FROM AUTOBACKUP

リストア・オブジェクト	自動バックアップの場所	SET DBID の実行	RESTORE への RECOVERY AREA の指定	RESTORE への DB_NAME または DB_UNIQUE_NAME の指定	SET CONTROLFILE AUTOBACKUP FORMAT の実行
SPFILE	リカバリ領域	実行しない	実行する	実行する	実行しない
SPFILE	リカバリ領域外	実行する	実行しない	実行しない	自動バックアップがデフォルトの場所でない場合のみ
制御ファイル	リカバリ領域	実行しない	自動バックアップが現行以外のリカバリ領域内にある場合のみ	自動バックアップが現行以外のリカバリ領域内にあり、また現行以外の DB_UNIQUE_NAME を使用している場合のみ	実行しない
制御ファイル	リカバリ領域外	実行する	実行しない	実行しない	自動バックアップがデフォルトの場所でない場合のみ

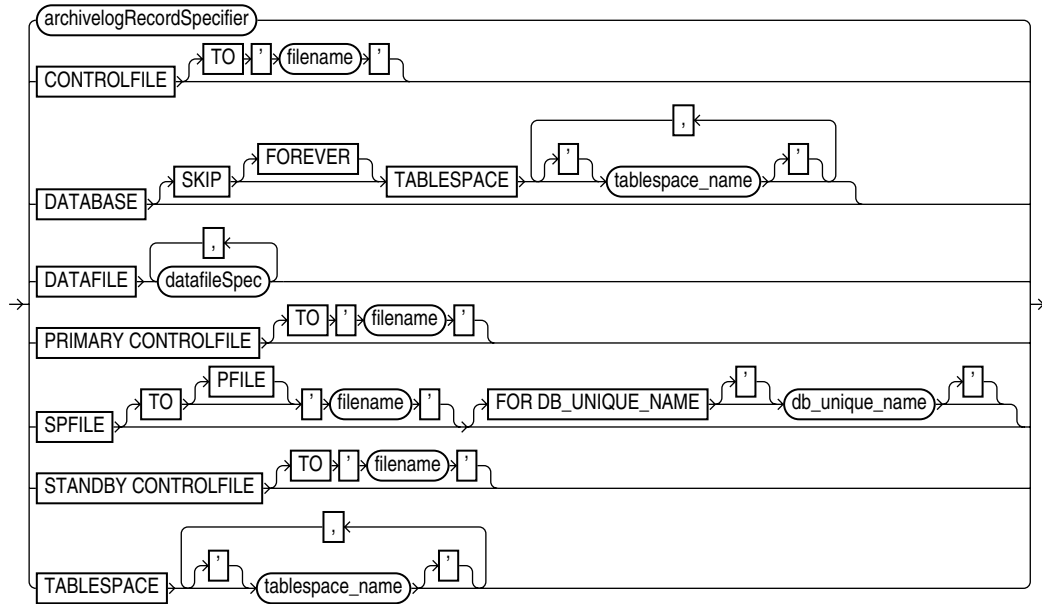
## 構文

**restore::=**



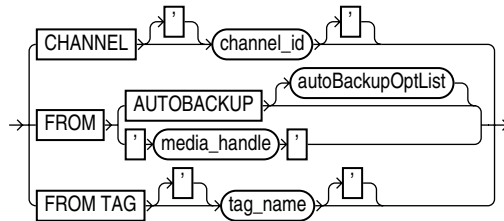
([restoreObject::=](#) (2-226 ページ)、[restoreSpecOperand::=](#) (2-226 ページ)、  
[deviceSpecifier::=](#) (3-15 ページ)、[untilClause::=](#) (3-38 ページ))

**restoreObject::=**

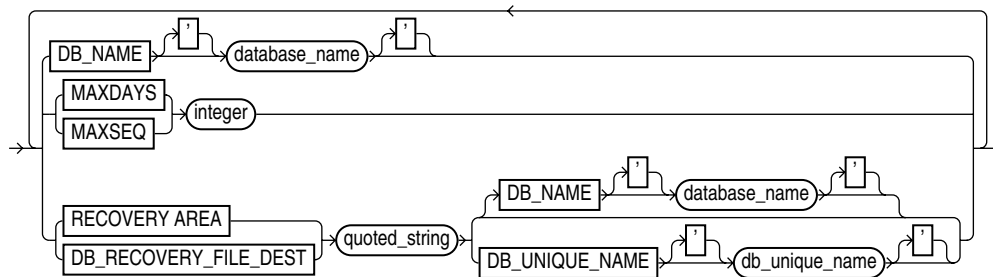


(`archivelogRecordSpecifier::=` (3-6 ページ)、`datafileSpec::=` (3-14 ページ))

**restoreSpecOperand::=**



**autoBackupOptList::=**



## セマンティクス

### restore

この句を使用すると、リストアするファイルを選択し、リストア操作の動作を制御するパラメータを指定できます。

構文の要素	説明
<code>restoreObject</code>	リストアするファイルを指定します。
<code>restoreSpecOperand</code>	<code>restoreObject</code> 句に対するオプションを指定します。
CHANNEL <code>channel_id</code>	<code>restoreSpecOperand</code> 句を参照してください。
CHECK LOGICAL	<p>物理的な破損チェックを通過したデータ・ブロックと索引ブロックについて、論理的な破損がないかどうかをテストします。たとえば、行ピースまたは索引エントリの破損がないかどうかを調べます。Recovery Manager は論理的な破損を見つけると、アラート・ログとサーバー・セッション・トレース・ファイルにそのブロックのログを書き込みます。</p> <p>あるファイルで検出された物理的な破損と論理的な破損の合計数が <code>SET MAXCORRUPT</code> の設定値以下の場合、Recovery Manager のコマンドは完了し、破損ブロック範囲が <code>V\$DATABASE_BLOCK_CORRUPTION</code> ビューに移入されます。MAXCORRUPT を超えている場合、コマンドはビューへの移入を行わずに終了します。</p> <p>バックアップ・データファイルのリストア時には、<code>DB_BLOCK_CHECKSUM</code> 初期化パラメータの設定が考慮されます。<code>DB_BLOCK_CHECKSUM</code> が <code>false</code> に設定されている場合は、チェックサムが消去されます。<code>typical</code> に設定されている場合は、バックアップからリストアしてデータファイルに書き込む際に、チェックサムが検証されます。初期化パラメータ <code>DB_BLOCK_CHECKSUM=typical</code> を設定し、MAXCORRUPT を設定していない場合に CHECK LOGICAL を指定すると、検出可能なすべてのタイプの破損が検出されます。</p> <p><b>注意:</b> MAXCORRUPT の設定値によって、ファイルに許容される物理的および論理的な破損の合計数が指定されます。</p>
DEVICE TYPE <code>deviceSpecifier</code>	<p>指定したデバイス・タイプ専用の自動チャンネルを割り当てます。たとえば、自動ディスクおよびテープ・チャンネルを構成して <code>RESTORE...DEVICE TYPE DISK</code> を実行すると、Recovery Manager ではディスク・チャンネルのみが割り当てられます。DEVICE TYPE オプションを指定する前に、<code>CONFIGURE</code> コマンドを使用してデバイス・タイプを構成する必要があります（事前構成される DISK 以外）。</p> <p><b>注意:</b> RUN ブロック内でチャンネルを手動で割り当ててから、DEVICE TYPE 句を使用して RESTORE を実行することはできません。</p> <p><b>関連項目:</b> 3-15 ページの「<code>deviceSpecifier</code>」を参照してください。</p>
FORCE	再起動可能なリストア機能をオーバーライドし、リストアが必要かどうかに関係なくすべてのファイルをリストアします。FORCE を指定しなければ、ヘッダー情報が制御ファイル内の情報と一致しない場合にのみ、Recovery Manager はファイルをリストアします。
FROM BACKUPSET	<p>Recovery Manager がバックアップ・セットからのみリストアを行うように指定します。デフォルトでは、RESTORE は最小のメディア・リカバリで済むファイル・コピーまたはバックアップ・セットを使用します。</p> <p>FROM BACKUPSET オプションを使用する場合は、リストアが必要となるバックアップ・セット用に、適切なタイプのストレージ・デバイスのチャンネルを割り当てる必要があります。たとえば、必要なバックアップがテープでのみ使用可能で、<code>sbt</code> チャンネルが割り当てられていない場合、Recovery Manager でリストア候補のバックアップ・セットを検出できないため、RESTORE コマンドは正常に実行されません。</p>

構文の要素	説明
FROM DATAFILECOPY	Recovery Manager がデータファイルのコピーのみをリストアするように指定します。デフォルトでは、RESTORE は最小のメディア・リカバリで済むファイル・コピーまたはバックアップ・セットを使用します。FROM DATAFILECOPY オプションを使用する場合、割当て済のチャンネルは DEVICE TYPE DISK タイプである必要があります。
FROM TAG <i>tag_name</i>	<a href="#">restoreSpecOperand</a> 句を参照してください。
PREVIEW	<p>Recovery Manager が指定した時刻のリストアおよびリカバリに使用できるバックアップとアーカイブ REDO ログをレポートします (リストアは行いません)。Recovery Manager ではメタデータの間合せを実行しますが、実際のバックアップ・ファイルの読取りは行いません。</p> <p>RESTORE ... PREVIEW の出力は、LIST BACKUP の出力と同じ形式です (2-236 ページの例 2-118 を参照)。</p> <p>いくつかのメディア・マネージャによって、オフサイトのバックアップを示すステータス情報が Recovery Manager に提供されます。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るため、メディアを入手しないと使用できません。</p> <p>オフサイトのバックアップは、バックアップをリストアする前にメディアをストレージから入手する必要があるにもかかわらず、Recovery Manager リポジトリでは AVAILABLE とマークされます。Recovery Manager がオフサイトのバックアップをリストアしようとする、リストア操作は失敗します。RESTORE ... PREVIEW は、入手する必要があるメディアに格納されている、RESTORE 操作に必要なバックアップを識別できます。出力には、バックアップがオフサイトで格納されているかどうかを示されます。</p> <p>必要なバックアップがオフサイトで格納されているのに、メディア・マネージャでオフサイトのバックアップを使用できない場合は、次のオプションを使用できません。</p> <ul style="list-style-type: none"> <li>CHANGE ... UNAVAILABLE を使用して、オフサイトにある必要なバックアップを Recovery Manager が選択できないようにし、RESTORE ... PREVIEW 操作を再試行して、Recovery Manager がオフサイトの別のバックアップを選択するかどうかを確認します。Recovery Manager がオフサイトのバックアップを選択しない場合は、リストア操作を実行できます。</li> <li>RECALL オプションを指定して、RESTORE ... PREVIEW を使用します。</li> </ul> <p><b>関連項目 :</b> 2-161 ページの「LIST」(特に、BACKUPS および SUMMARY オプションと、RECOVER ... VALIDATE HEADER コマンド) を参照してください。</p>
RECALL	<p>指定したリストア操作に必要なバックアップ・メディアをオフサイトのストレージから入手するようにメディア・マネージャに指示します (2-236 ページの例 2-119 を参照)。</p> <p><b>注意 :</b> このオプションが有効になるのは、メディア・マネージャでこの機能がサポートされている場合のみです。RESTORE ... PREVIEW を定期的を使用すると、必要なバックアップがローカルに格納されているかどうかを再度監視できます。</p>
SUMMARY	Recovery Manager によってリストアされるバックアップのサマリーを示します。この出力は、LIST BACKUPS ... SUMMARY コマンドの出力と同じ形式です。
SKIP READONLY	読取り専用ファイルはリストアしません。
TO RESTORE POINT <i>restore_point_name</i>	リストア・ポイントを作成した時点の SCN を上限として、リストア・ポイントを指定します。指定した値は含まれます。上限値が含まれるため、Recovery Manager は、リストア・ポイントに対応する SCN までファイルをリストアできるファイルのみを選択します。
<i>untilClause</i>	<p>選択範囲を、指定した時刻、SCN またはログ順序番号までの Point-in-Time リカバリに適したバックアップ・セットまたはファイル・コピーに制限します。</p> <p>他の基準がない場合、Recovery Manager は、リストアする最新のファイル・コピーまたはバックアップ・セットを選択します。UNTIL 句で指定した時点は、現行のデータベース・インカネーション内である必要があります。</p> <p><b>関連項目 :</b> 3-38 ページの「<i>untilClause</i>」を参照してください。</p>



構文の要素	説明
VALIDATE	<p>Recovery Manager で、リストアが必要なバックアップ・セット、データファイルのコピーおよびアーカイブ REDO ログ・ファイルを決定してから、これらを検証するように指定します (2-237 ページの例 2-120 を参照)。ファイルはリストアされません。</p> <p>ディスクとテープ両方のファイルについては、Recovery Manager がバックアップ・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。Recovery Manager は、オフサイト・バックアップも検証します。Recovery Manager での検証は実際のリストア操作と同じですが、出力ファイルの書き出しは実行しません。</p> <p><b>注意:</b> VALIDATE オプションを指定して RESTORE を使用すると、データファイルをオンラインにした状態でデータベースをオープンできます。</p> <p><b>関連項目:</b> 2-287 ページの「VALIDATE」を参照してください。</p>
HEADER	<p>Recovery Manager が指定した時刻のリストアに使用できるバックアップをレポートして検証します (リストアは行いません)。</p> <p>このオプションを指定すると、Recovery Manager は、PREVIEW オプションを指定して RESTORE を実行した場合と同じ機能を実行します。ただし、Recovery Manager は、リストアおよびリカバリに必要なファイルのリストに加えて、バックアップ・ファイルのヘッダーを検証し、ファイルがディスク上にあるか、または Recovery Manager リポジトリのメタデータに対応するメディア管理カタログにあるかを判別します。</p> <p><b>関連項目:</b> RESTORE PREVIEW オプションおよび RECOVER ... VALIDATE HEADER オプションの説明を参照してください。</p>

### restoreObject

この副次句は、リストアされるオブジェクト (制御ファイル、データファイル、アーカイブ REDO ログまたはサーバー・パラメータ・ファイル) を指定します。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。チェンジ・トラッキング・ファイルは、Recovery Manager によってデータベースのリストアおよびリカバリ後に再作成されます。リカバリ後の次の増分バックアップでは、このチェンジ・トラッキング・ファイルが使用されます。したがって、リストアおよびリカバリによるチェンジ・トラッキングへの影響は、ユーザーからは管理できません。

構文の要素	説明
<code>archiveLogRecordSpecifier</code>	<p>アーカイブ REDO ログの指定範囲をリストアします。</p> <p>デフォルトのリストアの場所は、DB_RECOVERY_FILE_DEST です (LOG_ARCHIVE_DEST_n のいずれかが、暗黙的または明示的に USE_DB_RECOVERY_FILE_DEST に構成されている場合)。それ以外の場合、デフォルトのリストア・ファイル名は、ターゲット・データベースの LOG_ARCHIVE_FORMAT および LOG_ARCHIVE_DEST_1 初期化パラメータで構成されます。これらのパラメータをポート固有の方法で組合せて、リストアされたログの名前を導出します。デフォルトの場所は、SET ARCHIVELOG DESTINATION コマンドでオーバーライドできます。</p> <p>RECOVER コマンドでは必要に応じてアーカイブ REDO ログが自動的にリストアされるため、手動によるリストアが必要になることはほとんどありません。アーカイブ REDO ログを手動でリストアする可能性が発生するのは、リカバリ時間を短縮する場合、ログを複数の宛先に書き込む場合、Point-in-Time リカバリ後にログの内容を分析する場合などです。データベースを停止せずに以前のインカネーションからログをリストアするには、FROM SCN または SCN BETWEEN ... AND ... 句を指定して RESTORE ARCHIVELOG を実行します。</p> <p><b>注意:</b> この操作では、データベースを起動、マウントまたはオープンできます。</p> <p><b>関連項目:</b> 3-6 ページの「archiveLogRecordSpecifier」を参照してください。</p>

構文の要素	説明
CONTROLFILE	<p>ターゲット・データベースのロールに応じて、スタンバイ制御ファイルまたはバックアップ制御ファイルをリストアします。</p> <p>制御ファイルが消失した場合は、制御ファイルをリストアし (2-224 ページの表 2-36 を参照)、リストアした制御ファイルをマウントしてからデータベースをリストアします。リストアした制御ファイルをマウントした後は、常に RECOVER コマンドを実行して、RESETLOGS オプションでデータベースをオープンする必要があります。</p> <p><b>注意:</b> ターゲット・データベースがマウントされていない状態で、Recovery Manager がリカバリ・カタログに接続していない場合は、RESTORE CONTROLFILE に FROM AUTOBACKUP 句を指定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形式である場合は、最初に SET CONTROLFILE AUTOBACKUP FORMAT コマンドを使用して形式を指定します。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場合は、RESTORE CONTROLFILE に TO filename 句を指定する必要があります。</p>
TO 'filename'	<p>リカバリ・カタログに接続中にバックアップ制御ファイルを使用して RESTORE を実行すると (2-234 ページの例 2-114 を参照)、制御ファイルは、リカバリ・カタログのメタデータに基づいて、リストアしたデータベースの構造が反映されるように Recovery Manager で自動的に更新されます。</p> <p>制御ファイルを指定されたファイル名にリストアします。</p> <p>TO 句を使用して制御ファイルをリストアする場合の Recovery Manager の動作については、2-224 ページの表 2-36 を参照してください。</p>
DATABASE	<p>オフラインのファイルを除いて、データベースのすべてのデータファイルをリストアします。デフォルトでは、Recovery Manager は読取り専用表領域のデータファイルをリストアします。</p> <p>BACKUP DATABASE とは異なり、RESTORE DATABASE では制御ファイルとサーバー・パラメータ・ファイルは自動的に含まれません。追加の RESTORE CONTROLFILE および RESTORE SPFILE コマンドを発行して、これらのファイルをリストアする必要があります。</p> <p><b>注意:</b> オフライン・データファイルをリストアするには、RESTORE DATAFILE または RESTORE TABLESPACE を使用する必要があります。</p>
SKIP [FOREVER] TABLESPACE	<p>指定した表領域をリストア操作から除外します。このオプションは、一時データを含む表領域のリストアを回避する場合に有効です。</p> <p>FOREVER を指定すると、Recovery Manager は表領域に属するデータファイルをリストア前にオフライン化するときに、ALTER DATABASE DATAFILE ... OFFLINE の DROP オプションを指定します。DROP オプションは、Recovery Manager がこれらのファイルをリカバリせず、データベースを再びオープンした後に、その表領域をデータベースから削除することを示します。したがって、FOREVER は、Recovery Manager がスキップした表領域にこれ以上の処理をしないことを意味します。</p>
DATAFILE <i>datafileSpec</i>	<p>ファイル名または絶対データファイル番号で指定したデータファイルをリストアします (2-234 ページの例 2-113 を参照)。</p> <p><b>注意:</b> リストア・ジョブでは、1 つのデータファイルを 2 回以上指定しないでください。たとえば、次のコマンド例は、データファイル 1 が明示的に指定されていると同時に、SYSTEM 表領域内で暗黙的に指定されているため、無効とみなされません。</p> <pre>RESTORE TABLESPACE SYSTEM DATAFILE 1;</pre> <p><b>関連項目:</b> 3-14 ページの「<i>datafileSpec</i>」を参照してください。</p>

構文の要素	説明
PRIMARY CONTROLFILE	<p>Data Guard 環境のプライマリ・データベースの制御ファイルをリストアします。</p> <p>Recovery Manager は、ターゲット・データベースのリカバリ・カタログに認識されている最新のデータベース・ロール (RC_SITE.DATABASE_ROLE) に応じて、適切に通常の制御ファイルまたはスタンバイ制御ファイルをリストアします。このオプションの目的は、最新のデータベース・ロールが古くなった場合に、デフォルトの設定をオーバーライドすることです。</p> <p>プライマリ・データベース dgny からスタンバイ・データベース dgsf への切替えを実行して、dgsf を新しいプライマリ・データベースにします。dgsf の制御ファイルをリストアする必要がありますが、リカバリ・カタログの再同期化が行われていないため、dgsf はまだスタンバイ・データベースとして表示されています。この場合、PRIMARY CONTROLFILE を指定すると、デフォルトの Recovery Manager の動作をオーバーライドし、通常の制御ファイルをリストアできます。</p>
SPFILE	<p>プライマリ・サーバー・パラメータ・ファイルまたはスタンバイ・サーバー・パラメータ・ファイルをバックアップ元にリストアします。Recovery Manager では、ターゲット・データベースで使用中のサーバー・パラメータ・ファイルは上書きできません。</p> <p>デフォルトでは、最新のサーバー・パラメータ・ファイルがリストアされます。UNTIL または TAG オプションを指定すると、古いバージョンのサーバー・パラメータ・ファイルをリストアできます。</p> <p>サーバー・パラメータ・ファイルが失われた場合は、Recovery Manager をターゲット・データベース (および使用している場合はリカバリ・カタログ) に接続し、SET DBID を実行します。RESTORE SPFILE を実行する前に、STARTUP FORCE NOMOUNT を実行します。次に STARTUP FORCE を実行し、リストアしたサーバー・パラメータ・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。</p> <p><b>注意:</b> ターゲット・データベースがマウントされていない状態で、Recovery Manager がリカバリ・カタログに接続していない場合は、RESTORE SPFILE に FROM AUTOBACKUP 句を指定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形式である場合は、最初に SET CONTROLFILE AUTOBACKUP FORMAT コマンドを使用して形式を指定します。ターゲット・データベースが起動、マウントまたはオープンされており、データベースの起動にサーバー・パラメータ・ファイルが使用されている場合は、RESTORE SPFILE に TO filename 句を指定する必要があります。</p>
TO [PFILE] 'filename'	<p>プライマリ・サーバー・パラメータ・ファイルまたはスタンバイ・サーバー・パラメータ・ファイルを、TO 句で指定された場所にリストアします。PFILE を指定して、サーバー・パラメータ・ファイルをテキストベースの初期化パラメータ・ファイルとして保存します。</p>
FOR DB_UNIQUE_NAME 'db_unique_name'	<p>インスタンスが起動されていないときは、ターゲット・データベースの DB_UNIQUE_NAME を指定します。このパラメータは、Data Guard 環境でのみ有効です。</p> <p>FOR DB_UNIQUE_NAME が指定されている場合、Recovery Manager は SPFILE がリストアされるホストの正しい Recovery Manager 構成を検索し、それらを使用して、バックアップ・デバイスにアクセスします。指定されていない場合、Recovery Manager は正しいチャネル構成が選択できず、RMAN-6758 エラーを戻します。</p> <p>Data Guard 環境では、プライマリ・ホストとスタンバイ・ホストに、関連 SBT バックアップおよびディスク・デバイスと通信するための異なるチャネル構成が設定されている場合があります。プライマリ・データベースとスタンバイ・データベースの両方がリカバリ・カタログで認識される場合は、両方のデータベースの構成設定はリカバリ・カタログに記録されています。プライマリ・データベースとスタンバイ・データベースには同じ DB_NAME が含まれているため、リカバリ・カタログのレコードは、DB_UNIQUE_NAME 初期化パラメータによってのみ区別できます。</p> <p><b>注意:</b> DB_NAME がリカバリ・カタログ内で一意でない場合に RESTORE SPFILE を使用すると、RMAN-6758 エラーが発生します。</p>
	<p><b>関連項目:</b> Data Guard 環境でサーバー・パラメータ・ファイルをリストアする手順の詳細は、『Oracle Data Guard 概要および管理』を参照してください。</p>

構文の要素	説明
TO 'filename'	スタンバイ制御ファイルを指定されたファイル名にリストアします。TO 句を使用して制御ファイルをリストアする場合の Recovery Manager の動作については、表 2-36 を参照してください。
STANDBY CONTROLFILE	<p>スタンバイ・データベースの制御ファイルをリストアします。Recovery Manager は、通常のバックアップ制御ファイルを透過的にリストアし、スタンバイ・データベースに対して使用できるようにします。</p> <p>Recovery Manager は、ターゲット・データベースのリカバリ・カタログに認識されている最新のデータベース・ロール (RC_SITE.DATABASE_ROLE) に応じて、適切に通常の制御ファイルまたはスタンバイ制御ファイルをリストアします。このオプションの目的は、最新のデータベース・ロールが古くなった場合に、デフォルトの設定をオーバーライドすることです。プライマリ・データベース dgny からスタンバイ・データベース dgsf への切替えを実行して、dgsf を新しいプライマリ・データベースにします。後で、dgny を dgsf のスタンバイ・データベースにします。dgny で制御ファイルをリストアする必要がありますが、リカバリ・カタログの再同期化が行われていないため、dgny はまだプライマリ・データベースとして表示されています。この場合、STANDBY CONTROLFILE を指定すると、デフォルトの Recovery Manager の動作をオーバーライドし、スタンバイ制御ファイルをリストアできます。</p> <p>DB_UNIQUE_NAME がリカバリ・カタログに認識されているデータベースの制御ファイルをリストアする場合、Recovery Manager は、制御ファイル内のすべてのファイル名をリカバリ・カタログに認識されるファイル名に更新します。ファイル名は、ALTER DATABASE RENAME FILE を使用して明示的に名前が変更されると、リカバリ・カタログのファイル名よりも優先されます。</p> <p><b>関連項目：</b>制限事項と使用上の注意は、表 2-36 を参照してください。</p> <p><b>注意：</b>リストアされた制御ファイルのマウント後は常に RECOVER コマンドを実行する必要があります。また、データベースは常に RESETLOGS オプションでオープンする必要があります。</p>
TABLESPACE 'tablespace_name'	<p>指定した表領域にあるすべてのデータファイルをリストアします (2-234 ページの例 2-112 を参照)。</p> <p>Recovery Manager は、表領域名をデータファイルのリストに内部的に変換します。表領域名を変更する場合 (users から customers への変更など) は、古い名前 (users) で追加の表領域が作成されていないかぎり、古い名前 (users) または新しい名前 (customers) のいずれかを表領域に使用できます。表領域名が変更されていて、次の再同期化時にリカバリ・カタログが更新されることが、Recovery Manager によって確認されます。</p> <p><b>注意：</b>Recovery Manager を使用して、ディクショナリ管理一時表領域はバックアップおよびリストアできますが、ローカル管理一時表領域はバックアップできません。ただし、ローカル管理一時表領域は、データベースのリストア後に自動的に再作成されます。</p>

### restoreSpecOperand

この副次句は、restoreObject 句に対するオプションを指定します。これらのパラメータは、RESTORE コマンドのレベルで同じ名前を持つパラメータをオーバーライドします。

構文の要素	説明
CHANNEL 'channel_id'	このリストア操作に使用するチャンネルの名前を指定します。このチャンネル名には大 / 小文字区別があります。チャンネル指定がないと、RESTORE は正しいデバイス・タイプで割り当てられた使用可能なチャンネルのいずれかを使用します。

構文の要素	説明
FROM AUTOBACKUP	<p>制御ファイルの自動バックアップをリストアします (2-235 ページの例 2-115 を参照)。</p> <p>このオプションは、RESTORE CONTROLFILE コマンドおよび RESTORE SPFILE コマンドでのみ有効です。どちらかのタイプのファイルを NOCATALOG モードでリストアする場合は、FROM AUTOBACKUP 句が必須です。</p> <p>Recovery Manager は、現在の日付または SET UNTIL で指定された日付から検索を開始します。検索初日は、順序番号 256 (または指定されている場合は MAXSEQ で指定された順序番号) で検索を開始し順序 0 に戻るまで降順にカウントします。自動バックアップが現在の日付または SET UNTIL 日に見つからなければ、Recovery Manager では順序 256 から 0 に戻るまで過去の日付が順番にチェックされます。検索は現在の日付または SET UNTIL 日より前に MAXDAYS 日間 (デフォルトは 7、最大 366) 継続されます。MAXDAYS 日以内に自動バックアップが見つからなければ、Recovery Manager はエラーを発行し、コマンドが停止します。</p> <p><b>関連項目:</b> 制限事項と使用上の注意は、表 2-36 を参照してください。</p>
<pre>autoBackupOptList ' media_handle '</pre>	<p>制御ファイルの自動バックアップの検索を制御するパラメータを指定します。</p> <p>制御ファイルのコピー名、または制御ファイルを含むバックアップ・ピースの名前を指定します。media_handle には、制御ファイルのバックアップを含む任意のバックアップ・ピースを指定できます。制御ファイルのバックアップは、自動バックアップでなくてもかまいません。</p> <p><b>関連項目:</b> 制限事項と使用上の注意は、表 2-36 を参照してください。</p>
FROM TAG tag_name	<p>最新のバックアップまたは使用可能なファイル・コピーに関するデフォルトの選択をオーバーライドします。このタグは、自動選択の対象を、指定したタグで作成されたバックアップ・セットまたはファイル・コピーに制限するために使用します。複数のバックアップ・セットまたはコピーに一致するタグが存在していると、Recovery Manager は最新の内容を選択します。タグ名には、大 / 小文字区別はありません。</p> <p><b>関連項目:</b> 多重化バックアップ・セットの個々のコピーにタグを適用する方法と、タグのデフォルト・ファイル名形式については、2-19 ページの「BACKUP」を参照してください。</p>

### autoBackupOptList

この副次句は、制御ファイルの自動バックアップの検索を制御するパラメータを指定します。

構文の要素	説明
DB_NAME database_name	<p>制御ファイルの自動バックアップの検索で使用する DB_NAME を指定します。このパラメータの設定方法については、2-225 ページの表 2-37 を参照してください。</p> <p>DB_UNIQUE_NAME 初期化パラメータのデフォルト値は、DB_NAME 初期化パラメータの設定値です。ターゲット・データベースに DB_UNIQUE_NAME 初期化パラメータが設定されていない場合は、RESTORE ... DB_NAME または RESTORE ... DB_UNIQUE_NAME を使用します。ターゲット・データベースの DB_UNIQUE_NAME 初期化パラメータ設定が DB_NAME と異なる場合は、RESTORE ... DB_UNIQUE_NAME を使用します。</p>
MAXDAYS integer	制御ファイルの自動バックアップの検索を過去の指定した日数内に制限します。
MAXSEQ integer	制御ファイルの自動バックアップの検索での最大順序番号を指定します。
RECOVERY AREA 'pathname'	<p>自動バックアップを検索するフラッシュ・リカバリ領域へのパスを指定します。RECOVERY AREA と DB_RECOVERY_FILE_DEST はシノニムです。このパラメータの設定方法については、2-225 ページの表 2-37 を参照してください。</p>
DB_RECOVERY_FILE_DEST 'pathname'	RECOVERY AREA と DB_RECOVERY_FILE_DEST はシノニムです。

構文の要素	説明
DB_NAME <i>database_name</i>	制御ファイルの自動バックアップの検索で使用する DB_NAME を指定します。このパラメータの設定方法については、2-225 ページの表 2-37 を参照してください。  DB_UNIQUE_NAME 初期化パラメータのデフォルト値は、DB_NAME 初期化パラメータの設定値です。ターゲット・データベースに DB_UNIQUE_NAME 初期化パラメータが設定されていない場合は、RESTORE ... DB_NAME または RESTORE ... DB_UNIQUE_NAME を使用します。ターゲット・データベースの DB_UNIQUE_NAME 初期化パラメータ設定が DB_NAME と異なる場合は、RESTORE ... DB_UNIQUE_NAME を使用します。
DB_UNIQUE_NAME <i>db_unique_name</i>	リストア操作のターゲットである、指定したフラッシュ・リカバリ領域内のデータベースの DB_UNIQUE_NAME を指定します。  DB_UNIQUE_NAME 初期化パラメータのデフォルト値は、DB_NAME 初期化パラメータの設定値です。ターゲット・データベースに DB_UNIQUE_NAME 初期化パラメータが設定されていない場合は、RESTORE ... DB_NAME または RESTORE ... DB_UNIQUE_NAME を使用します。ターゲット・データベースの DB_UNIQUE_NAME 初期化パラメータ設定が DB_NAME と異なる場合は、RESTORE ... DB_UNIQUE_NAME を使用します。

## 例

### 例 2-112 表領域のリストア

この例では、表領域をオフラインにし、リストアしてからメディア・リカバリを実行します。

```
SQL "ALTER TABLESPACE users OFFLINE IMMEDIATE";
RESTORE TABLESPACE users;
RECOVER TABLESPACE users;
SQL "ALTER TABLESPACE users ONLINE";
```

### 例 2-113 リストアされるデータファイルの新しい名前の設定

データファイル 9 を格納している /disk1 にメディア障害が発生したとします。この例では、データファイルに新しい名前を指定し、データファイルをリストアし、新しい名前を使用するように制御ファイルを更新してリカバリした後、オンライン化します。

```
RUN
{
  SQL "ALTER DATABASE DATAFILE 9 OFFLINE";
  SET NEWNAME FOR DATAFILE 9 TO '/disk2/users01.dbf';
  RESTORE DATAFILE 9;
  SWITCH DATAFILE ALL;
  RECOVER DATAFILE 9;
  SQL "ALTER DATABASE DATAFILE 9 ONLINE";
}
```

### 例 2-114 リカバリ・カタログ使用時の制御ファイルのリストア

monday\_cf\_backup というタグが付いている制御ファイルのバックアップをリストアするとします。Recovery Manager クライアントを起動し、ターゲット・データベースおよびリカバリ・カタログ・データベースに接続して、次のコマンドを実行します。

```
RUN
{ # SET DBID is not necessary when RMAN is connected to a recovery catalog
  STARTUP FORCE NOMOUNT;
  RESTORE CONTROLFILE FROM TAG 'monday_cf_backup';
  ALTER DATABASE MOUNT;
  RESTORE DATABASE;
  RECOVER DATABASE;
}
ALTER DATABASE OPEN RESETLOGS; # required after recovery with backup control file
```

Recovery Manager では、制御ファイルがデフォルト位置にリストアされ、それがすべての CONTROL\_FILES の位置に自動的にレプリケートされます。Recovery Manager は、制御ファイルをマウントし、データベースのリストアとリカバリを行います。Recovery Manager では、リカバリ・カタログのメタデータに基づいて、リストアしたデータベースの構造が反映されるように制御ファイルが自動的に更新されます。

#### 例 2-115 制御ファイルの自動バックアップを使用したデータベースのリカバリ

制御ファイルと一部のデータファイルが消失し、テープからリストアする必要があるとします。この例では、Recovery Manager はリカバリ・カタログを使用しないため、リストアする制御ファイルを特定するには SET DBID コマンドが必要です。次に、テー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てから、データベースのリストアとリカバリを行います。

```
CONNECT TARGET /
STARTUP FORCE NOMOUNT;
SET DBID 36508508; # required when restoring control file in NOCATALOG mode
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt;
  RESTORE CONTROLFILE FROM AUTOBACKUP;
  ALTER DATABASE MOUNT;
  RESTORE DATABASE;
  RECOVER DATABASE;
}
ALTER DATABASE OPEN RESETLOGS;
```

#### 例 2-116 デフォルト以外の位置への制御ファイルの自動バックアップのリストア

この例は、例 2-115 を部分的に変更したものです。この例では、制御ファイルの自動バックアップはデフォルト以外の場所にあるディスクに格納されています。Recovery Manager は、順序番号 20 を持つバックアップから始めて、過去 5 か月にさかのぼって検索します。

```
CONNECT TARGET /
STARTUP FORCE NOMOUNT
SET DBID 36508508; # required when restoring control file in NOCATALOG mode
RUN
{
  SET CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '/disk1/prod_cf_auto_%F';
  RESTORE CONTROLFILE TO '/tmp/cf_auto.dbf' FROM AUTOBACKUP
    MAXSEQ 20 MAXDAYS 150;
  ALTER DATABASE MOUNT;
  RESTORE DATABASE;
  RECOVER DATABASE;
}
ALTER DATABASE OPEN RESETLOGS;
```

#### 例 2-117 現行の位置へのサーバー・パラメータ・ファイルの自動バックアップのリストア

次の一連のコマンドは、現行のサーバー・パラメータ・ファイルを NOCATALOG モードでリストアしてから、リストアされたサーバー・パラメータ・ファイルを使用してインスタンスを起動します。

```
CONNECT TARGET /
SET DBID 1620189241; # set dbid to dbid of target database
STARTUP FORCE NOMOUNT; # start instance with dummy SPFILE
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt;
  RESTORE SPFILE FROM AUTOBACKUP; # FROM AUTOBACKUP needed in NOCATALOG mode
  STARTUP FORCE; # startup with restored SPFILE
}
```

**例 2-118 バックアップのプレビュー**

この例では、RESTORE ... PREVIEW コマンドの結果が表示されます。アーカイブ REDO ログのリストアに使用するために Recovery Manager で選択するバックアップ・セットが示されています。

```
RMAN> RESTORE ARCHIVELOG ALL DEVICE TYPE sbt PREVIEW;
```

```
Starting restore at 01-MAR-07
released channel: ORA_SBT_TAPE_1
allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=85 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: Oracle Secure Backup
```

```
List of Backup Sets
```

```
=====
```

BS Key	Size	Device Type	Elapsed Time	Completion Time
53	1.25M	SBT_TAPE	00:00:18	01-MAR-07
BP Key: 53 Status: AVAILABLE Compressed: NO Tag: TAG20070301T150155				
Handle: 2aibhej3_1_1 Media: RMAN-DEFAULT-000001				

```
List of Archived Logs in backup set 53
```

Thrd Seq	Low SCN	Low Time	Next SCN	Next Time
1 8	526376	01-MAR-07 527059	527059	01-MAR-07
1 9	527059	01-MAR-07 527074	527074	01-MAR-07
1 10	527074	01-MAR-07 527091	527091	01-MAR-07
1 11	527091	01-MAR-07 527568	527568	01-MAR-07
1 12	527568	01-MAR-07 527598	527598	01-MAR-07

```
validation succeeded for backup piece
```

```
Finished restore at 01-MAR-07
```

**例 2-119 オフサイト・ストレージからのオフサイト・バックアップの再呼出し**

バックアップのオフサイト・ストレージに関する情報をレポートし、オフサイト・バックアップの再呼出しをサポートするメディア・マネージャとともに使用すると、RESTORE ... PREVIEW RECALL は、バックアップからのアーカイブ・ログのリストアに必要なメディアをオフサイト・ストレージから再呼出しすることを要求します。

```
RMAN> RESTORE ARCHIVELOG ALL PREVIEW RECALL;
```

```
Starting restore at 10-JUN-06
using channel ORA_DISK_1
using channel ORA_SBT_TAPE_1
```

```
List of Backup Sets
```

```
=====
```

BS Key	Size	Device Type	Elapsed Time	Completion Time
31	12.75M	SBT_TAPE	00:00:02	10-JUN-06
BP Key: 33 Status: AVAILABLE Compressed: NO Tag: TAG20050610T152755				
Handle: 15gmknbs Media: /v1,15gmknbs				

```
List of Archived Logs in backup set 31
```

Thrd Seq	Low SCN	Low Time	Next SCN	Next Time
1 1	221154	06-JUN-06 222548	222548	06-JUN-06
1 2	222548	06-JUN-06 222554	222554	06-JUN-06
1 3	222554	06-JUN-06 222591	222591	06-JUN-06
1 4	222591	06-JUN-06 246629	246629	07-JUN-06
1 5	246629	07-JUN-06 262451	262451	10-JUN-06



```

BS Key   Size      Device Type Elapsed Time Completion Time
-----
32       256.00K   SBT_TAPE    00:00:01    10-JUN-06
          BP Key: 34   Status: AVAILABLE Compressed: NO   Tag: TAG20050610T153105
          Handle: 17gmknhp_1_1   Media: /v1,17gmknhp_1_1

```

List of Archived Logs in backup set 32

Thrd	Seq	Low SCN	Low Time	Next SCN	Next Time
1	6	262451	10-JUN-06	262547	10-JUN-06
1	7	262547	10-JUN-06	262565	10-JUN-06

Initiated recall for the following list of offsite backup files

=====

```

          Handle: 15gmknbs   Media: /v1,15gmknbs
Finished restore at 10-JUN-06

```

### 例 2-120 バックアップのリストアの検証

次の例では、RESTORE... VALIDATE を使用して、データベースのリストアに必要なバックアップがディスクまたはテープに存在し、読取り可能で破損していないことを確認する方法を示します。

```

RMAN> RESTORE DATABASE VALIDATE;

```

```

Starting restore at 01-MAR-07
using channel ORA_DISK_1
allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=85 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
/disk2/PROD/backupset/2007_03_01/o1_mf_nnndf_TAG20070301T161038_2ygtvzg0_.bkp
channel ORA_DISK_1: piece
handle=/disk2/PROD/backupset/2007_03_01/o1_mf_nnndf_TAG20070301T161038_2ygtvzg0_.bkp
tag=TAG20070301T161038
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: validation complete, elapsed time: 00:00:16
Finished restore at 01-MAR-07

```

## RESYNC CATALOG

### 用途

RESYNC CATALOG コマンドを使用すると、リカバリ・カタログ・スキーマとターゲット・データベース制御ファイルとのメタデータの完全再同期化を実行できます。また、FROM CONTROLFILECOPY 句を使用して、現行の制御ファイルを制御ファイルのコピー内の Recovery Manager メタデータと再同期化することもできます。

通常、次の場合に RESYNC CATALOG を実行します。

- 再同期化を自動的に実行する Recovery Manager コマンドを実行したときに、リカバリ・カタログが使用不可能な場合。
- ターゲット・データベースが ARCHIVELOG モードで実行されている場合。これは、オンライン REDO ログ・スイッチが発生するときや、REDO ログをアーカイブするときにリカバリ・カタログが自動的に更新されないためです。
- ターゲット・データベースの物理構造に、表領域の追加または削除などの変更を行った場合。ログのアーカイブの場合と同様に、物理スキーマを変更した場合も、リカバリ・カタログは自動的に更新されません。
- Recovery Manager が TARGET としてスタンバイ・データベースに接続されている場合。このデータベースで実行された Recovery Manager 操作に関するメタデータを使用してリカバリ・カタログを更新する場合。
- Recovery Manager が TARGET としてスタンバイ・データベースに接続されている場合。プライマリ・データベースでの物理的な変更に関するメタデータを使用してリカバリ・カタログを更新する場合 (2-241 ページの例 2-123 を参照)。

### 前提条件

Recovery Manager がマウント済またはオープン状態のデータベースに TARGET として接続され、またリカバリ・カタログ・データベースに CATALOG として接続されている必要があります。

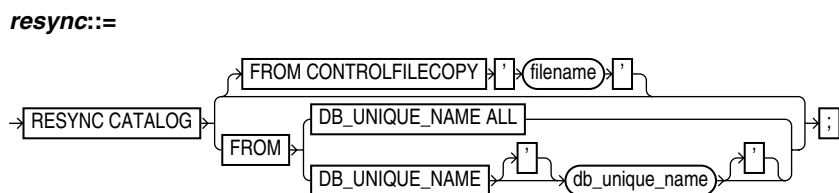
### 使用上の注意

完全同期化と部分同期化のどちらでも実行できます。完全再同期化を実行するときに、ターゲット・データベースにマウントされた現行の制御ファイルがある場合 (ただし、新しく作成された制御ファイルおよび以前に使用された制御ファイルより古い制御ファイルを除く)、Recovery Manager は物理スキーマについて変更があったすべてのレコード (データファイル、表領域、REDO スレッドおよびオンライン REDO ログ) を更新します。データベースがオープン状態の場合、Recovery Manager はロールバック・セグメントについてのデータも取得します。部分再同期化では、Recovery Manager は物理スキーマおよびロールバック・セグメントに関するメタデータの再同期化は行いません。

Recovery Manager コマンドの実行時にターゲット制御ファイルがマウントされており、カタログ・データベースが使用可能である場合、Recovery Manager は必要に応じてコマンドの実行時にリカバリ・カタログを自動的に再同期化します。データベース構造の変更 (データベース・ファイルの追加または削除、新規のインカネーションの作成など) 後、または Recovery Manager の永続構成の変更後に完全再同期化を実行します。

Oracle Database 11g 以上では、Data Guard 環境で単一のリカバリ・カタログ・スキーマがすべてのデータベースのデータベース・ファイル名を追跡できます。また、このカタログ・スキーマは、すべてのデータベースのオンライン REDO ログ、スタンバイ REDO ログ、一時ファイル、アーカイブ REDO ログ、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追跡します。Recovery Manager は、TARGET としてスタンバイ・データベースに接続されているときに、プライマリ・データベースのスタンバイ制御ファイルに物理スキーマの変更に関する情報が格納されていると、暗黙的に完全再同期化を実行します。

## 構文



## セマンティクス

構文の要素	説明
RESYNC CATALOG	<p>ターゲット・データベースの現行の制御ファイル（デフォルト）内の Recovery Manager メタデータで、リカバリ・カタログを更新します。</p> <p>Recovery Manager によって、制御ファイルの読取り一貫性ビューを取得するために、スナップショット制御ファイルが作成され、次に、スナップショットからの新しい情報でリカバリ・カタログが更新されます。RESYNC CATALOG コマンドでは、次のクラスまたはレコードが更新されます。</p> <ul style="list-style-type: none"> <li>■ ログ・スイッチの発生時に作成されたログ履歴レコード。ログ履歴レコードはオンライン・ログ・スイッチを表し、ログ・アーカイブは表さないことに注意してください。</li> <li>■ オンライン REDO ログのアーカイブ、既存のアーカイブ・ログのコピー、またはアーカイブ・ログのバックアップのリストアによって作成されたアーカイブ・ログに対応付けられているアーカイブ REDO ログ・レコード。</li> <li>■ バックアップ・セット、バックアップ・ピース、プロキシ・コピーおよびイメージ・コピーのレコードである、バックアップ・レコード。</li> <li>■ データファイルと表領域に対応付けられた物理スキーマ・レコード。ターゲット・データベースがオープン状態の場合は、ロールバック・セグメントの情報も更新されます。</li> </ul>
FROM CONTROLFILECOPY 'filename'	<p>制御ファイルのコピーからの Recovery Manager メタデータで、現行の制御ファイルおよびリカバリ・カタログを更新します（2-240 ページの例 2-122 を参照）。filename を使用して、再同期化に使用する制御ファイルのコピーの名前を指定します。</p> <p>FROM CONTROLFILECOPY は、主に、制御ファイルを再作成する場合に使用します。制御ファイルを再作成すると制御ファイルに格納されている Recovery Manager のレコードが失われます。ただし、新しく作成した制御ファイルは、古いコピーと再同期化できます。このオプションを使用すると、物理スキーマ情報は更新されません。</p> <p><b>注意：</b>制御ファイルのコピーは、現行のデータベース・インカネーション内に存在しているか、または以前のインカネーション（最新の OPEN RESETLOGS の前）で作成されています。</p>

構文の要素	説明
<pre>FROM DB_UNIQUE_NAME {ALL   db_unique_name}</pre>	<p>リカバリ・カタログを、指定した1つ以上のデータベースの制御ファイルのメタデータで再同期化します (2-241 ページの例 2-124 を参照)。</p> <p>リカバリ・カタログ内のデータベースは、<code>db_unique_name</code> を使用して1つのみ指定することも、ターゲット・データベースの DBID を共有するすべてのデータベースを ALL で指定することもできます。ALL を指定した場合は、Recovery Manager が Data Guard 環境でリカバリ・カタログに認識されているすべてのデータベースを再同期化します。</p> <p><b>注意:</b> CONFIGURE DB_UNIQUE_NAME ... CONNECT IDENTIFIER を使用して、FROM DB_UNIQUE_NAME で指定されたデータベースへの Oracle Net 接続に使用するネット・サービス名を指定する必要があります。</p> <p>指定されたデータベースに対して RESYNC FROM DB_UNIQUE_NAME を実行すると、Recovery Manager は、通常の再同期化と逆方向の再同期化を両方とも実行します。通常の再同期化では、Recovery Manager が制御ファイルのメタデータを使用してリカバリ・カタログを更新します。逆方向の再同期化では、制御ファイルの永続構成が、指定されたデータベースのリカバリ・カタログの情報と一致しない場合に、Recovery Manager がこの構成を更新します。</p> <p>例として、最近、Recovery Manager を TARGET としてプライマリ・データベースに接続し、CONFIGURE を実行してスタンバイ・データベース <code>standby_new</code> の Recovery Manager 構成を作成したとします。ただし、Recovery Manager は TARGET として <code>standby_new</code> に接続されていません。このような場合に、RESYNC CATALOG FROM DB_UNIQUE_NAME <code>standby_new</code> を実行できます。後で Recovery Manager を TARGET として <code>standby_new</code> に接続すると、Recovery Manager によってリカバリ・カタログから <code>standby_new</code> のマウント済の制御ファイルに構成が送信されます。</p>

## 例

### 例 2-121 ARCHIVELOG モードでのリカバリ・カタログの再同期化

この例では、アーカイブされていない REDO ログをすべてアーカイブしてから、ターゲット・データベースの完全再同期化を実行します。

```

RMAN> CONNECT TARGET /
RMAN> CONNECT CATALOG rman@catdb

recovery catalog database Password: password
connected to recovery catalog database

RMAN> SQL "ALTER SYSTEM ARCHIVE LOG CURRENT";
RMAN> RESYNC CATALOG;
```

### 例 2-122 制御ファイル・コピーからのリカバリ・カタログの再同期化

Recovery Manager クライアントを起動し、ターゲット・データベースとリカバリ・カタログに接続するとします。次のコマンドでは、ターゲット・データベースを停止してマウントし、バックアップ制御ファイルからのメタデータを使用して現行の制御ファイルの Recovery Manager リポジトリを更新した後、データベースをオープンします。

```

STARTUP FORCE MOUNT
RESYNC CATALOG FROM CONTROLFILECOPY '/disk1/cfile.dbf';
ALTER DATABASE OPEN;
```

### 例 2-123 構造を変更後のリカバリ・カタログの再同期化

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 standby3 を含む Data Guard 環境があるとします。次のように、SQL\*Plus を起動し、データベース prod に接続して、データファイルを表領域 users に追加します。

```
SQL> ALTER TABLESPACE users ADD DATAFILE '?:oradata/prod/users03.dbf'  
 2 SIZE 1M AUTOEXTEND ON  
 3 NEXT 10K MAXSIZE 10M";
```

目標は、この変更に関するメタデータを使用してリカバリ・カタログを更新することです。変更が standby3 に伝播された後に、Recovery Manager クライアントを起動し、TARGET として standby3 に接続し、リカバリ・カタログに接続します。次に、RESYNC コマンドを使用して、カタログをスタンバイ・データベースの制御ファイルと再同期させます。

```
RMAN> RESYNC CATALOG;
```

リカバリ・カタログは、データベース prod の users 表領域に追加されたデータファイルに関するメタデータを使用して更新されます。

### 例 2-124 リカバリ・カタログとスタンバイ・データベースの再同期化

Data Guard 環境に、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 dgprod3 が存在するとします。目標は、dgprod3 の Recovery Manager 構成を作成することです。

Recovery Manager を TARGET としてデータベース prod に接続してから、リカバリ・カタログに接続します。次のように、CONFIGURE を使用して、リカバリ・カタログで dgprod3 の Recovery Manager の永続構成を更新します。

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt FOR DB_UNIQUE_NAME dgprod3;  
CONFIGURE DB_UNIQUE_NAME dgprod3 CONNECT IDENTIFIER 'inst3';
```

dgprod3 ではまだバックアップ操作やその他の Recovery Manager 操作を実行していないため、dgprod3 の制御ファイルおよび dgprod3 のリカバリ・カタログ・メタデータは同期化されていません。次のように、同じ Recovery Manager セッションで、dgprod3 の制御ファイルをリカバリ・カタログと同期させます。

```
RESYNC CATALOG FROM DB_UNIQUE_NAME dgprod3;
```

Recovery Manager は、dgprod3 でデフォルトのデバイス・タイプを SBT に更新し、dgprod3 制御ファイルの名前を使用してリカバリ・カタログを更新します。

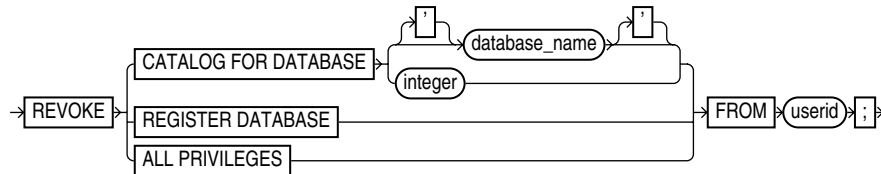
# REVOKE

## 用途

REVOKE コマンドを使用すると、GRANT コマンドで付与されたリカバリ・カタログ権限を取り消すことができます。

## 構文

**revoke::=**



## 前提条件

このコマンドは、RUN コマンドのカッコ内または Recovery Manager プロンプトで実行してください。

## 使用上の注意

仮想プライベート・カタログ・ユーザーに REGISTER DATABASE 権限が付与され、これにより、登録されているすべてのデータベースに対する CATALOG FOR DATABASE 権限が暗黙的に付与されたとします。このユーザーは複数のデータベースを登録します。REVOKE を使用してこのユーザーから REGISTER DATABASE 権限を取り消しても、このユーザーは引き続き、登録されているデータベースの CATALOG FOR DATABASE 権限を保持します。CATALOG 権限には、指定したデータベースの登録と登録解除が含まれます。

このユーザーがデータベースのメタデータへのアクセスやデータベースの追加登録を行えないようにするには、このユーザーに対して REVOKE ALL PRIVILEGES を実行します。このユーザーが登録したデータベースのサブセットに対する CATALOG 権限を取り消すには、サブセット内の各データベースに対して REVOKE CATALOG FOR DATABASE を実行します。

## セマンティクス

構文の要素	説明
CATALOG FOR DATABASE {databasename   integer}	指定したデータベースのリカバリ・カタログへのアクセス権を、指定したユーザーから取り消します。  データベースは、データベース名または DBID で指定できます。データベース名で指定した場合、その名前のデータベースがリカバリ・カタログに複数登録されていると、Recovery Manager によりエラーが戻されます。その場合は、DBID でデータベースを指定してください。
REGISTER DATABASE	指定したユーザーがこのリカバリ・カタログに新規データベースを登録できないようにします (2-243 ページの例 2-125 を参照)。
ALL PRIVILEGES	指定したユーザーからすべての CATALOG 権限および REGISTER 権限を取り消します。
FROM userid	権限を取り消すユーザーの名前を指定します。

## 例

### 例 2-125 仮想プライベート・カタログ・ユーザーからの権限の取消し

Recovery Manager をリカバリ・カタログ所有者 `rco` として基本リカバリ・カタログに接続するとします。基本カタログ所有者として、次のように、Recovery Manager の `GRANT` コマンドを使用して、`bckop2` に自分の仮想プライベート・カタログにすべてのデータベースを登録する権限を付与するとします。ただし、`bckop3` には、データ・センター内のデータベースのサブセットのみへのアクセス権を付与します。

```
RMAN> CONNECT CATALOG rco@catdb

recovery catalog database Password: password
connected to recovery catalog database

RMAN> GRANT REGISTER DATABASE TO bckop2;
RMAN> GRANT CATALOG FOR DATABASE prod TO bckop3;
RMAN> GRANT CATALOG FOR DATABASE prodb TO bckop3;
RMAN> EXIT;
```

その後、ユーザー `bckop2` の権限を制限し、このユーザーが新規データベースを登録できないようにします。したがって、`rco` として基本カタログに接続し、`REVOKE` コマンドを実行します。`bckop2` は引き続き、自分が登録したデータベースに対するカタログ権限を保持することに注意してください。

```
RMAN> CONNECT CATALOG rco@catdb

recovery catalog database Password: password
connected to recovery catalog database

RMAN> REVOKE REGISTER DATABASE FROM bckop2;
```

---

## RMAN

### 用途

RMAN コマンドを使用すると、オペレーティング・システムのコマンドラインから Recovery Manager を起動できます。

データベースへの SQL\*Plus 接続と同様に、データベースへの Recovery Manager 接続が指定され、認証されます。唯一異なるのは、ターゲット・データベースまたは補助データベースへの Recovery Manager 接続では SYSDBA 権限が必要なことです。AS SYSDBA キーワードは暗黙的に指定されており、明示的に指定することはできません。SQL\*Plus 使用時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。

---

**注意：** セキュリティ上、パスワードはコマンドラインにプレーン・テキストで入力しないでください。Recovery Manager プロンプトで要求された場合のみ、Recovery Manager に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。

---

**関連項目：** コマンドラインから Recovery Manager を起動する方法は、『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』を参照してください。

### 前提条件

RMAN コマンドと任意のオプションは、Recovery Manager プロンプトではなく、オペレーティング・システムのコマンドラインで発行する必要があります。

### 使用上の注意

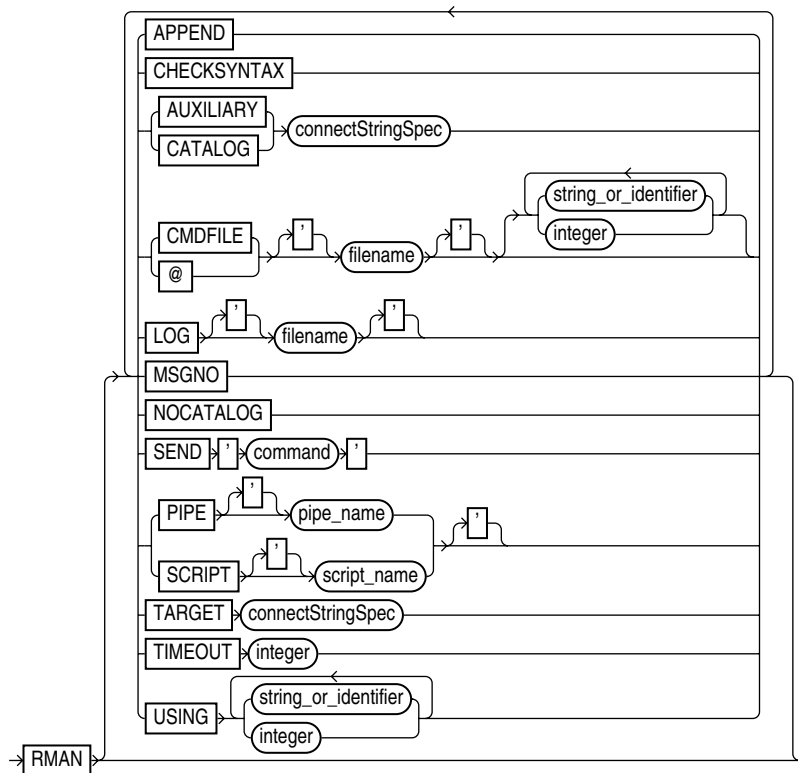
オペレーティング・システムのプロンプトで入力するコマンド名は、オペレーティング・システムによって異なります。たとえば、Linux システムおよび UNIX システムでは、小文字で rman と入力します。

オペレーティング・システムのコマンドラインで CATALOG または NOCATALOG を指定せずに Recovery Manager を起動した場合、CONNECT CATALOG コマンドを実行しないかぎり、Recovery Manager セッションは NOCATALOG モードになります (2-247 ページの例 2-126 を参照)。リカバリ・カタログを保持する場合は、Recovery Manager 操作を実行する前にリカバリ・カタログに接続することをお勧めします。



## 構文

cmdLine::=



## セマンティクス

### cmdLine

構文の要素	説明
APPEND	新規出力をメッセージ・ログ・ファイルの終わりに追加させます。このパラメータを指定せず、かつメッセージ・ログ・ファイルと同じ名前のファイルがすでにある場合、Recovery Managerはそのファイルを上書きします。
CHECKSYNTAX	入力されたコマンドに対して構文エラーをチェックするモードでRecovery Managerを起動させますが、それ以外の処理は実行しません(例 2-129を参照)。引数CMDFILEまたは@とともに使用すると、Recovery Managerクライアントが起動し、ファイル内のすべてのコマンドをチェックしてから終了します。コマンド・ファイルを指定せずに使用すると、Recovery Managerによって、入力が求められ、ユーザーがRecovery Managerクライアントを終了するまで各コマンドが解析されます。  Recovery Managerは、構文が正しくないそれぞれのコマンドに対してエラーRMAN-0558をレポートします。
AUXILIARY <i>connectStringSpec</i>	補助データベースへの接続文字列を指定します。たとえば、AUXILIARY SYS@dupdbのように指定します。  <b>関連項目</b> : 3-12 ページの「 <i>connectStringSpec</i> 」を参照してください。
CATALOG <i>connectStringSpec</i>	リカバリ・カタログを格納するデータベースへの接続文字列を指定します。たとえば、CATALOG catowner@inst2のように指定します。  <b>関連項目</b> : 3-12 ページの「 <i>connectStringSpec</i> 」を参照してください。

構文の要素	説明
CMDFILE <i>filename</i>	<p>ファイル内のすべての Recovery Manager コマンドを解析し、コンパイルしてから、順番に実行します。解析フェーズで構文エラーが発生するか、実行フェーズでランタイム・エラーが発生すると、Recovery Manager は終了します。エラーが見つからなければ、Recovery Manager はジョブの完了後に終了します。</p> <p>ファイル名の最初の文字がアルファベットの場合は、ファイル名を囲む引用符を省略できます。コマンド・ファイルの内容は、Recovery Manager プロンプトに入力した内容と同じにする必要があります。</p> <p><b>注意:</b> コマンド・ファイルをオペレーティング・システムのコマンドラインでオプションとして実行するのではなく、Recovery Manager プロンプトから実行すると、ファイルは 1 つのジョブとして実行されません。Recovery Manager は各行を順次読み込んで実行し、スクリプトの最終行に達した場合にのみ終了します。</p>
@ <i>filename</i>	CMDFILE と同じです。
{ <i>string_or_identifier</i>   <i>integer</i> }	USING 構文の後に指定されているオプションと同じです。
LOG <i>filename</i>	<p>Recovery Manager がその出力を記録するファイルを指定します。Recovery Manager 出力とは、処理したコマンドとその結果です。Recovery Manager はプロンプトにコマンド入力を表示しますが、コマンド出力は表示しません。コマンド出力はログ・ファイルに書き出されます。デフォルトでは、Recovery Manager はメッセージ・ログ・ファイルを標準出力に書き出します。</p> <p>また、Recovery Manager 出力は、V\$RMAN_OUTPUT ビュー（実行中のジョブのメモリ専用ビュー）および V\$RMAN_STATUS ビュー（完了したジョブおよび実行中のジョブの制御ファイル・ビュー）内にも格納されます。</p> <p>LOG パラメータを指定すると、指定したファイルをオープンできない場合にも Recovery Manager は終了しません。かわりに、Recovery Manager によって標準出力が書き込まれます。</p> <p><b>注意:</b> Recovery Manager の出力をログ・ファイルと標準出力の両方に送信する最も簡単な方法は、Linux の tee コマンドまたはこれに相当するコマンドを使用することです。次に例を示します。</p> <pre data-bbox="524 1142 773 1163">% rman   tee rman.log</pre>
MSGNO	Recovery Manager で、メッセージ番号を出力します。つまり、すべてのコマンドの出力に対して、RMAN-xxxx の形で出力します。デフォルトでは、Recovery Manager は RMAN-xxxx 接頭辞を出力しません。
NOCATALOG	リカバリ・カタログなしで Recovery Manager を使用するよう指定します。
SEND ' <i>command</i> '	<p>ベンダー固有のコマンド文字列を割り当てられたチャンネルすべてに送信します。</p> <p><b>関連項目:</b> この機能のサポートの有無は、メディア管理ソフトウェアのドキュメントおよび 2-252 ページの「SEND」を参照してください。</p>
PIPE <i>pipe_name</i>	<p>Recovery Manager パイプ・インタフェースを起動します。Recovery Manager では、コマンドの受信用と出力の送信用に 1 つずつ、2 つのパブリック・パイプが使用されます。パイプ名は PIPE パラメータの値から導出されます。たとえば、オプション PIPE rpi TARGET / を指定して Recovery Manager パイプ・インタフェースを起動できます。</p> <p>Recovery Manager はターゲット・データベース内で次のパイプをオープンします。</p> <ul data-bbox="524 1646 1365 1766" style="list-style-type: none"> <li>ORA\$RMAN_RPI_IN。Recovery Manager はこのパイプを使用してユーザー・コマンドを受信します。</li> <li>ORA\$RMAN_RPI_OUT。Recovery Manager はこのパイプを使用してすべての出力を送信します。</li> </ul> <p>入力パイプと出力パイプに関するメッセージは、すべて VARCHAR2 型です。</p> <p><b>関連項目:</b> パイプを通じて Recovery Manager にコマンドを渡す方法については、『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』を参照してください。</p>

構文の要素	説明
SCRIPT <i>script_name</i>	<p>ストアド・スクリプトの名前を指定します。</p> <p>Recovery Manager は、ターゲット・データベースおよびリカバリ・カタログ (TARGET および CATALOG オプションを使用して指定する必要がある) に接続した後、指定したストアド・スクリプトをリカバリ・カタログからターゲット・データベースに対して実行します。ターゲット・データベースに <i>script_name</i> という名前を持つグローバル・スクリプトとローカル・ストアド・スクリプトの両方が存在する場合、Recovery Manager はローカル・スクリプトを実行します。</p> <p>ストアド・スクリプト名が数字または Recovery Manager の予約語で始まる場合は、そのスクリプト名を一重引用符で囲む必要があります (1-4 ページの「<a href="#">Recovery Manager の予約語</a>」を参照)。数字で始まるスクリプト名または Recovery Manager の予約語と一致するスクリプト名は作成しないようにする必要があります。</p> <p><b>関連項目:</b> ストアド・スクリプトの詳細は、2-106 ページの「<a href="#">CREATE SCRIPT</a>」を参照してください。</p>
TARGET <i>connectStringSpec</i>	<p>ターゲット・データベースへの接続文字列を指定します。たとえば、TARGET / のように指定します。</p> <p><b>関連項目:</b> 3-12 ページの「<a href="#">connectStringSpec</a>」を参照してください。</p>
TIMEOUT <i>integer</i>	<p><i>integer</i> 秒以内に入力パイプから入力を受け取らなかった場合に、Recovery Manager を自動的に終了させます。TIMEOUT を使用する場合は、PIPE パラメータを指定する必要があります。</p> <p><b>関連項目:</b> パイプを通じて Recovery Manager にコマンドを渡す方法については、『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』を参照してください。</p>
USING { <i>string_or_identifier</i>   <i>integer</i> }	<p>コマンド・ファイルの置換変数で使用する値を 1 つ以上指定します。SQL*Plus の場合と同じく、&amp;1 は最初の値を配置する場所を示し、&amp;2 は 2 番目の値を配置する場所を示し、以降も同様に値が示されます。USING 句で指定した値を Recovery Manager コマンド・ファイルに渡す方法は、2-248 ページの例 2-128 で説明します。</p> <p>置換変数の構文は、&amp;<i>integer</i> の後にオプションでドットが続きます (&amp;1.3 など)。オプションのドットは変数の一部であり、値と置換されますので、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、置換変数 &amp;1.3 が含まれているコマンド・ファイルに値 mybackup を渡すと、その置換結果は mybackup3 になります。</p> <p><b>関連項目:</b> ストアド・スクリプトを実行する場合の USING 句の指定方法は、2-143 ページの「<a href="#">EXECUTE SCRIPT</a>」を参照してください。</p>

## 例

### 例 2-126 デフォルトの NOCATALOG モードでのターゲット・データベースへの Recovery Manager の接続

この例では、オペレーティング・システム・プロンプトでデータベース接続オプションを指定せずに Recovery Manager クライアントを起動します。Recovery Manager プロンプトで、CONNECT コマンドを実行してターゲット・データベースに接続します。CONNECT CATALOG が Recovery Manager プロンプトで実行されなかったため、Recovery Manager はリポジトリ接続を必要とする最初のコマンド (この場合は、BACKUP DATABASE コマンド) の実行時にデフォルトの NOCATALOG モードで接続します。

```
% rman
RMAN> CONNECT TARGET /
RMAN> BACKUP DATABASE;
```

### 例 2-127 補助データベース・インスタンスへの Recovery Manager の接続

この例では、ターゲット・データベース `prod` およびリカバリ・カタログ・データベース `catdb` にはネット・サービス名を使用して接続し、補助データベース・インスタンスにはオペレーティング・システム認証を使用して接続します。

```
% rman TARGET SYS@prod

Recovery Manager: Release 11.1.0.6.0 - Production

Copyright (c) 1982, 2007, Oracle. All rights reserved.

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

RMAN> CONNECT CATALOG rman@catdb

recovery catalog database Password: password
connected to recovery catalog database

RMAN> CONNECT AUXILIARY /
```

### 例 2-128 置換変数の指定

データベースをバックアップする Linux シェル・スクリプトを作成するとします。シェル変数を使用して、実行時に Recovery Manager のバックアップ・スクリプトに引数を渡すことができるようにします。置換変数を使用すると、この問題が解決します。最初に、次のような内容で、`whole_db.cmd` という名前のコマンド・ファイルを作成します。

```
cat > /tmp/whole_db.cmd <<EOF
# name: whole_db.cmd
CONNECT TARGET /
BACKUP TAG &1 COPIES &2 DATABASE FORMAT '/disk2/db_%U';
EXIT;
EOF
```

次に、Linux シェル・スクリプトを次のように記述します。このスクリプトでは、`csh` シェル変数を `tagname` および `copies` に設定します。シェル・スクリプトにより、Recovery Manager が起動され、ターゲット・データベース `prod1` に接続され、`whole_db.cmd` が実行されます。USING 句は、実行時に変数 `tagname` および `copies` の値を Recovery Manager コマンド・ファイルに渡します。

```
#!/bin/csh
# name: runbackup.sh
# usage: use the tag name and number of copies as arguments
set tagname = $argv[1]
set copies = $argv[2]
rman @'/tmp/whole_db.cmd' USING $tagname $copies LOG /tmp/runbackup.out
# note that the preceding line is equivalent to:
# rman @'/tmp/whole_db.cmd' $tagname $copies LOG /tmp/runbackup.out
```

最後に、次のように Linux シェルからシェル・スクリプト `runbackup.sh` を実行し、タグ `Q106` を使用してデータベースのバックアップを 2 つ作成します。

```
% runbackup.sh Q106 2
```

### 例 2-129 コマンド・ファイルの構文のチェック

次のように、コマンド・ファイル `backup_db.cmd` を作成するとします。

```
cat > /tmp/backup_db.cmd <<EOF
CONNECT TARGET /
BACKUP DATABASE;
EXIT;
EOF
```

次の例では、コマンド・ファイル backup\_db.cmd の内容を構文チェックします（例には出力例も含まれます）。

```
% rman CHECKSYNTAX @'/tmp/backup_db.cmd'
```

```
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.
```

```
RMAN> CONNECT TARGET *  
2> BACKUP DATABASE;  
3> EXIT;  
The cmdfile has no syntax errors
```

```
Recovery Manager complete.
```

### 例 2-130 ストアド・スクリプトの実行とメッセージ・ログへの出力の追加

この例では、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た後、ストアド・スクリプト wdbb を実行します。Recovery Manager により、出力がメッセージ・ログ /tmp/wdbb.log に書き込まれます。

```
% rman TARGET / SCRIPT wdbb LOG /tmp/wdbb.log
```

### 例 2-131 Recovery Manager パイプ・インタフェースの起動

この例では、タイムアウト・オプションで 90 秒を指定して、Recovery Manager パイプ newpipe を起動します。

```
% rman PIPE newpipe TARGET / TIMEOUT 90
```

# RUN

## 用途

RUN コマンドを使用すると、一連の **Recovery Manager** コマンドをブロックにグループ化して、順番に実行できます。RUN ブロックの閉じカッコを読み取ると、**Recovery Manager** は、ジョブ・コマンドのリストを 1 つ以上のジョブ手順にコンパイルした後、すぐにその手順を実行します。

## 前提条件

このコマンドは、**Recovery Manager** プロンプトでのみ実行してください。ジョブ・コマンドのリストの前には開きカッコ ( { )、後には閉じカッコ ( } ) が必要です。

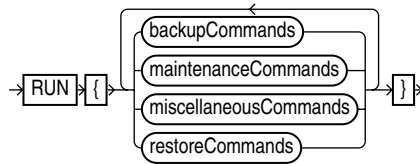
## 使用上の注意

RUN を使用すると、スクリプトでデフォルトの構成をオーバーライドできる有効範囲を作成できます。たとえば、**ALLOCATE CHANNEL** および **RELEASE CHANNEL** コマンドを使用して構成済チャンネルをオーバーライドしたり、**SET** コマンドを使用してその他のパラメータをオーバーライドすることができます (2-251 ページの例 2-132 を参照)。RUN ブロックにリストされているコマンドの実行後に、RUN ブロック内に割り当てられたチャンネルが解放され、設定がそれぞれの値に戻されます。

例 2-133 に示すように、RUN ブロック内で **EXECUTE SCRIPT** コマンドを使用する必要があります。

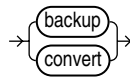
## 構文

**run::=**

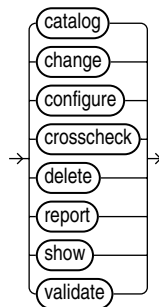


(**backupCommands::=** (2-250 ページ)、**maintenanceCommands::=** (2-250 ページ)、**miscellaneousCommands::=** (2-251 ページ)、**restoreCommands::=** (2-251 ページ))

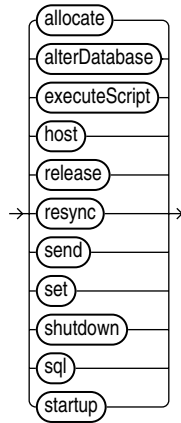
**backupCommands::=**



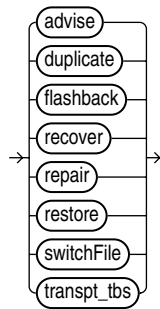
**maintenanceCommands::=**



**miscellaneousCommands::=**



**restoreCommands::=**



**セマンティクス**

Recovery Manager プロンプトから実行できるコマンドについての情報は、各コマンドの項目を参照してください。

**例**

**例 2-132 構成済の設定のオーバーライド**

デバイス構成が次のようになっているとします。

```

RMAN> SHOW DEVICE TYPE;
  
```

```

RMAN configuration parameters for database with db_unique_name PROD1 are:
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
CONFIGURE DEVICE TYPE SBT_TAPE PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
  
```

デフォルト以外のディレクトリにバックアップを作成します。構成を変更するかわりに、次のようにジョブ内でオーバーライドすることもできます。

```

RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE DISK FORMAT "/disk2/%U";
  BACKUP DATABASE PLUS ARCHIVELOG;
}
  
```

**例 2-133 Recovery Manager スクリプトの実行**

**CREATE SCRIPT** コマンドを使用して、`backup_db` という名前のバックアップ・スクリプトを作成するとします。この例では、次のストアド・スクリプトを実行します。

```

RUN { EXECUTE SCRIPT backup_db; }
  
```

## SEND

### 用途

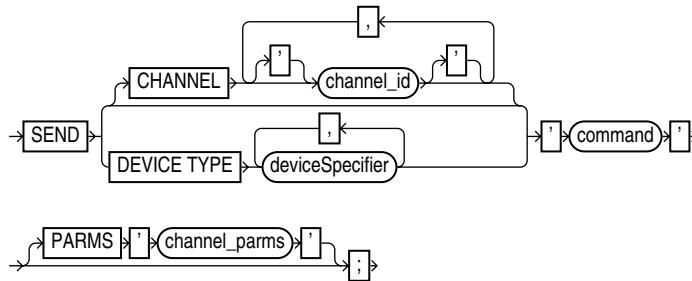
SEND コマンドを使用すると、ベンダー固有の文字列を、メディア・マネージャでサポートされている 1 つ以上のチャンネルに送信できます。どのコマンドがサポートされているかは、使用しているメディア管理のドキュメントを参照してください。

### 使用上の注意

DEVICE TYPE または CHANNEL を指定しないかぎり、Recovery Manager は割り当てられたすべてのチャンネルを使用します。

### 構文

**send::=**



([deviceSpecifier::=](#) (3-15 ページ))

### セマンティクス

構文の要素	説明
CHANNEL <i>channel_id</i>	どのチャンネルを使用するかを指定します。CHANNEL キーワードの後に、大 / 小文字区別があるチャンネル ID、つまりチャンネルの名前を指定する必要があります。データベースでは、このチャンネル ID が I/O エラーのレポートに使用されます。
DEVICE TYPE <i>deviceSpecifier</i>	ストレージ・デバイスのタイプを指定し、コマンドを指定したタイプのすべてのチャンネルに送信します。 <b>関連項目</b> : 3-15 ページの「 <a href="#">deviceSpecifier</a> 」を参照してください。
' <i>command</i> '	ベンダー固有のメディア管理コマンドを指定します。 <b>関連項目</b> : どのコマンドがサポートされているかは、使用しているメディア管理のドキュメントを参照してください。メディア・マネージャでサポートされているコマンドのみを送信してください。文字列の内容は、データベースでは解釈されず、そのままメディア管理サブシステムに渡されます。
PARMS ' <i>channel_parms</i> '	メディア・マネージャと通信するチャンネルのパラメータを指定します。



## 例

### 例 2-134 Oracle Secure Backup でのテープ・ドライブの指定

この例では、SEND コマンドを使用して、users 表領域の Oracle Secure Backup へのバックアップに使用するテープ・ドライブを指定します。パラメータ OB\_DEVICE とテープ・ドライブの名前の間には等号記号は挿入されないことに注意してください。

```
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt;
  SEND 'OB_DEVICE stapel';
  BACKUP TABLESPACE users;
}
```

## SET

### 用途

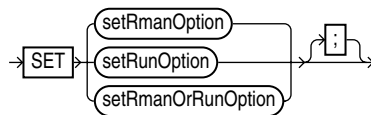
SET コマンドを使用すると、ジョブまたはセッション内の Recovery Manager の動作を制御できます。セッション全体で保持されるオプションを構成するには、[CONFIGURE](#) を使用します。

### 前提条件

SET コマンドは、Recovery Manager プロンプトまたは RUN ブロック内で使用できます。Recovery Manager プロンプトで使用する場合、SET で行われた変更は、Recovery Manager クライアントを終了するまで保持されます (2-255 ページの「[setRmanOption](#)」を参照)。RUN ブロック内で使用する場合、SET で行われた変更は、RUN ブロック、または同じ属性の値を変更する次の SET コマンドが終了するまで保持されます (2-257 ページの「[setRunOption](#)」を参照)。

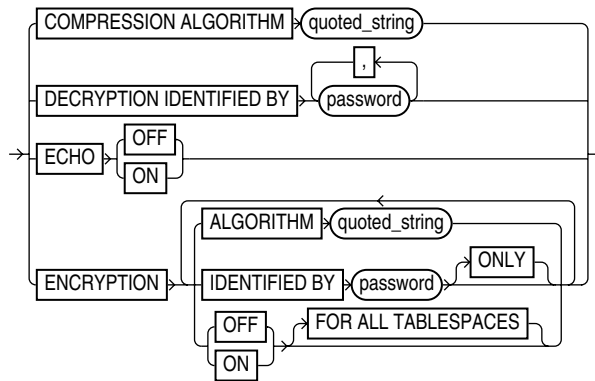
### 構文

**set::=**



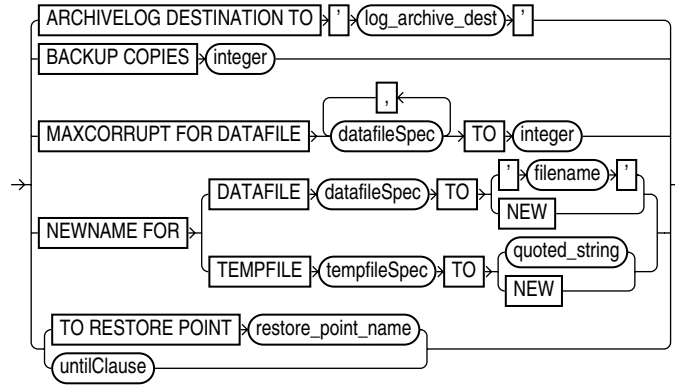
([setRmanOption::=](#) (2-254 ページ)、[setRunOption::=](#) (2-255 ページ))

**setRmanOption::=**



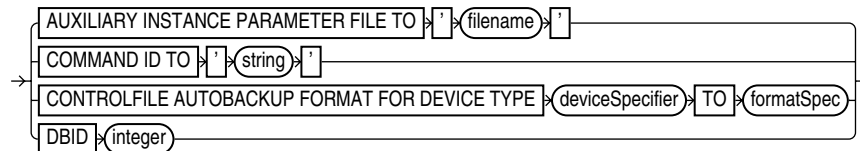
([deviceSpecifier::=](#) (3-15 ページ)、[formatSpec::=](#) (3-21 ページ))

**setRunOption::=**



(deviceSpecifier::= (3-15 ページ)、formatSpec::= (3-21 ページ)、  
datafileSpec::= (3-14 ページ)、tempfileSpec::= (3-37 ページ)、untilClause::= (3-38 ページ))

**setRmanOrRunOption::=**



(deviceSpecifier::= (3-15 ページ)、formatSpec::= (3-21 ページ))

## セマンティクス

### setRmanOption

この副次句では、**RUN** ブロック外で使用可能な **SET** オプションを指定します。

構文の要素	説明
COMPRESSION ALGORITHM 'algorithm_name'	バックアップ・セットの圧縮アルゴリズムを指定します。このコマンドは、現行の Recovery Manager セッションの現行の <b>CONFIGURE</b> COMPRESSION ALGORITHM 設定をオーバーライドします。デフォルトの圧縮アルゴリズムは、BZIP2 です。  ZLIB または BZIP2 を指定します。ZLIB 圧縮は非常に高速ですが、圧縮率は他のアルゴリズムより低くなります。BZIP2 では圧縮率が非常に高くなりますが、速度が少し遅くなります。つまり、ZLIB アルゴリズムは BZIP2 よりも処理が速くなりますが、速度が向上するかわりに圧縮率が少し低くなります。ZLIB 圧縮 (Oracle Advanced Compression オプションが必要) を使用する場合は、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。  <b>注意:</b> サポートされているアルゴリズムは、V\$RMAN_COMPRESSION_ALGORITHM ビューに示されています。

構文の要素	説明
DECRYPTION IDENTIFIED BY <i>password</i>	<p>デュアル・モードのバックアップまたはパスワードで暗号化されたバックアップを読み取る際に使用する 1 つ以上の復号化パスワードを指定します。</p> <p>パスワードで暗号化されたバックアップでは、正しいパスワードを入力しないとリストアできません。Recovery Manager では、暗号化されたバックアップ・ピースが読み取られると、そのバックアップ・ピースを複合化するための正しいパスワードが検出されるまでリスト内の各パスワードが試用されます。指定したいいずれのキーも有効でない場合は、Recovery Manager によりエラーが発行されます。</p> <p><b>注意:</b> 別の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、SET DECRYPTION コマンドで必要なパスワードをすべて指定します。各バックアップ・セットで正しいパスワードが自動的に使用されます。</p> <p><b>関連項目:</b> 2-20 ページの「バックアップ・セットの暗号化」を参照してください。</p>
ECHO {OFF   ON}	<p>Recovery Manager コマンドをメッセージ・ログに表示するかどうかを制御します。コマンド・ファイルからコマンドを読み込むとき、Recovery Manager はそれらのコマンドを自動的にメッセージ・ログに表示します。標準入力からコマンドを読み取る場合、デフォルトでは、Recovery Manager はそれらのコマンドをメッセージ・ログに表示しません。Recovery Manager でコマンドを表示するには、コマンド・ファイルを実行する前に SET ECHO ON コマンドを実行します。SET ECHO OFF を実行して、コマンド・ログへの表示を無効化します。</p> <p>このコマンドは、stdin および stdout がリダイレクトされた場合に有効です。たとえば、UNIX ではこの方法で Recovery Manager の入力と出力をリダイレクトできます。</p> <pre data-bbox="524 884 922 911">% rman TARGET / &lt; in_file &gt; out_file</pre> <p>in_file に SET ECHO ON を含めると、in_file に含まれているコマンドを out_file に表示できます。</p>
ENCRYPTION	<p>Recovery Manager セッションの実行時に、バックアップ・セットを作成する BACKUP コマンドを適用する暗号化関連オプションを指定します。</p> <p><b>関連項目:</b> 2-20 ページの「バックアップ・セットの暗号化」を参照してください。</p>
ALGORITHM ' <i>algorithm_name</i> '	<p>この Recovery Manager セッション実行時に使用するアルゴリズムを指定します。CONFIGURE ALGORITHM で指定されているデフォルトの構成済暗号化アルゴリズムをオーバーライドします。V\$RMAN_ENCRYPTION_ALGORITHMS に、使用可能な値が示されています。</p>

構文の要素	説明
IDENTIFIED BY <i>password</i> [ONLY]	<p>次の規則に従って、バックアップの暗号化でユーザー指定のパスワードを採用するかどうかを指定します。</p> <ul style="list-style-type: none"> <li>透過モードの暗号化バックアップを指定するには、IDENTIFIED BY <i>password</i> 句を省略します。</li> <li>パスワード・モードの暗号化バックアップを指定するには、IDENTIFIED BY <i>password</i> ONLY を使用します。</li> <li>デュアル・モードの暗号化バックアップを指定するには、ONLY を指定せずに IDENTIFIED BY <i>password</i> を使用します。</li> </ul> <p>安全なパスワードを作成します。詳細は、『Oracle Database セキュリティ・ガイド』を参照してください。</p> <p>パスワードは、引用符で囲まない場合、内部で大文字に変換されることに注意してください。したがって、次の句は、すべて IDENTIFIED BY "SOMEPWD" のシノニムとなります。</p> <ul style="list-style-type: none"> <li>IDENTIFIED BY somepwd</li> <li>IDENTIFIED BY Somepwd</li> <li>IDENTIFIED BY sOmEpWd</li> </ul> <p><b>注意:</b> Wallet ベースの暗号化は、パスワードが含まれないため、パスワード・ベースの暗号化よりも安全です。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、必要な場合のみ使用してください。</p> <p><b>関連項目:</b> 様々な暗号化モードの詳細は、2-20 ページの「バックアップ・セットの暗号化」を参照してください。</p>
{OFF   ON}	<p>バックアップ・セットを暗号化するかどうかを指定します。ON と指定すると、デフォルトでバックアップ・セットが暗号化されます。OFF と指定すると、デフォルトではバックアップ・セットは暗号化されません。</p> <p>このオプションは、CONFIGURE ENCRYPTION FOR コマンドで作成された設定をオーバーライドします。暗号化用に構成されているデータファイルがない場合は、明示的に ON を使用して必要なデータファイルを暗号化してください。</p> <p>FOR ALL TABLESPACES が指定されていない場合は、この設定によって、CONFIGURE ENCRYPTION FOR TABLESPACE <i>tablespace_name</i> が暗号化動作の制御に使用されていない表領域のバックアップの暗号化が制御されます。</p>
FOR ALL TABLESPACES	<p>すべての表領域の暗号化を制御し、CONFIGURE ENCRYPTION FOR TABLESPACE <i>tablespace_name</i> 設定をすべてオーバーライドします。</p>

### setRunOption

この副次句では、RUN ブロック内で使用可能な SET オプションを指定します。

構文の要素	説明
ARCHIVELOG DESTINATION TO 'log_archive_dest'	<p>後続の RESTORE および RECOVER コマンドでリストアされるアーカイブ REDO ログの名前を構成するときに、ターゲット・データベースの LOG_ARCHIVE_DEST_1 初期化パラメータをオーバーライドします。Recovery Manager は、'log_archive_dest' に指定した宛先にログをリストアします。</p> <p>このコマンドを使用すると、データベースのリストア中に、異なる位置にアーカイブ・ログを移動できます。Recovery Manager は新しくリストアされたアーカイブ・ログがどこにあるかを認識しています。アーカイブ・ログが LOG_ARCHIVE_DEST_1 によって指定された宛先にある必要はありません。たとえば、パラメータ・ファイルで指定した宛先とは異なる宛先を指定してアーカイブ・ログのバックアップをリストアする場合、後続のリストアおよびリカバリ操作では新しい位置が検出されます。</p> <p>すでにディスクには存在していないアーカイブ REDO ログのリストアに、このパラメータを使用します。Recovery Manager は、ログをバックアップからリストアする前に、それがディスク上にあるかどうかを必ず最初に調べます。</p>

構文の要素	説明
BACKUP COPIES <i>integer</i>	<p>チャンネルが作成する必要がある各バックアップ・ピースのコピー数として 1、2、3 または 4 を指定します (2-261 ページの例 2-136 を参照)。</p> <p>Recovery Manager は、バックアップをディスクまたはテープのいずれかに多重化できますが、テープとディスクに同時に多重化することはできません。テープにバックアップを行う場合は、コピー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ます。また、BACKUP COPIES が 2 以上の場合、ターゲット・データベースで BACKUP_TAPE_IO_SLAVES 初期化パラメータを有効にする必要があります。</p> <p>SET BACKUP COPIES コマンドは、SET BACKUP COPIES の (前ではなく) 後に発行された RUN ブロック内のすべての BACKUP コマンドに影響し、明示的に無効化する変更するまで有効になっています。SET BACKUP COPIES コマンドは、BACKUP コマンドのみに影響しますが、BACKUP AS COPY コマンドには適用されません。</p> <p>SET BACKUP COPIES コマンドは、セッションで割当て済のすべてのチャンネルに影響します。優先順位は次のとおりで、リストの上位にある設定で下位にある設定がオーバーライドされます。</p> <ol style="list-style-type: none"> <li>1. BACKUP COPIES</li> <li>2. SET BACKUP COPIES</li> <li>3. CONFIGURE ... BACKUP COPIES</li> </ol> <p>バックアップ・ピースの名前は、BACKUP コマンドの FORMAT 句に依存します。指定できる FORMAT 文字列は 4 つ以内です。Recovery Manager で 2 番目、3 番目および 4 番目の値が使用されるのは、BACKUP COPIES、SET BACKUP COPIES、または CONFIGURE ... BACKUP COPIES が有効な場合のみです。各バックアップ・ピースに使用する形式を選択すると、Recovery Manager では最初の形式値がコピー 1、2 番目の形式値がコピー 2 というように順番に使用されます。形式値の数がコピー数より多ければ、余分の形式は使用されません。形式値の数がコピー数より少なければ、Recovery Manager では最初の形式値から順番に再利用されます。</p> <p><b>注意:</b> BACKUP COPIES オプションは、ファイルがフラッシュ・リカバリ領域に作成されている場合は有効ではありません。フラッシュ・リカバリ領域へのバックアップは多重化できません。</p> <p><b>注意:</b> 制御ファイルのディスクへの自動バックアップは特殊ケースであり、多重化されることはありません。Recovery Manager が書き込むコピーは常に 1 つのみです。</p> <p>指定したデータファイルまたはデータファイルのグループ内でデータベースが許容する未検出のブロック破損数に制限を設定します。デフォルトは 0 (ゼロ) で、Recovery Manager が破損ブロックを許容しないことを意味します。</p> <p>SET MAXCORRUPT コマンドによって、バックアップまたはリストア・ジョブ中にデータファイルに許容される物理的および論理的な破損の合計数が指定されます。データファイルで検出された物理的な破損と論理的な破損の合計数が MAXCORRUPT の設定値以下の場合、BACKUP または RESTORE コマンドは最後まで実行されます。破損ブロックが MAXCORRUPT より多く存在する場合は、出力ファイルが生成されずに Recovery Manager が終了します。</p> <p>MAXCORRUPT の制限を超えたかどうかに関係なく、見つかった破損ブロックの範囲が V\$DATABASE_BLOCK_CORRUPTION ビューに移入されます。ただし、バックアップまたはリストア・ジョブは、MAXCORRUPT+1 個の破損ブロックが検出されると終了するため、この場合に Recovery Manager で記録される破損数は MAXCORRUPT+1 のみとなります。バックアップまたはリストア・ジョブの終了時点を超えるとブロック破損は記録されません。</p> <p><b>注意:</b> CHECK LOGICAL を指定した場合、MAXCORRUPT は、検出された論理的および物理的な破損の合計に適用されます。そうでない場合、MAXCORRUPT は、物理的な破損の数にのみ適用されます。</p> <p><b>関連項目:</b> 3-14 ページの「<a href="#">datafileSpec</a>」を参照してください。</p>
MAXCORRUPT FOR DATAFILE <i>datafileSpec</i> TO <i>integer</i>	

構文の要素	説明
NEWNAME FOR DATAFILE <i>datafileSpec</i>	<p>指定したデータファイルに影響を与える、後続のすべての <b>RESTORE</b> コマンドまたは <b>SWITCH</b> コマンドについて、デフォルト名を設定します。データファイル・リストア操作の前にこのコマンドを発行しない場合、<b>Recovery Manager</b> はファイルをそのデフォルトの位置にリストアします。</p> <p>データファイルを新しい位置にリストアすると、<b>SWITCH</b> を実行して制御ファイル内でファイルの名前を <b>NEWNAME</b> に変更できます。<b>SWITCH</b> を実行しなければ、リストアされたファイルはデータファイルのコピーとしてリポジトリに記録されません。</p> <p>複製データベースやスタンバイ・データベースの作成時、または <b>Recovery Manager</b> の <b>TSPITR</b> の実行時には、<b>SET NEWNAME TO NEW</b> を使用できません。</p> <p><b>注意:</b> <b>SET NEWNAME</b> コマンドは <b>ASM</b> ディスク・グループをサポートします。</p> <p><b>関連項目:</b> 3-14 ページの「<i>datafileSpec</i>」を参照してください。</p>
NEWNAME FOR TEMPFILE <i>tempfileSpec</i>	<p>後続の <b>SWITCH</b> コマンドに新しい一時ファイル名を指定します。このコマンドは指定した一時ファイルを指定した名前に変更します。</p> <p>複製データベースやスタンバイ・データベースの作成時、または <b>Recovery Manager</b> の <b>TSPITR</b> の実行時には、<b>SET NEWNAME TO NEW</b> を使用できません。</p> <p><b>注意:</b> <b>SET NEWNAME</b> コマンドは <b>ASM</b> ディスク・グループをサポートします。</p> <p><b>関連項目:</b> 3-37 ページの「<i>tempfileSpec</i>」を参照してください。</p>
TO 'filename'	<p>リストアされるデータファイルまたは一時ファイルに対して、ユーザー定義ファイル名または自動ストレージ管理ディスク・グループを指定します。<b>NEWNAME</b> をデータファイルのディスク・グループに設定して <b>RESTORE</b> を実行すると、<b>Recovery Manager</b> によってファイルがディスク・グループにリストアされます。一時ファイルのファイル名を指定すると、データベースがリカバリされて、オープンされた後、このファイル名は一時ファイルの新しい名前になります。</p>
TO NEW	<p><b>DB_CREATE_FILE_DEST</b> で指定したディレクトリに <b>Oracle</b> 管理ファイルを作成します。元のファイルが <b>Oracle</b> 管理ファイルであるか、または自動ストレージ管理ディスク・グループに存在する場合は、<b>Recovery Manager</b> によって元のファイルの削除が試行されます。一時ファイルに対して <b>TO NEW</b> を指定すると、データベースのオープン時に、<b>DB_CREATE_FILE_DEST</b> で一時ファイルが作成されます。</p> <p><b>DUPLICATE</b> コマンドを使用する場合、または <b>Recovery Manager</b> の <b>TSPITR</b> を実行する場合は、このオプションを使用できません。</p> <p><b>関連項目:</b> <b>Oracle Managed Files</b> の詳細は、『<b>Oracle Database 管理者ガイド</b>』を参照してください。</p>
TO RESTORE POINT <i>restore_point_name</i>	<p>リストア・ポイントを作成した時点の <b>SCN</b> を上限として、後続の <b>RESTORE</b> または <b>RECOVER</b> コマンドのリストア・ポイントを指定します。指定した値は含まれます。上限値が含まれるため、<b>Recovery Manager</b> は、リストア・ポイントに対応する <b>SCN</b> までファイルを選択またはリカバリできるファイルのみを選択します。</p> <p><b>注意:</b> 定義済みのリストア・ポイントは制御ファイルに記録されているため、<b>SET TO RESTORE POINT</b> はデータベースがマウントされている場合のみ使用できます。たとえば、<b>SET TO RESTORE POINT</b> は、<b>RESTORE CONTROLFILE</b> 操作のターゲット <b>SCN</b> の指定には使用できません。</p>
<i>untilClause</i>	<p>後続の <b>RESTORE</b> または <b>RECOVER</b> コマンドで使用する終了時刻、<b>SCN</b> またはログ順序番号を指定します。</p> <p><b>関連項目:</b> 3-38 ページの「<i>untilClause</i>」を参照してください。</p>

**setRmanOrRunOption**

この副次句では、**RUN** ブロックの内部または外部で使用可能な **SET** オプションを指定します。

構文の要素	説明
AUXILIARY INSTANCE PARAMETER FILE TO 'filename'	<p>インスタンスの起動に使用するパラメータ・ファイルへのパスを指定します。このパラメータは、自動補助インスタンスで <b>TSPITR</b> をカスタマイズする場合、または、<b>Recovery Manager</b> で <b>Recovery Manager</b> 表領域をクローニングする場合に使用できます。</p> <p><b>注意:</b> <i>filename</i> は、<b>Recovery Manager</b> クライアントを実行するホスト上にあります。</p> <p><b>関連項目:</b> <code>V\$SESSION.CLIENT_INFO</code> の詳細は、『<b>Oracle Database リファレンス</b>』を参照してください。</p>
COMMAND ID TO 'string'	<p>指定した文字列をすべてのチャンネルの <code>V\$SESSION.CLIENT_INFO</code> 列に入力します。この情報は、データベース・サーバー・セッションと <b>Recovery Manager</b> のチャンネルの対応関係の確認に使用します。<b>SET COMMAND ID</b> コマンドは、すでに割当て済のチャンネルにのみ適用されます。</p> <p><code>V\$SESSION.CLIENT_INFO</code> 列には、各 <b>Recovery Manager</b> サーバー・セッションに関する情報があります。データの形式は、次の形式のいずれかです。</p> <ul style="list-style-type: none"> <li>■ <code>id=string</code></li> <li>■ <code>id=string,ch=channel_id</code></li> </ul> <p>1 番目の形式は、<b>Recovery Manager</b> ターゲット・データベース接続で使用されます。2 番目の形式は、割り当てられたすべてのチャンネルについて使用されます。現行のジョブが完了すると、<code>V\$SESSION.CLIENT_INFO</code> 列は消去されます。</p> <p><b>関連項目:</b> <code>V\$SESSION.CLIENT_INFO</code> の詳細は、『<b>Oracle Database リファレンス</b>』を参照してください。</p>
CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE <i>deviceSpecifier</i> TO <i>formatSpec</i>	<p>指定したデバイス・タイプの制御ファイルの自動バックアップについて、デフォルトのファイル名形式をオーバーライドします。このコマンドは <b>RUN</b> コマンド内または <b>Recovery Manager</b> プロンプトで使用できます。優先順位は次のとおりです。</p> <ol style="list-style-type: none"> <li>1. <b>RUN</b> ブロック内で実行される <b>SET CONTROLFILE AUTOBACKUP</b></li> <li>Recovery Manager プロンプトで実行される</li> <li>1. <b>SET CONTROLFILE AUTOBACKUP</b></li> <li>2. <b>CONFIGURE CONTROLFILE AUTOBACKUP FORMAT</b></li> </ol> <p>新しい <i>formatSpec</i> には、置換変数 <code>%F</code> が必要です。他の置換変数は、制御ファイルの自動バックアップ <i>formatSpec</i> では有効ではありません。</p> <p><b>関連項目:</b> <code>%F</code> 置換変数の意味については、3-21 ページの「<i>formatSpec</i>」を参照してください。</p>
DBID <i>integer</i>	<p><b>DBID</b> を指定します。データベースの作成時に計算される一意で 32 ビットの識別番号です。</p> <p><b>Recovery Manager</b> は、ターゲット・データベースへの接続時に <b>DBID</b> を表示します。<b>DBID</b> を取得するには、<code>V\$DATABASE</code> ビューまたは <code>RC_DATABASE</code> および <code>RC_DATABASE_INCARNATION</code> リカバリ・カタログ・ビューに問い合わせます。</p> <p><b>SET DBID</b> コマンドは、次の特殊な状況でのみ実行してください。</p> <ul style="list-style-type: none"> <li>■ リカバリ・カタログに接続せずに、制御ファイルをリストアする必要がある場合 (2-261 ページの例 2-137 を参照)。データ・リカバリ・アドバイザを使用して制御ファイルの自動バックアップをリストアする場合は、同じ制限が適用されます。<b>CONFIGURE</b> を使用すると、<b>ADVISE FAILURE</b> の前に <b>SET DBID</b> が発行されている場合のみ、自動バックアップを検索してリストアできます。</li> <li>■ サーバー・パラメータ・ファイルのリストアを行う必要がある場合 (例 2-138 を参照)。</li> <li>■ リカバリ・カタログに接続しているがターゲット・データベースには接続しておらず、<b>CONFIGURE</b>、<b>LIST</b>、<b>REPORT</b>、<b>SHOW</b> または <b>UNREGISTER</b> コマンドで <code>FOR DB_UNIQUE_NAME</code> オプションを使用する必要がある場合。</li> </ul>



## 例

**例 2-135 コマンド ID の設定**

この例では、コマンド ID を rman に設定し、データベースをバックアップして、オンライン REDO ログをアーカイブします。コマンド ID を使用すると、WHERE LIKE '%rman%' を指定した V\$SESSION への問合せで、ジョブのステータス情報がわかります。

```
RUN
{
  ALLOCATE CHANNEL d1 DEVICE TYPE DISK FORMAT '/disk1/%U';
  ALLOCATE CHANNEL d2 DEVICE TYPE DISK FORMAT '/disk2/%U';
  SET COMMAND ID TO 'rman';
  BACKUP INCREMENTAL LEVEL 0 DATABASE;
  SQL 'ALTER SYSTEM ARCHIVE LOG CURRENT';
}
```

**例 2-136 バックアップ・セットの多重化**

現行の多重化構成が次のようになっているとします。

```
CONFIGURE ARCHIVELOG COPIES FOR DEVICE TYPE sbt TO 3;
CONFIGURE DATAFILE COPIES FOR DEVICE TYPE sbt TO 3;
```

テープ・ドライブのうち 1 台が故障し、使用できるテープ・ドライブが 2 台のみになりました。テープ・バックアップでは、一般に、チャンネル数とコピー数を掛けた数が、デバイスの数に等しくなるようにします。次の例では、多重化の永続構成を SET BACKUP COPIES でオーバーライドし、データベース・バックアップの 2 つのコピーを、機能している 2 つのテープ・ドライブに書き出します。

```
RUN
{
  ALLOCATE CHANNEL dev1 DEVICE TYPE sbt
    PARMS 'ENV=(OB_DEVICE_1=stape1,OB_DEVICE_2=stape2)';
  SET BACKUP COPIES 2;
  BACKUP DATABASE PLUS ARCHIVELOG;
}
```

**例 2-137 リストア中の制御ファイルの自動バックアップ形式の設定**

制御ファイルを格納しているディスクに障害が発生したとします。初期化パラメータ・ファイルの CONTROL\_FILES パラメータを編集して、新しい場所を示すようにします。

この例では、リカバリ・カタログへのアクセス権はありません。この例では、インスタンスを起動し、DBID を設定してから、制御ファイルの自動バックアップをリストアします。データベースをマウントした後は、データベースのリカバリを行うことができます。

```
CONNECT TARGET /
STARTUP FORCE NOMOUNT
SET DBID 28014364;
RUN
{
  SET CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '/disk2/cf_%F.bak';
  RESTORE CONTROLFILE FROM AUTOBACKUP MAXSEQ 100;
}
ALTER DATABASE MOUNT;
RECOVER DATABASE;
ALTER DATABASE OPEN RESETLOGS;
```

**例 2-138 サーバー・パラメータ・ファイルのリストア**

データベースのホストでメンテナンスを実行している間にデータベースが停止されたとします。このとき、サーバー・パラメータ・ファイルが誤って削除されました。Recovery Manager クライアントを起動し、TARGET としてデータベースに接続してから、リカバリ・カタログに接続します。次の例では、テープ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してから、インスタンスを再起動します。

```
SET DBID 3257174182; # set dbid so RMAN can identify the database
STARTUP FORCE NOMOUNT # rman starts database with a dummy server parameter file
RUN
{
  ALLOCATE CHANNEL t1 DEVICE TYPE sbt;
  RESTORE SPFILE FROM AUTOBACKUP;
}
STARTUP FORCE; # needed so that RMAN restarts database with restored server parameter file
```

## SHOW

### 用途

SHOW コマンドを使用すると、1つ以上のデータベースについて、現行の Recovery Manager 構成を設定するために使用された CONFIGURE コマンドを表示できます。Recovery Manager のデフォルト構成には、接尾辞 #default が付いています。

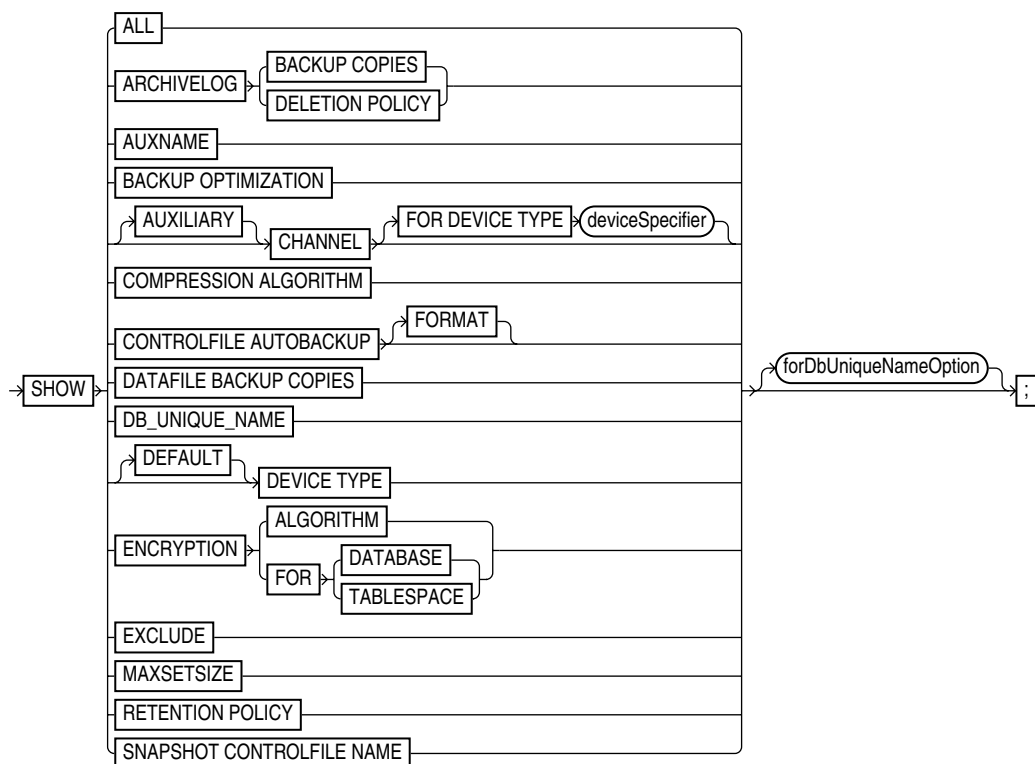
### 前提条件

このコマンドは、Recovery Manager プロンプトでのみ実行してください。次のいずれかの条件が満たされている必要があります。

- Recovery Manager がターゲット・データベースに接続していて、そのデータベースがマウントまたはオープン状態である必要があります。
- Recovery Manager はリカバリ・カタログに接続され、SET DBID が実行されている必要があります。

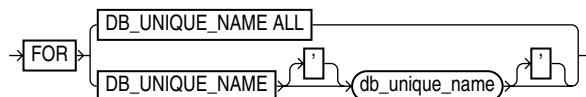
### 構文

**show::=**



(deviceSpecifier::= (3-15 ページ))

**forDbUniqueNameOption::=**



## セマンティクス

構文の要素	説明
ALL	ユーザーが入力したすべての <b>CONFIGURE</b> コマンドとデフォルト構成を表示します。
ARCHIVELOG BACKUP COPIES	アーカイブ REDO ログのバックアップに対して現在構成されている多重化の程度を表示します。
ARCHIVELOG DELETION POLICY	CONFIGURE <b>ARCHIVELOG DELETION POLICY</b> 設定を表示します。
AUXNAME	CONFIGURE AUXNAME 設定を表示します。
BACKUP OPTIMIZATION	CONFIGURE BACKUP OPTIMIZATION 設定を表示します。設定は ON または OFF (デフォルト) です。
[AUXILIARY] CHANNEL	CONFIGURE CHANNEL 設定を表示します。通常のチャンネルまたは AUXILIARY チャンネルを指定できます。
FOR DEVICE TYPE <i>deviceSpecifier</i>	チャンネルのデバイス・タイプを指定します。たとえば、SHOW CHANNEL FOR DEVICE TYPE DISK を使用すると、ディスク・チャンネルのチャンネル設定のみが表示されます。
COMPRESSION ALGORITHM	構成済のバックアップ圧縮アルゴリズムを表示します。
CONTROLFILE AUTOBACKUP	CONFIGURE CONTROLFILE AUTOBACKUP 設定を表示します。設定は ON または OFF です。
FORMAT	構成済デバイスについて、制御ファイルの自動バックアップ・ファイルの形式を表示します。
DATAFILE BACKUP COPIES	データファイルの CONFIGURE ...BACKUP COPIES 設定 (1、2、3 または 4) を表示します。
DB_UNIQUE_NAME	リカバリ・カタログで認識されている DB_UNIQUE_NAME 値を表示します。
[DEFAULT] DEVICE TYPE	構成済のデバイス・タイプと並列度の設定を表示します。DEFAULT を指定すると、SHOW ではデフォルトのデバイス・タイプと設定が表示されます。
ENCRYPTION	ALGORITHM または FOR {DATABASE   TABLESPACE} とともに使用すると、データベースまたはデータベース内の表領域に対して現在構成されている暗号化設定を表示します。
ALGORITHM	暗号化されたバックアップ・セットに書き込む場合に暗号化で使用するデフォルトの構成済アルゴリズムを表示します。V\$RMAN_ENCRYPTION_ALGORITHMS に、使用可能な値が示されています。
FOR DATABASE	データベースの現行の暗号化設定を表示します。
FOR TABLESPACE	各表領域の現行の暗号化設定を表示します。
EXCLUDE	除外するように指定した表領域のみを表示します。
MAXSETSIZE	CONFIGURE MAXSETSIZE 設定を表示します。
RETENTION POLICY	現行のターゲット・データベースに関する CONFIGURE RETENTION POLICY の設定を表示します。
SNAPSHOT CONTROLFILE NAME	CONFIGURE SNAPSHOT CONTROLFILE 設定を表示します。

構文の要素	説明
<code>forDbUniqueNameOption</code>	<p>リカバリ・カタログ内の一意の名前を持つデータベースの構成を表示します。Recovery Manager がこのデータベースに TARGET として接続されていない場合でも同様に表示します。データベースは <code>db_unique_name</code> で指定できますが、ALL を使用すると、一意の名前を持つデータベースをすべて指定できます。</p> <p>データベースの一意の名前は、DB_UNIQUE_NAME 初期化パラメータ設定の値です。FOR DB_UNIQUE_NAME 句は、Data Guard 環境でスタンバイ・データベースの構成を表示する場合に役立ちます。</p> <p>Recovery Manager は、リカバリ・カタログに接続する必要があります。Recovery Manager は、マウント済のターゲット・データベースに接続しているか、SET DBID でターゲット・データベースを指定する必要があります。たとえば、ターゲット・データベースの DBID を指定して SET DBID を実行し、リカバリ・カタログで認識されているすべてのスタンバイ・データベースの構成を表示できます (2-86 ページの例 2-46 を参照)。</p> <p><b>関連項目:</b> この句のオプションについては、3-18 ページの「<code>forDbUniqueNameOption</code>」を参照してください。</p>

## 例

### 例 2-139 ターゲット・データベースのすべての構成の表示

ターゲット・データベースの Recovery Manager のすべての永続構成を把握する必要があります。Recovery Manager クライアントを起動し、ターゲット・データベースおよびリカバリ・カタログに接続して、次のように SHOW コマンドを実行します (例には出力例も含まれません)。

```

RMAN> SHOW ALL;

RMAN configuration parameters for database with db_unique_name PROD1 are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP ON;
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '/disk1/oracle/dbs/%F';
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE SBT_TAPE TO '%F'; # 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 DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE SBT_TAPE TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE SBT_TAPE TO 1; # default
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE'
  PARMS 'SBT_LIBRARY=/usr/local/oracle/backup/lib/libobk.so';
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE ON;
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/disk1/oracle/dbs/cf_snap .f'

```

## SHUTDOWN

### 用途

SHUTDOWN コマンドを使用すると、**Recovery Manager** を終了せずに、ターゲット・データベースを停止できます。このコマンドは、SQL\*Plus の SHUTDOWN 文と同じです。

**関連項目：** データベースの起動と停止の方法は、『Oracle Database 管理者ガイド』を、SHUTDOWN コマンドの構文は、『SQL\*Plus ユーザーズ・ガイドおよびリファレンス』を参照してください。

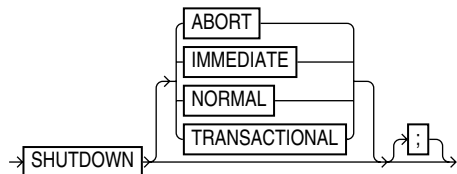
### 使用上の注意

リカバリ・カタログ・データベースの停止には、**Recovery Manager** の SHUTDOWN コマンドは使用できません。リカバリ・カタログ・データベースを停止するには、SQL\*Plus セッションを開始して SHUTDOWN 文を発行します。

データベースを NOARCHIVELOG モードで操作している場合は、データベースを正しく停止し、バックアップの作成前に **STARTUP MOUNT** を発行する必要があります。

### 構文

**shutdown::=**



### セマンティクス

構文の要素	説明
ABORT	<p>ターゲット・インスタンスを一貫性のない状態で停止します。次の結果になります。</p> <ul style="list-style-type: none"> <li>すべての現行のクライアントの SQL 文は、即時終了します。</li> <li>コミットされていないトランザクションは、次の起動までロールバックされません。</li> <li>すべての接続ユーザーは切断されます。</li> <li>次の起動時に、データベースのインスタンス・リカバリが実行されます。</li> </ul>
IMMEDIATE	<p>ターゲット・インスタンスを一貫性のとれた状態で即時停止します。次の結果になります。</p> <ul style="list-style-type: none"> <li>データベースで処理中の現行のクライアントの SQL 文が完了します。</li> <li>コミットされていないトランザクションはロールバックされます。</li> <li>すべての接続ユーザーは切断されます。</li> </ul>
NORMAL	<p>データベースを一貫性のとれた状態で <b>NORMAL</b> モード（デフォルトのオプション）で停止します。これには、次のような意味があります。</p> <ul style="list-style-type: none"> <li>文の発行後は、新しい接続ができません。</li> <li>データベースは、現在の接続ユーザーが切断するまで待機してから停止します。</li> <li>次のデータベース起動時には、インスタンス・リカバリの必要がありません。</li> </ul>

構文の要素	説明
TRANSACTIONAL	<p>ターゲット・データベースを一貫性のとれた状態でクライアントへの中断を最小化して停止します。次の結果になります。</p> <ul style="list-style-type: none"> <li>■ 現在トランザクションを進めているクライアントは、データベースの停止前にトランザクションをコミットするか、終了します。</li> <li>■ このインスタンス時には、どのクライアントも新しいトランザクションを開始できません。新しいトランザクションを開始しようとするクライアントは切断されます。</li> <li>■ すべてのトランザクションがコミットするかまたは終了した後に、接続中のクライアントが切断されます。</li> </ul>

## 例

### 例 2-140 IMMEDIATE オプションを指定したデータベースの停止

この例では、現行の SQL トランザクションが処理されるのを待ってデータベースを停止し、その後でデータベースをマウントします。

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

### 例 2-141 NOARCHIVELOG モードでのデータベースの停止

この例では、NOARCHIVELOG モードで実行中のデータベースをバックアップします。

```
STARTUP FORCE DBA;
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
# executing the preceding commands ensures that database is in proper state
# for NOARCHIVELOG backups
BACKUP DATABASE;
ALTER DATABASE OPEN;
```

## SPOOL

### 用途

SPOOL コマンドを使用すると、Recovery Manager による出力の書き込み先のログ・ファイルを指定できます。

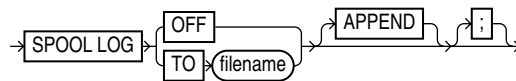
**関連項目：** LOG ファイルの詳細は、2-244 ページの「[RMAN](#)」を参照してください。

### 前提条件

SPOOL コマンドは、Recovery Manager プロンプトで実行してください。

### 構文

**spool::=**



### セマンティクス

構文の要素	説明
OFF	スプーリングをオフにします。
TO <i>filename</i>	Recovery Manager による出力の書き込み先のログ・ファイルの名前を指定します。Recovery Manager は、ファイルが存在しない場合は作成し、存在する場合は上書きします。指定したファイルが書き込み用にオープンできない場合、Recovery Manager は SPOOL を OFF に変更して動作を継続します。
APPEND	Recovery Manager の出力を既存のログの最後に追加します。

### 例

#### 例 2-142 ファイルへの Recovery Manager 出力のスプーリング

この例では、デフォルトのデバイス・タイプの構成に関する Recovery Manager 出力は標準出力に書き込み、[SHOW](#) コマンドの出力はログ・ファイル `current_config.log` にスプーリングし、データベース全体のバックアップの出力は `db_backup.log` にスプーリングします。

```

CONFIGURE DEFAULT DEVICE TYPE TO sbt;
SPOOL LOG TO '/tmp/current_config.log';
SHOW ALL;
SPOOL LOG OFF;
SPOOL LOG TO '/tmp/db_backup.log';
BACKUP DATABASE;
SPOOL LOG OFF;

```



## SQL

### 用途

SQL コマンドを使用すると、Recovery Manager 内から SQL 文または PL/SQL ストアド・プロシージャを実行できます。

**関連項目：**『Oracle Database SQL 言語リファレンス』

### 前提条件

なし。

### 構文

**sql::=**



### セマンティクス

構文の要素	説明
CHANNEL <i>channel_id</i>	<p><b>RUN</b> コマンド内で Recovery Manager のコマンドを実行するときに使用するチャンネルの名前を指定します。この名前は大 / 小文字が区別されます。</p> <p>チャンネルは、この RUN コマンド内の <b>ALLOCATE CHANNEL</b> で割り当てられている必要があります。このパラメータを設定しなかった場合、Recovery Manager はデフォルトのチャンネルを使用します。</p>
' <i>command</i> '	<p>SQL 文の実行を指定します (2-269 ページの例 2-143 を参照)。SELECT 文は使用できません。</p> <p>引用符で囲んだ文字列に同じ引用スタイルを使用する場合、引用符で囲んだ文字列に一重引用符を挿入するためには、2 組の一重引用符を使用する必要があります。たとえば、Recovery Manager が SQL に渡す文字列にファイル名がある場合は、ファイル名を 2 組の一重引用符で囲み、SQL キーワードに続く文字列全体を二重引用符で囲む必要があります (2-270 ページの例 2-144 を参照)。</p> <p><b>注意：</b> EXECUTE は SQL*Plus コマンドであるため、Recovery Manager の SQL コマンド内で EXECUTE を指定して PL/SQL プログラムを実行することはできません。かわりに、BEGIN および END キーワードを使用する必要があります。たとえば、SQL コマンドを使用して PL/SQL プロシージャ rman.rman_purge を実行するには、次のコマンドを発行します。</p> <pre>SQL 'BEGIN rman.rman_purge; END;';</pre>

### 例

#### 例 2-143 アーカイブされていないオンライン・ログのアーカイブ

この例では、表領域をバックアップしてから、アーカイブされていないオンライン REDO ログをすべてアーカイブします。

```
BACKUP TABLESPACE users;
SQL "ALTER SYSTEM ARCHIVE LOG CURRENT";
```

**例 2-144 引用符付き文字列内のファイル名の指定**

この例では、二重引用符付き文字列のコンテキスト内で、2組の一重引用符を使用してファイル名を指定します。

```
SQL "ALTER TABLESPACE users ADD DATAFILE '/disk1/oradata/users02.dbf'
    SIZE 100K AUTOEXTEND ON NEXT 10K MAXSIZE 100K";
```

## STARTUP

### 用途

STARTUP コマンドを使用すると、Recovery Manager 環境内からターゲット・データベースを起動できます。このコマンドは、SQL\*Plus の STARTUP コマンドと同じです。

また、Recovery Manager の STARTUP コマンドでは、サーバー・パラメータ・ファイルや初期化パラメータ・ファイルがない場合にも、NOMOUNT モードでインスタンスを起動できます。この機能は、消失したサーバー・パラメータ・ファイルのリストアを必要とする場合に役立ちます。

**関連項目：** データベースの起動と停止の方法は、『Oracle Database 管理者ガイド』を、SQL\*Plus の STARTUP コマンドの構文は、『SQL\*Plus ユーザーズ・ガイドおよびリファレンス』を参照してください。

### 前提条件

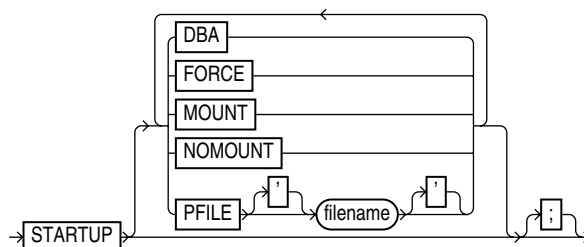
Recovery Manager がターゲット・データベースに接続している必要があります。このコマンドは、ターゲット・データベースの起動のみに使用できます。

### 使用上の注意

Recovery Manager の STARTUP コマンドでは、サーバー・パラメータ・ファイルや初期化パラメータ・ファイルがない場合にも、NOMOUNT モードでインスタンスを起動できます。この機能は、消失したサーバー・パラメータ・ファイルのリストアを必要とする場合に役立ちます (2-272 ページの例 2-146 を参照)。

### 構文

**startup::=**



### セマンティクス

構文の要素	説明
STARTUP	STARTUP のみを指定して他のオプションを指定しない場合、インスタンスはデフォルトのサーバー・パラメータ・ファイルを使用して起動され、制御ファイルがマウントされた後、データベースがオープンされます。
DBA	アクセスを RESTRICTED SESSION 権限を持つユーザーに制限します。
FORCE	データベースがオープン状態の場合、FORCE は、データベースを再オープンする前に SHUTDOWN ABORT 文で停止します。データベースがクローズ状態の場合、FORCE はデータベースをオープンします。
MOUNT	インスタンスを起動してからデータベースをマウントしますが、オープンはしません。

構文の要素	説明
NOMOUNT	データベースをマウントせずにインスタンスを起動します。パラメータ・ファイルが存在しない場合、Recovery Manager は一時パラメータ・ファイルでインスタンスを起動します。 <b>RESTORE SPFILE</b> を実行すると、バックアップ・サーバー・パラメータ・ファイルをリストアできます。
PFILE <i>filename</i>	ターゲット・データベースで使用するテキスト・ベースの初期化パラメータ・ファイルのファイル名を指定します。PFILE を指定しなければ、デフォルトの初期化パラメータ・ファイルの名前が使用されます。

## 例

### 例 2-145 パラメータ・ファイルの指定によるデータベースのマウント

この例では、SHUTDOWN ABORT を実行してから、非デフォルトの初期化パラメータ・ファイルの位置を指定し、制限付きアクセスでデータベースをマウントします。

```
CONNECT TARGET /
STARTUP FORCE MOUNT DBA PFILE=/tmp/initPROD.ora;
```

### 例 2-146 パラメータ・ファイルを使用しないインスタンスの起動

サーバー・パラメータ・ファイルが誤ってファイル・システムから削除されたとします。次の例では、パラメータ・ファイルを使用せずにインスタンスを起動して、**RESTORE SPFILE FROM AUTOBACKUP** を実行します。この例では、自動バックアップの場所がフラッシュ・リカバリ領域であるため、SET DBID は必要ありません。

```
CONNECT TARGET /
STARTUP FORCE NOMOUNT; # RMAN starts instance with dummy parameter file
RESTORE SPFILE TO '?/dbs/spfileprod.ora'
FROM AUTOBACKUP
RECOVERY AREA '/disk2' DB_NAME='prod';
STARTUP FORCE; # restart instance with restored server parameter file
```

## SWITCH

### 用途

SWITCH コマンドを使用すると、次の操作のいずれかを実行できます。

- データベース、表領域またはデータファイルのファイル名を、指定したファイルに使用可能な最新のイメージ・コピーに更新します。
- **SET NEWNAME** コマンドが発行されたデータファイルおよび一時ファイルのファイル名を更新します。

SWITCH は、SQL 文 ALTER DATABASE RENAME FILE を使用した場合と同じ結果になります。Recovery Manager リポジトリ内のファイルの名前は更新されますが、データベースは、オペレーティング・システム・レベルでは名前を変更しません。

### 前提条件

Recovery Manager がターゲット・データベースに接続している必要があります。表領域、データファイルまたは一時ファイルを切り替えるときは、ファイルがオフライン状態になっている必要があります。データベース全体を切り替えるときは、データベースをオープンしないでください。

### 使用上の注意

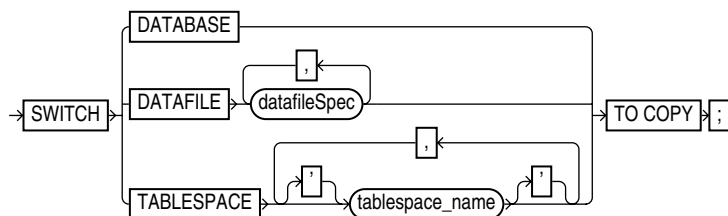
SWITCH コマンドによって、データファイルのコピーの Recovery Manager リポジトリ・レコードがリカバリ・カタログから削除され、制御ファイル・レコードの状態が DELETED に更新されます。

Recovery Manager がリカバリ・カタログに接続されていて、バックアップからリストアされた制御ファイルをデータベースで使用している場合は、SWITCH によって、リカバリ・カタログでは認識されているが、制御ファイルでは不明なデータファイルのレコードで制御ファイルが更新されます。

SWITCH ...TO COPY は Recovery Manager プロンプトでのみ実行できます。TO COPY を指定しない SWITCH は、**RUN** ブロック内でのみ使用できます。

### 構文

**switch::=**



([datafileSpec::=](#) (3-14 ページ))



構文の要素	説明
TEMPFILE <i>tempfileSpec</i>	名前を変更する一時ファイルを指定します。TO オプションを指定しないと、Recovery Manager は、このファイルに対して RUN ブロック内の以前の SET NEWNAME コマンドで切替え先として指定したファイル名を使用します。ターゲット・データベースはマウントする必要がありますが、オープンはしないでください。
TO ' <i>filename</i> '	一時ファイルの名前を指定した名前に変更します (2-275 ページの例 2-149 を参照)。ターゲット・データベースはマウントする必要がありますが、オープンはしないでください。

## 例

### 例 2-147 バックアップからのリストアを回避するためのイメージ・コピーへの切替え

ディスクに障害が発生し、users 表領域内のすべてのデータファイルにアクセスできなくなったとします。この表領域内のすべてのデータファイルのイメージ・コピーは、フラッシュ・リカバリ領域に存在します。Recovery Manager を起動し、TARGET としてデータベースに接続した後、SWITCH を実行して制御ファイルが新しいデータファイルを指し示すようにしてから、次のように RECOVER を実行します。

```
SQL "ALTER TABLESPACE users OFFLINE IMMEDIATE";
SWITCH TABLESPACE users TO COPY;
RECOVER TABLESPACE users;
SQL "ALTER TABLESPACE users ONLINE";
```

### 例 2-148 新しい場所へリストアした後のデータファイルのファイル名の切替え

ディスクに障害が発生し、データファイルを新しいディスクの場所へリストアする必要があるとします。Recovery Manager を起動し、TARGET としてデータベースに接続した後、SET NEWNAME コマンドを実行してデータファイル名を変更してから、RESTORE を実行して欠落しているデータファイルをリストアします。SWITCH を実行して制御ファイルが新しいデータファイルを指し示すようにしてから、RECOVER を実行します。この例では、ディスクとテーブの両方のチャンネルを割り当てます。

```
RUN
{
  ALLOCATE CHANNEL dev1 DEVICE TYPE DISK;
  ALLOCATE CHANNEL dev2 DEVICE TYPE sbt;
  SQL "ALTER TABLESPACE users OFFLINE IMMEDIATE";
  SET NEWNAME FOR DATAFILE '/disk1/oradata/prod/users01.dbf'
    TO '/disk2/users01.dbf';
  RESTORE TABLESPACE users;
  SWITCH DATAFILE ALL;
  RECOVER TABLESPACE users;
  SQL "ALTER TABLESPACE users ONLINE";
}
```

### 例 2-149 SET NEWNAME と SWITCH TEMPFILE ALL を使用した一時ファイルの名前の変更

この例では、SET NEWNAME を使用して複数の一時ファイルの新しい名前を指定し、SWITCH TEMPFILE ALL を使用して一時ファイル名を指定されたファイル名に変更します。データベースはこの手順を開始する際にクローズされた状態である必要があります。データベースがオープンされている場合は一時ファイルが新しい場所に再作成されます。

```
CONNECT TARGET /
STARTUP FORCE MOUNT
RUN
{
  SET NEWNAME FOR TEMPFILE 1 TO '/disk2/temp01.dbf';
  SET NEWNAME FOR TEMPFILE 2 TO '/disk2/temp02.dbf';
  SET NEWNAME FOR TEMPFILE 3 TO '/disk2/temp03.dbf';
  SWITCH TEMPFILE ALL;
  RESTORE DATABASE;
```

```
RECOVER DATABASE;  
ALTER DATABASE OPEN;  
}
```

#### 例 2-150 データファイルのコピーへの切替え

次のコマンドは、tools 表領域のデータファイルを、/disk2/tools.copy という名前のデータファイル・コピーに切り替えます。

```
RUN  
{  
  SQL "ALTER TABLESPACE tools OFFLINE IMMEDIATE";  
  SWITCH DATAFILE '/disk1/oradata/prod/tools01.dbf'  
  TO DATAFILECOPY '/disk2/tools.copy';  
  RECOVER TABLESPACE tools;  
  SQL "ALTER TABLESPACE tools ONLINE";  
}
```



# TRANSPORT TABLESPACE

## 用途

TRANSPORT TABLESPACE コマンドを使用すると、ソース・データベースのライブ・データファイルではなく、Recovery Manager のバックアップからトランスポート表領域セットを作成できます。

**関連項目：** Recovery Manager での表領域のトランスポート方法は、『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』を参照してください。

## 前提条件

『Oracle Database 管理者ガイド』で説明されているトランスポート表領域セットの作成に関する制限事項は、表領域を読取り専用にする要件を除いて、バックアップから表領域をトランスポートする場合に適用されます。

SYSAUX 表領域は、リカバリ・セット（トランスポートする表領域のセット）に含めないように注意してください。Recovery Manager は、SYSAUX 表領域を補助セット（データファイルと、表領域のトランスポートに必要な他のファイルを含む）に組み込みます。

TRANSPORT TABLESPACE では、エンディアン形式は変換しません。ターゲット・プラットフォームに別のエンディアン形式が含まれている場合は、TRANSPORT TABLESPACE の実行後に、CONVERT コマンドを使用して、トランスポート表領域のデータファイルのエンディアン形式を変換します。

表領域を削除すると、TRANSPORT TABLESPACE の SCN が表を削除したときの SCN より古い場合でも、後で TRANSPORT TABLESPACE を使用してこの表領域をトランスポート表領域セットに含めることはできません。表領域名を変更すると、TRANSPORT TABLESPACE を使用して、表領域名を変更した時点よりも前の時点でのトランスポート表領域セットは作成できません。

### バックアップおよびバックアップ・メタデータ

必要なすべての表領域（補助セットの表領域を含む）のバックアップ、およびターゲット時刻までリカバリするために必要なアーカイブ REDO ログ・ファイルが必要です。

リカバリ・カタログを使用せずに、必要なバックアップに関するメタデータを含む制御ファイル・レコードがデータベースで再使用された場合、バックアップを Recovery Manager で検索できないため、そのコマンドは失敗します。必要なバックアップが使用可能な場合は、CATALOG を使用してそれらを Recovery Manager リポジトリに追加できる場合があります。ただし、制御ファイル・レコードがデータベースですでに上書きされている場合は、他の必要なバックアップのレコードが失われる場合があります。

### データ・ポンプ・エクスポート/インポート

Recovery Manager ではデータ・ポンプ・エクスポート/インポート・ユーティリティが使用されるため、トランスポートする表領域で XMLType が使用されている場合、TRANSPORT TABLESPACE は使用できません。この場合は、『Oracle Database 管理者ガイド』の手順を使用する必要があります。

エクスポート・ダンプ・ファイルの名前の付いたファイルがすでに表領域のトランスポート先に格納されていると、TRANSPORT TABLESPACE はデータ・ポンプ・エクスポートの呼出し時に失敗します。以前の TRANSPORT TABLESPACE ジョブを繰り返す場合は、エクスポート・ダンプ・ファイルを含む以前の出力ファイルを削除してください。

### 表領域と列の暗号化

列レベルで機能する透過的データ暗号化と表領域暗号化のデータベース暗号化機能のいずれにも、ウォレットが使用されています。暗号化される表領域または暗号化された列を含む表領域には、次の制限が適用されることに注意してください。

- トランSPORTする表領域が暗号化されている場合は、トランSPORT先データベースにウォレットを手動でコピーする必要があります。
- 複製先データベースにウォレットがすでに存在する場合は、ソース・データベースのウォレットを複製先データベースにコピーすることはできません。したがって、ウォレットがすでに存在するデータベースには、暗号化データをトランSPORTできません。データの透過的暗号化機能で列を暗号化する場合は、パスワードで保護されたエクスポート・ファイルに列をエクスポートし、そのデータを複製先データベースにインポートできます。

**関連項目：** 透過的データ暗号化については、『Oracle Database Advanced Security 管理者ガイド』を参照してください。

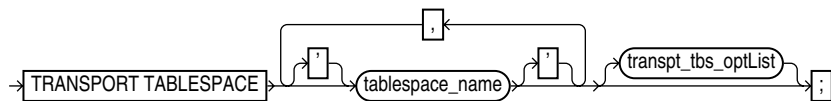
### 使用上の注意

Recovery Manager では同じノードでのリストアおよびリカバリに使用される自動補助インスタンスがソース・インスタンスとして作成されるため、TRANSPORT TABLESPACE コマンドの操作中にパフォーマンス・オーバーヘッドが発生します。

Recovery Manager がデータベースのバックアップ計画の一部分でない場合は、必要なデータファイル・コピーおよびアーカイブ・ログがディスクで使用可能なかぎり、TRANSPORT TABLESPACE を使用できます。CATALOG コマンドを使用してデータファイル・コピーおよびアーカイブ・ログを Recovery Manager リポジトリに記録します。その後、TRANSPORT TABLESPACE を使用できます。また、TRANSPORT TABLESPACE を使用できるように、Recovery Manager を使用してデータベースをバックアップするオプションもあります。

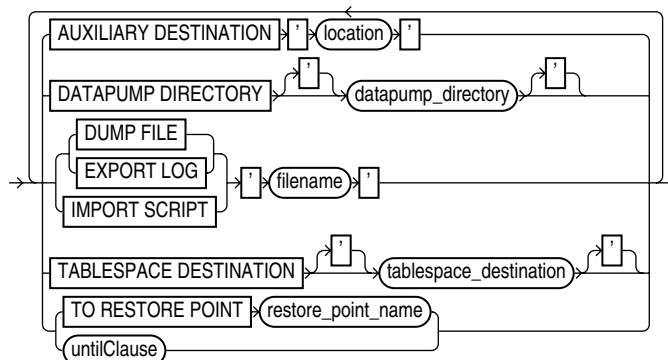
### 構文

**transpt\_tbs::=**



(transpt\_tbs\_optlist::= (2-278 ページ))

**transpt\_tbs\_optlist::=**



(untilClause::= (3-38 ページ))

## セマンティクス

### *transpt\_tbs*

構文の要素	説明
<i>tablespace_name</i>	トランスポートする各表領域の名前を指定します。  必要なすべての表領域（補助セットの表領域を含む）のバックアップ、および TRANSPORT TABLESPACE 操作のターゲット時刻までリカバリできる、Recovery Manager で使用可能なアーカイブ REDO ログ・ファイルが必要です。

### *transpt\_tbs\_optlist*

この副次句では、表領域のトランスポートに影響を与えるオプションのパラメータを指定します。

構文の要素	説明
AUXILIARY DESTINATION 'location'	補助インスタンス用のファイルの格納場所を指定します。  SET NEWNAME と CONFIGURE AUXNAME を使用すると、個々のファイルに対してこの引数をオーバーライドできます。独自の初期化パラメータ・ファイルを使用して補助インスタンスをカスタマイズする場合は、AUXILIARY DESTINATION のかわりに、DB_FILE_NAME_CONVERT および LOG_FILE_NAME_CONVERT 初期化パラメータを使用できます。  <b>関連項目：</b> 補助インスタンス・ファイルの様々なネーミング手法での相互作用の詳細は、『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』を参照してください。
DATAPUMP DIRECTORY <i>datapump_directory</i>	データ・ポンプ・エクスポートの出力が作成されるデータベース・ディレクトリ・オブジェクトを指定します（例 2-152 を参照）。指定しない場合、Recovery Manager は TABLESPACE DESTINATION で指定された場所にファイルを作成します。  <b>関連項目：</b> データ・ポンプ・エクスポートおよびデータベース・ディレクトリ・オブジェクトの詳細は、『Oracle Database ユーティリティ』を参照してください。
DUMP FILE ' <i>filename</i> '	データ・ポンプ・エクスポートのダンプ・ファイルを作成する場所を指定します。指定しない場合、エクスポート・ダンプ・ファイルは、dmpfile.dmp という名前で、DATAPUMP DIRECTORY 句で指定された場所または表領域のトランスポート先に格納されます。  <b>注意：</b> エクスポート・ダンプ・ファイルの名前の付いたファイルがすでに表領域のトランスポート先に格納されていると、TRANSPORT TABLESPACE はデータ・ポンプ・エクスポートの呼出し時に失敗することに注意してください。以前の TRANSPORT TABLESPACE ジョブを繰り返す場合は、エクスポート・ダンプ・ファイルを含む以前の出力ファイルを削除してください。
EXPORT LOG ' <i>filename</i> '	データ・ポンプ・エクスポートで生成されたログの場所を指定します。省略した場合、エクスポート・ログは、explog.log という名前で、DATAPUMP DIRECTORY 句で指定された場所または表領域のトランスポート先に格納されます。
IMPORT SCRIPT ' <i>filename</i> '	トランスポート先データベースのトランスポートされた表領域に接続する際に使用する、Recovery Manager で生成されるサンプル入力スクリプトのファイル名を指定します。省略した場合、インポート・スクリプトは impscript.sql という名前になります。このスクリプトは、表領域のトランスポート先に格納されます。
TABLESPACE DESTINATION <i>tablespace_destination</i>	表領域のトランスポート操作が完了した後の、トランスポートされた表領域のデータファイルの場所を指定します。
TO RESTORE POINT <i>restore_point_name</i>	リストア・ポイントを作成した時点の SCN を上限として、表領域のリストアおよびリカバリのリストア・ポイントを指定します。指定した値は含まれます。上限値が含まれるため、Recovery Manager は、リストア・ポイントに対応する SCN まで表領域をリストアまたはリカバリできるファイルのみを選択します。

構文の要素	説明
<code>untilClause</code>	<p>過去の時刻、SCN またはログ順序番号を指定します (2-280 ページの例 2-151 を参照)。指定した場合、<b>Recovery Manager</b> は、エクスポート前に補助インスタンスで表領域をその過去の時点における表領域の内容にリストアおよびリカバリします。</p> <p>表領域名を変更すると、このコマンドを使用して、表領域名を変更した時点よりも前の時点でのトランスポータブル表領域セットを作成することはできません。<b>Recovery Manager</b> には、前の表領域名に関する情報がありません。</p> <p>TRANSPORT TABLESPACE の UNTIL 時刻または SCN の時点の UNDO セグメントを含む表領域は、補助セットの一部である必要があります。制御ファイルには、現時点での UNDO セグメントを含む表領域のレコードのみが含まれています。UNDO セグメントを含む表領域セットが UNTIL 時刻や SCN でのものと異なる場合、TRANSPORT TABLESPACE は失敗します。したがって、NOCATALOG モードで <b>Recovery Manager</b> を使用し、UNTIL を指定する場合、TRANSPORT TABLESPACE を実行する時点の UNDO セグメントを含む表領域セットは、UNTIL 時刻または SCN での UNDO セグメントを含む表領域セットと同じである必要があります。</p>

## 例

### 例 2-151 過去の時刻を指定した TRANSPORT TABLESPACE の使用

この例では、トランスポータブル・セットの表領域は `example` および `tools` で、トランスポータブル・セットのファイルは `/disk1/transport_dest` に格納されます。また、トランスポータブル表領域は、15 分前の時刻までリカバリされます。

```
TRANSPORT TABLESPACE example, tools
  TABLESPACE DESTINATION '/disk1/transportdest'
  AUXILIARY DESTINATION '/disk1/auxdest'
  UNTIL TIME 'SYSDATE-15/1440';
```

次に出力例の一部を示します。

```
Creating automatic instance, with SID='egnr'

initialization parameters used for automatic instance:
db_name=PROD
compatible=11.0.0
db_block_size=8192
.
.
starting up automatic instance PROD
.
.
executing Memory Script

executing command: SET until clause

Starting restore at 07-JUN-07
allocated channel: ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: SID=44 device type=DISK

channel ORA_AUX_DISK_1: starting datafile backup set restore
channel ORA_AUX_DISK_1: restoring control file
.
.
output file name=/disk1/auxdest/cntrl_tspitr_PROD_egnr.f
Finished restore at 07-JUN-07

sql statement: alter database mount clone database
```

```

sql statement: alter system archive log current

sql statement: begin dbms_backup_restore.AutoBackupFlag(FALSE); end;

starting full resync of recovery catalog
full resync complete
.
.
.
executing Memory Script
.
.
.

Starting restore at 07-JUN-07
using channel ORA_AUX_DISK_1

channel ORA_AUX_DISK_1: starting datafile backup set restore
channel ORA_AUX_DISK_1: specifying datafile(s) to restore from backup set
channel ORA_AUX_DISK_1: restoring datafile 00001 to
/disk1/auxdest/TSPITR_PROD_EGNR/datafile/o1_mf_system_%u_.dbf
datafile 1 switched to datafile copy
.
.
.
starting media recovery
.
.
.
Finished recover at 07-JUN-07

database opened
.
.
.
executing Memory Script
.
.
.

sql statement: alter tablespace EXAMPLE read only
Removing automatic instance
shutting down automatic instance
Oracle instance shut down
Automatic instance removed
auxiliary instance file /disk1/auxdest/cntrl_tspitr_PROD_egnr.f deleted
.
.
.

```

#### 例 2-152 カスタマイズされたファイルの場所を指定した TRANSPORT TABLESPACE の使用

この例では、データ・ポンプ関連ファイル（ダンプ・ファイルなど）の場所を制御するオプションの引数の使用方法を示します。DATAPUMP DIRECTORY は、ターゲット・データベースに存在するオブジェクトを参照する必要があります。SQL 文の CREATE DIRECTORY を使用すると、ディレクトリ・オブジェクトを作成できます。

```

TRANSPORT TABLESPACE example
  TABLESPACE DESTINATION '/disk1/transportdest'
  AUXILIARY DESTINATION '/disk1/auxdest'
  DATAPUMP DIRECTORY mypumpdir
  DUMP FILE 'mydumpfile.dmp'
  IMPORT SCRIPT 'myimportscript.sql'
  EXPORT LOG 'myexportlog.log';

```

## UNREGISTER

### 用途

UNREGISTER コマンドを使用すると、登録されている 1 つ以上のデータベースの Recovery Manager のメタデータをリカバリ・カタログから削除できます。

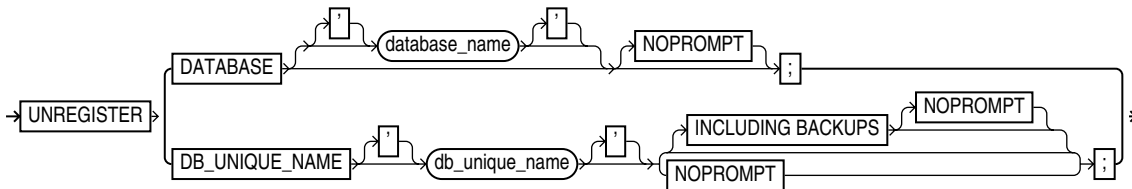
**関連項目：** 1 つのコマンドで、データベースを削除して登録解除する方法は、2-124 ページの「[DROP DATABASE](#)」を参照してください。

### 前提条件

このコマンドは、Recovery Manager プロンプトでのみ実行してください。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。登録解除するデータベースは、現在リカバリ・カタログに登録されている必要があります。

### 構文

**unregister** ::=



### セマンティクス

構文の要素	説明
DATABASE	<p>登録解除するプライマリ・データベースを指定します。Recovery Manager により、プライマリ・データベース、およびこれに対応付けられているスタンバイ・データベースが登録解除されます (2-283 ページの例 2-153 を参照)。</p> <p><code>database_name</code> を指定しない場合は、次の方法のうちいずれかでデータベースが識別される必要があります。</p> <ul style="list-style-type: none"> <li>Recovery Manager が TARGET としてデータベースに接続されている場合、ターゲット・データベースと対応付けられているスタンバイ・データベースが Recovery Manager によって登録解除されます。</li> <li>Recovery Manager が TARGET としてデータベースに接続されていない場合、または TARGET データベースの名前がリカバリ・カタログで一意ではない場合、Recovery Manager では SET DBID コマンドで指定された値を使用します (2-284 ページの例 2-154 を参照)。Recovery Manager により、この DBID を持つすべてのデータベースが登録解除されます。</li> </ul>
<code>database_name</code>	<p>登録解除するプライマリ・データベースの名前を指定します。</p> <p>このデータベース名は、リカバリ・カタログで一意である必要があります。データベース名が一意であるため、Recovery Manager はターゲット・データベースに接続したり、SET DBID コマンドを使用する必要がありません。</p>

構文の要素	説明
DB_UNIQUE_NAME db_unique_name	<p>db_unique_name で指定されている Data Guard データベースに対するバックアップのメタデータを除く、すべてのメタデータの削除を指定します。</p> <p>通常は、スタンバイ・データベースの登録解除を行うためにこの句を使用しますが、指定するデータベースはプライマリ・データベースでもスタンバイ・データベースでもかまいません 2-284 ページの例 2-155 を参照)。</p> <p>この句は、Recovery Manager がマウントまたはオープン状態のターゲット・データベースに接続されている場合、または SET DBID コマンドを実行済の場合のみ使用できます。通常、ターゲット・データベースは登録解除対象とはなりません。たとえば、TARGET として prod1 に接続し、DB_UNIQUE_NAME を使用して standby1 を登録解除することができます。SET DBID で使用される DBID は、プライマリ・データベースとその対応付けられているスタンバイ・データベースで同じです。</p> <p>データベースのバックアップには、リカバリ・カタログ内のバックアップ・ファイルの名前に対応付けられた DB_UNIQUE_NAME が設定されています。データベースが登録解除されると、これらのバックアップ・ファイルのデータベース名が null としてマークされます。データベース名とこれらのファイルを関連付けるには、CHANGE ...RESET DB_UNIQUE_NAME コマンドを使用します。</p>
INCLUDING BACKUPS	<p>db_unique_name で指定されたデータベースのバックアップ・メタデータも削除されるように指定します。</p> <p><b>注意:</b> 物理バックアップは、UNREGISTER コマンドで削除できないことに注意してください。物理バックアップを削除する場合は、最初に DELETE コマンドでバックアップを削除してから、INCLUDING BACKUPS オプションを指定して UNREGISTER を実行します。</p>
NOPROMPT	<p>確認のプロンプトを表示せずに、データベースを登録解除します。</p>

## 例

### 例 2-153 プライマリ・データベースとそのスタンバイ・データベースの登録解除

プライマリ・データベース prod には、対応付けられているスタンバイ・データベース dgprod3 および dgprod4 があるとします。この例では、データベース名がリカバリ・カタログ内で一意のターゲット・データベース prod に Recovery Manager を接続し、そのデータベースを登録解除します。Recovery Manager により、prod、dgprod3 および dgprod4 に関するすべてのメタデータがカタログから削除されます。例には出力例も含まれます。

```

RMAN> CONNECT TARGET /

connected to target database: PROD (DBID=1627709917)

RMAN> CONNECT CATALOG rman@catdb

recovery catalog database Password: password
connected to recovery catalog database

RMAN> UNREGISTER DATABASE NOPROMPT;

database name is "PROD" and DBID is 1627709917
database unregistered from the recovery catalog

RMAN> LIST DB_UNIQUE_NAME ALL;

RMAN>
```

**例 2-154 カタログ内で一意でないデータベースの登録解除**

リカバリ・カタログに登録されている2つのデータベースの名前が `prod` であるとして、目標は、DBID が 28014364 の `prod` データベースを登録解除することです。リカバリ・カタログで `prod` という名前を持つ複数のデータベースが登録されているため、また、Recovery Manager が 28014364 データベースに TARGET として接続されていない（このデータベースがファイル・システムからすでに削除されている）ため、UNREGISTER DATABASE の前に SET DBID を実行します。この例には、出力例も含まれます。

```
RMAN> CONNECT CATALOG rman@catdb

recovery catalog database Password: password
connected to recovery catalog database

RMAN> SET DBID 28014364;

executing command: SET DBID
database name is "PROD" and DBID is 28014364

RMAN> UNREGISTER DATABASE;

Do you really want to unregister the database (enter YES or NO)? YES
database unregistered from the recovery catalog
```

**例 2-155 スタンバイ・データベースの登録解除**

プライマリ・データベース `prod` には、対応付けられているスタンバイ・データベース `dgprod3` および `dgprod4` があります。`dgprod4` を登録解除しますが、このデータベースで作成されたバックアップのメタデータは削除しません。これらの場合バックアップは、その環境の他のデータベースで引き続き使用できるためです。この例では、SET DBID を使用してスタンバイ・データベースの DBID を指定してから、登録解除します（例には出力例も含まれません）。

```
RMAN> CONNECT CATALOG rman@catdb

recovery catalog database Password: password
connected to recovery catalog database

RMAN> SET DBID 1627367554;

executing command: SET DBID
database name is "PROD" and DBID is 1627367554

RMAN> UNREGISTER DB_UNIQUE_NAME dgprod4;

database db_unique_name is "dgprod4", db_name is "PROD" and DBID is 1627367554

Want to unregister the database with target db_unique_name (enter YES or NO)? YES
database with db_unique_name dgprod4 unregistered from the recovery catalog
```



## UPGRADE CATALOG

### 用途

UPGRADE CATALOG コマンドを使用すると、リカバリ・カタログ・スキーマを、旧バージョンから、Recovery Manager クライアントで必要なバージョンにアップグレードできます。

### 前提条件

Recovery Manager がリカバリ・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ていて、リカバリ・カタログ・データベースがオープン状態である必要があります。仮想プライベート・カタログへの接続中は、UPGRADE CATALOG コマンドを使用できません (CREATE CATALOG コマンドを参照)。アップグレードできるのは基本カタログのみです。

リカバリ・カタログが、Recovery Manager 実行可能ファイルに必要なバージョンより後のバージョンになっていないことが必要です。後のバージョンの場合は、エラーが発行されます。Recovery Manager は、アップグレード中に生成したエラー・メッセージをメッセージ・ログに表示します。

Oracle Database 10g リリース 1 のリカバリ・カタログ・スキーマには、CREATE TYPE 権限が必要です。10g リリース 1 より前のリリースでリカバリ・カタログ所有者を作成し、このユーザーに RECOVERY\_CATALOG\_OWNER ロールを付与しているとします。このロールに CREATE TYPE 権限が含まれていない場合は、アップグレードを実行する前に、このユーザーに明示的に CREATE TYPE を付与する必要があります。

### 使用上の注意

アップグレードの確認のため、UPGRADE CATALOG コマンドを続けて 2 回入力する必要があります。

リカバリ・カタログがすでに現行の状態になっていれば、Recovery Manager はコマンドの実行を認めます。その結果、必要に応じてパッケージの再作成ができます。

基本リカバリ・カタログへのアップグレードで既存の仮想プライベート・カタログへの変更が必要な場合は、Recovery Manager が次にその仮想プライベート・カタログに接続したときに、これらの変更が自動的に行われます。

UPGRADE CATALOG コマンドでは、アップグレード用のスクリプトは実行されません。かわりに、Recovery Manager は各種の SQL DDL 文をリカバリ・カタログに送信し、新規の表、ビュー、列などでリカバリ・カタログ・スキーマを更新します。

### 構文

**upgradeCatalog::=**

→ UPGRADE CATALOG 

### セマンティクス

なし。

## 例

### 例 2-156 リカバリ・カタログのアップグレード

この例では、Recovery Manager をリカバリ・カタログ・データベース catdb に接続し、それを新しいバージョンにアップグレードします。

```
RMAN> CONNECT CATALOG rco@catdb

recovery catalog database Password: password
connected to recovery catalog database
PL/SQL package rman.DEMS_RCVCAT version 10.01.00 in RCVCAT database is too old

RMAN> UPGRADE CATALOG;

recovery catalog owner is RCO
enter UPGRADE CATALOG command again to confirm catalog upgrade

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
```

## VALIDATE

### 用途

VALIDATE コマンドを使用すると、破損ブロックおよび欠落ファイルをチェックしたり、バックアップ・セットがリストア可能かどうかを判断することができます。

VALIDATE での検証中に問題が検出されると、Recovery Manager によってその問題が表示され、障害の評価が開始されます。障害が検出されたら、Recovery Manager によってその障害のログが自動診断リポジトリに書き込まれます。LIST FAILURE を使用すると、障害を表示できます。

### 前提条件

ターゲット・データベースはマウントまたはオープン状態である必要があります。

### 使用上の注意

VALIDATE コマンドのオプションは、BACKUP VALIDATE コマンドのオプションと同じ意味です。ただし、BACKUP VALIDATE とは異なり、VALIDATE では個々のバックアップ・セットとデータ・ブロックをチェックできます。

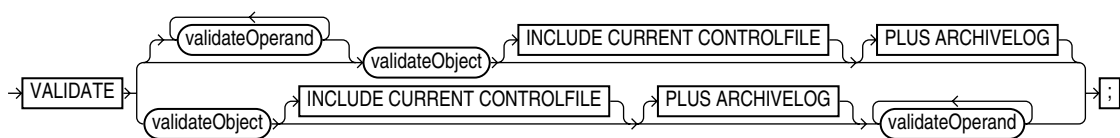
VALIDATE コマンドでは、一度も使用されていないブロックはスキップされます。Recovery Manager ではまた、COMPATIBLE パラメータが 10.2 以上に設定されている場合は、ローカル管理の表領域の（一度も使用されていないブロックではなく）現在使用されていないブロックがスキップされます。

物理的な破損が発生した場合、データベースでブロックが認識されません。論理的な破損が発生した場合は、ブロックの内容に論理的な一貫性がなくなります。デフォルトでは、VALIDATE コマンドを使用すると、物理的な破損のみがチェックされます。CHECK LOGICAL を指定すると、論理的な破損もチェックできます。Recovery Manager により、結果が V\$DATABASE\_BLOCK\_CORRUPTION ビューに移入されます。

ブロック破損はブロック間破損とブロック内破損に分類できます。ブロック内破損では、破損はブロック自体の中で発生し、物理的な破損または論理的な破損のいずれかです。ブロック間破損では、破損はブロック間で発生し、論理的な破損のみです。VALIDATE コマンドでは、ブロック内破損のみがチェックされます。

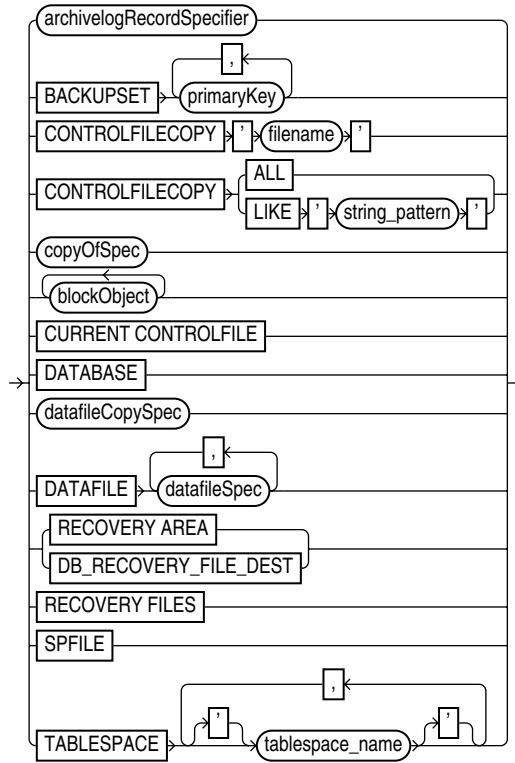
### 構文

**validate::=**



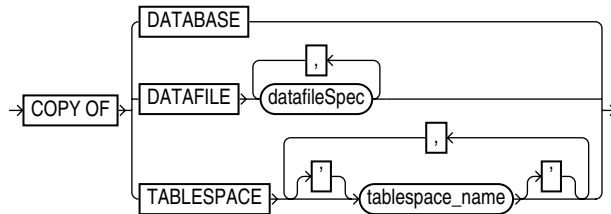
([validateObject::=](#) (2-288 ページ)、[validateOperand::=](#) (2-289 ページ))

**validateObject::=**



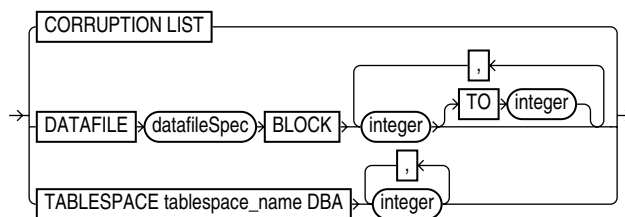
(archivelogRecordSpecifier::= (3-6 ページ)、 copyOfSpec::= (2-25 ページ)、  
 blockObject::= (2-189 ページ)、 datafileCopySpec::= (2-26 ページ)、  
 datafileSpec::= (3-14 ページ))

**copyOfSpec::=**

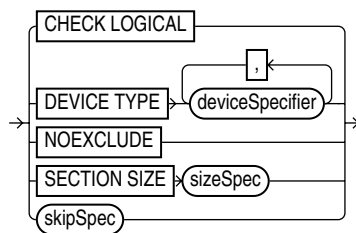


(datafileSpec::= (3-14 ページ))

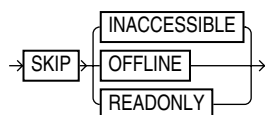
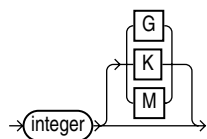
**blockObject::=**



(datafileSpec::= (3-14 ページ))

**validateOperand::=**

([deviceSpecifier::=](#) (3-15 ページ)、[sizeSpec::=](#) (2-289 ページ)、[skipSpec::=](#) (2-289 ページ))

**skipSpec::=****sizeSpec::=**

## セマンティクス

**validate**

この副次句では、検証するバックアップ・セットを指定します。構文は、2-287 ページの「[validate::=](#)」を参照してください。

構文の要素	説明
<a href="#">validateOperand</a>	検証方法を制御するオプションを指定します。 <b>関連項目</b> : 2-291 ページの「 <a href="#">validateOperand</a> 」を参照してください。
<a href="#">validateObject</a>	検証するファイルを指定します。 <b>関連項目</b> : 2-290 ページの「 <a href="#">validateObject</a> 」を参照してください。
INCLUDE CURRENT CONTROLFILE	現行の制御ファイルのスナップショットを作成し、検証を行います。
PLUS ARCHIVELOG	アーカイブ REDO ログも検証の対象にします。Recovery Manager によって次の手順が実行されます。 <ol style="list-style-type: none"> <li>ALTER SYSTEM ARCHIVE LOG CURRENT 文が実行されます。</li> <li>VALIDATE ARCHIVELOG ALL コマンドが実行されます。バックアップの最適化が使用可能になっている場合、Recovery Manager では、まだバックアップされていないログの検証のみが行われるため注意してください。</li> <li>VALIDATE コマンドで指定したファイルの検証が行われます。</li> <li>ALTER SYSTEM ARCHIVE LOG CURRENT 文が実行されます。</li> <li>残りのアーカイブ REDO ログの検証が行われます。</li> </ol>

**validateObject**

この副次句では、検証するデータベース・ファイルを指定します。構文は、2-288 ページの「[validateObject::=](#)」を参照してください。

構文の要素	説明
<a href="#">archiveLogRecordSpecifier</a>	アーカイブ REDO ログの範囲を検証します。VALIDATE ARCHIVELOG は BACKUP VALIDATE ARCHIVELOG と同じです。
BACKUPSET <a href="#">primary_key</a>	<p><a href="#">primary_key</a> で指定したバックアップ・セットが存在し、リストア可能であることをチェックします。</p> <p>バックアップ・セットの主キーを取得するには、<a href="#">LIST</a> 文を実行します。あるいは、リカバリ・カタログを使用している場合には、RC_BACKUP_SET リカバリ・カタログ・ビューに問い合わせます。</p> <p>VALIDATE BACKUPSET コマンドは、バックアップ・セットのすべてのブロックをチェックして、バックアップがリストア可能であることを確認します。Recovery Manager はブロックの破損を検出すると、エラーを発行し、検証を終了します。一方、<a href="#">CROSSCHECK</a> コマンドは、指定したファイルがディスク上にある場合はファイルのヘッダーを調べ、ファイルがテープ上にある場合はメディア管理カタログに問い合わせます。</p> <p>バックアップ・セットのうち 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疑いがあるときは、VALIDATE BACKUPSET を使用してください。テストするバックアップ・セットの選択には VALIDATE BACKUPSET を使用し、Recovery Manager にどのバックアップを検証するかを選択させるときは <a href="#">RESTORE</a> コマンドの VALIDATE オプションを使用します。イメージ・コピーを検証する場合は、RESTORE VALIDATE FROM DATAFILECOPY を実行します。</p> <p>自動チャンネルを構成していない場合は、VALIDATE BACKUPSET を実行する前に 1 つ以上のチャンネルを手動で割り当てます。</p> <p><b>注意:</b> バックアップ・セットのコピーが複数存在する場合は、Recovery Manager により最新のコピーのみが検証されます。VALIDATE コマンドでは、特定のコピーを検証するオプションはサポートされません。ただし、あるコピーが別のコピーとは違うデバイス上にある場合は、VALIDATE DEVICE TYPE を使用して、指定したデバイス上のコピーを検証できます。両方のコピーが同じデバイス上にある場合は、<a href="#">CHANGE</a> を使用して 1 つのコピーを一時的に UNAVAILABLE に設定してから、VALIDATE を再発行できます。</p>
CONTROLFILECOPY { 'filename'   ALL   LIKE 'string_pattern' }	<p>制御ファイルのコピーを検証します。次のいずれかの方法で制御ファイルのコピーを指定できます。</p> <ul style="list-style-type: none"> <li>■ 'filename' では、ファイル名で制御ファイルのコピーを指定します。</li> <li>■ ALL では、すべての制御ファイルのコピーのバックアップを指定します。</li> <li>■ LIKE 'pattern' では、ファイル名のパターンを指定します。パーセント記号 (%) は 0 文字以上を示すワイルド・カードで、アンダースコア (_) は 1 文字を示すワイルド・カードです。</li> </ul> <p>制御ファイルのコピーを作成するには、<a href="#">BACKUP AS COPY CURRENT CONTROLFILE</a> コマンドまたは SQL 文 ALTER DATABASE BACKUP CONTROLFILE TO '...' を使用します。</p>
<a href="#">copyOfSpec</a>	<p>データファイルおよび制御ファイルのイメージ・コピーを検証します。</p> <p><b>関連項目:</b> 詳細は、2-45 ページの「<a href="#">copyOfSpec</a>」を参照してください。</p>
<a href="#">blockObject</a>	<p>個々のデータ・ブロックを検証します。</p> <p><b>関連項目:</b> 2-192 ページの「<a href="#">blockObject</a>」を参照してください。</p>
CURRENT CONTROLFILE	<p>現行の制御ファイルを検証します。</p>

構文の要素	説明
DATABASE	データベースを検証します。  Recovery Manager によって、すべてのデータファイルおよび制御ファイルを検証します。データベースが現在サーバー・パラメータ・ファイルを使用している場合は、Recovery Manager がサーバー・パラメータ・ファイルを検証します。 <b>注意:</b> オンライン REDO ログ・ファイルおよび一時ファイルは検証されません。
<code>datafileCopySpec</code>	1 つ以上のデータファイル・イメージ・コピーを検証します。  データファイル・コピーの検証時に、Recovery Manager はブロックの破損をチェックしますが、破損が発見されても検証を終了しません。VALIDATE BACKUPSET とは異なり、Recovery Manager は検証を続行して、破損したブロックの数をレポートします。 <b>関連項目:</b> 詳細は、2-46 ページの「 <code>datafileCopySpec</code> 」を参照してください。
DATAFILE <code>datafileSpec</code>	検証を必要とするブロックを含む 1 つ以上のデータファイルのリストを指定します。 <b>注意:</b> 検証する場合は、対象となるデータファイルをオフライン化する必要はありません。 <b>関連項目:</b> 3-14 ページの「 <code>datafileSpec</code> 」を参照してください。
RECOVERY AREA	現行および前回のすべてのフラッシュ・リカバリ領域の指定先に作成されたリカバリ・ファイルを検証します。リカバリ・ファイルには、全体増分バックアップ・セット、制御ファイルの自動バックアップ、アーカイブ REDO ログおよびデータファイルのコピーが含まれます。フラッシュバック・ログ、現行の制御ファイルおよびオンライン REDO ログは検証されません。
DB_RECOVERY_FILE_DEST	RECOVERY AREA と同じ意味です。
RECOVERY FILES	ディスク上のすべてのリカバリ・ファイルに対して、フラッシュ・リカバリ領域に格納されているか、ディスクの別の場所に格納されているかに関係なく検証を行います。リカバリ・ファイルには、全体および増分のバックアップ・セット、制御ファイルの自動バックアップ、アーカイブ REDO ログおよびデータファイルのコピーが含まれます。フラッシュバック・ログは検証されません。
SPFILE	データベースで現在使用されているサーバー・パラメータ・ファイルを検証します。Recovery Manager では、サーバー・パラメータ・ファイルの他のコピーは検証できません。また、インスタンスの起動に初期化パラメータ・ファイルが使用された場合、サーバー・パラメータ・ファイルは検証できません。
TABLESPACE <code>tablespace_name</code>	指定した表領域を検証します。Recovery Manager は、その内部で表領域名をデータファイルのリストに変換してから、表領域を現在構成しているデータファイルをすべて検証します。Recovery Manager は、指定した表領域の現在メンバーになっているデータファイルをすべて検証します。

### **validateOperand**

この副次句では、検証する修飾子を指定します。構文は、2-289 ページの「`validateOperand::=`」を参照してください。

構文の要素	説明
CHECK LOGICAL	ファイルで物理的な破損チェックを通過したデータ・ブロックと索引ブロックについて、論理的な破損がないかどうかをテストします。たとえば、行ピースまたは索引エントリの破損がないかどうかを調べます。Recovery Manager は論理的な破損を見つけると、アラート・ログとサーバー・セッション・トレース・ファイルにそのブロックのログを書き込みます。Recovery Manager コマンドは完了し、V\$DATABASE_BLOCK_CORRUPTION に破損ブロックの範囲が移入されます。 <b>注意:</b> VALIDATE では MAXCORRUPT は使用されません。

構文の要素	説明
DEVICE TYPE <i>deviceSpecifier</i>	指定したデバイス・タイプ専用の自動チャンネルを割り当てます。このオプションが有効になるのは、構成済の自動チャンネルがあり、チャンネルを手動で割り当てていない場合のみです。たとえば、自動ディスクおよびテープ・チャンネルを構成して VALIDATE ... DEVICE TYPE DISK を実行すると、Recovery Manager ではディスク・チャンネルのみが割り当てられます。 <b>関連項目</b> : 3-15 ページの「 <i>deviceSpecifier</i> 」を参照してください。
NOEXCLUDE	VALIDATE DATABASE または VALIDATE COPY OF DATABASE コマンドで指定されている場合、Recovery Manager は、CONFIGURE EXCLUDE コマンドで入力されたものを含め、表領域をすべて検証します。このオプションで SKIP OFFLINE または SKIP READONLY がオーバーライドされることはありません。
SECTION SIZE <i>sizeSpec</i>	各ファイルを指定のセクション・サイズに分割して、検証をパラレル化します。 このパラメータを指定するのは、複数のチャンネルの構成または割当てが行われていて、これらのチャンネルで検証のパラレル化を行い、複数のチャンネルで 1 つのデータファイルを検証できるようにする場合のみです。このパラメータは、データファイルの検証時のみ適用されます。 ファイルのサイズより大きなセクション・サイズを指定すると、Recovery Manager はファイルの検証をパラレル化しません。256 より多くのセクションが作成される小さなセクション・サイズを指定すると、Recovery Manager は、セクション・サイズをちょうど 256 セクションが作成されるような値に増やします。 <b>関連項目</b> : マルチセクション・バックアップの作成方法は、「BACKUP SECTION SIZE」を参照してください。
<i>skipSpec</i>	指定したファイルを検証対象から除外します。

***skipSpec***

この副次句では、検証対象から除外するファイルを指定します。

構文の要素	説明
SKIP	データファイルまたはアーカイブ REDO ログがアクセス不能、オフラインまたは読取り専用である場合は検証対象から除外します。
INACCESSIBLE	I/O エラーのために読み取ることができないデータファイルおよびアーカイブ REDO ログを検証対象から除外するように指定します。 データファイルは、読取りが不可能な場合にのみアクセス不能とみなされます。一部のオフライン・データファイルは、ディスク上に残っているために読取りが可能です。他のデータファイルは削除または移動されたためにアクセス不可となり、読取り不可となります。
OFFLINE	オフライン・データファイルを検証対象から除外するように指定します。
READONLY	読取り専用データファイルを検証対象から除外するように指定します。

**VALIDATE コマンドの出力****表 2-38 データファイルのリスト**

列	指定対象
File	検証されるデータファイルの絶対番号。
Status	破損がない場合は OK、ブロックの破損が検出された場合は FAILED。



表 2-38 データファイルのリスト (続き)

列	指定対象
Marked Corrupt	破損としてマークされたブロックの数。これらは以前にデータベースにより破損としてマークされたブロックです。たとえば、データベースは、NOLOGGING 操作を含むリカバリ中に意図的にブロックを破損としてマークする場合があります。また、Recovery Manager のバックアップには、SET MAXCORRUPT コマンドにより許容されている範囲の破損ブロックが含まれている場合もあります。このバックアップをリストアすると、ファイルに破損としてマークされたブロックが含まれます。
Empty Blocks	一度も使用されたことのないブロックの数。
Blocks Examined	ファイル内の合計ブロック数。
High SCN	ファイルに記録されている最大の SCN。
File Name	検証されるファイルの名前。
Block Type	検証されるブロックのタイプ。Data、Index または Other。
Blocks Failing	破損チェックに失敗するブロック数。これらのブロックで新たに破損が発生しています。
Blocks Processed	破損をチェックするブロックの数。

表 2-39 制御ファイルおよび SPFILE のリスト

列	指定対象
File TYPE	ファイルのタイプ。SPFILE または Control File。
Status	破損がない場合は OK、ブロックの破損が検出された場合は FAILED。
Blocks Failing	破損チェックに失敗するブロック数。これらのブロックで新たに破損が発生しています。
Blocks Examined	ファイル内の合計ブロック数。

表 2-40 アーカイブ・ログのリスト

列	指定対象
Thrd	REDO のスレッド番号。
Seq	ログ順序番号。
Status	破損がない場合は OK、ブロックの破損が検出された場合は FAILED。
Blocks Failing	破損チェックに失敗するブロック数。これらのブロックで新たに破損が発生しています。
Blocks Examined	ファイル内の合計ブロック数。
Name	アーカイブ REDO ログ・ファイルの名前。

## 例

**例 2-157 バックアップ・セットの検証**

この例では、使用可能なバックアップ・セットをすべてリストしてから、これらを検証します。出力例に示すとおり、Recovery Manager は、バックアップのリストアが可能であることを確認します。

```

RMAN> LIST BACKUP SUMMARY;

List of Backups
=====
Key       TY LV S Device Type Completion Time #Pieces #Copies Compressed Tag
-----
3871      B F A DISK      08-MAR-07      1      1      NO      TAG20070308T092426
3890      B F A DISK      08-MAR-07      1      1      NO      TAG20070308T092534

RMAN> VALIDATE BACKUPSET 3871, 3890;

Starting validate at 08-MAR-07
using channel ORA_DISK_1
channel ORA_DISK_1: starting validation of datafile backup set
channel ORA_DISK_1: reading from backup piece
/disk2/PROD/backupset/2007_03_08/o1_mf_nnndf_TAG20070308T092426_426_2z0kpc72_.bkp
channel ORA_DISK_1: piece
handle=/disk2/PROD/backupset/2007_03_08/o1_mf_nnndf_TAG20070308T092426_2z0kpc72_.bkp tag=TAG20070308T092426
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: validation complete, elapsed time: 00:00:18
channel ORA_DISK_1: starting validation of datafile backup set
channel ORA_DISK_1: reading from backup piece
/disk2/PROD/autobackup/2007_03_08/o1_mf_s_616670734_2z0krhjb_.bkp
channel ORA_DISK_1: piece
handle=/disk2/PROD/autobackup/2007_03_08/o1_mf_s_616670734_2z0krhjb_.bkp tag=TAG20070308T092534
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: validation complete, elapsed time: 00:00:00
Finished validate at 08-MAR-07

```

**例 2-158 データベースの検証**

この例では、データベースを検証します。出力例も示します。検証では、データファイル 1 内に破損ブロックが 1 個検出されました。VALIDATE 出力では、指定されたトレース・ファイルで破損の詳細を参照するように示されています。

```

RMAN> VALIDATE DATABASE;

Starting validate at 26-FEB-07
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/oradata/prod/system01.dbf
input datafile file number=00002 name=/disk1/oradata/prod/sysaux01.dbf
input datafile file number=00003 name=/disk1/oradata/prod/undotbs01.dbf
input datafile file number=00004 name=/disk1/oradata/prod/cwmlite01.dbf
input datafile file number=00005 name=/disk1/oradata/prod/drsys01.dbf
input datafile file number=00006 name=/disk1/oradata/prod/example01.dbf
input datafile file number=00007 name=/disk1/oradata/prod/indx01.dbf
input datafile file number=00008 name=/disk1/oradata/prod/tools01.dbf
input datafile file number=00009 name=/disk1/oradata/prod/users01.dbf
channel ORA_DISK_1: validation complete, elapsed time: 00:01:25
List of Datafiles
=====
File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
1 FAILED 0 4140 57600 498288

```

```

File Name: /disk1/oradata/prod/system01.dbf
Block Type Blocks Failing Blocks Processed
-----
Data      1          41508
Index     0          7653
Other     0          4299

File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
2  OK      0          8918          20040          498237
File Name: /disk1/oradata/prod/sysaux01.dbf
Block Type Blocks Failing Blocks Processed
-----
Data      0          2473
Index     0          2178
Other     0          6471

File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
3  OK      0          36           2560          498293
File Name: /disk1/oradata/prod/undotbs01.dbf
Block Type Blocks Failing Blocks Processed
-----
Data      0          0
Index     0          0
Other     0          2524

File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
4  OK      0          1           1280          393585
File Name: /disk1/oradata/prod/cwmlite01.dbf
Block Type Blocks Failing Blocks Processed
-----
Data      0          0
Index     0          0
Other     0          1279

File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
5  OK      0          1           1280          393644
File Name: /disk1/oradata/prod/drsys01.dbf
Block Type Blocks Failing Blocks Processed
-----
Data      0          0
Index     0          0
Other     0          1279

File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
6  OK      0          1           1280          393690
File Name: /disk1/oradata/prod/example01.dbf
Block Type Blocks Failing Blocks Processed
-----
Data      0          0
Index     0          0
Other     0          1279

File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
7  OK      0          1           1280          393722
File Name: /disk1/oradata/prod/indx01.dbf
Block Type Blocks Failing Blocks Processed
-----
Data      0          0
Index     0          0
Other     0          1279

```

```
File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
8   OK      0          1          1280          393754
File Name: /disk1/oradata/prod/tools01.dbf
Block Type Blocks Failing Blocks Processed
-----
Data      0          0
Index     0          0
Other     0          1279
```

```
File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
9   OK      0          1272         1280          393785
File Name: /disk1/oradata/prod/users01.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 /disk2/oracle/log/diag/rdbms/prod/prod/trace/prod_ora_10609.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          506
Finished validate at 26-FEB-07
```

---

## Recovery Manager 副次句

この章では、Recovery Manager のコマンド内で参照される副次句をアルファベット順に説明します。Recovery Manager コマンドとコマンドライン・オプションの概要は、1-6 ページの「[Recovery Manager コマンドの概要](#)」を参照してください。

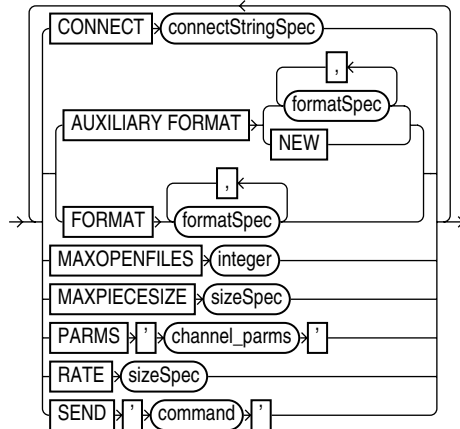
## allocOperandList

### 用途

`allocOperandList` 副次句を使用すると、Recovery Manager とデータベース・インスタンス間の接続であるチャンネルのオプションを制御できます。

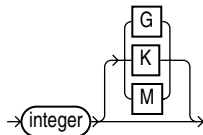
### 構文

**allocOperandList::=**



(`connectStringSpec::=` (3-12 ページ)、`formatSpec::=` (3-21 ページ))

**sizeSpec::=**



### セマンティクス

構文の要素	説明
CONNECT <code>connectStringSpec</code>	<p>Recovery Manager がバックアップまたはリストアの操作を行うデータベース・インスタンスへの接続文字列を指定します。Oracle RAC 構成で別々のインスタンスに操作を分散させる場合は、このパラメータを使用します。</p> <p>このパラメータを指定せず、AUXILIARY オプションも指定しなかった場合、Recovery Manager は、コマンドラインの CONNECT パラメータ、または CONNECT コマンドで指定したターゲット・データベース・インスタンスに対してすべての操作を実行します。一般的に、CONNECT パラメータと AUXILIARY オプションを一緒に使用することはできません。</p> <p><b>関連項目</b> : 3-12 ページの「<code>connectStringSpec</code>」および 2-244 ページの「RMAN」</p>
AUXILIARY FORMAT	<p>補助インスタンス上で作成されるイメージ・コピーの形式を指定します。</p> <p>Recovery Manager は補助インスタンスに CONNECT AUXILIARY で接続し、補助チャンネルにアクセスできる必要があります。</p>

構文の要素	説明
<i>formatSpec</i>	<p>補助インスタンス上のイメージ・コピー名のパターンを指定します。</p> <p>このパスは、補助ホスト上で有効であることが必要です。</p> <p><b>関連項目：</b>有効な置換変数については、3-21 ページの「<i>formatSpec</i>」を参照してください。</p>
NEW	<p>補助インスタンスの DB_CREATE_FILE_DEST 初期化パラメータで指定したディレクトリに、イメージ・コピーを作成します。</p> <p>イメージ・コピーは、Oracle Managed Files の形式で指定されます。</p>
FORMAT <i>formatSpec</i>	<p>このチャンネルで作成するバックアップ・ピースまたはイメージ・コピーの名前に使用する形式を指定します。例 3-1 でこの方法を説明します。</p> <p>FORMAT パラメータが役立つのは、複数のディスク・チャンネルを割り当てて、各チャンネルで別々のディレクトリに書き込む場合です。CONFIGURE CHANNEL または ALLOCATE CHANNEL で指定した FORMAT パラメータは、(2-47 ページの「forRecoveryOfSpec」の DATAFILECOPY FORMAT パラメータではなく) BACKUP コマンドで指定した FORMAT パラメータと同じ意味です。BACKUP コマンドで FORMAT パラメータを指定すると、CONFIGURE CHANNEL または ALLOCATE CHANNEL で指定した FORMAT パラメータがオーバーライドされます。</p> <p>FORMAT を指定しない場合、Recovery Manager はデフォルトの %U を使用します。これによって、バックアップ・ファイルの名前が一意であることが保証されます。フラッシュ・リカバリ領域が構成されている場合、Recovery Manager は、デフォルトのディスクの場所にバックアップ・ファイルを作成します。構成されていない場合、デフォルトのディスクの場所はオペレーティング・システム固有（たとえば、Solaris では ?/dbs）です。</p> <p>指定できる FORMAT 文字列は 4 つ以内です。Recovery Manager で 2 番目、3 番目および 4 番目の値が使用されるのは、BACKUP COPIES、SET BACKUP COPIES または CONFIGURE ... BACKUP COPIES が有効な場合のみです。各バックアップ・ピースに使用する形式を選択すると、Recovery Manager では最初の形式値がコピー 1、2 番目の形式値がコピー 2 というように順番に使用されます。形式値の数がコピー数より多ければ、余分の形式は使用されません。形式値の数がコピー数より少なければ、Recovery Manager では最初の形式値から順番に再利用されます。</p> <p>チャンネルはターゲット・データベース上のサーバー・セッションに対応するため、FORMAT 文字列ではクライアント・ホストではなくターゲット・ホストの表記規則を使用する必要があります。たとえば、Recovery Manager クライアントが Windows ホスト上で実行され、ターゲット・データベースが Linux ホスト上で実行されている場合は、Linux ファイル・システムまたは RAW デバイスのネーミング規則に従って FORMAT 文字列を指定する必要があります。</p> <p><b>関連項目：</b>使用可能な FORMAT パラメータについては、3-21 ページの「<i>formatSpec</i>」を参照してください。</p>
MAXOPENFILES <i>integer</i>	<p>BACKUP コマンドが一度にオープンできる入力ファイルの最大数（デフォルトは 8）を制御します。このパラメータは、多くのファイルのバックアップを 1 つのバックアップ・セットとして作成するときに、「オープンされているファイルが多すぎます。」というエラー・メッセージを出さないために使用します。</p>
MAXPIECESIZE <i>sizeSpec</i>	<p>このチャンネル上で作成される各バックアップ・ピースの最大サイズを指定します。例 3-2 にこの方法を示します。サイズはバイト単位、KB 単位 (k)、MB 単位 (M) または GB 単位 (G) で指定します。デフォルト設定はバイト単位で、KB 単位になるように切り捨てられます。たとえば、MAXPIECESIZE を 5000 に設定すると、最大ピース・サイズは 5000 バイトの KB 単位の下限である 4KB に設定されます。</p> <p><b>注意：</b>BACKUP...SECTION SIZE と MAXPIECESIZE を一緒に使用することはできません。</p>

構文の要素	説明
PARMS 'channel_parms'	<p>sbt チャンネルのパラメータを指定します。例 3-3 に、この方法を示します。DEVICE TYPE DISK を指定している場合は、このポート固有の文字列を使用しないでください。</p> <p>チャンネル・パラメータには、次の書式を使用できます。</p> <ul style="list-style-type: none"> <li>■ 'ENV=(var1=val1, var2=val2,...)'</li> </ul> <p>この Recovery Manager クライアントに対応するサーバー・セッションで、メディア・マネージャが必要とする 1 つ以上の環境変数を指定します。Recovery Manager はクライアント・プログラムであるため、ENV パラメータを使用すると、Recovery Manager クライアントのかわりに操作を実行するサーバー・セッション固有の変数を、PARMS 'ENV=(OB_DEVICE_1=tape1)' のように設定できます。</p> <ul style="list-style-type: none"> <li>■ 'SBT_LIBRARY= lib_name'</li> </ul> <p>この sbt チャンネルで使用するメディア・ライブラリを、PARMS="SBT_LIBRARY=/oracle/lib/mmv.so" のように指定します。デフォルト・ライブラリはオペレーティング・システム固有 (Linux では libobk.so、Windows では ORASBT.DLL など) です。</p> <p><b>関連項目:</b> メディア管理ライブラリの統合方法は、『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』を参照してください。</p>
RATE sizeSpec	<p>Recovery Manager がこのチャンネルで 1 秒に読み取ることができる最大バイト数 (デフォルト)、KB 数 (k)、MB 数 (M) または GB 数 (G) を設定します。このパラメータでは、Recovery Manager が必要以上にディスク帯域幅を使用してパフォーマンスが低下しないように、読取りバイト数の上限を設定します。</p>
SEND 'command'	<p>ベンダー固有のコマンド文字列を割り当てられたチャンネルすべてに送信します。たとえば、SEND 'OB_MEDIA_FAMILY datafile_mf' で、Oracle Secure Backup メディア・ファミリを指定できます。</p> <p><b>関連項目:</b> この機能のサポートの有無および使用する時期については、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。</p>

### sizeSpec

この副次句は、データのサイズを指定します。構文図は、3-2 ページの「sizeSpec::=」を参照してください。

構文の要素	説明
integer [G   K   M]	データのサイズを GB 単位 (G)、KB 単位 (K) または MB 単位 (M) で指定します。

## 例

### 例 3-1 ディスク・バックアップのデフォルトの場所の指定

この例では、デフォルト以外の形式を指定するディスク・チャンネルを割り当ててから、デフォルト以外の場所にデータベースをバックアップします。

```

RUN
{
  ALLOCATE CHANNEL d1 DEVICE TYPE DISK FORMAT = '/disk1/bkup_%U';
  BACKUP DATABASE;
}

```



### 例 3-2 バックアップ・ピースの最大サイズの設定

この例では、Oracle Secure Backup テープ・ドライブを指定する SBT チャンネルを手動で割り当てて、データベース全体のバックアップを作成します。テープに書き込まれるバックアップ・ピースのサイズが 800MB を超えないように、MAXPIECESIZE パラメータで指定します。

```
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt
    PARMS 'SBT_LIBRARY=/usr/local/oracle/backup/lib/libobk.so, ENV=(OB_DEVICE_1=stape1)'
    MAXPIECESIZE 800M;
  BACKUP DATABASE;
}
```

### 例 3-3 SBT チャンネル・パラメータの設定

この例では、stape1 という名前の Oracle Secure Backup テープ・ドライブを使用するようにデフォルトの SBT チャンネルを構成し、デフォルト・チャンネルでデータベース・バックアップを作成します。

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE CHANNEL DEVICE TYPE sbt PARMS 'ENV=(OB_DEVICE_1=stape1)';
BACKUP DATABASE;
```

後で、ドライブ stape2 にデータベースをバックアップすることになります。次の例では、手動で割り当てた SBT チャンネルを使用して、stape2 にデータベースをバックアップします。

```
RUN
{
  ALLOCATE CHANNEL st2 DEVICE TYPE sbt
    PARMS 'ENV=(OB_DEVICE_1=stape2)';
  BACKUP DATABASE;
}
```

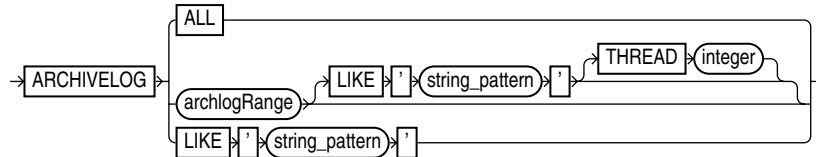
## archivelogRecordSpecifier

### 用途

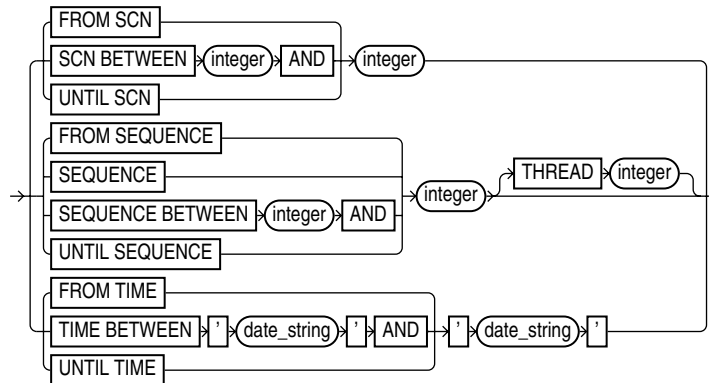
*archivelogRecordSpecifier* 副次句を使用すると、Recovery Manager の操作で使用されるアーカイブ REDO ログ・セットを指定できます。

### 構文

#### *archivelogRecordSpecifier*::=



#### *archlogRange*::=



### セマンティクス

#### *archivelogRecordSpecifier*

この副次句では、すべてのアーカイブ REDO ログ、または指定したパターンと一致するファイル名のログのいずれかを指定します。

構文の要素	説明
ALL	使用可能なすべてのアーカイブ・ログを指定します。
LIKE ' <i>string_pattern</i> '	指定した <i>string_pattern</i> と一致するすべてのアーカイブ・ログを指定します。SQL 言語の LIKE 演算子で有効なのと同じパターン一致文字を使用して、複数ファイルを一致させることができます。 <b>関連項目</b> : Oracle RAC 構成での Recovery Manager の使用方法については、『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。

## archlogRange

この副次句では、SCN、時刻、リストア・ポイント（タイムスタンプまたはSCNに対するラベル）またはログ順序番号で、アーカイブ REDO ログの範囲を指定します。この副次句は、データベースのリカバリ可能な時点を識別するのに有効です。

Recovery Manager は V\$ARCHIVED\_LOG または RC\_ARCHIVED\_LOG ビューに問合せを行って、範囲に含まれるログを特定します。時刻、SCN またはリストア・ポイントを指定した場合、Recovery Manager はアーカイブ REDO ログの内容から範囲を特定します。ログが作成またはバックアップされた時刻からではありません。ログ順序番号で範囲を指定した場合、Recovery Manager は順序番号を使用して範囲を特定します。

Recovery Manager で、範囲内に含まれるログを特定する方法を表 3-1 に示します。FIRST\_TIME、NEXT\_TIME などの列は、V\$ARCHIVED\_LOG の列を表しています。たとえば、FROM SCN 1000 UNTIL SCN 2000 と指定した場合、Recovery Manager では V\$ARCHIVED\_LOG.NEXT\_SCN の値が 1000 より大きく、V\$ARCHIVED\_LOG.FIRST\_SCN の値が 2000 以下のすべてのログが範囲内となります。

表 3-1 範囲に含まれるログの特定方法

副次句	FIRST_TIME	NEXT_TIME	FIRST_SCN	NEXT_SCN	SEQUENCE#
FROM TIME t1	N/A	t1 より後	N/A	N/A	N/A
FROM TIME t1 UNTIL TIME t2	t2 以前	t1 より後	N/A	N/A	N/A
UNTIL TIME t2	t2 以前	N/A	N/A	N/A	N/A
FROM SCN s1	N/A	N/A	N/A	s1 より大きい	N/A
FROM SCN s1 UNTIL SCN s2	N/A	N/A	s2 以下	s1 より大きい	N/A
UNTIL SCN s2	N/A	N/A	s2 以下	N/A	N/A
FROM SEQUENCE q1	N/A	N/A	N/A	N/A	q1 以上、最大順序番号以下
FROM SEQUENCE q1 UNTIL SEQUENCE q2	N/A	N/A	N/A	N/A	q1 以上、q2 以下
UNTIL SEQUENCE q2	N/A	N/A	N/A	N/A	0 以上、q2 以下

時刻は、現在有効なグローバル化セッション・テクノロジーの日付書式指定に従って書式化されている必要があります。現行のグローバル化セッション設定で時刻が指定されていない場合、日付文字列のデフォルトは 00 時、00 分、00 秒です。

date\_string には、SYSDATE などの DATE 型の SQL 式を使用することができます。TO\_DATE 関数を使用すると、現行のグローバル化セッション・テクノロジー設定に関係なく動作するハードコードされた日付を指定できます。SYSDATE の精度は、グローバル化セッション設定に関係なく常に秒単位です。したがって、今日が 2007 年 3 月 15 日の場合、SYSDATE-10 は 05-MAR-07 および TO\_DATE('03/15/2007', 'MM/DD/YYYY') -10 とは異なります。

アーカイブ REDO ログの順序を指定した場合に、すべての REDO データがその順序で Recovery Manager に含まれるという保証はありません。たとえば、使用可能な最新のアーカイブ・ログ・ファイルの最終時点が順序の最後より前の場合や、範囲内のアーカイブ・ログ・ファイルがすべてのアーカイブ先から欠落している場合があります。Recovery Manager は、検索されたアーカイブ REDO ログは選択しますが、ファイルの欠落については警告を発行しません。

**関連項目：** NLS\_LANG および NLS\_DATE\_FORMAT 環境変数を使用して時刻の書式を指定する方法については、『Oracle Database リファレンス』を参照してください。

構文の要素	説明
FROM SCN <i>integer</i>	アーカイブ REDO ログ・ファイルの範囲に対する開始 SCN を指定します。UNTIL SCN パラメータを指定しない場合、Recovery Manager は FROM SCN パラメータで指定した SCN から始まるすべての使用可能なログ・ファイルを組み込みます。
SCN BETWEEN <i>integer</i> AND <i>integer</i>	ログの範囲に対して開始 SCN と終了 SCN を指定します。この構文は、FROM SCN <i>integer1</i> UNTIL SCN <i>integer2</i> と同じ意味です。
UNTIL SCN <i>integer</i>	アーカイブ REDO ログ・ファイルの範囲に対して終了 SCN を指定します（範囲決定のルールは、表 3-1 を参照）。
FROM SEQUENCE <i>integer</i>	一連のアーカイブ REDO ログ・ファイルに対して開始ログ順序番号を指定します（範囲決定のルールは、表 3-1 を参照）。 <b>注意：</b> 例 3-6 に示す構文を使用して、スレッド内のすべてのログ順序番号を指定できます。
SEQUENCE <i>integer</i>	単一のログ順序番号を指定します。
SEQUENCE BETWEEN <i>integer</i> AND <i>integer</i>	ログ順序番号の範囲を指定します。この構文は、FROM SEQUENCE <i>integer1</i> UNTIL SEQUENCE <i>integer2</i> と同じ意味です。
UNTIL SEQUENCE <i>integer</i>	一連のアーカイブ REDO ログ・ファイルに対して終了ログ順序番号を指定します（範囲決定のルールは、表 3-1 を参照）。
THREAD <i>integer</i>	組み込むアーカイブ REDO ログ・ファイルが含まれているスレッドを指定します。このパラメータは、Oracle RAC 構成のデータベースの場合のみ必要です。  SEQUENCE パラメータに THREAD の指定は必須ではありませんが、ログ順序を指定すると、常にスレッドも暗黙的に指定したことになります。スレッドを指定しなければ、デフォルトで 1 になります。V\$ARCHIVED_LOG を問合せ、ログのスレッド番号を確認してください。
FROM TIME ' <i>date_string</i> '	アーカイブ REDO ログの範囲に対して開始時刻を指定します（範囲決定のルールは、表 3-1 を参照）。例 3-5 に、LIST コマンドでの FROM TIME の使用例があります。
TIME BETWEEN ' <i>date_string</i> ' AND ' <i>date_string</i> '	時刻の範囲を指定します。この構文は、FROM TIME ' <i>date_string</i> ' UNTIL TIME ' <i>date_string</i> ' と同じ意味です。
UNTIL TIME ' <i>date_string</i> '	アーカイブ REDO ログの範囲に対して終了時間を指定します（範囲決定のルールは表 3-1 を参照）。例 3-4 に、BACKUP コマンドでの UNTIL TIME の使用例があります。

## 例

### 例 3-4 リカバリの時点によるレコードの指定

現在より 5 日前の時点までデータベースのリカバリが可能だとします。この Point-in-Time リカバリを可能にするアーカイブ REDO ログの範囲をバックアップすることが必要です。

UNTIL TIME 'SYSDATE-5' 句を使用すると、最初の変更のタイムスタンプ (V\$ARCHIVED\_LOG.FIRST\_TIME の値) が SYSDATE-5 以前になっているすべてのログを範囲に含むように指定できます。この関数は、現在より 5 日前の時刻に秒単位の精度で戻します。

```
CONNECT TARGET /
BACKUP ARCHIVELOG UNTIL TIME 'SYSDATE-5';
```

**例 3-5 アーカイブ・ログ・バックアップの時刻によるリスト**

3-7 ページの表 3-1 に示すように、アーカイブ REDO ログの範囲に対して `date_string` を指定する場合、バックアップ時刻またはログの作成時刻は指定しません。アーカイブ・ログ 32 の次の変更のタイムスタンプが 3 月 6 日になっているとします。

```
SQL> SELECT SEQUENCE#, NEXT_TIME
       2 FROM V$ARCHIVED_LOG
       3 WHERE SEQUENCE#='32';
```

```
SEQUENCE# NEXT_TIME
-----
50 06-MAR-07
```

3 月 8 日に、次の BACKUP コマンドおよび LIST コマンドを実行します。

```
RMAN> BACKUP ARCHIVELOG SEQUENCE 32;
```

```
Starting backup at 08-MAR-07
allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=109 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: Oracle Secure Backup
channel ORA_SBT_TAPE_1: starting archived log backup set
channel ORA_SBT_TAPE_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=32 RECID=125 STAMP=616528547
channel ORA_SBT_TAPE_1: starting piece 1 at 08-MAR-07
channel ORA_SBT_TAPE_1: finished piece 1 at 08-MAR-07
piece handle=6kic3fkm_1_1 tag=TAG20070308T111014 comment=API Version 2.0,MMS Version 10.1.0.3
channel ORA_SBT_TAPE_1: backup set complete, elapsed time: 00:00:25
Finished backup at 08-MAR-07
```

```
Starting Control File and SPFILE Autobackup at 08-MAR-07
piece handle=c-28014364-20070308-08 comment=API Version 2.0,MMS Version 10.1.0.3
Finished Control File and SPFILE Autobackup at 08-MAR-07
```

```
RMAN> LIST BACKUP OF ARCHIVELOG FROM TIME 'SYSDATE-1';
```

```
RMAN>
```

ログ 32 の次の変更のタイムスタンプは FROM TIME 句で指定した時刻範囲より古いいため、前述の LIST BACKUP コマンドでは、アーカイブ・ログ 32 のバックアップは表示されません。

**例 3-6 REDO スレッドでのすべてのログのクロスチェック**

2 つの REDO スレッドを使用する Oracle RAC データベースを管理しているとします。この例では、スレッド 1 でのみすべてのアーカイブ REDO ログをクロスチェックします。

```
CROSSCHECK ARCHIVELOG FROM SEQUENCE 0 THREAD 1;
```

## completedTimeSpec

### 用途

`completedTimeSpec` 副次句を使用すると、バックアップまたはコピーが完了する時刻を指定できます。

### 使用上の注意

すべての日付文字列は、次のいずれかにしてください。

- 現在有効なグローバル化バージョン・テクノロジーの日付書式指定に従って書式化されていること
- 次の例のように、DATE 値を戻す SQL 式で作成されること

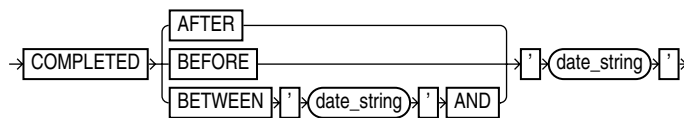
- 'SYSDATE-30'

- TO\_DATE('09/30/2000 08:00:00', 'MM/DD/YY HH24:MI:SS')

TO\_DATE 関数により、現行のグローバル化バージョン・テクノロジーの環境変数の設定に関係なく日付を指定できます。

### 構文

**completedTimeSpec::=**



### セマンティクス

構文の要素	説明
AFTER 'date_string'	その時刻より後に完了したバックアップを指定するための時刻です (例 3-7 を参照)。
BEFORE 'date_string'	その時刻より前に完了したバックアップを指定するための時刻です (例 3-9 を参照)。
BETWEEN 'date_string' AND 'date_string'	その時間範囲の間に完了したバックアップを指定するための時間範囲です (例 3-8 を参照)。BETWEEN 'date1' AND 'date2' は、AFTER 'date1' BEFORE 'date2' と同じであることに注意してください。

### 例

#### 例 3-7 時間範囲内のバックアップのクロスチェック

この例では、前月に作成したデータベースのバックアップ・セットをクロスチェックします。

```
CROSSCHECK BACKUP OF DATABASE
COMPLETED BETWEEN 'SYSDATE-62' AND 'SYSDATE-31';
```

#### 例 3-8 期限切れのバックアップの削除

この例では、最近 2 週間に作成したアーカイブ・ログの期限切れのバックアップを削除します。

```
DELETE EXPIRED BACKUP OF ARCHIVELOG ALL
COMPLETED AFTER 'SYSDATE-14';
```

### 例 3-9 コピーのリスト

この例では、2007年3月9日より前に作成された、データファイル  
/disk1/oradata/prod/users01.dbf のイメージ・コピーをリストします。

```
RMAN> LIST COPY OF DATAFILE '/disk1/oradata/prod/users01.dbf' COMPLETED BEFORE '9-MAR-07';
```

```
List of Datafile Copies
```

```
=====
```

Key	File S	Completion Time	Ckp SCN	Ckp Time
3794	28 A	06-MAR-07	1010097	06-MAR-07
	Name: /disk1/oradata/prod/users01.dbf			
3793	28 A	06-MAR-07	1009950	06-MAR-07
	Name: /disk2/PROD/datafile/o1_mf_users_2yvg4v6o_.dbf			
	Tag: TAG20070306T105314			

## connectStringSpec

### 用途

`connectStringSpec` 副次句を使用すると、ユーザー名、パスワードおよびネット・サービス名を指定してターゲット・データベース、リカバリ・カタログ・データベースまたは補助データベースに接続できます。この接続は、ユーザーの認証とデータベースの識別のために必要です。

### 前提条件

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するには、SYSDBA 権限が必要です。リカバリ・カタログ・データベースに接続するときは、ユーザー SYS で接続しないでください。

### 使用上の注意

データベースへの SQL\*Plus 接続と同様に、データベースへの Recovery Manager 接続が指定され、認証されます。唯一異なるのは、ターゲット・データベースまたは補助データベースへの Recovery Manager 接続では SYSDBA 権限が必要なことです。AS SYSDBA キーワードは暗黙的に指定されており、明示的に指定することはできません。SQL\*Plus 使用時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。

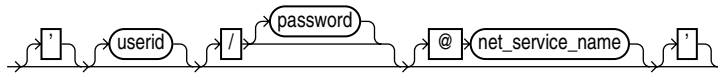
---

**注意：** セキュリティ上、パスワードはコマンドラインにプレーン・テキストで入力しないでください。Recovery Manager プロンプトで要求された場合のみ、Recovery Manager に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。

---

### 構文

`connectStringSpec ::=`



### セマンティクス

構文の要素	説明
/	ユーザー ID またはパスワードを指定せずにターゲット・データベースに接続した場合、スラッシュ (/) が、オペレーティング・システム認証機能を使用して、ユーザー SYS での接続を確立します (例 3-12 を参照)。 <b>注意：</b> スラッシュ (/) は、プラットフォーム固有の環境変数に依存します。
userid	指定したユーザーに対してデータベース接続を確立します。パスワードを指定しない場合、Recovery Manager はプロンプトを表示して対話方式でパスワードを取得します (例 3-11 を参照)。文字は端末には表示されません。 <b>注意：</b> 接続文字列に空白を含めることはできませんが、スラッシュ (/) やアットマーク (@) などの文字を含めることはできます。



構文の要素	説明
<code>/password</code>	パスワードを使用して指定したユーザーに対して接続を確立します。ターゲット・データベースがオープンされていない場合は、パスワード・ファイルの存在が必要です。 <b>注意:</b> コマンドラインにプレーン・テキストで入力したパスワードは、セキュリティ上脆弱です。プロンプトに対してパスワードを入力するなど、より安全な方法が利用できます。
<code>@net_service_name</code>	オプションの Oracle Net ネット・サービス名を使用してデータベースへの接続を確立します (例 3-10 を参照)。

## 例

### 例 3-10 リカバリ・カタログを使用しないターゲット・データベースへの接続

この例では、データベース接続を指定せずに Recovery Manager を起動します。CONNECT コマンドでは、Oracle Net サービス名 prod を使用して、デフォルトの NOCATALOG モードでターゲット・データベースに接続します。プロンプトに対して SYS パスワードを入力します。

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

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

### 例 3-11 オペレーティング・システムのコマンドラインでのターゲット・データベースへの接続

この例では、パスワードを指定せずに、オペレーティング・システムのコマンドラインでユーザー SYS としてターゲット・データベースに接続します。Recovery Manager によって、パスワードの入力を求めるプロンプトが表示されます。

```
% rman TARGET SYS

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.

target database Password: password
```

### 例 3-12 オペレーティング・システム認証を使用したターゲット・データベースへの接続

この例では、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 prod に接続します。また、ネット・サービス名を使用して、リカバリ・カタログ・データベース catdb にも接続します。

```
% rman
RMAN> CONNECT TARGET /

connected to target database: PROD (DBID=39525561)

RMAN> CONNECT CATALOG rman@catdb

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

## datafileSpec

### 用途

*datafileSpec* 副次句を使用すると、ファイル名または絶対ファイル番号でデータファイルを指定できます。

### 構文

***datafileSpec*::=**



### セマンティクス

構文の要素	説明
'filename'	絶対パスまたは相対ファイル名のいずれかでデータファイルを指定します。相対ファイル名を指定した場合、ファイル名はターゲット・データベースによってポート固有の方法で修飾されます。Oracle ホームを表すには?、Oracle SIDを表すには@を使用できます (例 3-14 を参照)。 二重引用符と一重引用符はどちらも有効です (構文図では、一重引用符のみを使用しています)。 <b>関連項目:</b> 一重引用符と二重引用符の相違点、および Recovery Manager での引用符付き文字列における環境変数の動作については、1-3 ページの「 <a href="#">Recovery Manager 構文内の引用符</a> 」を参照してください。
integer	データファイルを絶対ファイル番号で指定します (例 3-13 を参照)。ファイル番号は、V\$DATAFILE、V\$DATAFILE_COPY、V\$DATAFILE_HEADER ビューまたは REPORT SCHEMA コマンドの出力から取得します。

### 例

#### 例 3-13 ファイル名によるデータファイルの指定

この例では、ファイル名を指定して、データファイル /disk1/oradata/prod/users01.dbf をディスクにコピーします。

```
BACKUP AS COPY
  DATAFILE '/disk1/oradata/prod/users01.dbf'
  FORMAT '/disk2/users01.cpy';
```

#### 例 3-14 絶対ファイル番号によるデータファイルの指定

この例では、ファイル番号を指定して、データファイル 1 および 2 をディスクにコピーします。

```
BACKUP AS COPY
  DATAFILE 1 FORMAT '/disk2/df1.cpy'
  DATAFILE 2 FORMAT '/disk2/df1.cpy';
```

## deviceSpecifier

### 用途

`deviceSpecifier` 副次句を使用すると、バックアップに使用するストレージのタイプを指定できます。

### 構文

**deviceSpecifier ::=**



### セマンティクス

構文の要素	説明
DISK	ディスク・ストレージ・デバイスを指定します (3-15 ページの例 3-16 を参照)。
<code>media_device</code>	<p>ストレージとして使用する順次 I/O デバイスまたはアクセス方法を指定します (3-15 ページの例 3-15 を参照)。</p> <p><code>media_device</code> 変数では、メディア・マネージャの名前 (大 / 小文字は区別されません) を指定します。順次 I/O デバイス・タイプの構文とセマンティクスはプラットフォームによって異なります。最も一般的な値は <code>sbt</code> または <code>sbt_tape</code> です (これらの値はシノニムです)。</p> <p><b>注意:</b> Recovery Manager は、値 <code>sbt</code> を <code>sbt_tape</code> としてリカバリ・カタログに格納して、下位互換性を保持します。</p>

### 例

#### 例 3-15 テープ・チャンネルの割当て

この例では、メディア管理デバイスにメンテナンス・チャンネルを割り当てます。

```
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt;
CROSSCHECK BACKUP;
RELEASE CHANNEL;
```

#### 例 3-16 データベースのディスクへのバックアップ

この例では、データベースをディスクにバックアップします。

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

## fileNameConversionSpec

### 用途

`fileNameConversionSpec` 副次句を使用すると、古いファイル名を基にして新しいファイル名を生成する際に使用される 1 つ以上のパターンを指定できます。出力ファイル名の生成方法の 1 つとして、**BACKUP**、**CONVERT** および **DUPLICATE** とともに使用されます。

### 使用上の注意

文字列パターンのルールと、このルールがファイルの命名方法に与える影響は、初期化パラメータ `DB_FILE_NAME_CONVERT` の場合と同じです。カッコ内には、偶数の文字列パターンを指定してください。

**Recovery Manager** は、古い名前に基づいて新しいファイル名を生成するときに、文字列パターンの各ペアの最初のメンバーと元のファイル名を比較します。元の名前のサブストリングであるパターンが最初に検出されると、マッチしたサブストリングがペアの 2 番目のメンバーで置換されて新しいファイル名が生成されます。

パターン・マッチは、ファイル名の 1 文字目からである必要はありません。次のコマンドでは、データファイル `/disk1/dbs/users.dbf` のイメージ・コピーが `/disk1/newdbs/users.dbf` として作成されます。

```
BACKUP AS COPY TABLESPACE users
  DB_FILE_NAME_CONVERT = ('dbs', 'newdbs');
```

指定された変換ファイル名に複数のマッチがある場合、**Recovery Manager** は、パターン・リスト内の最初のマッチを使用して、新しいファイル名を生成します。次のコマンドの場合、前述の例と同じ結果になります。パターン `'dbs'` がファイル名にマッチし、2 番目のパターン `'/disk1'` と比較されないためです。

```
BACKUP AS COPY TABLESPACE users
  DB_FILE_NAME_CONVERT = ('dbs', 'newdbs', '/disk1', '/newdisk');
```

**CONVERT TABLESPACE**、**CONVERT DATABASE** および **BACKUP AS COPY** コマンドでは、操作対象のソース・ファイルが **Oracle Managed Files** の場合、`fileNameConversionSpec` を使用してソース・ファイル名を新しい出力ファイル名に変換することはできません。自動ストレージ管理 (ASM) または通常ファイル・システム・ストレージにある **Oracle Managed Files** の場合、データベースで出力ファイルの名前が生成できることが必要です。たとえば、ASM 以外のストレージのデータファイルの OMF ファイル名は、次の形式です。

```
/private/boston/datafile/01_mf_system_ab12554_.dbf
```

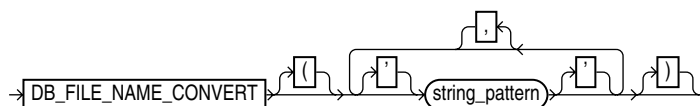
ASM ストレージの OMF ファイル名は、次の形式です。

```
+DISK/boston/datafile/system.256.4543080
```

OMF ファイル名の生成および管理はデータベースでのみ行えます。通常、異なるディスク・グループまたは異なる OMF の場所の名前を OMF ファイル名に置き換えても、新しい場所に有効なファイル名は生成されません。異なる OMF の場所にあるストレージの OMF ファイル名を変換するには、これらのコマンドで **FORMAT** 句などの代替方法を使用して新しい出力場所を指定し、データベースが特定の出力ファイル名を管理できるようにします。

### 構文

**fileNameConversionSpec::=**



## セマンティクス

構文の要素	説明
<code>string_pattern</code>	<p>ファイル名の変換に使用される文字列のペアを指定します。必要な数のプライマリとスタンバイの置換文字列のペアを使用できます。たとえば、次のように値を文字列パターンとして設定できます。</p> <pre>DB_FILE_NAME_CONVERT = ('str1','str2','str3', 'str4' ...)</pre> <p>この例では、<code>str1</code> は元のファイル名と一致するパターンで、<code>str2</code> は <code>str1</code> の置換文字列として生成されるパターンです。また、<code>str3</code> は元のファイル名と一致するパターンで、<code>str4</code> は <code>str3</code> の置換文字列として生成されるパターンです。</p>

## 例

### 例 3-17 単一の変換ペアを持つ DB\_FILE\_NAME\_CONVERT の使用

この例では、表領域 `users` にはディレクトリ `/disk1/oradata/prod/` にあるデータファイルが含まれており、表領域 `tools` には `/disk1/oradata/prod/` にあるデータファイルが含まれています。変換対象の各データファイルに対して、`disk1/oradata/prod` がファイル名のサブストリングである場合、その文字列を `disk2` に置換して、イメージ・コピー名が作成されます。

```
BACKUP AS COPY
  DB_FILE_NAME_CONVERT = ('disk1/oradata/prod','disk2')
  TABLESPACE users, tools;
```

### 例 3-18 複数の変換ペアを持つ DB\_FILE\_NAME\_CONVERT の使用

この例では、例 3-17 で説明したデータファイルと同じデータファイルのイメージ・コピーを作成します。各ペアの最初の文字列で、ソース・データファイルの名前と比較するパターンを指定します。各ペアの 2 番目の文字列は、イメージ・コピーの名前を生成するときに使用する置換パターンです。

```
BACKUP AS COPY
  DB_FILE_NAME_CONVERT=('/disk1/oradata/prod/users','/disk2/users',
                        '/disk1/oradata/prod/tools','/tmp/tools')
  TABLESPACE tools, users;
```

このコマンドの次のサンプル出力は、Recovery Manager で、変換ペアを使用して出力イメージ・コピーに名前を付ける方法を示しています。

```
Starting backup at 08-MAR-07
using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile copy
input datafile file number=00027 name=/disk1/oradata/prod/tools01.dbf
output file name=/tmp/tools01.dbf tag=TAG20070308T143300 RECID=33 STAMP=616689181
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile copy
input datafile file number=00028 name=/disk1/oradata/prod/users01.dbf
output file name=/disk2/users01.dbf tag=TAG20070308T143300 RECID=34 STAMP=616689182
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
Finished backup at 08-MAR-07

Starting Control File and SPFILE Autobackup at 08-MAR-07
piece handle=/disk2/PROD/autobackup/2007_03_08/o1_mf_s_616689184_2z13s1kx_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 08-MAR-07
```

## forDbUniqueNameOption

### 用途

*forDbUniqueNameOption* 副次句を使用すると、Data Guard 環境ですべてのデータベースまたは特定のデータベースを指定できます。

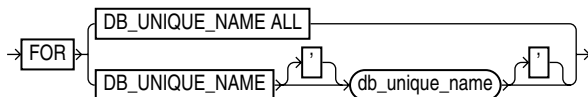
### 使用上の注意

プライマリ・データベースの DBID は、これに関連付けられているフィジカル・スタンバイ・データベースの DBID と同じです。リカバリ・カタログ内のデータベースは、DBID およびデータベースの DB\_UNIQUE\_NAME 初期化パラメータの値によって一意に識別されます。

任意のコマンドに *forDbUniqueNameOption* を指定すると、Recovery Manager は操作の対象を、DB\_UNIQUE\_NAME が指定されたデータベースにのみ関連付けられているオブジェクトに制限します。たとえば、LIST コマンドでこのオプションを使用すると、Recovery Manager では、DB\_UNIQUE\_NAME が指定されたデータベースにのみ関連付けられているオブジェクトのみがリストされます。いずれのデータベースにも関連付けら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）はリストされないことに注意してください。

### 構文

**forDbUniqueNameOption::=**



### セマンティクス

構文の要素	説明
FOR DB_UNIQUE_NAME ALL	ターゲット・データベースの DBID（または SET DBID コマンドで指定された DBID）を共有する、リカバリ・カタログのすべてのプライマリ・データベースおよびスタンバイ・データベースを指定します。
FOR DB_UNIQUE_NAME db_unique_name	db_unique_name が指定された、リカバリ・カタログのプライマリ・データベースまたはスタンバイ・データベースを指定します。

### 例

#### 例 3-19 スタンバイ・データベースに関連付けられた期限切れバックアップのリスト

この例では、リカバリ・カタログに接続し、Data Guard 環境に DBID を設定します。この環境のすべてのプライマリ・データベースおよびスタンバイ・データベースが、同じ DBID を共有します。LIST コマンドは、データベース standby1 に関連付けられたすべての期限切れバックアップをリストします。

```

RMAN> CONNECT CATALOG rman@catdb;

recovery catalog database Password: password
connected to recovery catalog database

RMAN> SET DBID 3257174182;
RMAN> LIST EXPIRED BACKUP FOR DB_UNIQUE_NAME standby1;
  
```

## foreignlogRecordSpecifier

### 用途

*foreignlogRecordSpecifier* 副次句を使用すると、Recovery Manager の操作で使用される外部アーカイブ REDO ログ・セットを指定できます。

### 使用上の注意

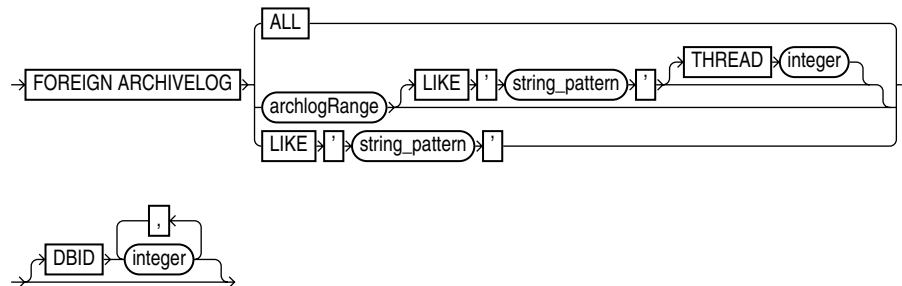
外部アーカイブ REDO ログは、LogMiner セッション中にロジカル・スタンバイ・データベースが受け取ります。通常のアーカイブ・ログとは異なり、外部アーカイブ・ログには別の DBID が使用されています。このため、それらのログをロジカル・スタンバイ・データベースでバックアップまたはリストアすることはできません。

次の条件が満たされる場合、ロジカル・スタンバイ・データベースではフラッシュ・リカバリ領域に外部アーカイブ・ログが作成されます。

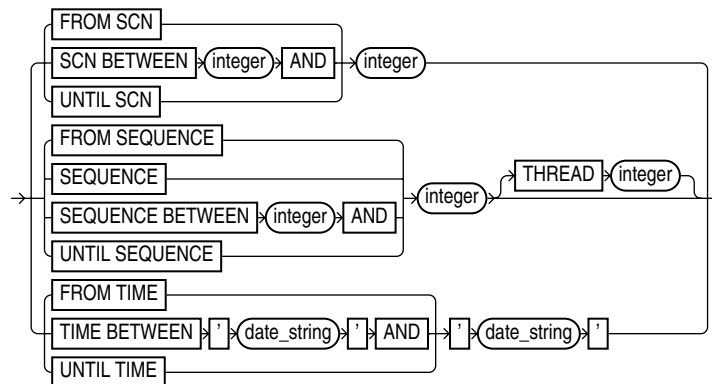
- フラッシュ・リカバリ領域がロジカル・スタンバイ・データベースで構成されている。
- LOG\_ARCHIVE\_DEST\_n 初期化パラメータの 1 つが、外部アーカイブ・ログを受け取るために、適切な VALID\_FOR 設定を使用して 'LOCATION=USE\_DB\_RECOVERY\_FILE\_DEST' に設定されている。
- COMPATIBLE 初期化パラメータが 11.0.0 以上に設定されている。

### 構文

**foreignlogRecordSpecifier::=**



**archlogRange::=**



## セマンティクス

ログが外部アーカイブ REDO ログであることを除いて、セマンティクスは 3-6 ページの [archivelogRecordSpecifier](#) と同じです。

## 例

### 例 3-20 外部アーカイブ REDO ログのクロスチェック

この例では、すべての外部アーカイブ REDO ログがクロスチェックされます。

```
RMAN> CROSSCHECK FOREIGN ARCHIVELOG ALL;
```



## formatSpec

### 用途

*formatSpec* 副次句を使用して、バックアップ・ピースまたはイメージ・コピーに対するファイル名の形式または自動ストレージ管理ディスク・グループを指定します。FORMAT パラメータに値を指定しない場合、Recovery Manager は、フラッシュ・リカバリ領域（使用可能な場合）にバックアップを作成するか、またはフラッシュ・リカバリ領域が使用不可の場合は、プラットフォーム固有のディレクトリ（たとえば UNIX 上の `*/dbs` など）にバックアップを作成します。いずれの場合も、Recovery Manager は変数 `%U` を使用してバックアップを指定します。

**ヒント：** 自動ストレージ管理のディスク・グループの作成方法および命名方法は、『Oracle Database SQL 言語リファレンス』を参照してください。

### 使用上の注意

各バックアップ・ピースまたはバックアップ・コピーの名前が一意であり、かつプラットフォーム上の順次ファイル名として有効であればどのような名前でも指定できます。ディスクにバックアップする場合は、一意の有効なファイル名であれば、どのようなディスク・ファイル名も使用できます。

Oracle Managed Files のファイル名は、バックアップ用の形式として指定できません。たとえば、`+DISK1/datafile/system.732.609791431` が OMF ファイル名の場合、そのファイル名を FORMAT パラメータに指定することはできません。

*format\_string* 全体がターゲット・インスタンスによりポート固有の方法で処理され、最終的なバックアップ・ピース名が導出されます。一意のファイル名を生成するために、3-22 ページの「[セマンティクス](#)」に示す置換変数を FORMAT 文字列で使用できます。この情報の書式設定は、プラットフォームに応じて異なります。

指定できる FORMAT 文字列は 4 つ以内です。Recovery Manager で 2 番目、3 番目および 4 番目の値が使用されるのは、[BACKUP COPIES](#)、[SET BACKUP COPIES](#)、または [CONFIGURE ... BACKUP COPIES](#) が有効な場合のみです。各バックアップ・ピースに使用する形式を選択すると、Recovery Manager では最初の形式値がコピー 1、2 番目の形式値がコピー 2 というように順番に使用されます。形式値の数がコピー数より多ければ、余分の形式は使用されません。形式値の数がコピー数より少なければ、Recovery Manager では最初の形式値から順番に再利用されます。

次の位置のいずれかに *format\_string* を指定します。優先順位は次のようになります。

1. [backupSpec](#) 句
2. [BACKUP](#) コマンド
3. [ALLOCATE CHANNEL](#) コマンド
4. [CONFIGURE CHANNEL](#) コマンド

前述の場所のうち、2 箇所以上に指定した場合、Recovery Manager は前述の順序で FORMAT パラメータを検索します。

### 構文

**formatSpec::=**

```
→ 'format_string'
```

## セマンティクス

### formatSpec

次の表に、書式文字列が有効な Recovery Manager 置換変数を示します。

構文の要素	説明
%a	データベースのアクティブ ID を指定します。
%c	1 セットの二重バックアップ・ピース内でバックアップ・ピースのコピー番号を指定します。バックアップを多重化していなければ、この変数はバックアップ・セットの場合は 1、プロキシ・コピーの場合は 0 となります。これらのコマンドの 1 つが使用可能になっている場合、この変数はコピー番号を示します。%c の最大値は 256 です。
%d	データベースの名前を指定します (3-23 ページの例 3-22 を参照)。
%D	グレゴリオ暦による月の現在の日付を DD 形式で指定します。
%e	アーカイブ・ログ順序番号を指定します。
%f	絶対ファイル番号を指定します (3-23 ページの例 3-22 を参照)。
%F	DBID、日、月、年および順序を組み合わせて固有で反復可能な生成名にします。この変数は、 <code>c-#####-YYYYMMDD-QQ</code> に変換されます。各項目の意味は次のとおりです。 <ul style="list-style-type: none"> <li>##### は DBID を表します。DBID は 10 進形式で出力されるため、ターゲット・データベースに簡単に対応付けることができます。</li> <li>YYYYMMDD は、バックアップの生成日を示すグレゴリオ暦のタイムスタンプです。</li> <li>QQ は、00 ~ 'FF' (256) の 16 進数による順序です。</li> </ul>
%h	アーカイブ REDO ログ・スレッド番号を指定します。
%I	DBID を指定します。
%M	グレゴリオ暦による月を MM 形式で指定します。
%N	表領域名を指定します。この置換変数は、データファイルをイメージ・コピーとしてバックアップする場合のみ有効です。
%n	データベース名を指定します。データベース名は合計 8 文字の桁数になるまで、右側に x が埋め込まれます。たとえば、prod1 がデータベース名であれば、埋込み後の名前は prod1xxx となります。
%p	バックアップ・セット内のピース番号を指定します。この値は、バックアップ・セットごとに 1 から始まり、バックアップ・ピースの作成ごとに 1 ずつ増えます。 <b>注意:</b> PROXY を指定した場合は、%p 変数を FORMAT 文字列内の %U に明示的または暗黙的に含める必要があります。
%s	バックアップ・セット番号を指定します。この番号は制御ファイルにあるカウンタで、バックアップ・セットごとに加算されます。カウンタ値は 1 から始まり、制御ファイルの存続期間中は一意です。バックアップ制御ファイルをリストアした場合は、重複した値になることがあります。また、CREATE CONTROLFILE によって、カウンタの値は 1 に初期化されます。
%t	バックアップ・セットのタイムスタンプを指定します。4 バイトの内部的な基準日時以降の経過秒数を表す値です。バックアップ・セットの一意の名前を構成するには、%s と %t の組合せを使用できます。
%T	グレゴリオ暦による年、月および日を YYYYMMDD 形式で指定します。
%u	バックアップ・セット番号またはイメージ・コピー番号と、これらの作成時刻を短く表現する 8 文字の名前を指定します。

構文の要素	説明
%U	<p>システムによって生成される一意のファイル名（デフォルト）を指定します。</p> <p>%Uの意味は、イメージ・コピーとバックアップ・ピースで異なります。バックアップ・ピースの場合は、%Uによって、生成されたバックアップ・ファイル名の一意性を保証する %u_%p_%c の短縮形が指定されます。データファイルのイメージ・コピーの場合、%Uには、次の意味があります。</p> <p>data-D-%d_id-%I_TS-%N_FNO-%f_%u</p> <p>アーカイブ REDO ログのイメージ・コピーの場合、%Uには、次の意味があります。</p> <p>arch-D_%d-id-%I_S-%e_T-%h_A-%a_%u</p> <p>制御ファイルのイメージ・コピーの場合、%Uには、次の意味があります。</p> <p>cf-D_%d-id-%I_%u</p>
%Y	年を YYYY 形式で指定します。
%%	パーセント (%) 文字を指定します。たとえば、%%Y は文字列 %Y に変換されます。

## 例

### 例 3-21 ASM ディスク・グループの指定

この例では、データベースを ASM ディスク・グループ DISK1 にコピーします。

```
BACKUP AS COPY DATABASE FORMAT '+DATAFILE';
```

### 例 3-22 データファイルのコピーの形式指定

この例では、LATESTCOPY タグの付いた 2 つのデータファイルを /disk2 ディレクトリにコピーします。

```
BACKUP AS COPY
COPY OF DATAFILE 27, 28
FROM TAG 'LATESTCOPY'
FORMAT '/disk2/Datafile%f_Database%d';
```

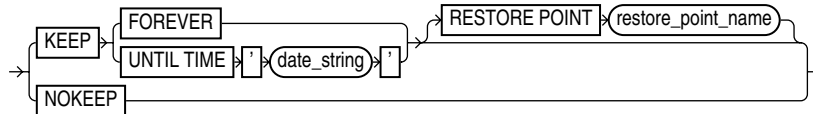
## keepOption

### 用途

`keepOption` 副次句を使用すると、バックアップまたはコピーの状態を保存方針に基づいて指定できます。

### 構文

**keepOption::=**



### 使用上の注意

Recovery Manager は、KEEP オプションが指定されたバックアップ・ピースをバックアップの保存方針の計算から除外します。使用可能であれば、Recovery Manager は、これらのバックアップを障害時リカバリのリストア操作で使用します。この目的は、別のシステムにリストアできるデータベースのスナップショットを作成してテストまたは履歴の使用状況を確認することです。

## セマンティクス

構文の要素	説明
KEEP	<p>バックアップを<b>アーカイブ・バックアップ</b>に指定します。これは、構成済の保存方針から除外される自己完結型のバックアップです。</p> <p>アーカイブ・バックアップには、バックアップをリストアして一貫性のある状態にリカバリするために必要なすべてのファイルが含まれているため、自己完結型となっています。バックアップ中にオープンされているデータベースがある場合、Recovery Manager は、データベースの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 REDO ログの生成およびバックアップを自動的に行います（2-53 ページの例 2-26 を参照）。</p> <p>Recovery Manager は、KEEP オプションが指定されたバックアップ・ピースを保存方針の計算から除外します。使用可能であれば、Recovery Manager は、これらのバックアップを障害時リカバリのリストア操作で使用します。この目的は、別のシステムにリストアできるデータベースのスナップショットを作成してテストまたは履歴の使用状況を確認することです。</p> <p><b>注意：</b>フラッシュ・リカバリ領域に格納されているファイルの保存方針は、KEEP を使用してオーバーライドすることはできません。リカバリ領域へのバックアップ中に KEEP を指定すると、Recovery Manager はエラーを発行します。</p> <p>KEEP が指定されると、Recovery Manager は複数のバックアップ・セットを作成します。Recovery Manager は、最初の <code>backupOperand</code> でオプションで指定された、データファイル、アーカイブ REDO ログ、制御ファイルおよびサーバー・パラメータ・ファイルをバックアップします。すべてのバックアップに <code>FORMAT</code>、<code>POOL</code> および <code>TAG</code> パラメータを使用します。このような理由から、<code>FORMAT</code> 文字列に複数のバックアップ・ピースを作成できることが必要です。この要件を満たす最も簡単な方法は、<code>%U</code> を指定することです。</p> <p><b>注意：</b>リカバリ・カタログは KEEP FOREVER の場合のみ必要になります。他の KEEP オプションの場合、カタログは不要です。</p>

構文の要素	説明
FOREVER	バックアップまたはコピーが不要とされないように指定します (2-53 ページの例 2-27 を参照)。最終的にはバックアップ・レコードが制御ファイルより古くなるため、FOREVER を指定する場合にリカバリ・カタログが必要になります。
UNTIL TIME 'date_string'	バックアップまたはコピーの保存期限を指定します。バックアップの保存方針の設定に関係なく、この期限を過ぎるとバックアップは不要とされます。  現行の NLS_DATE_FORMAT、または 'SYSDATE+365' などの SQL DATE 式を使用すると、特定の時刻を指定できます。01-JAN-07 などの KEEP TIME を指定すると、この日付の午前 0 時の 1 秒後にはバックアップは不要とされます。午後 9 時などの KEEP 時刻を指定すると、午後 9 時 1 分にバックアップは不要とされます。
RESTORE POINT restore_point_name	SCN に一致する通常のリストア・ポイントを作成します。これは、Recovery Manager がバックアップを一貫性のある状態までリカバリする必要がある SCN です (2-53 ページの例 2-26 を参照)。リストア・ポイント名はまだ存在していないものである必要があります。  SCN は、データファイルのバックアップが完了した直後に取得されます。リストア・ポイントとは、このアーカイブ・バックアップのリストアおよびリカバリが可能な SCN を示すラベルのことで、これによりデータベースをオープンできるようになります。一方 UNTIL TIME 句は、バックアップの保存期限の日付を指定します。  <b>注意:</b> RESTORE POINT パラメータは CHANGE コマンドでは無効です。
NOKEEP	いずれの KEEP 属性もバックアップに適用しないことを指定します。したがって、バックアップは、構成されたバックアップ保存方針が適用される通常のバックアップになります。これは、KEEP オプションを指定しない場合のデフォルト動作です。

## 例

### 例 3-23 アーカイブのための一貫性データベース・バックアップの作成

この例では、タグ Q107 の付いたデータベース・バックアップを作成し、バックアップが不要とみなされないように指定します (サンプル出力はその一部を示しています)。データファイルに一貫性を持たせるために必要なアーカイブ REDO ログがバックアップ・セットに含まれます。

```
RMAN> BACKUP TAG Q107 DATABASE KEEP FOREVER;
```

```
Starting backup at 24-JAN-07
```

```
current log archived
allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=105 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: Oracle Secure Backup
backup will never be obsolete
archived logs required to recover from this backup will be backed up
channel ORA_SBT_TAPE_1: starting full datafile backup set
channel ORA_SBT_TAPE_1: specifying datafile(s) in backup set
.
.
.
```

**例 3-24 バックアップのための KEEP 属性の削除**

この例では、すべてのアーカイブ REDO ログをバックアップします。KEEP 句では、2008 年 1 月 1 日の午前 0 時 1 秒に、バックアップが不要とみなされるように指定します。

```
RMAN> BACKUP KEEP UNTIL TIME '01-JAN-08' ARCHIVELOG ALL;
```

次のコマンドを実行すると、すべてのアーカイブ REDO ログのバックアップから KEEP 属性が削除されます (サンプル出力を示します)。

```
RMAN> CHANGE BACKUP OF ARCHIVELOG ALL NOKEEP;
```

```
using channel ORA_SBT_TAPE_1
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=77 device type=DISK
keep attributes for the backup are deleted
backup set key=330 RECID=19 STAMP=612722760
keep attributes for the backup are deleted
backup set key=397 RECID=22 STAMP=612722884
```

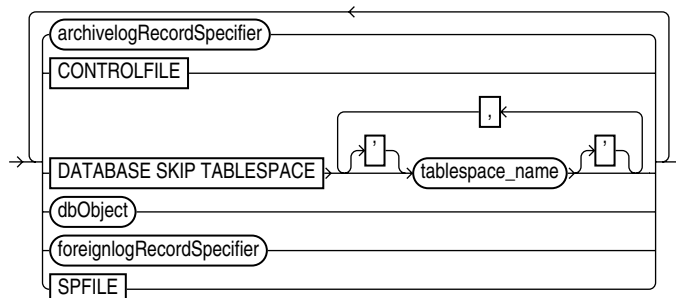
## listObjList

### 用途

*listObjList* 副次句を使用すると、データベース・ファイルおよびアーカイブ REDO ログを指定できます。

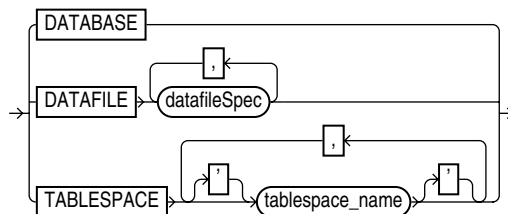
### 構文

**listObjList::=**



(*archivelogRecordSpecifier::=* (3-6 ページ))

**dbObject::=**



(*datafileSpec::=* (3-14 ページ))

## セマンティクス

### *listObjList*

構文の要素	説明
<i>archivelogRecordSpecifier</i>	アーカイブ REDO ログの範囲を指定します。 <b>関連項目</b> : 3-6 ページの「 <i>archivelogRecordSpecifier</i> 」を参照してください。
CONTROLFILE	現行の制御ファイルを指定します。
DATABASE SKIP TABLESPACE 'tablespace_name'	指定した表領域を DATABASE 指定から除外します。
<i>foreignlogRecordSpecifier</i>	指定した外部アーカイブ REDO ログを処理します。 <b>関連項目</b> : 3-6 ページの「 <i>foreignlogRecordSpecifier</i> 」を参照してください。
SPFILE	現行のサーバー・パラメータ・ファイルを指定します。

**dbObject**

構文の要素	説明
DATABASE	現行のデータベースの全ファイルのバックアップ・セットまたはイメージ・コピーを指定します。
DATAFILE <i>datafileSpec</i>	ファイル名またはファイル番号でデータファイルを指定します。この句では、データファイルの少なくとも1つが含まれているデータファイル・イメージ・コピーまたはバックアップ・セットを指定します。 <b>関連項目</b> : 3-14 ページの「 <i>datafileSpec</i> 」を参照してください。
TABLESPACE <i>tablespace_name</i>	表領域名を指定します。この句では、指定した表領域からのデータファイルの少なくとも1つが含まれているデータファイル・イメージ・コピーまたはバックアップ・セットを指定します。

**例****例 3-25 データファイルのコピーのリスト**

次のコマンドは、データベースにある全ファイルのイメージ・コピーを、ディクショナリ管理の一時表領域である temp 表領域をスキップしてリストします。

```
LIST COPY OF DATABASE
  SKIP TABLESPACE temp;
```

**例 3-26 アーカイブ REDO ログのクロスチェック**

次の例では、サーバー・パラメータ・ファイルと最近3か月間に作成したアーカイブ REDO ログのバックアップの状態について、メディア・マネージャに問い合わせます。この例では、サンプル出力も示しています。

```
RMAN> ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt;

allocated channel: ORA_MAINT_SBT_TAPE_1
channel ORA_MAINT_SBT_TAPE_1: SID=103 device type=SBT_TAPE
channel ORA_MAINT_SBT_TAPE_1: Oracle Secure Backup

RMAN> CROSSCHECK BACKUP OF SPFILE ARCHIVELOG FROM TIME 'SYSDATE-90';

crosschecked backup piece: found to be 'AVAILABLE'
backup piece handle=8cic4031_1_1 RECID=195 STAMP=616693857
crosschecked backup piece: found to be 'AVAILABLE'
backup piece handle=c-28014364-20070308-15 RECID=196 STAMP=616693875
Crosschecked 2 objects

RMAN> RELEASE CHANNEL;

released channel: ORA_MAINT_SBT_TAPE_1
```

**例 3-27 期限切れのバックアップの削除**

次のコマンドでは、すべてのバックアップのクロスチェックを実行します。1つのバックアップが期限切れであることが検出されます。この例では、期限切れのすべてのバックアップを削除します（サンプル出力を示します）。

```
RMAN> CROSSCHECK BACKUP;

allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=104 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: Oracle Secure Backup
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=103 device type=DISK
crosschecked backup piece: found to be 'EXPIRED'
backup piece handle=/disk2/PROD/autobackup/2007_03_08/o1_mf_s_616690991_2z15kl5h_.bkp
```



```
RECID=191 STAMP=616690994
crosschecked backup piece: found to be 'AVAILABLE'
backup piece handle=8cic4031_1_1 RECID=195 STAMP=616693857
crosschecked backup piece: found to be 'AVAILABLE'
backup piece handle=c-28014364-20070308-15 RECID=196 STAMP=616693875
Crosschecked 3 objects
```

```
RMAN> DELETE EXPIRED BACKUP;
```

```
using channel ORA_SBT_TAPE_1
using channel ORA_DISK_1
```

```
List of Backup Pieces
```

BP Key	BS Key	Pc#	Cp#	Status	Device Type	Piece Name
7678	7677	1	1	EXPIRED	DISK	/disk2/PROD/autobackup/2007_03_08/o1_mf_s_616690991_2z15k15h_.bkp

```
Do you really want to delete the above objects (enter YES or NO)? YES
```

```
deleted backup piece
```

```
backup piece handle=/disk2/PROD/autobackup/2007_03_08/o1_mf_s_616690991_2z15k15h_.bkp
```

```
RECID=191 STAMP=616690994
```

```
Deleted 1 EXPIRED objects
```

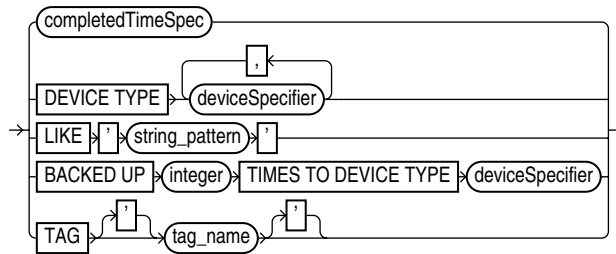
## maintQualifier

### 用途

*maintQualifier* 副次句を使用すると、データベース・ファイルおよびアーカイブ REDO ログを指定できます。

### 構文

***maintQualifier::=***



(*completedTimeSpec::=* (3-10 ページ)、*deviceSpecifier::=* (3-15 ページ))

### セマンティクス

***maintQualifier***

構文の要素	説明
<i>completedTimeSpec</i>	バックアップまたはコピー完了のための時刻範囲を指定します。 <b>関連項目</b> : 3-10 ページの「 <i>completedTimeSpec</i> 」を参照してください。
DEVICE TYPE <i>deviceSpecifier</i>	指定したデバイス・タイプ専用の自動チャンネルを割り当てます。このオプションが有効になるのは、構成済の自動チャンネルがあり、チャンネルを手動で割り当てていない場合のみです。たとえば、自動ディスクおよびテープ・チャンネルを構成して CHANGE...DEVICE TYPE DISK を発行すると、Recovery Manager ではディスク・チャンネルのみが割り当てられます。 <b>関連項目</b> : 3-15 ページの「 <i>deviceSpecifier</i> 」を参照してください。
LIKE ' <i>string_pattern</i> '	ファイル名パターンを指定して、データファイルのコピーを限定します。パターンには、Oracle のパターン一致文字であるパーセント記号 (%) およびアンダースコア ( _ ) を使用できます。Recovery Manager は、パターンと一致する名前のファイルのみを操作します。 <b>注意</b> : LIKE オプションは、LIST...ARCHIVELOG コマンドまたはバックアップ・ピースでは使用できません。
BACKED UP <i>integer</i> TIMES TO DEVICE TYPE <i>deviceSpecifier</i>	コマンドの対象を、指定したメディアに <i>integer</i> 回以上正常にバックアップされたアーカイブ・ログに制限します。このオプションはアーカイブ REDO ログにのみ適用されます。 BACKED UP オプションが DELETE ARCHIVELOG コマンドとともに使用されると、Recovery Manager は、構成済の設定ではなく BACKED UP オプションの設定を使用して、アーカイブ・ログが削除可能かどうかを判断します。つまり、CONFIGURE ARCHIVELOG DELETION POLICY はオーバーライドされます。この構成の他に、メディアとリポジトリ間の不一致をオーバーライドするには、DELETE ARCHIVELOG コマンドとともに FORCE を使用します。

構文の要素	説明
TAG tag_name	<p>データファイルのコピーとバックアップ・セットをタグで指定します。タグ名には大 / 小文字区別がなく、すべて大文字で表示されます。</p> <p><b>関連項目:</b> 多重化バックアップ・セットの個々のコピーにタグを適用する方法と、タグのデフォルト・ファイル名形式については、2-19 ページの「<a href="#">BACKUP</a>」を参照してください。</p>

## 例

### 例 3-28 特定の場所に格納されているバックアップのリスト

次のコマンドは、ディレクトリ /disk2 に格納されているすべてのイメージ・コピーをリストします。

```

RMAN> LIST COPY LIKE '/disk2/%';

List of Datafile Copies
=====

Key      File S Completion Time Ckp SCN    Ckp Time
-----
9855     1    A 08-MAR-07          1394701   08-MAR-07
Name: /disk2/data_D-PROD_I-28014364_TS-SYSTEM_FNO-1_8eic410j
Tag: TAG20070308T160643

9856     2    A 08-MAR-07          1394735   08-MAR-07
Name: /disk2/data_D-PROD_I-28014364_TS-SYSAUX_FNO-2_8fic412a
Tag: TAG20070308T160643

```

### 例 3-29 バックアップ済のアーカイブ・ログの削除

次のコマンドは、テープに 2 回以上正常にバックアップされたアーカイブ・ログのみを削除します。この例では、Seq 36 のアーカイブ・ログのみがこれらの条件を満たしています。

```

RMAN> DELETE ARCHIVELOG ALL BACKED UP 2 TIMES TO DEVICE TYPE sbt;

released channel: ORA_SBT_TAPE_1
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=104 device type=DISK
RMAN-08138: WARNING: archived log not deleted - must create more backups
archived log file name=/disk1/oradata/prod/arch/archiver_1_37_616443024.arc thread=1
sequence=37
List of Archived Log Copies for database with db_unique_name PROD
=====

Key      Thrd Seq      S Low Time
-----
9940     1    36      A 08-MAR-07
Name: /disk1/oradata/prod/arch/archiver_1_36_616443024.arc

Do you really want to delete the above objects (enter YES or NO)? Y
deleted archived log
archived log file name=/disk1/oradata/prod/arch/archiver_1_36_616443024.arc RECID=129
STAMP=616695115
Deleted 1 objects

```

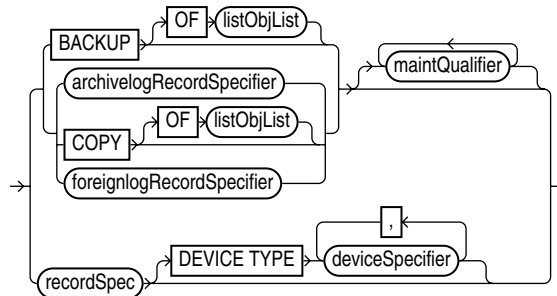
## maintSpec

### 用途

*maintSpec* 副次句を使用すると、**CHANGE**、**CROSSCHECK** および **DELETE** コマンドで処理されるバックアップ・ファイルを指定できます。

### 構文

**maintSpec**::=



(*listObjList*::= (3-27 ページ)、 *maintQualifier*::= (3-30 ページ)、  
*archiveLogRecordSpecifier*::= (3-6 ページ)、 *recordSpec*::= (3-35 ページ)、  
*deviceSpecifier*::= (3-15 ページ))

### セマンティクス

**maintSpec**

構文の要素	説明
BACKUP	バックアップ・セットおよびプロキシ・コピーを処理します。 <b>CHANGE</b> BACKUP で OF 句が使用されない場合、Recovery Manager はリポジトリに記録されているすべてのバックアップおよびプロキシ・コピーを変更します。 <b>CROSSCHECK</b> BACKUP で OF 句が使用されない場合、Recovery Manager はデータベース全体のバックアップのクロスチェックを行います。 <b>DELETE</b> BACKUP で OF 句が使用されない場合、Recovery Manager はデータベース全体のバックアップを削除します。
OF <i>listObjList</i>	操作するファイルのリストを <i>listObjList</i> 句で指定したオブジェクト型に限定します。 <b>関連項目</b> : 3-27 ページの「 <i>listObjList</i> 」を参照してください。
<i>archiveLogRecordSpecifier</i>	指定したアーカイブ REDO ログを処理します。 BACKED UP 句を使用せず <b>DELETE</b> ARCHIVELOG が指定された場合、Recovery Manager は、構成済の設定 ( <b>CONFIGURE</b> ARCHIVELOG DELETION POLICY TO BACKED UP) を使用して、アーカイブ・ログが削除可能かどうかを判断します。BACKED UP 句とともに <b>DELETE</b> ARCHIVELOG が指定された場合、Recovery Manager は <b>DELETE</b> 設定を使用して、アーカイブ・ログが削除可能かどうかを判断します。削除方針の構成およびメディアとリポジトリ間の不一致をオーバーライドするには、 <b>DELETE</b> ARCHIVELOG コマンドとともに <b>FORCE</b> を使用します。 <b>関連項目</b> : 3-6 ページの「 <i>archiveLogRecordSpecifier</i> 」を参照してください。

構文の要素	説明
COPY	<p>データファイルのコピー、制御ファイルのコピー、アーカイブ REDO ログを処理します。</p> <p><b>CHANGE COPY</b> にオプションを指定しない場合は、このコマンドによって、リポジトリに記録されたすべてのイメージ・コピーが処理されます。<b>CROSSCHECK COPY</b> を使用している場合、デフォルトでは、データベース内で状態が <b>AVAILABLE</b> または <b>EXPIRED</b> であるすべてのファイルのすべてのイメージ・コピーがチェックされます。<b>DELETE COPY</b> を使用している場合、デフォルトでは、データベース内のすべてのファイルのコピーが <b>COPY</b> によって削除されます。<b>EXPIRED</b> オプションを指定すると、リポジトリ内で <b>EXPIRED</b> とマークされたコピーのみが削除されます。</p>
OF <i>listObjList</i>	<p>操作するオブジェクトのリストを <i>listObjList</i> 句で指定したオブジェクト型に限定します。オブジェクトを指定しなければ、コマンドはデフォルトですべてのコピーになります。<b>CHANGE COPY OF DATABASE</b> では、制御ファイルではなくデータファイルが含まれるため注意してください。</p> <p><b>関連項目</b> : 3-27 ページの「<i>listObjList</i>」を参照してください。</p>
<i>foreignlogRecordSpecifier</i>	<p>指定した外部アーカイブ REDO ログを処理します。</p> <p><b>関連項目</b> : 3-6 ページの「<i>foreignlogRecordSpecifier</i>」を参照してください。</p>
<i>maintQualifier</i>	<p>指定したオプションに基づいてコマンドを制限します。</p> <p><b>関連項目</b> : 3-30 ページの「<i>maintQualifier</i>」を参照してください。</p>
<i>recordSpec</i>	<p>メンテナンス処理を実行するファイルを指定します。</p> <p><i>recordSpec</i> で <b>BACKUPSET</b> パラメータを使用すると、<b>CHANGE</b>、<b>CROSSCHECK</b> および <b>DELETE</b> コマンドで使用するバックアップ・セットがキーによって識別されます。詳細は、2-167 ページの「<b>LIST コマンドの出力</b>」にある、<b>LIST</b> の出力表の列ヘッダーの説明を参照してください。出力の <b>KEY</b> 列を使用して、<b>CHANGE</b> および <b>DELETE</b> コマンドで使用可能な主キーを取得します。</p> <p><b>関連項目</b> : 3-35 ページの「<i>recordSpec</i>」を参照してください。</p>
DEVICE TYPE <i>deviceSpecifier</i>	<p>指定したデバイス・タイプ専用の自動チャネルを割り当てます。このオプションが有効になるのは、構成済の自動チャネルがあり、チャネルを手動で割り当てていない場合のみです。たとえば、自動ディスクおよびテープ・チャネルを構成して <b>CROSSCHECK...DEVICE TYPE DISK</b> を実行すると、<b>Recovery Manager</b> ではディスク・チャネルのみが割り当てられます。</p> <p><b>関連項目</b> : 3-15 ページの「<i>deviceSpecifier</i>」を参照してください。</p>

## 例

### 例 3-30 バックアップのクロスチェック

次のコマンドは、アーカイブ REDO ログのバックアップをクロスチェックします。

```

RMAN> CROSSCHECK BACKUP OF ARCHIVELOG ALL;

allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=103 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: Oracle Secure Backup
using channel ORA_DISK_1
crosschecked backup piece: found to be 'AVAILABLE'
backup piece handle=8cic4031_1_1 RECID=195 STAMP=616693857
crosschecked backup piece: found to be 'AVAILABLE'
backup piece handle=8oic41ad_1_1 RECID=198 STAMP=616695118
crosschecked backup piece: found to be 'AVAILABLE'
backup piece handle=8qic41c3_1_1 RECID=200 STAMP=616695171
Crosschecked 3 objects

```

## obsOperandList

### 用途

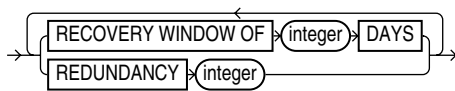
*obsOperandList* 副次句を使用して、バックアップを不要とマークするための条件を指定します。

### 使用上の注意

単一の REPORT OBSOLETE コマンドまたは DELETE OBSOLETE コマンドで、RECOVERY WINDOW と REDUNDANCY の両方を使用することはできません。

### 構文

***obsOperandList*::=**



### セマンティクス

***obsOperandList***

構文の要素	説明
RECOVERY WINDOW OF <i>integer</i> DAYS	<p><i>integer</i> で指定した日数以内の過去のいずれかの時点までデータベースをリカバリするのに必要ではないバックアップ・セットとイメージ・コピーを、Recovery Manager で不要としてレポートするように指定します。</p> <p><b>関連項目：</b>リカバリ期間の説明は、2-68 ページの「<a href="#">CONFIGURE</a>」を参照してください。</p>
REDUNDANCY <i>integer</i>	<p>バックアップ・セットまたはイメージ・コピーを不要とみなすために必要な最小レベルの冗長性を指定します。</p> <p>データファイルのコピーが不要と見なされるのは、このファイルの最近のバックアップ・セットまたはイメージ・コピーが <i>integer</i> で指定した数以上ある場合です。データファイルのバックアップ・セットは、バックアップ・セットに含まれている各ファイルに、<i>integer</i> で指定した数以上の最近のバックアップ・セットまたはイメージ・コピーが存在している場合に不要とみなされます。たとえば、REDUNDANCY 2 の場合、他のバックアップ・セットまたはイメージ・コピーを不要とみなすには、データファイルの新しいバックアップ・セットまたはイメージ・コピーが 2 つ以上必要です。</p>

### 例

#### 例 3-31 不要なバックアップの削除

このコマンドは、データベースを過去 30 日以内の SCN までリカバリするためには不要なバックアップとコピーをすべて削除します。

```
DELETE OBSOLETE RECOVERY WINDOW OF 30 DAYS;
```

## recordSpec

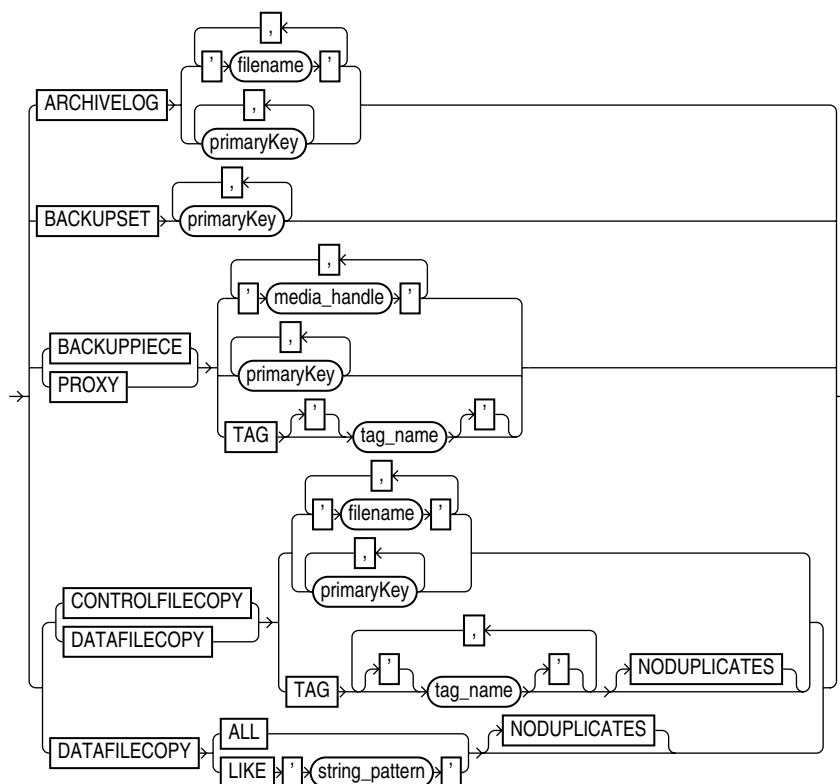
### 用途

`recordSpec` 副次句を使用すると、**CHANGE**、**CROSSCHECK**、**DELETE** および **LIST** コマンドの対象となるバックアップまたはコピーを指定できます。

ほとんどの `recordSpec` オプションでは、主キーを指定できます。LIST コマンドの出力を使用して、主キーを取得します。

### 構文

`recordSpec::=`



### セマンティクス

構文の要素	説明
ARCHIVELOG	主キーまたはファイル名のいずれかで、アーカイブ REDO ログを指定します。
BACKUPSET	主キーでバックアップ・セットを指定します。
BACKUPPIECE	メディア・ハンドル、主キーまたはタグ名でバックアップ・ピースを指定します。
PROXY	メディア・ハンドル、主キーまたはタグ名でプロキシ・コピーを指定します。
CONTROLFILECOPY	主キー、ファイル名のパターン ('filename') または TAG tag_name で制御ファイルのコピーを指定します。制御ファイルのコピーのクロスチェックを行う場合は、主キーではなくファイル名を指定する必要があります。

構文の要素	説明
DATAFILECOPY	主キー、ファイル名のパターン ('filename')、タグ (TAG tag_name) または一致する文字列 (LIKE 'string_pattern') でデータファイルのコピーを指定します。Recovery Manager リポジトリに記録されているすべてのデータファイルのコピーを示すには、ALL を指定します。
NODUPLICATES	複数のコピーが存在する場合でも、残りの句で指定された制御ファイルまたはデータファイルのコピーのみが操作の対象になるように指定します。

## 例

### 例 3-32 バックアップのクロスチェック

この例では、主キーで指定したバックアップ・セットをクロスチェックします。

```
RMAN> LIST BACKUP SUMMARY;
```

```
List of Backups
```

```
=====
```

Key	TY	LV	S	Device	Type	Completion Time	#Pieces	#Copies	Compressed	Tag
8504	B	A	A	SBT_TAPE		08-MAR-07	1	1	NO	TAG20070308T155057
8558	B	F	A	SBT_TAPE		08-MAR-07	1	1	NO	TAG20070308T155114
9872	B	F	A	DISK		08-MAR-07	1	1	NO	TAG20070308T160830
9954	B	A	A	SBT_TAPE		08-MAR-07	1	1	NO	TAG20070308T161157
9972	B	F	A	SBT_TAPE		08-MAR-07	1	1	NO	TAG20070308T161224
10021	B	A	A	SBT_TAPE		08-MAR-07	1	1	NO	TAG20070308T161251
10042	B	F	A	SBT_TAPE		08-MAR-07	1	1	NO	TAG20070308T161308
10185	B	F	A	DISK		08-MAR-07	1	1	NO	TAG20070308T170532
10210	B	F	A	DISK		08-MAR-07	1	1	NO	TAG20070308T170535

```
RMAN> CROSSCHECK BACKUPSET 9872, 10185, 10210;
```

```
allocated channel: ORA_SBT_TAPE_1
```

```
channel ORA_SBT_TAPE_1: SID=103 device type=SBT_TAPE
```

```
channel ORA_SBT_TAPE_1: Oracle Secure Backup
```

```
using channel ORA_DISK_1
```

```
crosschecked backup piece: found to be 'AVAILABLE'
```

```
backup piece handle=/disk2/PROD/autobackup/2007_03_08/o1_mf_s_616694910_2z19d0wg_.bkp
```

```
RECID=197 STAMP=616694912
```

```
crosschecked backup piece: found to be 'AVAILABLE'
```

```
backup piece
```

```
handle=/disk2/PROD/backupset/2007_03_08/o1_mf_nnsnf_TAG20070308T170532_2z1dpwz6_.bkp
```

```
RECID=202 STAMP=616698332
```

```
crosschecked backup piece: found to be 'AVAILABLE'
```

```
backup piece handle=/disk2/PROD/autobackup/2007_03_08/o1_mf_s_616698335_2z1dq0d0_.bkp
```

```
RECID=203 STAMP=616698336
```

```
Crosschecked 3 objects
```

### 例 3-33 データファイルのコピーの削除

この例では、指定したデータファイルのコピーを削除します。

```
RMAN> DELETE NOPROMPT DATAFILECOPY '/disk1/oradata/prod/users01.dbf';
```



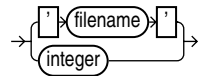
## tempfileSpec

### 用途

*tempfileSpec* 副次句を使用すると、ファイル名または絶対ファイル番号で一時ファイルを指定できます。

### 構文

**tempfileSpec ::=**



### セマンティクス

構文の要素	説明
'filename'	絶対パスまたは相対ファイル名のいずれかでデータファイルを指定します。相対ファイル名を指定した場合、ファイル名はターゲット・データベースによってプラットフォーム固有の方法で修飾されます。  絶対パス名または Oracle ホームに対する相対パス名を指定できます。二重引用符と一重引用符はどちらも有効です（構文図では、一重引用符のみを使用しています）。Oracle ホームを表すには疑問符 (?) を使用し、Oracle SID を表すにはアットマーク (@) を使用します。
integer	絶対ファイル番号でデータファイルを指定します。ファイル番号は、V\$TEMPFILE ビューまたは <b>REPORT SCHEMA</b> の出力から取得します。

### 例

#### 例 3-34 ファイル名による一時ファイルの指定

この例では、ファイル名を指定して、一時ファイル /disk1/oradata/prod/temp01.dbf の名前を /disk2/temp01.dbf に変更します。

```

SHUTDOWN IMMEDIATE
STARTUP MOUNT
RUN
{
  SWITCH TEMPFILE '/disk1/oradata/prod/temp01.dbf'
    TO '/disk2/temp01.dbf';
}
ALTER DATABASE OPEN;
  
```

この例を実行する場合は、データベースをマウントする必要があります。

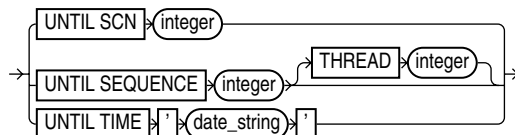
## untilClause

### 用途

*untilClause* 副次句を使用すると、各種の Recovery Manager 操作に対して時刻、SCN、リストア・ポイントまたはログ順序番号による上限を指定できます。

### 構文

***untilClause* ::=**



### セマンティクス

構文の要素	説明
UNTIL SCN <i>integer</i>	SCN を上限として指定します。指定した SCN は含まれません。  Recovery Manager は、指定した SCN の前までのリストアまたはリカバリに使用できるファイルのみを選択します (3-39 ページの例 3-35 を参照)。たとえば、 <b>RESTORE</b> DATABASE UNTIL SCN 1000 を実行すると、SCN 999 までのリカバリに使用できるバックアップのみが選択されます。
UNTIL SEQUENCE <i>integer</i>	REDO ログ順序番号とスレッドを上限として指定します。指定した値は含まれません。  Recovery Manager は、指定した順序番号の前までのリストアまたはリカバリに使用できるファイルのみを選択します。たとえば、 <b>REPORT OBSOLETE UNTIL SEQUENCE</b> 8000 を実行すると、ログ順序 7999 までのリカバリに使用できるバックアップのみが選択されます。
THREAD <i>integer</i>	REDO スレッドの番号を指定します。

構文の要素	説明
UNTIL TIME 'date_string'	<p>時刻を上限として指定します。指定した時刻は含まれません (3-39 ページの例 3-36 を参照)。</p> <p>Recovery Manager は、指定した時刻の直前までのリストアまたはリカバリに使用できるファイルのみを選択します。たとえば、<code>LIST BACKUP UNTIL TIME 'SYSDATE-7'</code> を実行すると、1 週間前の時点までのリカバリに使用できるバックアップがすべてリストされます。</p> <p>Recovery Manager コマンドで日付を指定する場合は、次のいずれかの日付文字列を使用する必要があります。</p> <ul style="list-style-type: none"> <li>■ NLS_DATE_FORMAT の設定と一致する形式のリテラル文字列。</li> <li>■ 'SYSDATE-10' または "TO_DATE('01/30/2007', 'MM/DD/YYYY')" などの DATE 型の SQL 式。2 番目の例には、専用の日付書式マスクが含まれているため、現行の NLS_DATE_FORMAT 設定とは関係がないことに注意してください。</li> </ul> <p>Recovery Manager における日付設定の代表例を次に示します。</p> <pre>BACKUP ARCHIVELOG FROM TIME 'SYSDATE-31' UNTIL TIME 'SYSDATE-14'; RESTORE DATABASE UNTIL TIME "TO_DATE('09/20/06', 'MM/DD/YY')";</pre> <p><b>注意:</b> 時間ベース・リカバリの粒度は、REDO ログのタイムスタンプによって異なります。たとえば、次のコマンドを指定するとします。</p> <pre>RECOVER DATABASE UNTIL TIME '2007-07-26 17:45:00';</pre> <p>17:45:00 のタイムスタンプを持つ REDO が書き込まれなかった場合は、それ以降の次の REDO タイムスタンプが検出されるまでリカバリは続行します。たとえば、次の REDO タイムスタンプが 17:45:04 であるとします。V\$LOG_HISTORY TABLE の FIRST_TIME および FIRST_CHANGE# 列を問い合わせることで、特定の SCN に最も近い時間を確認できます。</p>

## 例

### 例 3-35 指定した SCN までの不完全リカバリの実行

この例では、マウントされたデータベースを想定して、指定した SCN の前までデータベースをリカバリします。

```
STARTUP FORCE MOUNT
RUN
{
  SET UNTIL SCN 1418901; # set to 1418901 to recover database through SCN 1418900
  RESTORE DATABASE;
  RECOVER DATABASE;
}
ALTER DATABASE OPEN RESETLOGS;
```

### 例 3-36 不要なバックアップのレポート

この例は、過去 1 週間以内のいずれかの時点までリカバリできることを前提条件としています。1 週間前の時点までデータベースをリカバリするために使用できるバックアップは、すべて不要とみなされます。

```
REPORT OBSOLETE UNTIL TIME 'SYSDATE-7';
```



---

---

## リカバリ・カタログ・ビュー

この章では、リカバリ・カタログ・ビューを説明します。リカバリ・カタログが作成されている場合にかぎり、リカバリ・カタログ・ビューにアクセスできます（「[CREATE CATALOG](#)」を参照）。リカバリ・カタログ・ビューの概要は、4-2 ページの「[Recovery Manager のリカバリ・カタログ・ビューの概要](#)」を参照してください。

---

---

**注意：** これらのビューは正規化されていませんが、Recovery Manager および Enterprise Manager で使用するための最適化は行われています。したがって、ほとんどのカタログ・ビューには、基礎となるいくつかの表を結合したために生じた冗長な値があります。

通常、Enterprise Manager での使用を目的としたビューは直接問合せの機能が他のビューほど有効ではありません。

---

---

## Recovery Manager のリカバリ・カタログ・ビューの概要

次の表に Recovery Manager のリカバリ・カタログ・ビューの機能の概要を示します。

表 4-1 リカバリ・カタログ・ビュー

リカバリ・カタログ・ビュー	対応する VS ビュー	カタログ・ビューの内容
<a href="#">RC_ARCHIVED_LOG</a>	V\$ARCHIVED_LOG	アーカイブ REDO ログとアーカイブされていない REDO ログ。
<a href="#">RC_BACKUP_ARCHIVELOG_DETAILS</a>	V\$BACKUP_ARCHIVELOG_DETAILS	Enterprise Manager のアーカイブ REDO ログ・バックアップに関する詳細。
<a href="#">RC_BACKUP_ARCHIVELOG_SUMMARY</a>	V\$BACKUP_ARCHIVELOG_SUMMARY	Enterprise Manager のアーカイブ REDO ログ・バックアップに関する情報のサマリー。
<a href="#">RC_BACKUP_CONTROLFILE</a>	V\$BACKUP_CONTROLFILE	バックアップ・セットにバックアップされた制御ファイル。
<a href="#">RC_BACKUP_CONTROLFILE_DETAILS</a>	V\$BACKUP_CONTROLFILE_DETAILS	Enterprise Manager の制御ファイルのバックアップに関する詳細。
<a href="#">RC_BACKUP_CONTROLFILE_SUMMARY</a>	V\$BACKUP_CONTROLFILE_SUMMARY	Enterprise Manager の制御ファイルのバックアップに関する情報のサマリー。
<a href="#">RC_BACKUP_COPY_DETAILS</a>	V\$BACKUP_COPY_DETAILS	Enterprise Manager のデータファイルのイメージ・コピーのバックアップに関する詳細。
<a href="#">RC_BACKUP_COPY_SUMMARY</a>	V\$BACKUP_COPY_SUMMARY	Enterprise Manager のデータファイルのイメージ・コピーのバックアップに関する情報のサマリー。
<a href="#">RC_BACKUP_CORRUPTION</a>	V\$BACKUP_CORRUPTION	データファイル・バックアップ内の破損ブロック範囲。
<a href="#">RC_BACKUP_DATAFILE</a>	V\$BACKUP_DATAFILE	バックアップ・セット内のデータファイル。
<a href="#">RC_BACKUP_DATAFILE_DETAILS</a>	V\$BACKUP_DATAFILE_DETAILS	Enterprise Manager のデータファイルのバックアップに関する詳細。
<a href="#">RC_BACKUP_DATAFILE_SUMMARY</a>	V\$BACKUP_DATAFILE_SUMMARY	Enterprise Manager のデータファイルのバックアップに関する情報のサマリー。
<a href="#">RC_BACKUP_FILES</a>	V\$BACKUP_FILES	リポジトリで認識される Recovery Manager のバックアップおよびコピー。
<a href="#">RC_BACKUP_PIECE</a>	V\$BACKUP_PIECE	バックアップ・ピース。
<a href="#">RC_BACKUP_PIECE_DETAILS</a>	V\$BACKUP_PIECE_DETAILS	Enterprise Manager のバックアップ・ピースに関する詳細。
<a href="#">RC_BACKUP_REDOLOG</a>	V\$BACKUP_REDOLOG	バックアップ・セット内のアーカイブ REDO ログ。
<a href="#">RC_BACKUP_SET</a>	V\$BACKUP_SET	カタログに登録されたデータベースのすべてのインカネーションのバックアップ・セット。
<a href="#">RC_BACKUP_SET_DETAILS</a>	V\$BACKUP_SET_DETAILS	Enterprise Manager のバックアップ・セットに関する詳細。
<a href="#">RC_BACKUP_SET_SUMMARY</a>	V\$BACKUP_SET_SUMMARY	Enterprise Manager のバックアップ・セットに関する情報のサマリー。
<a href="#">RC_BACKUP_SPFILE</a>	V\$BACKUP_SPFILE	バックアップ内のサーバー・パラメータ・ファイル。

表 4-1 リカバリ・カタログ・ビュー (続き)

リカバリ・カタログ・ビュー	対応する V\$ ビュー	カタログ・ビューの内容
RC_BACKUP_SFFILE_DETAILS	V\$BACKUP_SFFILE_DETAILS	Enterprise Manager のサーバー・パラメータ・ファイルのバックアップに関する詳細。
RC_BACKUP_SFFILE_SUMMARY	V\$BACKUP_SFFILE_SUMMARY	Enterprise Manager のサーバー・パラメータ・ファイルのバックアップに関する情報のサマリー。
RC_CHECKPOINT	N/A	RC_RESYNC により、使用不可になりました。
RC_CONTROLFILE_COPY	V\$CONTROLFILE_COPY	ディスク上の制御ファイルのコピー。
RC_COPY_CORRUPTION	V\$COPY_CORRUPTION	データファイル・コピー内の破損ブロック範囲。
RC_DATABASE	V\$DATABASE	リカバリ・カタログに登録されているデータベース。
RC_DATABASE_BLOCK_CORRUPTION	V\$DATABASE_BLOCK_CORRUPTION	最新の Recovery Manager バックアップまたはコピーで破損としてマークされたデータベース・ブロック。
RC_DATABASE_INCARNATION	V\$DATABASE_INCARNATION	リカバリ・カタログに登録されているデータベース・インカーネーション。
RC_DATAFILE	V\$DATAFILE	リカバリ・カタログに登録されているデータファイル。
RC_DATAFILE_COPY	V\$DATAFILE_COPY	ディスク上のデータファイルのコピー。
RC_LOG_HISTORY	V\$LOG_HISTORY	ログ・スイッチの発生時刻を示すオンライン REDO ログの履歴。
RC_OFFLINE_RANGE	V\$OFFLINE_RANGE	データファイルのオフライン範囲。
RC_PROXY_ARCHIVEDLOG	V\$PROXY_ARCHIVEDLOG	プロキシ・コピー機能を使用して作成されたアーカイブ・ログのバックアップ。
RC_PROXY_ARCHIVELOG_DETAILS	V\$PROXY_ARCHIVELOG_DETAILS	Enterprise Manager のプロキシ・アーカイブ REDO ログに関する詳細。
RC_PROXY_ARCHIVELOG_SUMMARY	V\$PROXY_ARCHIVELOG_SUMMARY	Enterprise Manager のプロキシ・アーカイブ REDO ログに関する情報のサマリー。
RC_PROXY_CONTROLFILE	V\$PROXY_CONTROLFILE	プロキシ・コピー機能を使用して作成された制御ファイルのバックアップ。
RC_PROXY_COPY_DETAILS	V\$PROXY_COPY_DETAILS	Enterprise Manager のデータファイルのプロキシ・コピーに関する詳細。
RC_PROXY_COPY_SUMMARY	V\$PROXY_COPY_SUMMARY	Enterprise Manager のデータファイルのプロキシ・コピーに関する情報のサマリー。
RC_PROXY_DATAFILE	V\$PROXY_DATAFILE	プロキシ・コピー機能を使用して作成されたデータファイルのバックアップ。
RC_REDO_LOG	V\$LOG および V\$LOGFILE	前回のカタログ再同期化以降の、データベースのすべてのインカーネーションのオンライン REDO ログ。
RC_REDO_THREAD	V\$THREAD	前回のカタログ再同期化以降の、データベースのすべてのインカーネーションの全 REDO スレッド。
RC_RESTORE_POINT	V\$RESTORE_POINT	前回のカタログ再同期化以降の、データベースのすべてのインカーネーションの全リストア・ポイント。
RC_RESYNC	N/A	リカバリ・カタログの再同期化。

表 4-1 リカバリ・カタログ・ビュー (続き)

リカバリ・カタログ・ビュー	対応する V\$ ビュー	カタログ・ビューの内容
RC_RMAN_BACKUP_JOB_DETAILS	V\$RMAN_BACKUP_JOB_DETAILS	Enterprise Manager のバックアップ・ジョブに関する詳細。
RC_RMAN_BACKUP_SUBJOB_DETAILS	V\$RMAN_BACKUP_SUBJOB_DETAILS	Enterprise Manager のバックアップ・サブジョブに関する詳細。
RC_RMAN_BACKUP_TYPE	V\$BACKUP_TYPE	Enterprise Manager 内部で使用。
RC_RMAN_CONFIGURATION	V\$RMAN_CONFIGURATION	Recovery Manager 構成の設定。
RC_RMAN_OUTPUT	V\$RMAN_OUTPUT	Enterprise Manager で使用する Recovery Manager コマンドからの出力。
RC_RMAN_STATUS	V\$RMAN_STATUS	Recovery Manager 操作に関する履歴のステータス情報。
RC_SITE	N/A	Data Guard 環境でのデータベース。
RC_STORED_SCRIPT	N/A	リカバリ・カタログに格納されているスクリプトの名前。
RC_STORED_SCRIPT_LINE	N/A	リカバリ・カタログに格納されているスクリプトの内容。
RC_TABLESPACE	V\$TABLESPACE	リカバリ・カタログに登録されているすべての表領域、すべての削除済表領域および古いインカネーションに属している表領域。
RC_TEMPFILE	V\$TEMPFILE	リカバリ・カタログに登録されているすべての一時ファイル。
RC_UNUSABLE_BACKUPFILE_DETAILS	V\$UNUSABLE_BACKUPFILE_DETAILS	リカバリ・カタログに登録されている使用不可能なバックアップ・ファイル。



## RC\_ARCHIVED\_LOG

このビューには、アーカイブ REDO ログとアーカイブされていない REDO ログの履歴情報が含まれます。このビューは、ターゲット・データベースの制御ファイル内の V\$ARCHIVED\_LOG ビューに対応しています。

Oracle は、オンライン REDO ログが正常にアーカイブされた後にアーカイブ REDO ログ・レコードを挿入します。アーカイブされていないログが消去された場合、レコードは NULL に設定された NAME 列に挿入されます。

ログが複数回アーカイブされた場合、このビューには、THREAD#、SEQUENCE# および RESETLOGS\_CHANGE# が同じで名前が異なる複数のアーカイブ・ログ・レコードが含まれます。

アーカイブ・ログ・レコードは、アーカイブ・ログがバックアップ・セットまたはコピーからリストアされたときにも挿入されます。

レコードが制御ファイルより古い場合、アーカイブ・ログ・レコードはなくなるため注意してください。

列	データ型	説明
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	このレコードが属しているターゲット・データベースのインカーネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカーネーションの DB_NAME。
AL_KEY	NUMBER	リカバリ・カタログ内のアーカイブ REDO ログの主キー。Recovery Manager がリカバリ・カタログに接続している間に LIST コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	NUMBER	V\$ARCHIVED_LOG からのアーカイブ REDO ログの RECID。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
STAMP	NUMBER	V\$ARCHIVED_LOG からのアーカイブ REDO ログの STAMP。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
NAME	VARCHAR2 (1024)	アーカイブ REDO ログのファイル名。
THREAD#	NUMBER	REDO スレッドの番号。
SEQUENCE#	NUMBER	ログ順序番号。
RESETLOGS_CHANGE#	NUMBER	レコード作成時における最新の RESETLOGS の SCN。
RESETLOGS_TIME	DATE	レコード作成時における最新の RESETLOGS のタイムスタンプ。
FIRST_CHANGE#	NUMBER	この REDO ログの最初の SCN。
FIRST_TIME	DATE	REDO ログに切り替えた時刻。
NEXT_CHANGE#	NUMBER	スレッド内にある次の REDO ログの最初の SCN。
NEXT_TIME	DATE	スレッド内にある次の REDO ログの最初のタイムスタンプ。
BLOCKS	NUMBER	このアーカイブ・ログのサイズ (オペレーティング・システム・ブロック数)。
BLOCK_SIZE	NUMBER	ブロック・サイズ (バイト)。
COMPLETION_TIME	DATE	REDO ログをアーカイブまたはコピーした時刻。

列	データ型	説明
ARCHIVED	VARCHAR2 (3)	ログがアーカイブされたかどうかを、YES (REDO ログがアーカイブされた場合) または NO (オンライン REDO ログのファイル・ヘッダーが検査され、レコードが V\$ARCHIVED_LOG に追加された場合) で示します。オンライン・ログを検査すると、そのアーカイブ・ログ・レコードが作成され、Recovery Manager のリカバリ中に適用できます。Oracle は、ARCHIVED を NO に設定し、オンライン・ログがバックアップされないようにします。
STATUS	VARCHAR2 (1)	アーカイブ REDO ログの状態。A (使用可能)、U (使用不可能)、D (削除済) または X (期限切れ)。
IS_STANDBY	VARCHAR2 (3)	このログをアーカイブしたデータベース。スタンバイ・データベースに属している場合は Y、プライマリ・データベースに属している場合は N。
DICTIONARY_BEGIN	VARCHAR2 (3)	このアーカイブ・ログに LogMiner ディクショナリの開始が含まれているかどうかを YES または NO で示します。  DICTIONARY_BEGIN と DICTIONARY_END の両方が YES の場合、このログには LogMiner ディクショナリ全体が含まれています。DICTIONARY_BEGIN が YES でも、DICTIONARY_END が NO であれば、このログにはディクショナリの開始が含まれていますが、このスレッドの後続の各ログへと継続し、DICTIONARY_END が YES になっているログで終了します。
DICTIONARY_END	VARCHAR2 (3)	このアーカイブ・ログに LogMiner ディクショナリの終了が含まれているかどうかを YES または NO で示します。この値の解釈は、DICTIONARY_BEGIN の説明を参照してください。
IS_RECOVERY_DEST_FILE	VARCHAR2 (3)	このコピーはフラッシュ・リカバリ領域に格納されます。YES または NO。
COMPRESSED	VARCHAR2 (3)	内部使用のみ。
CREATOR	VARCHAR2 (7)	次のアーカイブ REDO ログの作成者。 <ul style="list-style-type: none"> <li>■ ARCH: アーカイバ・プロセス</li> <li>■ FGRD: フォアグラウンド・プロセス</li> <li>■ RMAN: Recovery Manager</li> <li>■ SRMN: スタンバイ時の Recovery Manager</li> <li>■ LGWR: ログ・ライター・プロセス</li> </ul>
TERMINAL	VARCHAR2 (3)	このログが、スタンバイ・データベースのターミナル・リカバリ中に作成されたログかどうかを示します。値は YES または NO です。
SITE_KEY	NUMBER	このファイルと関連付けられている Data Guard データベースの主キー。Data Guard 環境にある各データベースの SITE_KEY の値は一意です。SITE_KEY を使用して RC_SITE ビューと結合すると、データベースの DB_UNIQUE_NAME を取得できます。

## RC\_BACKUP\_ARCHIVELOG\_DETAILS

RC\_BACKUP\_ARCHIVELOG\_DETAILS ビューは、アーカイブ REDO ログ・ファイルのバックアップの詳細情報を表示します。

このビューは、主に Enterprise Manager で内部的に使用します。

列	データ型	説明
BTYPE	CHAR (9)	バックアップ・タイプ・コンテナ (BACKUPSET、IMAGECOPY または PROXYCOPY)。
BTYPE_KEY	NUMBER	バックアップ・タイプに対する一意の識別子。バックアップ・セットの場合は VS_KEY、イメージ・コピーの場合は COPY_KEY、プロキシ・コピーの場合は XAL_KEY です。
SESSION_KEY	NUMBER	このセッションに対する一意の識別子。 RC_RMAN_BACKUP_JOB_DETAILS との結合に使用します。
SESSION_RECID	NUMBER	この Recovery Manager セッションに対応するターゲット・データベースの制御ファイルからの RECID。
SESSION_STAMP	NUMBER	この Recovery Manager セッションに対応するターゲット・データベースの制御ファイルからの STAMP。
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカネーションの DB_NAME。
ID1	NUMBER	バックアップ・セット内のアーカイブ・ログでは、この列に SET_STAMP が含まれます。プロキシ・コピーまたはイメージ・コピーのバックアップでは、この列には制御ファイルからの RECID が含まれます。
ID2	NUMBER	バックアップ・セット内のアーカイブ・ログでは、この列に SET_COUNT が含まれます。イメージ・コピーまたはプロキシ・コピーのバックアップでは、この列には STAMP が含まれます。
THREAD#	NUMBER	このアーカイブ REDO ログ・ファイルのスレッド番号。
SEQUENCE#	NUMBER	このアーカイブ REDO ログ・ファイルの順序番号。
RESETLOGS_CHANGE#	NUMBER	このアーカイブ・ログの OPEN RESETLOGS ブランチの SCN。
RESETLOGS_TIME	DATE	このアーカイブ・ログの OPEN RESETLOGS ブランチの時刻。
FIRST_CHANGE#	NUMBER	このアーカイブ・ログ・ファイルの開始 SCN。
FIRST_TIME	DATE	このアーカイブ REDO ログ・ファイルの開始 SCN に対応する時刻。
NEXT_CHANGE#	NUMBER	このアーカイブ REDO ログ・ファイルの終了 SCN。
NEXT_TIME	DATE	このアーカイブ REDO ログ・ファイルの終了 SCN に対応する時刻。
FILESIZE	NUMBER	バックアップされた REDO ログ・ファイルのサイズ (バイト)。
COMPRESSION_RATIO	NUMBER	このバックアップの圧縮率。
FILESIZE_DISPLAY	VARCHAR2 (4000)	FILESIZE と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。

## RC\_BACKUP\_ARCHIVELOG\_SUMMARY

RC\_BACKUP\_ARCHIVELOG\_SUMMARY には、単一または複数の Recovery Manager ジョブについて、アーカイブ REDO ログ・ファイルのバックアップに関するサマリーが含まれます。

このビューは、主に Enterprise Manager で内部的に使用します。

列	データ型	説明
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカネーションの DB_NAME。
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
NUM_FILES_BACKED	NUMBER	バックアップされたアーカイブ REDO ログ・ファイルの数。1つのアーカイブ・ログが複数のバックアップ・ジョブに含まれている場合、Recovery Manager では各バックアップを個別に数えます。 たとえば、このビューに2つの Recovery Manager バックアップ・ジョブのサマリーが含まれている場合、それぞれのバックアップがアーカイブ・ログ 1000 のみであれば、この列の値は2になり、NUM_DISTINCT_FILES_BACKED の値は1になります。
NUM_DISTINCT_FILES_BACKED	NUMBER	バックアップされた個々のアーカイブ REDO ログ・ファイルの数。この場合、REDO ログは一意のログ順序番号、スレッド番号、RESETLOGS ブランチで識別されます。 たとえば、このビューに2つの Recovery Manager バックアップ・ジョブのサマリーが含まれている場合、それぞれのバックアップがアーカイブ・ログ 1000 のみであれば、この列の値は1になり、NUM_FILES_BACKED の値は2になります。
MIN_FIRST_CHANGE#	NUMBER	バックアップされたアーカイブ REDO ログ・ファイルの範囲内の最小 SCN。
MAX_NEXT_CHANGE#	NUMBER	バックアップされたアーカイブ REDO ログ・ファイルの範囲内の最大 SCN。
MIN_FIRST_TIME	DATE	バックアップされたアーカイブ REDO ログ・ファイルが適用される最も古い時点。
MAX_NEXT_TIME	DATE	バックアップされたアーカイブ REDO ログ・ファイルが適用される最新の時点。
INPUT_BYTES	NUMBER	バックアップされたすべてのアーカイブ REDO ログ・ファイルの合計サイズ (バイト)。
OUTPUT_BYTES	NUMBER	このジョブで生成されたすべての出力ピースの合計サイズ。
COMPRESSION_RATIO	NUMBER	このバックアップの圧縮率。
INPUT_BYTES_DISPLAY	VARCHAR2 (4000)	INPUT_BYTES と値は同じですが、ユーザーが表示可能な書式 (nM、nG、nT、nP など) に変換されます。
OUTPUT_BYTES_DISPLAY	VARCHAR2 (4000)	OUTPUT_BYTES と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。

## RC\_BACKUP\_CONTROLFILE

このビューは、バックアップ・セット内の制御ファイルに関する情報をリストします。V\$BACKUP\_DATAFILE ビューには、データファイルと制御ファイルのレコードが含まれていることに注意してください。ファイル番号が 0 のバックアップ・データファイル・レコードは、バックアップ制御ファイルを表します。リカバリ・カタログでは、RC\_BACKUP\_CONTROLFILE ビューには制御ファイル・レコードのみが含まれ、RC\_BACKUP\_DATAFILE ビューにはデータファイル・レコードのみが含まれます。

列	データ型	説明
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	ターゲット・データベースのインカーネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカーネーションの DB_NAME。
BCF_KEY	NUMBER	リカバリ・カタログ内にある制御ファイル・バックアップの主キー。Recovery Manager がリカバリ・カタログに接続している間に LIST コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	NUMBER	V\$BACKUP_DATAFILE からの RECID の値。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
STAMP	NUMBER	V\$BACKUP_DATAFILE からの STAMP の値。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
BS_KEY	NUMBER	リカバリ・カタログ内でこのレコードが属しているバックアップ・セットの主キー。この列を使用して、RC_BACKUP_SET または RC_BACKUP_PIECE との結合を構成します。
SET_STAMP	NUMBER	V\$BACKUP_SET からの SET_STAMP の値。SET_STAMP と SET_COUNT は、ターゲット・データベース制御ファイル内でこのレコードが属しているバックアップ・セットを一意に識別する連結主キーを構成します。
SET_COUNT	NUMBER	V\$BACKUP_SET からの SET_COUNT の値。SET_STAMP と SET_COUNT は、ターゲット・データベース制御ファイル内でこのレコードが属しているバックアップ・セットを一意に識別する連結主キーを構成します。
RESETLOGS_CHANGE#	NUMBER	レコード作成時における最新の RESETLOGS の SCN。
RESETLOGS_TIME	DATE	レコード作成時における最新の RESETLOGS のタイムスタンプ。
CHECKPOINT_CHANGE#	NUMBER	制御ファイルのチェックポイント SCN。
CHECKPOINT_TIME	DATE	制御ファイルのチェックポイント時刻。
CREATION_TIME	DATE	制御ファイルの作成時刻。
BLOCK_SIZE	NUMBER	ブロック・サイズ (バイト)。
OLDEST_OFFLINE_RANGE	NUMBER	内部使用のみ。
STATUS	VARCHAR2 (1)	バックアップ・セットの状態。A (使用可能)、U (使用不可能) または D (削除済)。
BS_RECID	NUMBER	このバックアップ制御ファイルを含むバックアップ・セットの制御ファイル RECID。
BS_STAMP	NUMBER	この制御ファイルを含むバックアップ・セットの制御ファイル・スタンプ。
BS_LEVEL	NUMBER	このバックアップ制御ファイルを含むバックアップ・セットの増分レベル (NULL、0、1)。増分バックアップ・セットには制御ファイルを含めることができますが、常に制御ファイルの完全コピーが含まれます。増分制御ファイル・バックアップの場合、このようなことはありません。
COMPLETION_TIME	DATE	制御ファイルのバックアップが完了した日付。

---

列	データ型	説明
CONTROLFILE_TYPE	VARCHAR2(1)	制御ファイルのバックアップのタイプ。B (通常のバックアップ) または S (スタンバイ・バックアップ)。
BLOCKS	NUMBER	ファイル内のブロック数。
AUTOBACKUP_DATE	DATE	制御ファイルの自動バックアップの日付。
AUTOBACKUP_SEQUENCE	NUMBER	制御ファイルの自動バックアップの順序。1 ~ 255。

---

## RC\_BACKUP\_CONTROLFILE\_DETAILS

RC\_BACKUP\_CONTROLFILE\_DETAILS ビューは、制御ファイルのイメージ・コピー、バックアップ・セットおよびプロキシ・コピーのバックアップを含む、リストア可能な制御ファイルのバックアップに関する詳細情報を表示します。

このビューは、主に Enterprise Manager で内部的に使用します。

列	データ型	説明
BTYPE	CHAR (9)	この制御ファイルのバックアップのタイプ。可能な値は BACKUPSET、IMAGECOPY または PROXYCOPY です。
BTYPE_KEY	NUMBER	バックアップ・タイプの一意の識別子。BTYPE が BACKUPSET の場合、この値はバックアップ・セットの BS_KEY の値になります。BTYPE が IMAGECOPY の場合、この値はコピーの COPY_KEY の値になります。BTYPE が PROXYCOPY の場合、この値はプロキシ・コピーの XCF_KEY の値になります。
SESSION_KEY	NUMBER	セッションの識別子。RC_RMAN_OUTPUT および RC_RMAN_BACKUP_JOB_DETAILS との結合に使用します。
SESSION_RECID	NUMBER	この Recovery Manager セッションに対応するターゲット・データベースの制御ファイルからの RECID。
SESSION_STAMP	NUMBER	この Recovery Manager セッションに対応するターゲット・データベースの制御ファイルからの STAMP。
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカネーションの DB_NAME。
ID1	NUMBER	バックアップ・セットとして取られるバックアップでは、この列には SET_STAMP が含まれます。プロキシ・コピーまたはイメージ・コピーのバックアップでは、この列には制御ファイルからの RECID が含まれます。
ID2	NUMBER	バックアップ・セットとして取られるバックアップでは、この列には SET_COUNT が含まれます。プロキシ・コピーまたはイメージ・コピーのバックアップでは、この列には STAMP の値が含まれます。
CREATION_TIME	DATE	バックアップされた制御ファイルの作成時刻。
RESETLOGS_CHANGE#	NUMBER	この制御ファイルがバックアップされた RESETLOGS ブランチの SCN。
RESETLOGS_TIME	DATE	この制御ファイルがバックアップされた RESETLOGS ブランチの時刻。
CHECKPOINT_CHANGE#	NUMBER	バックアップされた制御ファイルの最新チェックポイント変更時の SCN。
CHECKPOINT_TIME	DATE	バックアップされた制御ファイルの最新チェックポイント時刻。
FILESIZE	NUMBER	この制御ファイルのバックアップ出力用ファイル・サイズ (バイト)。
COMPRESSION_RATIO	NUMBER	このバックアップの圧縮率。
FILESIZE_DISPLAY	VARCHAR2 (4000)	FILESIZE 列と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。

## RC\_BACKUP\_CONTROLFILE\_SUMMARY

RC\_BACKUP\_CONTROLFILE\_SUMMARY ビューは、制御ファイルのイメージ・コピー、バックアップ・セットおよびプロキシ・コピーのバックアップを含む、リストア可能な制御ファイルのバックアップに関するサマリーを表示します。

このビューは、主に Enterprise Manager で内部的に使用します。

列	データ型	説明
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカンセーションの DB_NAME。
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
NUM_FILES_BACKED	NUMBER	制御ファイルのバックアップの合計数。
NUM_DISTINCT_FILES_BACKED	NUMBER	個々の制御ファイルのバックアップ数。
MIN_CHECKPOINT_CHANGE#	NUMBER	バックアップされたすべての制御ファイルで最小のチェックポイント SCN。
MAX_CHECKPOINT_CHANGE#	NUMBER	バックアップされたすべての制御ファイルで最大のチェックポイント SCN。
MIN_CHECKPOINT_TIME	DATE	サマリー内のすべての制御ファイルの中で最も古いチェックポイント時刻。
MAX_CHECKPOINT_TIME	DATE	サマリーにあるすべての制御ファイルの中で最新のチェックポイント時刻。
INPUT_BYTES	NUMBER	入力ファイルの合計サイズ (バイト)。
OUTPUT_BYTES	NUMBER	このジョブで生成されたすべての出力ピースの合計サイズ。
COMPRESSION_RATIO	NUMBER	このバックアップの圧縮率。
INPUT_BYTES_DISPLAY	VARCHAR2 (4000)	INPUT_BYTES と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。
OUTPUT_BYTES_DISPLAY	VARCHAR2 (4000)	OUTPUT_BYTES と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。



## RC\_BACKUP\_COPY\_DETAILS

RC\_BACKUP\_COPY\_DETAILS ビューには、AVAILABLE 制御ファイルおよびデータファイルのコピーに関する詳細な情報が含まれます。列 SESSION\_KEY、SESSION\_RECID、SESSION\_STAMP および COPY\_KEY は、Recovery Manager のセッションとデータファイルのコピーを一意に識別します。このビューの他の列の意味は、RC\_DATAFILE\_COPY と同じです。

このビューは、主に Enterprise Manager で内部的に使用します。

列	データ型	説明
SESSION_KEY	NUMBER	セッションの識別子。RC_RMAN_OUTPUT および RC_RMAN_BACKUP_JOB_DETAILS との結合に使用します。
SESSION_RECID	NUMBER	SESSION_STAMP とともに使用して、このバックアップ・ジョブの出力を RC_RMAN_OUTPUT から一意に識別します。
SESSION_STAMP	NUMBER	SESSION_RECID とともに使用して、このバックアップ・ジョブの出力を RC_RMAN_OUTPUT から一意に識別します。
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカンネーションの DB_NAME。
RSR_KEY	NUMBER	このコピーを作成したジョブに対応する RC_RMAN_STATUS の行に対する一意キー。
COPY_KEY	NUMBER	このデータファイルまたは制御ファイルのコピーの一意の識別子。
FILE#	NUMBER	データファイルまたはデータファイルのコピーの絶対ファイル番号。
NAME	VARCHAR2 (1024)	データファイルまたは制御ファイルのコピーのファイル名。
TAG	VARCHAR2 (32)	このデータファイルまたは制御ファイルのコピーのタグ (存在する場合)。
CREATION_CHANGE#	NUMBER	データファイルに対するデータファイルの作成時の SCN。
CREATION_TIME	DATE	ファイルの作成時刻。
CHECKPOINT_CHANGE#	NUMBER	最新データファイル・チェックポイント SCN。
CHECKPOINT_TIME	DATE	最新のデータファイル・チェックポイント時刻。
MARKED_CORRUPT	NUMBER	RC_DATABASE_BLOCK_CORRUPTION ビューに基づいた、破損としてマークされたデータファイルのブロック数。
OUTPUT_BYTES	NUMBER	このジョブで生成されたすべての出力ピースの合計サイズ。
COMPLETION_TIME	DATE	このファイルのコピー完了時刻。
CONTROLFILE_TYPE	VARCHAR2 (1)	制御ファイルのコピーに対する、制御ファイルのバックアップのタイプ。B (通常のコピー) または S (スタンバイ・コピー)。これ以外の場合は NULL です。
KEEP	VARCHAR2 (3)	このコピーの保存方針が CONFIGURE RETENTION POLICY の値と異なる場合は YES。それ以外の場合は NO。
KEEP_UNTIL	DATE	BACKUP コマンドの KEEP UNTIL TIME 句を指定した場合は、この列にこのファイル・コピーが不要になる直前の日付が表示されます。列が NULL で、KEEP_OPTIONS が NULL でなければ、コピーが不要とされることはありません。

列	データ型	説明
KEEP_OPTIONS	VARCHAR2(11)	このデータファイルのコピーに指定された KEEP オプション。可能な値は、NOLOGS、BACKUP_LOGS、LOGS および NULL です。NOLOGS は、データベースがマウントされたときに一貫性のあるバックアップが作成されたことを示します。BACKUP_LOGS は、データベースがオープンされているときにバックアップが作成されたことを示します。したがって、一貫性のある状態にするには、アーカイブ・ログ・バックアップを適用する必要があります。LOGS は、LOGS キーワードを使用して長期間バックアップが作成されたことを示します。現在、LOGS キーワードは使用不可です。NULL は、このバックアップには KEEP オプションが指定されておらず、バックアップの保存方針に基づいて不要とされることを示します。
IS_RECOVERY_DEST_FILE	VARCHAR2(3)	このコピーがフラッシュ・リカバリ領域に格納される場合は YES。それ以外の場合は NO。
OUTPUT_BYTES_DISPLAY	VARCHAR2(4000)	OUTPUT_BYTES と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。

## RC\_BACKUP\_COPY\_SUMMARY

RC\_BACKUP\_COPY\_SUMMARY には、各データベースのすべての AVAILABLE 制御ファイルおよびデータファイルのコピーに関するサマリー情報が含まれます。

このビューは、主に Enterprise Manager で内部的に使用します。

列	データ型	説明
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカーネーションの DB_NAME。
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
NUM_COPIES	NUMBER	イメージ・コピーのバックアップの合計数。
NUM_DISTINCT_COPIES	NUMBER	固有のイメージ・コピーのバックアップ数。
MIN_CHECKPOINT_CHANGE#	NUMBER	このビューに示すすべてのイメージ・コピーのバックアップでの最小チェックポイント SCN。
MAX_CHECKPOINT_CHANGE#	NUMBER	このビューに示すすべてのイメージ・コピーのバックアップでの最大チェックポイント SCN。
MIN_CHECKPOINT_TIME	DATE	このビューに示すすべてのコピーでの最も古いチェックポイント時刻。
MAX_CHECKPOINT_TIME	DATE	このビューに示すすべてのコピーでの最新のチェックポイント時刻。
OUTPUT_BYTES	NUMBER	データファイルと制御ファイルのすべてのコピーの合計サイズ。
OUTPUT_BYTES_DISPLAY	VARCHAR2 (4000)	OUTPUT_BYTES と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。

## RC\_BACKUP\_CORRUPTION

このビューは、データファイル・バックアップ内の破損ブロック範囲をリストします。このビューは、制御ファイルの V\$BACKUP\_CORRUPTION ビューに対応しています。制御ファイルとアーカイブ REDO ログのバックアップには破損が許容されないことに注意してください。

列	データ型	説明
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	ターゲット・データベースのインカーネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカーネーションの DB_NAME。
RECID	NUMBER	V\$BACKUP_CORRUPTION からのレコード識別子。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
STAMP	NUMBER	V\$BACKUP_CORRUPTION から伝播した STAMP。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
BS_KEY	NUMBER	リカバリ・カタログ内でこのレコードが属しているバックアップ・セットの主キー。この列を使用して、RC_BACKUP_SET または RC_BACKUP_PIECE との結合を構成します。
SET_STAMP	NUMBER	V\$BACKUP_SET からの SET_STAMP の値。SET_STAMP と SET_COUNT は、ターゲット・データベース制御ファイル内でこのレコードが属しているバックアップ・セットを一意に識別する連結主キーを構成します。
SET_COUNT	NUMBER	V\$BACKUP_SET からの SET_COUNT の値。SET_STAMP と SET_COUNT は、ターゲット・データベース制御ファイル内でこのレコードが属しているバックアップ・セットを一意に識別する連結主キーを構成します。
PIECE#	NUMBER	この破損ブロックを含むバックアップ・ピース。
BDF_KEY	NUMBER	リカバリ・カタログ内にあるデータファイルのバックアップまたはコピーに対する主キー。この主キーを使用して RC_BACKUP_DATAFILE と結合します。Recovery Manager がリカバリ・カタログに接続している間に LIST コマンドを発行すると、この値が出力の KEY 列に表示されます。
BDF_RECID	NUMBER	V\$BACKUP_DATAFILE からの RECID の値。
BDF_STAMP	NUMBER	V\$BACKUP_DATAFILE からの STAMP の値。
FILE#	NUMBER	破損ブロックを含むデータファイルの絶対ファイル番号。
CREATION_CHANGE#	NUMBER	破損ブロックを含むデータファイルの作成時の SCN。
BLOCK#	NUMBER	この破損ブロック範囲内の最初の破損ブロックのブロック番号。
BLOCKS	NUMBER	BLOCK# 以降に見えられた破損ブロックの数。
CORRUPTION_CHANGE#	NUMBER	メディア破損ブロックの場合、この値は 0 です。論理破損ブロックの場合、この値はこの破損範囲内のブロックの最小 SCN です。
MARKED_CORRUPT	VARCHAR2 (3)	破損ブロックが以前に Oracle で検出されていない場合は YES、この破損ブロックがすでに検出済で、データベース内で破損としてマークされている場合は NO。バックアップ中に破損ブロックが検出されても、まだ Oracle により破損としてマークされていないければ、バックアップ・プロセスでは本番データファイル内でもそのブロックは破損としてマークされないため注意してください。したがって、このフィールドには、複数のバックアップ・セット内で同じブロックについて YES となることがあります。
CORRUPTION_TYPE	VARCHAR2 (9)	RC_DATABASE_BLOCK_CORRUPTION.CORRUPTION_TYPE と同じです。

## RC\_BACKUP\_DATAFILE

このビューは、バックアップ・セットのデータファイルに関する情報をリストします。このビューは、V\$BACKUP\_DATAFILE ビューに対応しています。バックアップ・データファイルは BDF\_KEY によって一意に識別されます。

列	データ型	説明
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	ターゲット・データベースのインカーネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカーネーションの DB_NAME。
BDF_KEY	NUMBER	リカバリ・カタログ内のデータファイル・バックアップの主キー。 Recovery Manager がリカバリ・カタログに接続している間に LIST コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	NUMBER	V\$BACKUP_DATAFILE からのバックアップ・データファイルの RECID。 RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
STAMP	NUMBER	V\$BACKUP_DATAFILE からのバックアップ・データファイルの STAMP。 RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
BS_KEY	NUMBER	リカバリ・カタログ内でこのレコードが属しているバックアップ・セットの主キー。この列を使用して、RC_BACKUP_SET または RC_BACKUP_PIECE との結合を構成します。
SET_STAMP	NUMBER	V\$BACKUP_SET からの SET_STAMP の値。SET_STAMP と SET_COUNT は、ターゲット・データベース制御ファイル内でこのレコードが属しているバックアップ・セットを一意に識別する連結主キーを構成します。
SET_COUNT	NUMBER	V\$BACKUP_SET からの SET_COUNT の値。SET_STAMP と SET_COUNT は、ターゲット・データベース制御ファイル内でこのレコードが属しているバックアップ・セットを一意に識別する連結主キーを構成します。
BS_RECID	NUMBER	V\$BACKUP_SET からの RECID。
BS_STAMP	NUMBER	V\$BACKUP_SET からの STAMP。
BACKUP_TYPE	VARCHAR2 (1)	バックアップのタイプ。D (全体バックアップまたはレベル 0 増分) または I (増分レベル 1)。
INCREMENTAL_LEVEL	NUMBER	増分バックアップのレベル。NULL、0 または 1。
COMPLETION_TIME	DATE	バックアップの完了時刻。
FILE#	NUMBER	データファイルの絶対ファイル番号。FILE# が 0 の場合は、レコードは制御ファイルを参照することに注意してください。
CREATION_CHANGE#	NUMBER	データファイルの作成時の SCN。
RESETLOGS_CHANGE#	NUMBER	データファイル・ヘッダー内の最新 RESETLOGS の SCN。
RESETLOGS_TIME	DATE	データファイル・ヘッダー内の最新 RESETLOGS のタイムスタンプ。
INCREMENTAL_CHANGE#	NUMBER	ブロックを増分バックアップに組み込むかどうかを判断する SCN。ブロックは、ブロック・ヘッダーにある SCN が INCREMENTAL_CHANGE# 以上の場合にかぎり組み込まれます。 増分バックアップが適用される REDO の範囲は、INCREMENTAL_CHANGE# から始まり、CHECKPOINT_CHANGE# で終わります。
CHECKPOINT_CHANGE#	NUMBER	このバックアップ・セット内のこのデータファイルのチェックポイント SCN。
CHECKPOINT_TIME	DATE	CHECKPOINT_CHANGE# に対応付けられている時刻。
ABSOLUTE_FUZZY_CHANGE#	NUMBER	絶対ファジー SCN。

列	データ型	説明
DATAFILE_BLOCKS	NUMBER	データファイル内のデータ・ブロック数。
BLOCKS	NUMBER	バックアップに書き込まれたデータ・ブロック数。通常、この値は DATAFILE_BLOCKS より小さくなります。これは、全体バックアップでは、使用されたことのないブロックはバックアップに含まれず、増分バックアップでは、変更されたことのないブロックはバックアップに含まれないためです。この値が DATAFILE_BLOCKS より大きくなることはありません。
BLOCK_SIZE	NUMBER	データ・ブロック・サイズ (バイト)。
STATUS	VARCHAR2 (1)	バックアップ・セットの状態。A (すべてのピースが使用可能)、D (すべてのピースが削除済)、o (一部のピースを除いて使用不可能なため、バックアップ・セットは使用不可)。
BS_LEVEL	NUMBER	このバックアップの作成時に指定された増分レベル (NULL、0 または 1)。ここには、INCREMENTAL_LEVEL 列とは異なる値を指定できます。これは、たとえばレベル 1 の増分バックアップを実行しても、一部のファイルに以前のレベル 0 のバックアップがなければ、そのファイルには自動的にレベル 0 のバックアップが使用されるためです。この場合、BS_LEVEL は 1 で、INCREMENTAL_LEVEL は 0 です。
PIECES	NUMBER	このバックアップ・データファイルを含むバックアップ・セット内のバックアップ・ピース数。
BLOCKS_READ	NUMBER	このバックアップを取る間にスキャンされたブロック数。このバックアップが増分バックアップで、バックアップを最適化するためにチェンジ・トラッキングが使用された場合は、この列の値は DATAFILE_BLOCKS より小さくなります。そうでない場合は、この列の値は DATAFILE_BLOCKS と同じになります。チェンジ・トラッキング・データが使用されていても、チェンジ・トラッキングによって読み取られたデータは増分バックアップの作成処理中にさらに絞り込まれるため、この列の値は BLOCKS より大きくなる場合があります。
CREATION_TIME	DATE	データファイル作成時のタイムスタンプ。
MARKED_CORRUPT	NUMBER	破損としてマークされたブロックの数。
USED_CHANGE_TRACKING	VARCHAR2 (3)	この増分バックアップを高速化するためにチェンジ・トラッキング・データが使用された (YES) か、使用されなかった (NO) かを示します。
USED_OPTIMIZATION	VARCHAR2 (3)	バックアップ最適化が適用された (YES) か、適用されなかった (NO) かを示します。
PCT_NOTREAD	NUMBER	このバックアップ中にスキップされたファイルの割合。増分バックアップの場合、この値はブロック・チェンジ・トラッキング・ファイルの効果を示します。
FOREIGN_DBID	NUMBER	このデータファイルのトランスポート元であるデータベースの外部 DBID。バックアップされたファイルが外部データベース・ファイルでない場合、値は 0 になります。
PLUGGED_READONLY	VARCHAR2 (3)	トランスポートされた読取り専用外部ファイルのバックアップである場合は YES。そうでない場合は NO。
PLUGIN_CHANGE#	NUMBER	外部データファイルがデータベースにトランスポートされたときの SCN。このファイルが外部データベース・ファイルでない場合、値は 0 です。
PLUGIN_RESETLOGS_CHANGE#	NUMBER	この外部ファイルがトランスポートされたインカネーションに対する RESETLOGS 操作の SCN。このファイルが外部データベース・ファイルでない場合、値は 0 です。
PLUGIN_RESETLOGS_TIME	DATE	この外部ファイルがトランスポートされたインカネーションに対する RESETLOGS 操作の時刻。このファイルが外部データベース・ファイルでない場合、値は 0 です。
SECTION_SIZE	NUMBER	マルチセクション・バックアップの各セクションに含まれるブロック数を指定します。ファイル全体のバックアップの場合、値は 0 です。

## RC\_BACKUP\_DATAFILE\_DETAILS

RC\_BACKUP\_DATAFILE\_DETAILS ビューは、リカバリ・カタログに登録されているデータベースの使用可能なデータファイルのバックアップに関する詳細情報を表示します。

このビューは、主に Enterprise Manager で内部的に使用します。

列	データ型	説明
BTYPE	CHAR (9)	バックアップ・タイプ・コンテナ (BACKUPSET、IMAGECOPY または PROXYCOPY)。
BTYPE_KEY	NUMBER	バックアップ・タイプの一意の識別子。BTYPE が BACKUPSET の場合、この値はバックアップ・セットの BS_KEY の値になります。BTYPE が IMAGECOPY の場合、この値はコピーの COPY_KEY の値になります。
SESSION_KEY	NUMBER	セッションの識別子。RC_RMAN_OUTPUT および RC_RMAN_BACKUP_JOB_DETAILS との結合に使用します。
SESSION_RECID	NUMBER	SESSION_STAMP とともに使用して、このバックアップ・ジョブの出力を RC_RMAN_OUTPUT から一意に識別します。
SESSION_STAMP	NUMBER	SESSION_RECID とともに使用して、このバックアップ・ジョブの出力を RC_RMAN_OUTPUT から一意に識別します。
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカンションの DB_NAME。
ID1	NUMBER	バックアップ・セットとして取られるバックアップでは、この列には SET_STAMP が含まれます。プロキシ・コピーまたはイメージ・コピーのバックアップでは、この列には制御ファイルからの RECID が含まれます。
ID2	NUMBER	バックアップ・セットとして取られるバックアップでは、この列には SET_COUNT が含まれます。プロキシ・コピーまたはイメージ・コピーのバックアップの場合、この列には STAMP が含まれます。
FILE#	NUMBER	このデータファイルの番号。
CREATION_CHANGE#	NUMBER	このデータファイルが作成されたときのチェックポイント SCN。
CREATION_TIME	DATE	このデータファイルが作成された時刻。
RESETLOGS_CHANGE#	NUMBER	このデータファイルに影響する最新の RESETLOGS 操作のチェックポイント SCN。
RESETLOGS_TIME	DATE	このデータファイルに影響する最新の RESETLOGS 操作の時刻。
INCREMENTAL_LEVEL	NUMBER	増分バックアップにおける、増分バックアップのレベル (0 または 1)。これ以外の場合は NULL です。
INCREMENTAL_CHANGE#	NUMBER	増分バックアップでは、増分バックアップの SCN です。これ以外の場合は NULL です。
CHECKPOINT_CHANGE#	NUMBER	データファイルをバックアップした時点のデータファイルの現行のチェックポイント SCN。
CHECKPOINT_TIME	DATE	データファイルをバックアップした時点のデータファイルの現行のチェックポイントに対応する時刻。
MARKED_CORRUPT	NUMBER	破損としてマークされたデータファイル・ブロックの数。
FILESIZE	NUMBER	データファイルをバックアップした時点のデータファイルのサイズ。
COMPRESSION_RATIO	NUMBER	このバックアップの圧縮率。
TS#	NUMBER	表領域の番号。
TSNAME	VARCHAR2 (30)	このデータファイルが含まれている表領域の名前。
FILESIZE_DISPLAY	VARCHAR2 (4000)	FILESIZE と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。

## RC\_BACKUP\_DATAFILE\_SUMMARY

RC\_BACKUP\_DATAFILE\_SUMMARY ビューは、使用可能なデータファイルのバックアップに関するサマリーを表示します。

このビューは、主に Enterprise Manager で内部的に使用します。

列	データ型	説明
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカネーションの DB_NAME。
NUM_FILES_BACKED	NUMBER	この DB_KEY および DB_NAME の値に対するバックアップしたデータファイルの数。
NUM_DISTINCT_FILES_BACKED	NUMBER	この DB_KEY および DB_NAME の値に対するバックアップした個々のデータファイルの数。
NUM_DISTINCT_TS_BACKED	NUMBER	この DB_KEY および DB_NAME の値に対するバックアップした個々の表領域の数。
MIN_CHECKPOINT_CHANGE#	NUMBER	この DB_KEY および DB_NAME の値に対するバックアップした任意のデータファイルの最小チェックポイント。
MAX_CHECKPOINT_CHANGE#	NUMBER	この DB_KEY および DB_NAME の値に対するバックアップした任意のデータファイルの最大チェックポイント変更番号。
MIN_CHECKPOINT_TIME	DATE	この DB_KEY および DB_NAME の値に対するバックアップした任意のデータファイルの最小チェックポイント時刻。
MAX_CHECKPOINT_TIME	DATE	この DB_KEY および DB_NAME の値に対するバックアップした任意のデータファイルの最大チェックポイント時刻。
INPUT_BYTES	NUMBER	この DB_KEY および DB_NAME の値に対するバックアップしたすべてのファイルで読み取られた合計入力バイト数。
OUTPUT_BYTES	NUMBER	この DB_KEY および DB_NAME の値に対するバックアップで生成されたすべての出力ピースの合計サイズ。
COMPRESSION_RATIO	NUMBER	すべてのバックアップの圧縮率。
INPUT_BYTES_DISPLAY	VARCHAR2 (4000)	INPUT_BYTES と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。
OUTPUT_BYTES_DISPLAY	VARCHAR2 (4000)	OUTPUT_BYTES と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。



## RC\_BACKUP\_FILES

このビューは、Recovery Manager リポジトリで認識されるバックアップを、リカバリ・カタログに反映されているとおりにリストします。このビューは、V\$BACKUP\_FILES 制御ファイル・ビューに対応しています。

### 注意：

- 通常、Recovery Manager 内から LIST BACKUP および LIST COPY コマンドを使用すると、この情報に簡単にアクセスできます。
- リカバリ・カタログに登録されているデータベースが 1 つのみの場合も、このビューを使用する前に DBMS\_RCVMAN.SetDatabase を使用して、リカバリ・カタログ・スキーマからデータベースを選択する必要があります。このタスクの実行方法は、『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』の説明を参照してください。

列	データ型	説明
PKEY	NUMBER	バックアップに対する主キー。
BACKUP_TYPE	VARCHAR2 (32)	バックアップのタイプ。BACKUP SET、COPY または PROXY COPY。
FILE_TYPE	VARCHAR2 (32)	バックアップされたファイルのタイプ。DATAFILE、CONTROLFILE、SPFILE、REDO LOG、COPY (イメージ・コピーのバックアップ用) または PIECE (バックアップ・ピース用)。
KEEP	VARCHAR2 (3)	このバックアップに、バックアップの保存方針をオーバーライドする KEEP 属性が設定されているかどうか。値は YES または NO です。
KEEP_UNTIL	DATE	その日付を過ぎると、このバックアップが不要とみなされる日付。
KEEP_OPTIONS	VARCHAR2 (13)	このバックアップの保存に影響する属性。可能な値は、NOLOGS、BACKUP_LOGS、LOGS および NULL です。NOLOGS は、データベースがマウントされたときに一貫性のあるバックアップが作成されたことを示します。BACKUP_LOGS は、オープン・モードでバックアップが作成されたことを示します。したがって、一貫性のある状態にするには、アーカイブ・ログ・バックアップを適用する必要があります。LOGS は、LOGS キーワードを使用して長期間バックアップが作成されたことを示します。現在、LOGS キーワードは使用不可です。NULL は、このバックアップには KEEP オプションが指定されておらず、保存方針に基づいて不要とされることを示します。
STATUS	VARCHAR2 (16)	バックアップの状態。可能な値は AVAILABLE、UNAVAILABLE、EXPIRED です。
FNAME	VARCHAR2 (1024)	このピース、コピーのファイル名、またはこのバックアップ・セットに含まれるファイルのファイル名。たとえば、BACKUP_TYPE が BACKUP SET、FILE_TYPE が DATAFILE の列の場合、FNAME はバックアップ内のデータファイル名です。一方、BACKUP_TYPE が BACKUP SET、FILE_TYPE が PIECE の場合は、FNAME はバックアップ・ピースの名前を示します。
TAG	VARCHAR2 (32)	バックアップ・ピースまたはイメージ・コピーのタグ。FILE_TYPE が PIECE または COPY の場合のみ、この列に値が保持されます。
MEDIA	VARCHAR2 (80)	バックアップが格納されるメディアのメディア ID。BACKUP_TYPE が BACKUP SET で、FILE_TYPE が PIECE の場合のみ、この列に値が保持されます。
RECID	NUMBER	この行に対応する制御ファイル・レコードの ID。
STAMP	NUMBER	この行に対応する制御ファイル・レコードのタイムスタンプ。
DEVICE_TYPE	VARCHAR2 (255)	バックアップが格納されるデバイス・タイプ。FILE_TYPE が PIECE または COPY の場合のみ、この列に値が移入されます。

列	データ型	説明
BLOCK_SIZE	NUMBER	バックアップまたはコピーのブロック・サイズ (バイト)。
COMPLETION_TIME	NUMBER	バックアップが完了した時刻。FILE_TYPE が PIECE または COPY の場合のみ、この列に値が移入されます。
COMPRESSED	VARCHAR2 (3)	この行で表されるバックアップ・ピースが圧縮されているかどうか。FILE_TYPE が PIECE の場合のみ、この列に値が移入されます (本来、イメージ・コピーは圧縮できないため)。
OBSOLETE	VARCHAR2 (3)	このバックアップ・ピースまたはコピーが不要かどうか。可能な値: YES、NO。FILE_TYPE が PIECE または COPY の場合のみ、この列に値が移入されます。
BYTES	NUMBER	この行に示されるファイルのサイズ。BACKUP_TYPE が BACKUP SET の場合、これはバックアップ・セットの合計サイズです。FILE_TYPE が PIECE または COPY の場合、これは個々のファイルのサイズです。FILE_TYPE が DATAFILE、ARCHIVED LOG、SPFILE または CONTROL FILE の場合、この値はバックアップ・セットに取り込まれたデータ量を表します (ただし、バックアップ・セットが圧縮されて作成された場合は、対応するバックアップ・セットが小さくなる場合があります)。
BS_KEY	NUMBER	バックアップ・セットのキー。BACKUP_TYPE が BACKUP SET の場合のみ、この列に値が移入されます。この列を使用して、RC_BACKUP_SET または RC_BACKUP_PIECE との結合を構成します。
BS_COUNT	NUMBER	バックアップ・セットの数。BACKUP_TYPE が BACKUP SET の場合のみ、この列に値が移入されます。
BS_STAMP	NUMBER	バックアップ・セットのタイムスタンプ。BACKUP_TYPE が BACKUP SET の場合のみ、この列に値が移入されます。
BS_TYPE	VARCHAR2 (32)	バックアップ・セットの内容のタイプ (データファイルまたはアーカイブ REDO ログ)。BACKUP_TYPE が BACKUP SET の場合のみ、この列に値が移入されます。
BS_INCR_TYPE	VARCHAR (32)	バックアップ・セットの増分タイプ (全体または全体以外)。BACKUP_TYPE が BACKUP SET の場合のみ、この列に値が移入されます。
BS_PIECES	NUMBER	バックアップ・セット内のピースの数。BACKUP_TYPE が BACKUP SET の場合のみ、この列に値が移入されます。
BS_COPIES	NUMBER	バックアップ・セットのコピーの数。BACKUP_TYPE が BACKUP SET の場合のみ、この列に値が移入されます。
BS_COMPLETION_TIME	DATE	バックアップ・セットの完了時刻。BACKUP_TYPE が BACKUP SET の場合のみ、この列に値が移入されます。
BS_STATUS	VARCHAR2 (16)	バックアップ・セットの状態。可能な値は AVAILABLE、UNAVAILABLE、EXPIRED または OTHER です。(OTHER は、バックアップ・セットのピースが異なる状態で存在していることを意味します。この状態は、一部が AVAILABLE で、その他が UNAVAILABLE の場合などに発生します)。BACKUP_TYPE が BACKUP SET の場合のみ、この列に値が移入されます。
BS_BYTES	NUMBER	バックアップ・セット内のすべてのバックアップ・ピースの合計サイズ。BACKUP_TYPE が BACKUP SET の場合のみ、この列に値が移入されます。
BS_COMPRESSED	VARCHAR2 (3)	このバックアップ・セットのバックアップ・ピースが圧縮されているかどうか。BACKUP_TYPE が BACKUP SET の場合のみ、この列に値が移入されます。
BS_TAG	VARCHAR2 (1024)	このバックアップ・セットのバックアップ・ピースのタグ。ピースのタグが異なる場合、BS_TAGS には、バックアップ・セット内のすべてのピースのタグのカンマで区切られたリストが含まれます。BACKUP_TYPE が BACKUP SET の場合のみ、この列に値が移入されます。
BS_DEVICE_TYPE	VARCHAR2 (255)	バックアップ・セットが格納されるデバイス・タイプ。このバックアップ・セットのコピーが複数存在し、異なるデバイスに格納される場合、このフィールドには、すべてのデバイス・タイプのカンマで区切られたリストが含まれます。たとえば、バックアップ・セットがディスク上に存在し、テープにもバックアップされている場合、BS_DEVICE_TYPE には DISK、SBT_TAPE が含まれます。BACKUP_TYPE が BACKUP SET の場合のみ、この列に値が移入されます。

列	データ型	説明
BP_PIECE#	NUMBER	このバックアップ・セットを構成するバックアップ・ピースの数。BACKUP_TYPE が BACKUP SET の場合のみ、この列に値が移入されます。
BP_COPY#	NUMBER	バックアップ・セットのコピーの数。BACKUP_TYPE が BACKUP SET の場合のみ、この列に値が移入されます。
DF_FILE#	NUMBER	この行に示されるデータファイルのファイル数。FILE_TYPE が DATAFILE の場合のみ、この列に値が移入されます。
DF_TABLESPACE	VARCHAR2 (30)	この行に示されるデータファイルの表領域名。FILE_TYPE が DATAFILE の場合のみ、この列に値が移入されます。
DF_RESETLOGS_CHANGE#	NUMBER	この行に示されるデータファイルの RESETLOGS の変更。FILE_TYPE が DATAFILE の場合のみ、この列に値が移入されます。
DF_CREATION_CHANGE#	NUMBER	この行に示されるデータファイルの作成時の変更番号。FILE_TYPE が DATAFILE の場合のみ、この列に値が移入されます。
DF_CHECKPOINT_CHANGE#	NUMBER	この行に示されるデータファイルのチェックポイント時の変更番号。FILE_TYPE が DATAFILE の場合のみ、この列に値が移入されます。
DF_CKP_MOD_TIME	DATE	この行に示されるデータファイルのチェックポイント時刻。FILE_TYPE が DATAFILE の場合にのみ有効です。
RL_THREAD#	NUMBER	この行に示されるアーカイブ REDO ログの REDO ログ・スレッド番号。FILE_TYPE が ARCHIVED LOG の場合にのみ有効です。
RL_SEQUENCE#	NUMBER	この行に示されるアーカイブ REDO ログの REDO ログ順序番号。FILE_TYPE が ARCHIVED LOG の場合にのみ有効です。
RL_RESETLOGS_CHANGE#	NUMBER	この行に示されるアーカイブ REDO ログの RESETLOGS の変更番号。FILE_TYPE が ARCHIVED LOG の場合にのみ有効です。
RL_FIRST_CHANGE#	NUMBER	この行に示されるアーカイブ REDO ログの最初の変更番号。FILE_TYPE が ARCHIVED LOG の場合にのみ有効です。
RL_FIRST_TIME	DATE	この行に示されるアーカイブ REDO ログの最初の変更時刻。FILE_TYPE が ARCHIVED LOG の場合にのみ有効です。
RL_NEXT_CHANGE#	NUMBER	この行に示されるアーカイブ REDO ログ後の変更番号。FILE_TYPE が ARCHIVED LOG の場合にのみ有効です。
RL_NEXT_TIME	DATE	この行に示されるアーカイブ・ログ直後の変更時刻。FILE_TYPE が ARCHIVED LOG の場合にのみ有効です。

## RC\_BACKUP\_PIECE

このビューは、バックアップ・ピースに関する情報をリストします。このビューは、V\$BACKUP\_PIECE ビューに対応しています。各バックアップ・セットには1つ以上のバックアップ・ピースが含まれています。

同じバックアップ・ピースのコピーを複数持つことができますが、制御ファイルには各コピーのレコード、ビューには各コピーの行があります。

列	データ型	説明
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DB_ID	NUMBER	データベース識別子。
BP_KEY	NUMBER	リカバリ・カタログ内のバックアップ・ピースに対する主キー。Recovery Manager がリカバリ・カタログに接続している間に LIST コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	NUMBER	V\$BACKUP_PIECE からのバックアップ・ピースの RECID。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
STAMP	NUMBER	V\$BACKUP_PIECE から伝播したバックアップ・ピースの STAMP。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
BS_KEY	NUMBER	リカバリ・カタログ内でこのレコードが属しているバックアップ・セットの主キー。この列を使用して、RC_BACKUP_SET、RC_BACKUP_CONTROLFILE、RC_BACKUP_DATAFILE などとの結合を構成します。
SET_STAMP	NUMBER	V\$BACKUP_SET からの SET_STAMP の値。SET_STAMP と SET_COUNT は、ターゲット・データベース制御ファイル内でこのレコードが属しているバックアップ・セットを一意に識別する連結主キーを構成します。
SET_COUNT	NUMBER	V\$BACKUP_SET からの SET_COUNT の値。SET_STAMP と SET_COUNT は、ターゲット・データベース制御ファイル内でこのレコードが属しているバックアップ・セットを一意に識別する連結主キーを構成します。
BACKUP_TYPE	VARCHAR2(1)	バックアップのタイプ。D (全体バックアップまたはレベル0増分)、I (増分レベル1)、L (アーカイブ REDO ログ)。
INCREMENTAL_LEVEL	NUMBER	増分バックアップのレベル。NULL、0 または 1。
PIECE#	NUMBER	バックアップ・ピースの番号。最初のピースの値は 1 です。
COPY#	NUMBER	バックアップ・ピースのコピー番号。
DEVICE_TYPE	VARCHAR2(255)	DISK など、バックアップ・デバイスのタイプ。
HANDLE	VARCHAR2(1024)	バックアップ・ピースのファイル名。
COMMENTS	VARCHAR2(255)	バックアップ・ピースについてのコメント。
MEDIA	VARCHAR2(80)	このバックアップを作成したメディア・マネージャについての詳細コメント。
MEDIA_POOL	NUMBER	バックアップを格納するメディア・プールの数。
CONCUR	VARCHAR2(3)	バックアップ・メディアが同時アクセスをサポートするかどうかを、YES または NO で指定します。
TAG	VARCHAR2(32)	バックアップ・ピースのタグ。タグ名のデフォルトの書式は、BACKUP の説明を参照してください。
START_TIME	DATE	Recovery Manager がバックアップ・ピースへの書込みを開始した時刻。
COMPLETION_TIME	DATE	バックアップ・ピースが完了した時刻。
ELAPSED_SECONDS	NUMBER	バックアップ・ピース作成の実行時間。

列	データ型	説明
STATUS	VARCHAR2 (1)	バックアップ・ピースの状態。A (使用可能)、U (使用不可能)、D (削除済) または X (期限切れ)。状態 D が表示されるのは、古いリカバリ・カタログがアップグレードされている場合のみであることに注意してください。
BYTES	NUMBER	バックアップ・ピースのサイズ (バイト)。
IS_RECOVERY_DEST_FILE	VARCHAR2 (3)	このバックアップ・ピースはフラッシュ・リカバリ領域に格納されます。YES または NO。
RSR_KEY	NUMBER	このバックアップ・ピースを作成した RC_RMAN_STATUS の行に対する一意のキー。
COMPRESSED	VARCHAR2 (3)	バックアップ・ピースが圧縮されている (YES) か、圧縮されていない (NO) かを示します。
SITE_KEY	NUMBER	このファイルと関連付けられている Data Guard データベースの主キー。Data Guard 環境にある各データベースの SITE_KEY の値は一意です。SITE_KEY を使用して RC_SITE ビューと結合すると、データベースの DB_UNIQUE_NAME を取得できます。
ENCRYPTED	VARCHAR2 (3)	バックアップ・ピースが暗号化されている (YES) か、暗号化されていない (NO) かを示します。
BACKED_BY_OSB	VARCHAR2 (3)	バックアップ・ピースが Oracle Secure Backup にバックアップされている (YES) か、バックアップされていない (NO) かを示します。

## RC\_BACKUP\_PIECE\_DETAILS

RC\_BACKUP\_PIECE\_DETAILS ビューには、リカバリ・カタログに登録されている使用可能なすべてのバックアップ・ピースに関する詳細情報が含まれています。このビューの大部分の列の意味は、リカバリ・カタログ・ビュー RC\_BACKUP\_PIECE と同じです。

このビューは、主に Enterprise Manager で内部的に使用します。

列	データ型	説明
SESSION_KEY	NUMBER	セッションの識別子。RC_RMAN_OUTPUT および RC_RMAN_BACKUP_JOB_DETAILS との結合に使用します。
SESSION_RECID	NUMBER	SESSION_STAMP とともに使用して、このバックアップ・ジョブの出力を RC_RMAN_OUTPUT から一意に識別します。
SESSION_STAMP	NUMBER	SESSION_RECID とともに使用して、このバックアップ・ジョブの出力を RC_RMAN_OUTPUT から一意に識別します。
DB_NAME	VARCHAR2(8)	このレコードが属しているデータベース・インカネーションの DB_NAME。
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DB_ID	NUMBER	このレコードが属しているデータベース・インカネーションの DBID。
BP_KEY	NUMBER	リカバリ・カタログ内のバックアップ・ピースに対する主キー。Recovery Manager がリカバリ・カタログに接続している間に LIST コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	NUMBER	RC_BACKUP_PIECE からのバックアップ・ピース RECID。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
STAMP	NUMBER	V\$BACKUP_PIECE から伝播したバックアップ・ピースの STAMP。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
BS_KEY	NUMBER	リカバリ・カタログ内でこのレコードが属しているバックアップ・セットの主キー。この列を使用して、RC_BACKUP_SET または RC_BACKUP_PIECE との結合を構成します。
SET_STAMP	NUMBER	V\$BACKUP_SET からの SET_STAMP の値。SET_STAMP と SET_COUNT は、ターゲット・データベース制御ファイル内でこのレコードが属しているバックアップ・セットを一意に識別する連結キーを構成します。
SET_COUNT	NUMBER	V\$BACKUP_SET からの SET_COUNT の値。SET_STAMP と SET_COUNT は、ターゲット・データベース制御ファイル内でこのレコードが属しているバックアップ・セットを一意に識別する連結キーを構成します。
BACKUP_TYPE	VARCHAR2(1)	バックアップのタイプ。可能な値は、データファイルまたは制御ファイルのバックアップでは D、増分バックアップでは I、アーカイブ・ログ・ファイルのバックアップでは L です。
INCREMENTAL_LEVEL	NUMBER	増分バックアップにおける、増分バックアップのレベルを示します。可能な値は、NULL (完全バックアップの場合)、0 または 1 です。
PIECE#	NUMBER	バックアップ・ピースの番号。最初のピースの値は 1 です。
COPY#	NUMBER	多重化を有効にして作成されたバックアップ・ピースのコピー番号を示します。バックアップ・ピースが多重化されていない場合は 1 です。
DEVICE_TYPE	VARCHAR2(255)	バックアップ・ピースが格納されているデバイスのタイプ。ディスク上のバックアップ・セットの場合は DISK に設定します。
HANDLE	VARCHAR2(1024)	バックアップ・ピースのファイル名。
COMMENTS	VARCHAR2(255)	バックアップ・ピースについてのコメント。
MEDIA	VARCHAR2(80)	このバックアップを作成したメディア・マネージャについての詳細コメント。

列	データ型	説明
MEDIA_POOL	NUMBER	バックアップを格納するメディア・プールの数。0は、メディア・プールがないことを示します。
CONCUR	VARCHAR2 (3)	バックアップ・メディアが同時アクセスをサポートするかどうかを、YES または NO で指定します。
TAG	VARCHAR2 (32)	このバックアップ・ピースに関連するタグ。
START_TIME	DATE	Recovery Manager がバックアップ・ピースへの書き込みを開始した時刻。
COMPLETION_TIME	DATE	バックアップ・ピースが完了した時刻。
ELAPSED_SECONDS	NUMBER	バックアップ・ピース作成の実行時間。
STATUS	VARCHAR2 (1)	バックアップ・ピースの状態。バックアップ・ピースが AVAILABLE の場合は A です。(このビューは使用可能なバックアップ・ピースのみを表示するため、値は常に A になります。)
BYTES	NUMBER	バックアップ・ピースのサイズ (バイト)。
IS_RECOVERY_DEST_FILE	VARCHAR2 (3)	このバックアップ・ピースがフラッシュ・リカバリ領域に格納される場合は YES。それ以外の場合は NO。
RSR_KEY	NUMBER	このバックアップ・ピースを作成したジョブに対応する RC_RMAN_STATUS の行に対する一意のキー。
COMPRESSED	VARCHAR2 (3)	このバックアップ・ピースが圧縮されている場合は YES。それ以外の場合は NO。
SITE_KEY	NUMBER	このファイルと関連付けられている Data Guard データベースの主キー。Data Guard 環境にある各データベースの SITE_KEY の値は一意です。SITE_KEY を使用して RC_SITE ビューと結合すると、データベースの DB_UNIQUE_NAME を取得できます。
ENCRYPTED	VARCHAR2 (3)	バックアップ・ピースが暗号化されている (YES) か、暗号化されていない (NO) かを示します。
BACKED_BY_OS	VARCHAR2 (3)	バックアップ・ピースが Oracle Secure Backup にバックアップされている (YES) か、バックアップされていない (NO) かを示します。
PIECES_PER_SET	NUMBER	このバックアップ・ピースを含むバックアップ・セット内のバックアップ・ピース数。
SIZE_BYTES_DISPLAY	VARCHAR2 (4000)	BYTES と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。

## RC\_BACKUP\_REDOLOG

このビューは、バックアップ・セットのアーカイブ REDO ログに関する情報をリストします。このビューは、V\$BACKUP\_REDOLOG ビューに対応しています。

オンライン・ログを直接バックアップすることはできません。最初にオンライン・ログをディスクにアーカイブし、次にバックアップを作成します。アーカイブ・ログのバックアップ・セットには、1 つ以上のアーカイブ・ログが含まれています。

列	データ型	説明
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	ターゲット・データベースのインカーネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2(8)	このレコードが属しているデータベース・インカーネーションの DB_NAME。
BRL_KEY	NUMBER	リカバリ・カタログ内のアーカイブ REDO ログの主キー。Recovery Manager がリカバリ・カタログに接続している間に LIST コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	NUMBER	V\$BACKUP_REDOLOG から伝播したレコード識別子。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
STAMP	NUMBER	V\$BACKUP_REDOLOG からの STAMP。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
BS_KEY	NUMBER	リカバリ・カタログ内でこのレコードが属しているバックアップ・セットの主キー。この列を使用して、RC_BACKUP_SET または RC_BACKUP_PIECE との結合を構成します。
SET_STAMP	NUMBER	V\$BACKUP_SET からの SET_STAMP の値。SET_STAMP と SET_COUNT は、ターゲット・データベース制御ファイル内でこのレコードが属しているバックアップ・セットを一意に識別する連結主キーを構成します。
SET_COUNT	NUMBER	V\$BACKUP_SET からの SET_COUNT の値。SET_STAMP と SET_COUNT は、ターゲット・データベース制御ファイル内でこのレコードが属しているバックアップ・セットを一意に識別する連結主キーを構成します。
BACKUP_TYPE	VARCHAR2(1)	バックアップのタイプ。L (アーカイブ REDO ログ)。
COMPLETION_TIME	DATE	バックアップが完了した時刻。
THREAD#	NUMBER	REDO ログのスレッド番号。
SEQUENCE#	NUMBER	ログ順序番号。
RESETLOGS_CHANGE#	NUMBER	レコード作成時における最新の RESETLOGS の SCN。
RESETLOGS_TIME	DATE	レコード作成時における最新の RESETLOGS のタイムスタンプ。
FIRST_CHANGE#	NUMBER	REDO ログへの切替時に生成された SCN。
FIRST_TIME	DATE	REDO ログに切り替えた時刻。
NEXT_CHANGE#	NUMBER	スレッド内にある次の REDO ログの最初の SCN。
NEXT_TIME	DATE	スレッド内にある次の REDO ログの最初のタイムスタンプ。
BLOCKS	NUMBER	バックアップに書き込まれたオペレーティング・システムのブロック数。
BLOCK_SIZE	NUMBER	この REDO ログの各ブロックのバイト数。
STATUS	VARCHAR2(1)	バックアップ・セットの状態。A (すべてのピースが使用可能)、D (すべてのピースが削除済)、O (一部のピースを除いて使用不可能なため、バックアップ・セットは使用不可)。
BS_RECID	NUMBER	V\$BACKUP_SET からの RECID の値。



列	データ型	説明
BS_STAMP	NUMBER	V\$BACKUP_SET からの STAMP の値。BS_STAMP が SET_STAMP とは異なることに注意してください。BS_STAMP は制御ファイル内で作成された時点のバックアップ・セット・レコードのスタンプで、SET_STAMP は SET_COUNT と結合して一意の識別子となります。
PIECES	NUMBER	バックアップ・セット内のピースの数。
TERMINAL	VARCHAR2 (3)	このログが、スタンバイ・データベースのターミナル・リカバリ中に作成されたログかどうかを示します。値は YES または NO です。

## RC\_BACKUP\_SET

このビューは、データベースの全インカネーションのバックアップ・セットに関する情報をリストします。このビューは、V\$BACKUP\_SET ビューに対応しています。バックアップ・セット・レコードは、バックアップが正常に完了した後に挿入されます。

列	データ型	説明
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DB_ID	NUMBER	一意のデータベース識別子。
BS_KEY	NUMBER	リカバリ・カタログ内のバックアップ・セットの主キー。Recovery Manager がリカバリ・カタログに接続している間に LIST コマンドを発行すると、この値が出力の KEY 列に表示されます。この列を使用して、RC_BACKUP_PIECE との結合を構成します。
RECID	NUMBER	V\$BACKUP_SET からのバックアップ・セットの RECID。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。RECID と STAMP、または SET_STAMP と SET_COUNT を使用して、V\$BACKUP_SET にアクセスします。
STAMP	NUMBER	V\$BACKUP_SET からのバックアップ・セットの STAMP。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。RECID と STAMP、または SET_STAMP と SET_COUNT を使用して、V\$BACKUP_SET にアクセスします。
SET_STAMP	NUMBER	V\$BACKUP_SET からの SET_STAMP の値。SET_STAMP と SET_COUNT は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。RECID と STAMP、または SET_STAMP と SET_COUNT を使用して、V\$BACKUP_SET にアクセスします。
SET_COUNT	NUMBER	V\$BACKUP_SET からの SET_COUNT の値。SET_STAMP と SET_COUNT は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。RECID と STAMP、または SET_STAMP と SET_COUNT を使用して、V\$BACKUP_SET にアクセスします。
BACKUP_TYPE	VARCHAR2 (1)	バックアップのタイプ。D (全体バックアップまたはレベル 0 増分)、I (増分レベル 1)、L (アーカイブ REDO ログ)。
INCREMENTAL_LEVEL	NUMBER	増分バックアップのレベル。NULL、0 または 1。
PIECES	NUMBER	バックアップ・セット内のバックアップ・ピースの数。
START_TIME	DATE	バックアップを開始した時刻。
COMPLETION_TIME	DATE	バックアップが完了した時刻。
ELAPSED_SECONDS	NUMBER	バックアップ実行時間 (秒)。
STATUS	VARCHAR2 (1)	バックアップ・セットの状態。A (すべてのバックアップ・ピースが使用可能)、D (すべてのバックアップ・ピースが削除済)、O (一部のバックアップ・ピースを除いて使用不可能なため、バックアップ・セットは使用不可)。
CONTROLFILE_INCLUDED	VARCHAR2 (7)	可能な値は、NONE (バックアップ・セットにはバックアップ制御ファイルが含まれない場合)、BACKUP (バックアップ・セットに通常のバックアップ制御ファイルが含まれる場合) および STANDBY (バックアップ・セットにスタンバイ制御ファイルが含まれる場合) です。
INPUT_FILE_SCAN_ONLY	VARCHAR2 (3)	このバックアップ・セット・レコードは、BACKUP VALIDATE コマンドにより作成されたものです。実際のバックアップ・セットは存在しません。このレコードは、スキャンされるデータファイルと各ファイルで見つかった破損ブロック (存在する場合) を追跡するためのプレースホルダにすぎません。  COMPATIBLE が 11.0.0 以上に設定されている場合、Recovery Manager はこの列に値を移入しません。

列	データ型	説明
KEEP	VARCHAR2 (3)	このバックアップ・セットの保存方針が CONFIGURE RETENTION POLICY の値とは異なるかどうかを示します。可能な値は YES および NO です。
KEEP_UNTIL	DATE	BACKUP コマンドの KEEP UNTIL TIME 句を指定した場合は、この列にバックアップが不要になる直前の日付が表示されます。列が NULL で KEEP OPTIONS が NULL でなければ、バックアップが不要とされることはありません。
KEEP_OPTIONS	VARCHAR2 (11)	このバックアップ・セットに指定された KEEP オプション。可能な値は、NOLOGS、BACKUP_LOGS、LOGS および NULL です。NOLOGS は、データベースがマウントされたときに一貫性のあるバックアップが作成されたことを示します。BACKUP_LOGS は、オープン・モードでバックアップが作成されたことを示します。したがって、一貫性のある状態にするには、アーカイブ・ログ・バックアップを適用する必要があります。LOGS は、LOGS キーワードを使用して長期間バックアップが作成されたことを示します。現在、LOGS キーワードは使用不可です。NULL は、このバックアップには KEEP オプションが指定されておらず、保存方針に基づいて不要とされることを示します。
BLOCK_SIZE	NUMBER	バックアップ・セットのブロック・サイズ。
SITE_KEY	NUMBER	このバックアップ・セットと関連付けられている Data Guard データベースの主キー。Data Guard 環境にある各データベースの SITE_KEY の値は一意です。SITE_KEY を使用して RC_SITE ビューと結合すると、データベースの DB_UNIQUE_NAME を取得できます。
MULTI_SECTION	VARCHAR2 (3)	マルチセクション・バックアップの場合は Y、それ以外の場合は NULL。

## RC\_BACKUP\_SET\_DETAILS

RC\_BACKUP\_SET\_DETAILS ビューは、BACKUP BACKUPSET コマンドを使用して作成したバックアップ・セットを含め、現在使用可能なバックアップ・セットに関する詳細を表示します。

このビューは、主に Enterprise Manager で内部的に使用します。

列	データ型	説明
SESSION_KEY	NUMBER	セッションの識別子。RC_RMAN_OUTPUT および RC_RMAN_BACKUP_JOB_DETAILS との結合に使用します。
SESSION_RECID	NUMBER	SESSION_STAMP とともに使用して、このバックアップ・ジョブの出力を RC_RMAN_OUTPUT から一意に識別します。
SESSION_STAMP	NUMBER	SESSION_RECID とともに使用して、このバックアップ・ジョブの出力を RC_RMAN_OUTPUT から一意に識別します。
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカンネーションの DB_NAME。
BS_KEY	NUMBER	リカバリ・カタログ内でこのレコードが属しているバックアップ・セットの主キー。この列を使用して、RC_BACKUP_SET または RC_BACKUP_PIECE との結合を構成します。
RECID	NUMBER	バックアップ・セットの RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
STAMP	NUMBER	バックアップ・セットの RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
SET_STAMP	NUMBER	V\$BACKUP_SET からの SET_STAMP の値。SET_STAMP と SET_COUNT は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結キーを構成します。
SET_COUNT	NUMBER	V\$BACKUP_SET からの SET_COUNT の値。SET_STAMP と SET_COUNT は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結キーを構成します。
BACKUP_TYPE	VARCHAR2 (1)	バックアップのタイプ。D (全体バックアップまたはレベル 0 増分)、I (増分レベル 1)、L (アーカイブ REDO ログ)。
CONTROLFILE_INCLUDED	VARCHAR2 (7)	可能な値は、NONE (バックアップ・セットにはバックアップ制御ファイルが含まれない場合)、BACKUP (バックアップ・セットに通常のバックアップ制御ファイルが含まれる場合) および STANDBY (バックアップ・セットにスタンバイ制御ファイルが含まれる場合) です。
INCREMENTAL_LEVEL	NUMBER	増分バックアップのレベル。NULL、0 または 1。
PIECES	NUMBER	バックアップ・セット内のバックアップ・ピースの数。
START_TIME	DATE	バックアップを開始した時刻。
COMPLETION_TIME	DATE	バックアップが完了した時刻。
ELAPSED_SECONDS	NUMBER	バックアップ実行時間 (秒)。
BLOCK_SIZE	VARCHAR2	バックアップ・セットにバックアップ・ピースを作成するときを使用したブロック・サイズ。
KEEP	VARCHAR2 (3)	このバックアップ・セットの保存方針が CONFIGURE RETENTION POLICY の値とは異なるかどうかを示します。可能な値は YES および NO です。

列	データ型	説明
KEEP_UNTIL	DATE	BACKUP コマンドの KEEP UNTIL TIME 句を指定した場合は、この列にバックアップが不要になる直前の日付が表示されます。列が NULL で、KEEP_OPTIONS が NULL でない場合、バックアップが不要とされることはありません。
KEEP_OPTIONS	VARCHAR2 (11)	このバックアップ・セットに指定された KEEP オプション。可能な値は、NOLOGS、BACKUP_LOGS、LOGS および NULL です。NOLOGS は、データベースがマウントされたときに一貫性のあるバックアップが作成されたことを示します。BACKUP_LOGS は、オープン・モードでバックアップが作成されたことを示します。したがって、一貫性のある状態にするには、アーカイブ・ログ・バックアップを適用する必要があります。LOGS は、LOGS キーワードを使用して長期間バックアップが作成されたことを示します。現在、LOGS キーワードは使用不可です。NULL は、このバックアップには KEEP オプションが指定されておらず、保存方針に基づいて不要とされることを示します。
DEVICE_TYPE	VARCHAR2 (255)	バックアップが格納されるデバイス・タイプ。バックアップ・セットが1つ以上のタイプのデバイスに格納されている場合（たとえば、バックアップをディスク上に作成し、現在もディスクにあるバックアップ・セットを BACKUP BACKUPSET を使用してテープにもバックアップした場合）、この列にはアスタリスク (*) が含まれます。値は DISK または SBT_TAPE です。
COMPRESSED	VARCHAR2 (3)	バックアップ・セットの作成に Recovery Manager のバイナリ圧縮を使用した場合は YES。それ以外の場合は NO。
NUM_COPIES	NUMBER	バックアップ中に作成されたこのバックアップ・セットの同一コピーの数（多重化が使用された場合など）。
OUTPUT_BYTES	NUMBER	このジョブで生成されたすべての出力ピースの合計サイズ。
ORIGINAL_INPUT_BYTES	NUMBER	このジョブでバックアップされたすべての入力ファイルの合計サイズ。
COMPRESSION_RATIO	NUMBER	このバックアップの圧縮率。
STATUS	CHAR (1)	バックアップ・セットの状態。このビューには使用可能なバックアップ・セットのみが反映されるため、常に A（すべてのバックアップ・ピースが使用可能）になります。
ORIGINAL_INPRATE_BYTES	NUMBER	バックアップ・セットが最初に作成されたときの毎秒の読取りバイト数。
OUTPUT_RATE_BYTES	NUMBER	バックアップ・セットが最初に作成されたときの毎秒の書込みバイト数。
ORIGINAL_INPUT_BYTES_DISPLAY	VARCHAR2 (4000)	ORIGINAL_INPUT_BYTES と値は同じですが、ユーザーが表示可能な書式（798.01M または 5.25G など）に変換されます。
OUTPUT_BYTES_DISPLAY	VARCHAR2 (4000)	OUTPUT_BYTES と値は同じですが、ユーザーが表示可能な書式（798.01M または 5.25G など）に変換されます。
ORIGINAL_INPRATE_BYTES_DISPLAY	VARCHAR2 (4000)	ORIGINAL_INPRATE_BYTES と値は同じですが、ユーザーが表示可能な書式（798.01M または 5.25G など）に変換されます。
OUTPUT_RATE_BYTES_DISPLAY	VARCHAR2 (4000)	OUTPUT_RATE_BYTES と値は同じですが、ユーザーが表示可能な書式（798.01M または 5.25G など）に変換されます。
TIME_TAKEN_DISPLAY	VARCHAR2 (4000)	ELAPSED_SECONDS と値は同じですが、ユーザーが表示可能な書式（時間、分および秒）に変換されます。
ENCRYPTED	VARCHAR2 (3)	バックアップ・ピースが暗号化されている（YES）か、暗号化されていない（NO）かを示します。
BACKED_BY_OS	VARCHAR2 (3)	バックアップ・ピースが Oracle Secure Backup にバックアップされている（YES）か、バックアップされていない（NO）かを示します。

## RC\_BACKUP\_SET\_SUMMARY

RC\_BACKUP\_SET\_SUMMARY ビューは、リカバリ・カタログに登録されている、各データベースで使用可能なバックアップ・セットに関する集計情報を表示します。

このビューは、主に Enterprise Manager で内部的に使用します。

列	データ型	説明
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカネーションの DB_NAME。
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
NUM_BACKUPSETS	NUMBER	リカバリ・カタログに登録されている、このデータベースで使用可能なバックアップ・セットの合計数。
OLDEST_BACKUP_TIME	DATE	リカバリ・カタログに登録されている、このデータベースで使用可能な最も古いバックアップ・セットの作成時刻。
NEWEST_BACKUP_TIME	DATE	リカバリ・カタログに登録されている、このデータベースで使用可能な最新のバックアップ・セットの作成時刻。
OUTPUT_BYTES	NUMBER	リカバリ・カタログに登録されている、このデータベースで使用可能なすべてのバックアップ・セットのすべてのバックアップ・ピースの合計サイズ。
ORIGINAL_INPUT_BYTES	NUMBER	リカバリ・カタログに登録されている、このデータベースで使用可能なすべてのバックアップ・セットのすべての入力ファイルの合計サイズ。
ORIGINAL_INPRATE_BYTES	NUMBER	リカバリ・カタログに登録されている、このデータベースで使用可能なすべてのバックアップ・セット作成の平均入力速度 (バイト)。
OUTPUT_RATE_BYTES	NUMBER	リカバリ・カタログに登録されている、このデータベースで使用可能なすべてのバックアップ・セット作成の平均出力速度 (バイト)。
COMPRESSION_RATIO	NUMBER	リカバリ・カタログに登録されている、このデータベースで使用可能なバックアップ・セットの集計圧縮率。
ORIGINAL_INPUT_BYTES_DISPLAY	VARCHAR2 (4000)	リカバリ・カタログに記録されている、このデータベースで使用可能なすべてのバックアップ・セットに格納されているすべての入力ファイルの合計サイズ。
OUTPUT_BYTES_DISPLAY	VARCHAR2 (4000)	OUTPUT_BYTES と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。
ORIGINAL_INPRATE_BYTES_DISPLAY	VARCHAR2 (4000)	ORIGINAL_INPRATE_BYTES と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。
OUTPUT_RATE_BYTES_DISPLAY	VARCHAR2 (4000)	OUTPUT_RATE_BYTES と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。

## RC\_BACKUP\_SPFILE

このビューは、バックアップ・セット内のサーバー・パラメータ・ファイルに関する情報をリストします。

列	データ型	説明
DB_KEY	NUMBER	ターゲット・データベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
BSF_KEY	NUMBER	リカバリ・カタログ内にあるサーバー・パラメータ・ファイルの主キー。Recovery Manager がリカバリ・カタログに接続している間に LIST コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	NUMBER	V\$BACKUP_SPFILE から伝播したレコード識別子。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
STAMP	NUMBER	V\$BACKUP_SPFILE からのスタンプ。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
BS_KEY	NUMBER	リカバリ・カタログ内でこのレコードが属しているバックアップ・セットの主キー。この列を使用して、RC_BACKUP_SET との結合を構成します。
SET_STAMP	NUMBER	V\$BACKUP_SET からの SET_STAMP の値。SET_STAMP と SET_COUNT は、ターゲット・データベース制御ファイル内でこのレコードが属しているバックアップ・セットを一意に識別する連結主キーを構成します。
SET_COUNT	NUMBER	V\$BACKUP_SET からの SET_COUNT の値。SET_STAMP と SET_COUNT は、ターゲット・データベース制御ファイル内でこのレコードが属しているバックアップ・セットを一意に識別する連結主キーを構成します。
MODIFICATION_TIME	DATE	サーバー・パラメータ・ファイルの最終変更時刻。
STATUS	VARCHAR2 (1)	バックアップ・セットの状態。A (すべてのバックアップ・ピースが使用可能)、D (すべてのバックアップ・ピースが削除済)、O (一部のバックアップ・ピースを除いて使用不可能なため、バックアップ・セットは使用不可)。
BS_RECID	NUMBER	V\$BACKUP_SET からの RECID の値。
BS_STAMP	NUMBER	V\$BACKUP_SET からの STAMP の値。BS_STAMP が SET_STAMP とは異なることに注意してください。BS_STAMP は制御ファイル内で作成された時点のバックアップ・セット・レコードのスタンプで、SET_STAMP は SET_COUNT と結合して一意の識別子となります。
COMPLETION_TIME	DATE	バックアップ・セットが完了した時刻。
BYTES	NUMBER	バックアップ・セットのサイズ (バイト単位)。
DB_UNIQUE_NAME	VARCHAR2 (30)	このレコードが属しているデータベースの DB_UNIQUE_NAME。

## RC\_BACKUP\_SPFILE\_DETAILS

RC\_BACKUP\_SPFILE\_DETAILS ビューは、リカバリ・カタログに登録されているデータベースの使用可能なデータファイルのバックアップに関する詳細情報を表示します。

このビューは、主に Enterprise Manager で内部的に使用します。

列	データ型	説明
SESSION_KEY	NUMBER	セッションの識別子。RC_RMAN_OUTPUT および RC_RMAN_BACKUP_JOB_DETAILS との結合に使用します。
SESSION_RECID	NUMBER	SESSION_STAMP とともに使用して、このバックアップ・ジョブの出力を RC_RMAN_OUTPUT から一意に識別します。
SESSION_STAMP	NUMBER	SESSION_RECID とともに使用して、このバックアップ・ジョブの出力を RC_RMAN_OUTPUT から一意に識別します。
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DB_NAME	VARCHAR2(8)	このレコードが属しているデータベース・インカネーションの DB_NAME。
BS_KEY	NUMBER	一意のバックアップ・セットの識別子。この列を使用して、RC_BACKUP_SET または RC_BACKUP_PIECE との結合を構成します。
SET_STAMP	NUMBER	セット STAMP。
SET_COUNT	NUMBER	セット・カウント。
MODIFICATION_TIME	DATE	変更時刻。
FILESIZE	NUMBER	バックアップされた SPFILE のサイズ (バイト)。
FILESIZE_DISPLAY	VARCHAR2(4000)	FILESIZE と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。



## RC\_BACKUP\_SPFILE\_SUMMARY

RC\_BACKUP\_SPFILE\_SUMMARY ビューは、リカバリ・カタログに登録されているデータベースに対するサーバー・パラメータ・ファイルのバックアップに関するサマリーを表示します。

このビューは、主に Enterprise Manager で内部的に使用します。

列	データ型	説明
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカネーションの DB_NAME。
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
NUM_FILES_BACKED	NUMBER	バックアップされたファイル数。
NUM_DISTINCT_FILES_BACKED	NUMBER	バックアップされた個別のファイル数 (異なる変更のタイムスタンプに基づく)。
MIN_MODIFICATION_TIME	DATE	このデータベースに対してバックアップされた任意の SPFILE の最も古い変更時刻。
MAX_MODIFICATION_TIME	DATE	このデータベースに対してバックアップされた任意の SPFILE の最新の変更時刻。
INPUT_BYTES	NUMBER	バックアップされた入力ファイルの合計バイト数。
INPUT_BYTES_DISPLAY	VARCHAR2 (4000)	INPUT_BYTES と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。

## RC\_CHECKPOINT

このビューは使用不可になりました。かわりに 4-66 ページの「[RC\\_RESYNC](#)」を参照してください。

## RC\_CONTROLFILE\_COPY

このビューは、ディスク上の制御ファイル・コピーに関する情報をリストします。  
V\$DATAFILE\_COPY ビューで制御ファイル・コピーを表示するには、ファイル番号 0 のデータファイルのコピーのレコードを使用します。

列	データ型	説明
DB_KEY	NUMBER	ターゲット・データベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	ターゲット・データベースのインカーネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカーネーションの DB_NAME。
CCF_KEY	NUMBER	リカバリ・カタログ内にある制御ファイル・コピーの主キー。Recovery Manager がリカバリ・カタログに接続している間に LIST コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	NUMBER	V\$DATAFILE_COPY からのレコード識別子。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
STAMP	NUMBER	V\$DATAFILE_COPY からの STAMP。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
NAME	VARCHAR2 (1024)	制御ファイル・コピーのファイル名。
TAG	VARCHAR2 (32)	制御ファイル・コピーのタグ。タグを使用しない場合は、NULL。
RESETLOGS_CHANGE#	NUMBER	レコード作成時における最新の RESETLOGS の SCN。
RESETLOGS_TIME	DATE	レコード作成時における最新の RESETLOGS のタイムスタンプ。
CHECKPOINT_CHANGE#	NUMBER	制御ファイルのチェックポイント SCN。
CHECKPOINT_TIME	DATE	制御ファイルのチェックポイント時刻。
CREATION_TIME	DATE	制御ファイルの作成時刻。
BLOCKS	NUMBER	制御ファイル内のブロック数。
BLOCK_SIZE	NUMBER	ブロック・サイズ (バイト)。
MIN_OFFR_RECID	NUMBER	内部使用のみ。
OLDEST_OFFLINE_RANGE	NUMBER	内部使用のみ。
COMPLETION_TIME	DATE	コピーを生成したときの時刻。
STATUS	VARCHAR2 (1)	コピーの状態。A (使用可能)、U (使用不可能)、X (期限切れ) または D (削除済)。
CONTROLFILE_TYPE	VARCHAR2 (1)	制御ファイル・コピーのタイプ。B (通常のコピー) または S (スタンバイ・コピー)。
KEEP	VARCHAR2 (3)	このコピーの保存方針が CONFIGURE RETENTION POLICY の値とは異なるかどうかを示します。可能な値は YES および NO です。
KEEP_UNTIL	DATE	COPY コマンドの KEEP UNTIL TIME 句を指定した場合は、この列にこのファイルが不要になる直前の日付が表示されます。列が NULL で、KEEP OPTIONS が NULL でなければ、ファイルが不要とされることはありません。

列	データ型	説明
KEEP_OPTIONS	VARCHAR2(11)	この制御ファイル・コピーに指定された KEEP オプション。可能な値は、NOLOGS、BACKUP_LOGS、LOGS および NULL です。NOLOGS は、データベースがマウントされたときに一貫性のあるバックアップが作成されたことを示します。BACKUP_LOGS は、オープン・モードでバックアップが作成されたことを示します。したがって、一貫性のある状態にするには、アーカイブ・ログ・バックアップを適用する必要があります。LOGS は、LOGS キーワードを使用して長期間バックアップが作成されたことを示します。現在、LOGS キーワードは使用不可です。NULL は、このバックアップには KEEP オプションが指定されておらず、保存方針に基づいて不要とされることを示します。
IS_RECOVERY_DEST_FILE	VARCHAR2(3)	このコピーはフラッシュ・リカバリ領域に格納されます。YES または NO。
RSR_KEY	NUMBER	このバックアップ・ピースを作成した RC_RMAN_STATUS の行に対する一意のキー。
SITE_KEY	NUMBER	このファイルと関連付けられている Data Guard データベースの主キー。Data Guard 環境にある各データベースの SITE_KEY の値は一意です。SITE_KEY を使用して RC_SITE ビューと結合すると、データベースの DB_UNIQUE_NAME を取得できます。

## RC\_COPY\_CORRUPTION

このビューは、データファイルのコピーの破損ブロック範囲をリストします。このビューは、V\$COPY\_CORRUPTION ビューに対応しています。

列	データ型	説明
DB_KEY	NUMBER	ターゲット・データベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	ターゲット・データベースのインカーネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカーネーションの DB_NAME。
RECID	NUMBER	V\$COPY_CORRUPTION からのレコード識別子。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
STAMP	NUMBER	V\$COPY_CORRUPTION から伝播したスタンプ。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
CDF_KEY	NUMBER	リカバリ・カタログ内のデータファイルのコピーの主キー。Recovery Manager がリカバリ・カタログに接続している間に LIST コマンドを発行すると、この値が出力の KEY 列に表示されます。この列を使用して、RC_DATAFILE_COPY との結合を構成します。
COPY_RECID	NUMBER	RC_DATAFILE_COPY からの RECID。この値は、制御ファイルから伝播されます。
COPY_STAMP	NUMBER	RC_DATAFILE_COPY からの STAMP。この値は、制御ファイルから伝播されます。
FILE#	NUMBER	データファイルの絶対ファイル番号。
CREATION_CHANGE#	NUMBER	このデータファイルの作成時の SCN。ファイル番号は再利用できるため、データベースの存続期間中に指定したファイルを一意に識別するには、FILE# および CREATION_CHANGE# の両方が必要です。
BLOCK#	NUMBER	ファイル内にある最初の破損ブロックのブロック番号。
BLOCKS	NUMBER	BLOCK# 以降に見えられた破損ブロックの数。
CORRUPTION_CHANGE#	NUMBER	メディア破損ブロックの場合、この値は 0 です。論理破損ブロックの場合、この値はこの破損範囲内のブロックの最小 SCN です。
MARKED_CORRUPT	VARCHAR2 (3)	データベース・サーバーでこの破損をこれまでに検出していない場合は YES、データベース・サーバーですでに認識済の場合は NO。
CORRUPTION_TYPE	VARCHAR2 (9)	RC_DATABASE_BLOCK_CORRUPTION.CORRUPTION_TYPE と同じです。

---

## RC\_DATABASE

このビューは、リカバリ・カタログに登録されているデータベースに関する情報を表示します。  
このビューは、V\$DATABASE ビューに対応しています。

列	データ型	説明
DB_KEY	NUMBER	データベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	現行のインカネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DBID	NUMBER	V\$DATABASE からのデータベースの一意の識別子。
NAME	VARCHAR2(8)	現行のインカネーションに対するデータベースの DB_NAME。
RESETLOGS_CHANGE#	NUMBER	現行のデータベース・インカネーションの RESETLOGS の SCN。
RESETLOGS_TIME	DATE	現行のデータベース・インカネーションの RESETLOGS のタイムスタンプ。

## RC\_DATABASE\_BLOCK\_CORRUPTION

このビューは、前回のバックアップ後に破損したデータベース・ブロックに関する情報を表示します。このビューは、V\$DATABASE\_BLOCK\_CORRUPTION ビューに対応しています。

列	データ型	説明
DB_KEY	NUMBER	データベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	現行のインカーネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
FILE#	NUMBER	データファイルの絶対ファイル番号。
BLOCK#	NUMBER	この破損ブロック範囲内の最初の破損ブロックのブロック番号。
BLOCKS	NUMBER	BLOCK# 以降に見えられた破損ブロックの数。
CORRUPTION_CHANGE#	NUMBER	メディア破損ブロックの場合、この値は 0 です。論理破損ブロックの場合、この値はこの破損範囲内のブロックの最小 SCN です。
CORRUPTION_TYPE	VARCHAR2 (9)	<p>データファイル内のブロック破損のタイプ。可能な値は次のとおりです。</p> <ul style="list-style-type: none"> <li>■ ALL ZERO。ディスク上のブロック・ヘッダーには 0 (ゼロ) のみが含まれています。ブロックにデータが入れられたことがない場合や、Oracle7 のファイル内のブロックの場合は、有効な可能性があります。バッファは、空のブロックに関する Oracle8 の標準に合わせて再フォーマットされます。</li> <li>■ FRACTURED。ブロック・ヘッダーは妥当であるように見えますが、ブロックの前後のバージョンが異なります。</li> <li>■ CHECKSUM。オプションのチェック値は、ブロックに自己一貫性が無いことを示します。チェック値が失敗した正確な原因は判断できませんが、ブロックの途中にあるセクターのバージョンが異なることが原因であると思われます。</li> <li>■ CORRUPT。ブロックが正しく識別されないか、データ・ブロックではありません (データ・ブロック・アドレスが欠落している場合など)。</li> <li>■ LOGICAL。論理的な破損ブロックの範囲であると指定します。CORRUPTION_CHANGE# は 0 以外の値になります。</li> </ul>

## RC\_DATABASE\_INCARNATION

このビューは、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・インカーネーションに関する情報をリストします。Oracle では、管理者がデータベースを RESETLOGS オプションでオープンすると、必ず新しいインカーネーションが作成されます。V\$DATABASE ビューには、現行と直前のインカーネーションに関するレコードも含まれます。

列	データ型	説明
DB_KEY	NUMBER	データベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBID	NUMBER	データベースの一意の識別子。
DBINC_KEY	NUMBER	インカーネーションに対する主キー。
NAME	VARCHAR2(8)	RESETLOGS 時点でのデータベースの DB_NAME。値は、ターゲット・データベースを Recovery Manager に登録する前に 1 回以上 RESETLOGS を実行している場合は UNKNOWN です。これは、Recovery Manager では、RESETLOGS までは DB_NAME が認識されないためです。
RESETLOGS_CHANGE#	NUMBER	このインカーネーションを作成した RESETLOGS の SCN。
RESETLOGS_TIME	DATE	このインカーネーションを作成した RESETLOGS のタイムスタンプ。
CURRENT_INCARNATION	VARCHAR2(3)	現行のインカーネーションの場合は YES、それ以外の場合は NO。
PARENT_DBINC_KEY	NUMBER	このデータベースに対する前のインカーネーションの DBINC_KEY。データベースの最初に記録されたインカーネーションの場合、値は NULL になります。
PRIOR_RESETLOGS_CHANGE#	NUMBER	このインカーネーションの親を作成した RESETLOGS 操作の SCN。
PRIOR_RESETLOGS_TIME	DATE	このインカーネーションの親を作成した RESETLOGS 操作のタイムスタンプ。
STATUS	VARCHAR2(8)	このインカーネーションが現行のデータベース・インカーネーションの場合は CURRENT。現行のインカーネーションの直接の祖先であり、現行以外のインカーネーションの場合は PARENT。現行のインカーネーションの直接の祖先ではない、現行以外のインカーネーションの場合は ORPHAN。



## RC\_DATAFILE

このビューは、リカバリ・カタログに登録されているすべてのデータファイルに関する情報をリストします。このビューは、V\$DATAFILE ビューに対応しています。データファイルは、その表領域が削除された場合、削除されたものとして表示されます。

列	データ型	説明
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	ターゲット・データベースのインカーネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカーネーションの DB_NAME。
TS#	NUMBER	データファイルが属している表領域の番号。表領域を削除し、再作成した場合は、同じインカーネーションに複数の TS# が存在します。
TABLESPACE_NAME	VARCHAR2 (30)	表領域名。表領域を削除し、再作成した場合は、同じインカーネーションにその名前が複数存在している可能性があります。
FILE#	NUMBER	データファイルの絶対ファイル番号。表領域を削除し、再作成した場合は、同じインカーネーションに複数の同一データファイル番号が存在しません。
CREATION_CHANGE#	NUMBER	データファイル作成時の SCN。
CREATION_TIME	DATE	データファイル作成の時刻。
DROP_CHANGE#	NUMBER	データファイルを削除したときに記録された SCN。同じファイル番号を持つ新規のデータファイルが発見された場合、DROP_CHANGE# はそのデータファイルの CREATION_CHANGE# に設定されます。それ以外の場合、値は RC_CHECKPOINT.CKP_SCN に設定されます。
DROP_TIME	DATE	データファイルを削除したときの時刻。同じファイル番号を持つ新規のデータファイルが発見された場合、DROP_TIME はそのデータファイルの CREATION_TIME に設定されます。他の場合は、値は RC_CHECKPOINT.CKP_TIME に設定されます。
BYTES	NUMBER	データファイルのサイズ (バイト)。
BLOCKS	NUMBER	データファイルのサイズ (ブロック数)。
BLOCK_SIZE	NUMBER	データ・ブロック・サイズ (バイト)。
NAME	VARCHAR2 (1024)	データファイルのファイル名。
STOP_CHANGE#	NUMBER	オフラインまたは読取り専用のデータファイルの場合は、このファイルに適用される同一以上の SCN で REDO ストリームに変更のない SCN の値。
STOP_TIME	DATE	オフラインの通常または読取り専用のデータファイルの場合、このデータファイルに適用される REDO ストリーム内で変更のない最後の時刻。
READ_ONLY	NUMBER	ファイルが読取り専用の場合は 1、それ以外の場合は 0。
RFILE#	NUMBER	表領域内でのこのデータファイルの相対ファイル番号。
INCLUDED_IN_DATABASE_BACKUP	VARCHAR2 (3)	この表領域がデータベース全体のバックアップに含まれるかどうかを YES または NO で示します。NO 値が発生するのは、このデータファイルを所有する表領域に対して CONFIGURE EXCLUDE が実行された場合のみです。
AUX_NAME	VARCHAR2 (1024)	CONFIGURE AUXNAME で設定されたデータファイルの補助名を示します。
ENCRYPT_IN_BACKUP	VARCHAR2 (3)	このデータファイルが、バックアップ時に透過的に暗号化されるように構成されている場合は YES、それ以外の場合は NULL。

列	データ型	説明
SITE_KEY	NUMBER	このファイルと関連付けられている Data Guard データベースの主キー。Data Guard 環境にある各データベースの SITE_KEY の値は一意です。SITE_KEY を使用して RC_SITE ビューと結合すると、データベースの DB_UNIQUE_NAME を取得できます。
DB_UNIQUE_NAME	VARCHAR2 (512)	このレコードが属しているデータベース・インカネーションの DB_UNIQUE_NAME。Data Guard 環境のすべてのデータベースで同一の DBID が共有されていますが、DB_UNIQUE_NAME の値は異なります。特定のファイルに対するデータベース名が不明の場合、この列の値は NULL です。たとえば、Oracle Database 11g より前のバージョンの Recovery Manager で管理されているデータベースの行は NULL です。
FOREIGN_DBID	NUMBER	このデータファイルの外部 DBID。このファイルが外部データベース・ファイルでない場合、値は 0 です。
FOREIGN_CREATION_CHANGE#	NUMBER	外部データファイル作成時の SCN。このファイルが外部データベース・ファイルでない場合、値は 0 です。
FOREIGN_CREATION_TIME	DATE	外部データファイル作成時の時刻。このファイルが外部データベース・ファイルでない場合、値は 0 です。
PLUGGED_READONLY	VARCHAR2 (3)	トランスポートされた読取り専用外部ファイルである場合は YES、そうでない場合は NO。
PLUGIN_CHANGE#	NUMBER	外部データファイルがデータベースにトランスポートされたときの SCN。このファイルが外部データベース・ファイルでない場合、値は 0 です。
PLUGIN_RESETLOGS_CHANGE#	NUMBER	この外部ファイルがトランスポートされたインカネーションに対する RESETLOGS 操作の SCN。このファイルが外部データベース・ファイルでない場合、値は 0 です。
PLUGIN_RESETLOGS_TIME	DATE	この外部ファイルがトランスポートされたインカネーションに対する RESETLOGS 操作の時刻。このファイルが外部データベース・ファイルでない場合、値は 0 です。

## RC\_DATAFILE\_COPY

このビューは、ディスク上のデータファイルのコピーに関する情報をリストします。このビューは、V\$DATAFILE\_COPY ビューに対応しています。

列	データ型	説明
DB_KEY	NUMBER	ターゲット・データベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	ターゲット・データベースのインカーネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカーネーションの DB_NAME。
CDF_KEY	NUMBER	リカバリ・カタログ内のデータファイルのコピーの主キー。Recovery Manager がリカバリ・カタログに接続している間に LIST コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	NUMBER	V\$DATAFILE_COPY からのデータファイルのコピーのレコード。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
STAMP	NUMBER	V\$DATAFILE_COPY からのデータファイルのコピーの STAMP。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
NAME	VARCHAR2 (1024)	データファイルのコピーのファイル名。
TAG	VARCHAR2 (32)	データファイルのコピーのタグ。
FILE#	NUMBER	データファイルの絶対ファイル番号。
CREATION_CHANGE#	NUMBER	データファイルの作成時の SCN。
CREATION_TIME	DATE	データファイル作成時のタイムスタンプ。
RESETLOGS_CHANGE#	NUMBER	データファイル作成時における最新の RESETLOGS の SCN。
RESETLOGS_TIME	DATE	データファイル・ヘッダー内の最新 RESETLOGS のタイムスタンプ。
INCREMENTAL_LEVEL	NUMBER	コピーの増分レベル。0 または NULL。
CHECKPOINT_CHANGE#	NUMBER	最新データファイル・チェックポイント SCN。
CHECKPOINT_TIME	DATE	最新のデータファイル・チェックポイント時刻。
ABSOLUTE_FUZZY_CHANGE#	NUMBER	認識されている場合は、ファイルのいずれかのブロック内にある一番大きい値の SCN。ファイルがファジーにならないように、リカバリは少なくともこの SCN まで進行させる必要があります。
RECOVERY_FUZZY_CHANGE#	NUMBER	ファイルをファジーにしないために、リカバリ操作が到達する必要がある SCN。NULL でない場合は、このファイルでデータベースをオープンする前に、少なくとも指定された SCN までのリカバリが必要です。
RECOVERY_FUZZY_TIME	DATE	RECOVERY_FUZZY_CHANGE# に対応付けられている時刻。
ONLINE_FUZZY	VARCHAR2 (3)	YES または NO。YES に設定されている場合、このコピーはインスタンス障害または OFFLINE IMMEDIATE の後に作成されたものです（またはデータベースがオープン状態で不適切に作成されたコピーです）。ファイルに一貫性を持たせるには、次のクラッシュ・リカバリ・マーカまでのすべての REDO を適用するリカバリが必要です。
BACKUP_FUZZY	VARCHAR2 (3)	YES または NO。YES に設定されている場合、このコピーは BEGIN BACKUP/END BACKUP により作成されています。このコピーに一貫性を持たせるには、ALTER TABLESPACE END BACKUP コマンドを使用するときに REDO ストリームに置かれたマーカまでの、すべての REDO を適用するリカバリ・プロセスが必要です。
BLOCKS	NUMBER	データファイルのコピーにあるブロックの数（およびコピー作成時のデータファイルのサイズ）。

列	データ型	説明
BLOCK_SIZE	NUMBER	ブロック・サイズ (バイト)。
COMPLETION_TIME	DATE	コピーが完了した時刻。
STATUS	VARCHAR2 (1)	コピーの状態。A (使用可能)、U (使用不可能)、X (期限切れ) または D (削除済)。
KEEP	VARCHAR2 (3)	このコピーの保存方針が CONFIGURE RETENTION POLICY の値とは異なるかどうかを示します。可能な値は YES および NO です。
KEEP_UNTIL	DATE	COPY コマンドの KEEP UNTIL TIME 句を指定した場合は、この列にこのデータファイルのコピーが不要になる直前の日付が表示されます。列が NULL で、KEEP OPTIONS が NULL でなければ、コピーが不要とされることはありません。
KEEP_OPTIONS	VARCHAR2 (11)	このデータファイルのコピーに指定された KEEP オプション。可能な値は、NOLOGS、BACKUP_LOGS、LOGS および NULL です。NOLOGS は、データベースがマウントされたときに一貫性のあるバックアップが作成されたことを示します。BACKUP_LOGS は、オープン・モードでバックアップが作成されたことを示します。したがって、一貫性のある状態にするには、アーカイブ・ログ・バックアップを適用する必要があります。LOGS は、LOGS キーワードを使用して長期間バックアップが作成されたことを示します。現在、LOGS キーワードは使用不可です。NULL は、このバックアップには KEEP オプションが指定されておらず、保存方針に基づいて不要とされることを示します。
SCANNED	VARCHAR2 (3)	<p>Recovery Manager がこのファイルをスキャンしたかどうか (YES または NO)。YES の場合、このコピーは、Recovery Manager の COPY または RESTORE コマンドなど、ファイル内の各ブロックを検査したサーバー・プロセスによって作成されたものです。NO の場合、Recovery Manager 以外が生成したイメージ・コピーを検査したり、プロキシ・コピーを使用してリストアするときとは異なり、ファイル内の各ブロックを検査していません。</p> <p>Recovery Manager は、データファイルのコピーを作成またはリストアするときに、ファイル内で破損ブロックを検出すると、V\$DATABASE_BLOCK_CORRUPTION ビューと RC_DATABASE_BLOCK_CORRUPTION ビューに行を追加します。ファイル全体のスキャンを完了しても、このコピーに関する破損レコードがない場合は、ファイルに破損ブロックが存在しないことを意味します。ファイルをスキャンしておらず、破損レコードがない場合は、破損ブロックが存在する可能性も存在しない可能性もあることを意味します。</p>
IS_RECOVERY_DEST_FILE	VARCHAR2 (3)	このデータファイルのコピーはフラッシュ・リカバリ領域に格納されません。YES または NO。
RSR_KEY	NUMBER	このバックアップ・ピースを作成した RC_RMAN_STATUS の行に対する一意のキー。
MARKED_CORRUPT	NUMBER	このバックアップの処理中に、破損していることが検出されたブロックの数。これらのブロックは、破損検出済ブロックとして再フォーマットされてイメージ・コピーに出力されています。
SITE_KEY	NUMBER	このファイルと関連付けられている Data Guard データベースの主キー。Data Guard 環境にある各データベースの SITE_KEY の値は一意です。SITE_KEY を使用して RC_SITE ビューと結合すると、データベースの DB_UNIQUE_NAME を取得できます。
DB_UNIQUE_NAME	VARCHAR2 (512)	このレコードが属しているデータベース・インカンションの DB_UNIQUE_NAME。Data Guard 環境のすべてのデータベースで同一の DBID が共有されていますが、DB_UNIQUE_NAME の値は異なります。特定のファイルに対するデータベース名が不明の場合、この列の値は NULL です。たとえば、Oracle Database 11g より前のバージョンの Recovery Manager で管理されているデータベースの行は NULL です。
FOREIGN_DBID	NUMBER	このデータファイルのトランスポート元であるデータベースの外部 DBID。バックアップされたファイルが外部データベース・ファイルでない場合、値は 0 になります。
PLUGGED_READONLY	VARCHAR2 (3)	トランスポートされた読取り専用外部ファイルのコピーである場合は YES、そうでない場合は NO。

列	データ型	説明
PLUGIN_CHANGE#	NUMBER	外部データファイルがデータベースにトランスポートされたときの SCN。このファイルが外部データベース・ファイルでない場合、値は 0 です。
PLUGIN_RESETLOGS_CHANGE#	NUMBER	この外部ファイルがトランスポートされたインカネーションに対する RESETLOGS 操作の SCN。このファイルが外部データベース・ファイルでない場合、値は 0 です。
PLUGIN_RESETLOGS_TIME	DATE	この外部ファイルがトランスポートされたインカネーションに対する RESETLOGS 操作の時刻。このファイルが外部データベース・ファイルでない場合、値は 0 です。

## RC\_LOG\_HISTORY

このビューは、オンライン REDO ログについての履歴情報をリストします。Oracle がオンライン REDO ログから切り替わるたびに、Recovery Manager はカタログの再同期化中に新規の行を追加します。このカタログ・ビューは、V\$LOG\_HISTORY ビューに対応しています。

列	データ型	説明
DB_KEY	NUMBER	ターゲット・データベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	ターゲット・データベースのインカーネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカーネーションの DB_NAME。
RECID	NUMBER	V\$LOG_HISTORY からの REDO ログの履歴の RECID。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
STAMP	NUMBER	V\$LOG_HISTORY からの REDO ログの履歴の STAMP。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
THREAD#	NUMBER	オンライン REDO ログのスレッド番号。
SEQUENCE#	NUMBER	REDO ログのログ順序番号。
FIRST_CHANGE#	NUMBER	REDO ログへの切替時に生成された SCN。
FIRST_TIME	DATE	REDO ログに切り替えたときのタイムスタンプ。
NEXT_CHANGE#	NUMBER	スレッド内にある次の REDO ログの最初の SCN。
CLEARED	VARCHAR2 (3)	REDO ログが  ALTER DATABASE CLEAR LOGFILE 文で消去された場合は YES、それ以外の場合は NULL。この文を使用すると、ログを最初にアーカイブしなくても初期化できます。

## RC\_OFFLINE\_RANGE

このビューはデータファイルのオフライン範囲をリストします。このビューは、V\$OFFLINE\_RANGE ビューに対応しています。

データファイルにオフライン範囲が作成されるのは、最初にその表領域が NORMAL モードによるオフラインまたは読取り専用に変更され、その後にオンラインまたは読取り / 書込み可能に変更されたときです。データファイル自体がオフラインに変更された場合、あるいは表領域が IMMEDIATE モードでオフラインに変更される場合、オフライン範囲は作成されません。

列	データ型	説明
DB_KEY	NUMBER	ターゲット・データベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	ターゲット・データベースのインカーネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカーネーションの DB_NAME。
RECID	NUMBER	V\$OFFLINE_RANGE からのオフライン範囲のレコード識別子。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
STAMP	NUMBER	V\$OFFLINE_RANGE からのオフライン範囲のスタンプ。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
FILE#	NUMBER	データファイルの絶対ファイル番号。
CREATION_CHANGE#	NUMBER	データファイル作成時の SCN。
OFFLINE_CHANGE#	NUMBER	データファイルをオフラインにしたときに記録された SCN。
ONLINE_CHANGE#	NUMBER	オンライン・チェックポイント SCN。
ONLINE_TIME	DATE	オンライン・チェックポイント時刻。
CF_CREATE_TIME	DATE	制御ファイル作成の時刻。

## RC\_PROXY\_ARCHIVEDLOG

このビューでは、プロキシ・コピー機能を使用して作成したアーカイブ・ログのバックアップを説明します。このビューは、V\$PROXY\_ARCHIVEDLOG ビューに対応しています。

プロキシ・コピーでは、メディア・マネージャがデータのバックアップとリストアの操作を行います。各行で1つの制御ファイルの1つのバックアップを説明します。

列	データ型	説明
DB_KEY	NUMBER	ターゲット・データベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	ターゲット・データベースのインカーネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカーネーションの DB_NAME。
XAL_KEY	NUMBER	リカバリ・カタログのプロキシ・コピーの主キー。Recovery Manager がリカバリ・カタログに接続している間に LIST コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	NUMBER	V\$PROXY_ARCHIVEDLOG からのプロキシ・コピー・レコードの識別子。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
STAMP	NUMBER	V\$PROXY_ARCHIVEDLOG からのプロキシ・コピーの STAMP。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
TAG	VARCHAR2 (32)	プロキシ・コピーのタグ。
DEVICE_TYPE	VARCHAR2 (255)	プロキシ・コピーを格納するメディア・デバイスのタイプ。
HANDLE	VARCHAR2 (1024)	プロキシ・コピーの名前またはハンドル。Recovery Manager は、この値を、アーカイブ REDO ログのプロキシ・コピーを作成したメディア・マネージャに渡します。
COMMENTS	VARCHAR2 (255)	プロキシ・コピーについてのコメント。
MEDIA	VARCHAR2 (80)	このバックアップを作成したメディア・マネージャについての詳細コメント。
MEDIA_POOL	NUMBER	プロキシ・コピーを格納するメディア・プールの数。
STATUS	VARCHAR2 (1)	バックアップ・セットの状態。A (使用可能)、U (使用不可能)、X (期限切れ) または D (削除済)。
THREAD#	NUMBER	REDO スレッドの番号。
SEQUENCE#	NUMBER	ログ順序番号。
RESETLOGS_CHANGE#	NUMBER	このアーカイブ・ログが属しているデータベース・インカーネーションの RESETLOGS の SCN。
RESETLOGS_TIME	DATE	このアーカイブ・ログが属しているデータベース・インカーネーションの RESETLOGS のタイムスタンプ。
FIRST_CHANGE#	NUMBER	この REDO ログの最初の SCN。
FIRST_TIME	DATE	REDO ログに切り替えた時刻。
NEXT_CHANGE#	NUMBER	スレッド内にある次の REDO ログの最初の SCN。
NEXT_TIME	DATE	スレッド内にある次の REDO ログの最初のタイムスタンプ。
BLOCKS	NUMBER	このアーカイブ REDO ログのサイズ (オペレーティング・システム・ブロック数)。
BLOCK_SIZE	NUMBER	コピーのブロック・サイズ (バイト)。
DEVICE_TYPE	VARCHAR2 (255)	順次メディア・デバイスのタイプ。
START_TIME	DATE	プロキシ・コピーの開始時刻。



列	データ型	説明
COMPLETION_TIME	DATE	プロキシ・コピーの完了時刻。
ELAPSED_SECONDS	NUMBER	プロキシ・コピーの実行時間。
RSR_KEY	NUMBER	Recovery Manager のステータス・レコードの主キー。この列を使用して、RC_RMAN_STATUS と結合します。
TERMINAL	VARCHAR2 (3)	このレコードが、V\$ARCHIVED_LOG に定義されたターミナル・アーカイブ REDO ログに対応している場合は YES。
KEEP	VARCHAR2 (3)	このプロキシ・コピーの保存方針が CONFIGURE RETENTION POLICY の値とは異なるかどうかを示します。可能な値は YES および NO です。
KEEP_OPTIONS	VARCHAR2 (11)	このプロキシ・コピーに指定された KEEP オプション。可能な値は、NOLOGS、BACKUP_LOGS、LOGS および NULL です。NOLOGS は、データベースがマウントされたときに一貫性のあるバックアップが作成されたことを示します。BACKUP_LOGS は、オープン・モードでバックアップが作成されたことを示します。したがって、一貫性のある状態にするには、アーカイブ・ログ・バックアップを適用する必要があります。LOGS は、LOGS キーワードを使用して長期間バックアップが作成されたことを示します。現在、LOGS キーワードは使用不可です。NULL は、このバックアップには KEEP オプションが指定されておらず、保存方針に基づいて不要とされることを示します。
KEEP_UNTIL	DATE	BACKUP コマンドの KEEP UNTIL TIME 句を指定した場合は、この列にプロキシ・コピーが不要になる直前の日付が表示されます。列が NULL で KEEP_OPTIONS が NULL でなければ、プロキシ・コピーが不要とされることはありません。
SITE_KEY	NUMBER	このファイルと関連付けられている Data Guard データベースの主キー。Data Guard 環境にある各データベースの SITE_KEY の値は一意です。SITE_KEY を使用して RC_SITE ビューと結合すると、データベースの DB_UNIQUE_NAME を取得できます。

## RC\_PROXY\_ARCHIVELOG\_DETAILS

RC\_PROXY\_ARCHIVELOG\_DETAILS ビューは、リカバリ・カタログに登録されている各データベースに対するアーカイブ REDO ログのプロキシ・コピーのバックアップに関する詳細情報を表示します。

このビューは、リカバリ・カタログに登録されている各データベースに対して1つのレコードを表示します。そのため、登録されているデータベースが1つのみの場合、このビューで表示される行は、アーカイブ REDO ログのプロキシ・コピーの実行回数に関係なく1行です。このビューは、主に Enterprise Manager で内部的に使用します。

列	データ型	説明
SESSION_KEY	NUMBER	セッションの識別子。RC_RMAN_OUTPUT および RC_RMAN_BACKUP_JOB_DETAILS との結合に使用します。
SESSION_RECID	NUMBER	SESSION_STAMP とともに使用して、このバックアップ・ジョブの出力を RC_RMAN_OUTPUT から一意に識別します。
SESSION_STAMP	NUMBER	SESSION_RECID とともに使用して、このバックアップ・ジョブの出力を RC_RMAN_OUTPUT から一意に識別します。
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカネーションの DB_NAME。
COPY_KEY	NUMBER	このプロキシ・コピーに対する一意の識別子。
THREAD#	NUMBER	バックアップされたアーカイブ REDO ログ・ファイルの REDO スレッド番号。
SEQUENCE#	NUMBER	バックアップされたアーカイブ REDO ログ・ファイルのログの順序番号。
RESETLOGS_CHANGE#	NUMBER	バックアップされたアーカイブ REDO ログ・ファイルのデータベースのインカネーションに対する OPEN RESETLOGS のチェックポイント SCN。
RESETLOGS_TIME	DATE	RESETLOGS_CHANGE# に対応する時刻。
HANDLE	VARCHAR2 (1024)	プロキシ・コピーのファイル名。
MEDIA	VARCHAR2 (80)	このバックアップを作成したメディア・マネージャについての詳細コメント。
MEDIA_POOL	NUMBER	バックアップを格納するメディア・プールの数。
TAG	VARCHAR2 (32)	このバックアップに指定されたタグ。
FIRST_CHANGE#	NUMBER	アーカイブ REDO ログ・ファイルに含まれている最初の変更 SCN。
NEXT_CHANGE#	NUMBER	このアーカイブ REDO ログ・ファイルの後で次に変更された SCN。
FIRST_TIME	DATE	FIRST_CHANGE# に対応する時刻。
NEXT_TIME	DATE	NEXT_CHANGE# に対応する時刻。
OUTPUT_BYTES	NUMBER	このジョブで生成されたすべての出力ピースの合計サイズ。
COMPLETION_TIME	DATE	ジョブの完了時刻。
OUTPUT_BYTES_DISPLAY	VARCHAR2 (4000)	OUTPUT_BYTES と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。

## RC\_PROXY\_ARCHIVELOG\_SUMMARY

RC\_PROXY\_ARCHIVELOG\_SUMMARY ビューには、アーカイブ REDO ログのプロキシ・コピーのバックアップのサマリーが含まれています。

このビューは、主に Enterprise Manager で内部的に使用します。

列	データ型	説明
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカンネーションの DB_NAME。
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
NUM_FILES_BACKED	NUMBER	バックアップされたアーカイブ REDO ログ・ファイルの合計数。
NUM_DISTINCT_FILES_BACKED	NUMBER	バックアップされた個別アーカイブ REDO ログ・ファイルの数。
MIN_FIRST_CHANGE#	NUMBER	このサマリー内の任意の REDO ログ・ファイルの最初の SCN の最小値。
MAX_NEXT_CHANGE#	NUMBER	このサマリー内の任意の REDO ログ・ファイルの NEXT_CHANGE# SCN の最大値。
MIN_FIRST_TIME	DATE	任意の REDO ログの最初の変更時刻の最小値。MAX_NEXT_TIME とともに使用して REDO の範囲を構成します。
MAX_NEXT_TIME	DATE	このセッション内の任意の REDO ログの後で次に変更された時刻の最大値。
OUTPUT_BYTES	NUMBER	このジョブで生成されたすべての出力ピースの合計サイズ。
OUTPUT_BYTES_DISPLAY	VARCHAR2 (4000)	OUTPUT_BYTES と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。

## RC\_PROXY\_CONTROLFILE

このビューは、プロキシ・コピー機能を使用して作成した制御ファイルのバックアップを説明します。このビューは、V\$PROXY\_DATAFILE ビューに対応しています。

プロキシ・コピーでは、メディア・マネージャがデータのバックアップとリストアの操作を行います。各行で1つの制御ファイルの1つのバックアップを説明します。

列	データ型	説明
DB_KEY	NUMBER	ターゲット・データベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	ターゲット・データベースのインカーネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2(8)	このレコードが属しているデータベース・インカーネーションの DB_NAME。
XCF_KEY	NUMBER	リカバリ・カタログのプロキシ・コピーの主キー。Recovery Manager がリカバリ・カタログに接続している間に LIST コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	NUMBER	V\$PROXY_DATAFILE からのプロキシ・コピー・レコードの識別子。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
STAMP	NUMBER	V\$PROXY_DATAFILE からのプロキシ・コピーの STAMP。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
TAG	VARCHAR2(32)	プロキシ・コピーのタグ。
RESETLOGS_CHANGE#	NUMBER	このデータファイルが属しているデータベース・インカーネーションの RESETLOGS の SCN。
RESETLOGS_TIME	DATE	このデータファイルが属しているデータベース・インカーネーションの RESETLOGS のタイムスタンプ。
CHECKPOINT_CHANGE#	NUMBER	このコピーの作成時におけるデータファイル・チェックポイント SCN。
CHECKPOINT_TIME	DATE	このコピーの作成時におけるデータファイル・チェックポイント時刻。
CREATION_TIME	DATE	制御ファイルの作成時刻。
BLOCK_SIZE	NUMBER	コピーのブロック・サイズ (バイト)。
BLOCKS	NUMBER	コピー内のブロック数。
MIN_OFFR_RECID	NUMBER	内部使用のみ。
OLDEST_OFFLINE_RANGE	NUMBER	内部使用のみ。
DEVICE_TYPE	VARCHAR2(255)	順次メディア・デバイスのタイプ。
HANDLE	VARCHAR2(1024)	プロキシ・コピーの名前またはハンドル。Recovery Manager は、この値を、ファイルを識別するオペレーティング・システム固有のレイヤーに渡します。
COMMENTS	VARCHAR2(255)	プロキシ・コピーについてのコメント。
MEDIA	VARCHAR2(80)	このバックアップを作成したメディア・マネージャについての詳細コメント。
MEDIA_POOL	NUMBER	プロキシ・コピーを格納するメディア・プールの数。
START_TIME	DATE	プロキシ・コピーの開始時刻。
COMPLETION_TIME	DATE	プロキシ・コピーの完了時刻。
ELAPSED_SECONDS	NUMBER	プロキシ・コピーの実行時間。
STATUS	VARCHAR2(1)	バックアップ・セットの状態。A (使用可能)、U (使用不可能)、X (期限切れ) または D (削除済)。

列	データ型	説明
KEEP	VARCHAR2 (3)	このプロキシ・コピーの保存方針が CONFIGURE RETENTION POLICY の値とは異なるかどうかを示します。可能な値は YES および NO です。
KEEP_OPTIONS	VARCHAR2 (11)	この制御ファイル・バックアップに指定された KEEP オプション。可能な値は、NOLOGS、BACKUP_LOGS、LOGS および NULL です。NOLOGS は、データベースがマウントされたときに一貫性のあるバックアップが作成されたことを示します。BACKUP_LOGS は、オープン・モードでバックアップが作成されたことを示します。したがって、一貫性のある状態にするには、アーカイブ・ログ・バックアップを適用する必要があります。LOGS は、LOGS キーワードを使用して長期間バックアップが作成されたことを示します。現在、LOGS キーワードは使用不可です。NULL は、このバックアップには KEEP オプションが指定されておらず、保存方針に基づいて不要とされることを示します。
KEEP_UNTIL	DATE	BACKUP コマンドの KEEP UNTIL TIME 句を指定した場合は、この列に制御ファイル・バックアップが不要になる直前の日付が表示されます。列が NULL で、KEEP_OPTIONS が NULL でなければ、バックアップが不要とされることはありません。
CONTROLFILE_TYPE	VARCHAR2 (1)	制御ファイル・コピーのタイプ。B (通常のコピー) または S (スタンバイ・コピー)。
RSR_KEY	NUMBER	このバックアップ・ピースを作成した RC_RMAN_STATUS の行に対する一意のキー。
SITE_KEY	NUMBER	このファイルと関連付けられている Data Guard データベースの主キー。Data Guard 環境にある各データベースの SITE_KEY の値は一意です。SITE_KEY を使用して RC_SITE ビューと結合すると、データベースの DB_UNIQUE_NAME を取得できます。

## RC\_PROXY\_COPY\_DETAILS

RC\_PROXY\_COPY\_DETAILS ビューには、リカバリ・カタログに登録されているデータベースのプロキシ・コピーのバックアップに関する詳細情報が含まれています。

このビューは、主に Enterprise Manager で内部的に使用します。

列	データ型	説明
SESSION_KEY	NUMBER	セッションの識別子。RC_RMAN_OUTPUT および RC_RMAN_BACKUP_JOB_DETAILS との結合に使用します。
SESSION_RECID	NUMBER	SESSION_STAMP とともに使用して、このプロキシ・コピー操作による出力を RC_RMAN_OUTPUT から一意に識別します。
SESSION_STAMP	NUMBER	SESSION_RECID とともに使用して、このプロキシ・コピー操作による出力を RC_RMAN_OUTPUT から一意に識別します。
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DB_NAME	VARCHAR2(8)	このレコードが属しているデータベース・インスタンスの DB_NAME。
RSR_KEY	NUMBER	このプロキシ・コピーを作成したジョブに対応する RC_RMAN_STATUS の行に対する一意のキー。
COPY_KEY	NUMBER	このプロキシ・コピーに対する一意の識別子。
FILE#	NUMBER	プロキシ・コピーが作成されたデータファイルの絶対ファイル番号。
HANDLE	VARCHAR2(1024)	プロキシ・コピーのハンドルはリストア操作のためのコピーを識別します。
COMMENTS	VARCHAR2(255)	このバックアップを格納したメディア・マネージャについての詳細コメント。
MEDIA	VARCHAR2(80)	このバックアップを格納するメディア・マネージャを識別します。
MEDIA_POOL	NUMBER	コピーを格納するメディア・プールの数。これは Recovery Manager の BACKUP コマンドの POOL オペランドに入力された値と同じ値です。
TAG	VARCHAR2(32)	このプロキシ・コピーに関連付けられているタグ。
CREATION_CHANGE#	NUMBER	データファイル作成時の SCN。
CREATION_TIME	DATE	CREATION_CHANGE# に対応する時刻。
CHECKPOINT_CHANGE#	NUMBER	プロキシ・コピー作成時におけるチェックポイント SCN。
CHECKPOINT_TIME	DATE	CHECKPOINT_CHANGE# に対応する時刻。
OUTPUT_BYTES	NUMBER	このプロキシ・コピー操作で生成されたすべての出力ピースの合計サイズ。
COMPLETION_TIME	DATE	プロキシ・コピーの完了時刻。
CONTROLFILE_TYPE	VARCHAR2(1)	可能な値は、通常の制御ファイルの場合は B、スタンバイ制御ファイルの場合は S です。
KEEP	VARCHAR2(3)	このバックアップの保存方針が CONFIGURE RETENTION POLICY の値とは異なるかどうかを示します。可能な値は YES および NO です。
KEEP_UNTIL	DATE	KEEP UNTIL TIME 句を指定した場合は、この列にこのバックアップが不要になる直前の日付が表示されます。列が NULL で、KEEP_OPTIONS が NULL でない場合、バックアップが不要とされることはありません。

列	データ型	説明
KEEP_OPTIONS	VARCHAR2 (11)	このバックアップに指定された KEEP オプション。可能な値は、NOLOGS、BACKUP_LOGS、LOGS および NULL です。NOLOGS は、データベースがマウントされたときに一貫性のあるバックアップが作成されたことを示します。BACKUP_LOGS は、オープン・モードでバックアップが作成されたことを示します。したがって、一貫性のある状態にするには、アーカイブ・ログ・バックアップを適用する必要があります。LOGS は、LOGS キーワードを使用して長期間バックアップが作成されたことを示します。現在、LOGS キーワードは使用不可です。NULL は、このバックアップには KEEP オプションが指定されておらず、保存方針に基づいて不要とされることを示します。
OUTPUT_BYTES_DISPLAY	VARCHAR2 (4000)	OUTPUT_BYTES と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。

## RC\_PROXY\_COPY\_SUMMARY

RC\_PROXY\_COPY\_SUMMARY ビューには、リカバリ・カタログに登録されているデータベースで使用可能なすべてのプロキシ・コピーのバックアップに関する集計情報が含まれています。

このビューは、主に Enterprise Manager で内部的に使用します。

列	データ型	説明
DB_KEY	NUMBER	このデータベースに対する主キー。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカネーションの DB_NAME。
NUM_COPIES	NUMBER	このデータベースに対するすべてのプロキシ・コピー操作で作成されたコピー数。
NUM_DISTINCT_COPIES	NUMBER	このデータベースに対するすべてのプロキシ・コピー操作で作成された個々のコピー数。
MIN_CHECKPOINT_CHANGE #	NUMBER	このデータベースに対する任意のプロキシ・コピーにおける最小チェックポイント SCN。
MAX_CHECKPOINT_CHANGE #	NUMBER	このデータベースに対する任意のプロキシ・コピーにおける最大チェックポイント SCN。
MIN_CHECKPOINT_TIME	DATE	このデータベースに対する任意のプロキシ・コピーにおける最も古いチェックポイント時刻。
MAX_CHECKPOINT_TIME	DATE	このデータベースに対する任意のプロキシ・コピーにおける最新のチェックポイント時刻。
OUTPUT_BYTES	NUMBER	プロキシ・コピーで生成されたすべての出力ピースの合計サイズ。
OUTPUT_BYTES_DISPLAY	VARCHAR2 (4000)	OUTPUT_BYTES と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。



## RC\_PROXY\_DATAFILE

このビューでは、プロキシ・コピー機能を使用して作成したデータファイルのバックアップを説明します。このビューは、V\$PROXY\_DATAFILE ビューに対応しています。

プロキシ・コピーでは、メディア・マネージャがデータのバックアップとリストアの操作を行います。各行が1データベース・ファイルの1バックアップを表します。

列	データ型	説明
DB_KEY	NUMBER	ターゲット・データベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	ターゲット・データベースのインカーネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカーネーションの DB_NAME。
XDF_KEY	NUMBER	リカバリ・カタログのプロキシ・コピーの主キー。Recovery Manager がリカバリ・カタログに接続している間に LIST コマンドを発行すると、この値が出力の KEY 列に表示されます。
RECID	NUMBER	V\$PROXY_DATAFILE からのプロキシ・コピー・レコードの識別子。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
STAMP	NUMBER	V\$PROXY_DATAFILE からのプロキシ・コピーの STAMP。RECID と STAMP は、ターゲット・データベース制御ファイル内でこのレコードを一意に識別する連結主キーを構成します。
TAG	VARCHAR2 (32)	プロキシ・コピーのタグ。
FILE#	NUMBER	プロキシ・コピーが作成されたデータファイルの絶対ファイル番号。
CREATION_CHANGE#	NUMBER	データファイル作成時の SCN。
CREATION_TIME	DATE	データファイル作成時の時刻。
RESETLOGS_CHANGE#	NUMBER	データファイル・ヘッダー内の最新 RESETLOGS の SCN。
RESETLOGS_TIME	DATE	データファイル・ヘッダー内の最新 RESETLOGS のタイムスタンプ。
INCREMENTAL_LEVEL	NUMBER	このコピーが増分バックアップの一部の場合は0、それ以外の場合は NULL。
CHECKPOINT_CHANGE#	NUMBER	コピー作成時におけるチェックポイント SCN。
CHECKPOINT_TIME	DATE	コピー作成時におけるチェックポイント時刻。
ABSOLUTE_FUZZY_CHANGE#	NUMBER	認識されている場合は、ファイルのいずれかのブロック内にある一番大きい値の SCN。ファイルがファジーにならないように、リカバリは少なくともこの SCN まで進行させる必要があります。
RECOVERY_FUZZY_CHANGE#	NUMBER	ファイルをファジーにしないために、リカバリ操作が到達する必要がある SCN。NULL でない場合は、このファイルでデータベースをオープンする前に、少なくとも指定された SCN までのリカバリが必要です。
RECOVERY_FUZZY_TIME	DATE	RECOVERY_FUZZY_CHANGE# に対応付けられている時刻。
ONLINE_FUZZY	VARCHAR2 (3)	YES または NO。YES に設定されている場合、このコピーはインスタンス障害または OFFLINE IMMEDIATE の後に作成されたものです（またはデータベースがオープン状態で不適切に作成されたコピーのコピーです）。ファイルに一貫性を持たせるには、次のクラッシュ・リカバリ・マーカまでのすべての REDO を適用するリカバリが必要です。
BACKUP_FUZZY	VARCHAR2 (3)	YES または NO。YES に設定されている場合、このコピーは BEGIN BACKUP/END BACKUP のバックアップ方式により作成されています。このコピーに一貫性を持たせるには、ALTER TABLESPACE END BACKUP 文を発行するときに REDO ストリームに置かれたマーカまでのすべての REDO を適用するリカバリが必要です。

列	データ型	説明
BLOCKS	NUMBER	ブロック単位のデータファイルのコピーのサイズ (およびコピーを作成したときのデータファイルのサイズ)。
BLOCK_SIZE	NUMBER	コピーのブロック・サイズ (バイト)。
DEVICE_TYPE	VARCHAR2 (255)	順次メディア・デバイスのタイプ。
HANDLE	VARCHAR2 (1024)	プロキシ・コピーの名前またはハンドル。Recovery Manager は、この値を、ファイルを識別するオペレーティング・システム固有のレイヤーに渡します。
COMMENTS	VARCHAR2 (255)	プロキシ・コピーについてのコメント。
MEDIA	VARCHAR2 (80)	このバックアップを作成したメディア・マネージャについての詳細コメント。
MEDIA_POOL	NUMBER	プロキシ・コピーを格納するメディア・プールの数。
START_TIME	DATE	プロキシ・コピーの開始時刻。
COMPLETION_TIME	DATE	プロキシ・コピーの完了時刻。
ELAPSED_SECONDS	NUMBER	プロキシ・コピーの実行時間。
STATUS	VARCHAR2 (1)	バックアップ・セットの状態。A (使用可能)、U (使用不可能)、X (期限切れ) または D (削除済)。
KEEP	VARCHAR2 (3)	このプロキシ・コピーの保存方針が CONFIGURE RETENTION POLICY の値とは異なるかどうか (YES または NO) を示します。
KEEP_UNTIL	DATE	BACKUP コマンドの KEEP UNTIL TIME 句を指定した場合は、この列にデータファイル・バックアップが不要になる直前の日付が表示されます。列が NULL で、KEEP OPTIONS が NULL でなければ、バックアップが不要とされることはありません。
KEEP_OPTIONS	VARCHAR2 (11)	このバックアップに指定された KEEP オプション。可能な値は、NOLOGS、BACKUP_LOGS、LOGS および NULL です。NOLOGS は、データベースがマウントされたときに一貫性のあるバックアップが作成されたことを示します。BACKUP_LOGS は、オープン・モードでバックアップが作成されたことを示します。したがって、一貫性のある状態にするには、アーカイブ・ログ・バックアップを適用する必要があります。LOGS は、LOGS キーワードを使用して長期間バックアップが作成されたことを示します。現在、LOGS キーワードは使用不可です。NULL は、このバックアップには KEEP オプションが指定されておらず、保存方針に基づいて不要とされることを示します。
RSR_KEY	NUMBER	このプロキシ・コピーを作成したジョブに対応する RC_RMAN_STATUS の行に対する一意のキー。
SITE_KEY	NUMBER	このファイルと関連付けられている Data Guard データベースの主キー。Data Guard 環境にある各データベースの SITE_KEY の値は一意です。SITE_KEY を使用して RC_SITE ビューと結合すると、データベースの DB_UNIQUE_NAME を取得できます。
FOREIGN_DBID	NUMBER	このデータファイルのトランスポート元であるデータベースの外部 DBID。バックアップされたファイルが外部データベース・ファイルでない場合、値は 0 になります。
PLUGGED_READONLY	VARCHAR2 (3)	トランスポートされた読取り専用外部ファイルのプロキシ・コピーである場合は YES、そうでない場合は NO。
PLUGIN_CHANGE#	NUMBER	外部データファイルがデータベースにトランスポートされたときの SCN。このファイルが外部データベース・ファイルでない場合、値は 0 です。
PLUGIN_RESETLOGS_CHANGE#	NUMBER	この外部ファイルがトランスポートされたインカネーションに対する RESETLOGS 操作の SCN。このファイルが外部データベース・ファイルでない場合、値は 0 です。
PLUGIN_RESETLOGS_TIME	DATE	この外部ファイルがトランスポートされたインカネーションに対する RESETLOGS 操作の時刻。このファイルが外部データベース・ファイルでない場合、値は 0 です。

## RC\_REDO\_LOG

このビューは、前回のカタログ再同期化以降の、データベースのすべてのインカーネーションのオンライン REDO ログに関する情報をリストします。このビューは、V\$LOG と V\$LOGFILE ビューの結合に対応しています。

列	データ型	説明
DB_KEY	NUMBER	ターゲット・データベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	ターゲット・データベースのインカーネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカーネーションの DB_NAME。
THREAD#	NUMBER	REDO スレッドの番号。
GROUP#	NUMBER	オンライン REDO ログ・グループの番号。
NAME	VARCHAR2 (1024)	オンライン REDO ログ・ファイルの名前。
SITE_KEY	NUMBER	このファイルと関連付けられている Data Guard データベースの主キー。Data Guard 環境にある各データベースの SITE_KEY の値は一意です。SITE_KEY を使用して RC_SITE ビューと結合すると、データベースの DB_UNIQUE_NAME を取得できます。
BYTES	NUMBER	ファイルのサイズ (バイト)。
TYPE	VARCHAR2 (7)	REDO ログのタイプ: ONLINE または STANDBY。

## RC\_REDO\_THREAD

このビューは、前回のカタログ再同期化以降の、データベースの全インカネーションの全 REDO スレッドに関するデータをリストします。このビューは `V$THREAD` に対応しています。

列	データ型	説明
DB_KEY	NUMBER	ターゲット・データベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	ターゲット・データベースのインカネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカネーションの DB_NAME。
THREAD#	NUMBER	データベース・インカネーションの REDO スレッド番号。
STATUS	VARCHAR2 (1)	REDO スレッドの状態。D (使用禁止)、E (使用可能) または O (オープン)。
SEQUENCE#	NUMBER	最後に割り当てられたログ順序番号。
ENABLE_CHANGE#	NUMBER	このスレッドが使用可能になったときの SCN。
ENABLE_TIME	DATE	このスレッドが使用可能になった時刻。
DISABLE_CHANGE#	NUMBER	このスレッドが使用禁止になった時点での最新 SCN。スレッドがまだ使用禁止になっている場合、このスレッドに関してこの SCN 以降の REDO は存在しません。スレッドが使用可能になっている場合、このスレッドに関して DISABLE_CHANGE# から ENABLE_CHANGE# までの REDO は存在しません。
DISABLE_TIME	DATE	このスレッドが使用禁止になった最新時刻。

## RC\_RESTORE\_POINT

このビューは、前回のカタログ再同期化以降の、データベースのすべてのインカーネーションの全リストア・ポイントをリストします。このビューは V\$RESTORE\_POINT に対応しています。

列	データ型	説明
DBINC_KEY	NUMBER	ターゲット・データベースのインカーネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
RECID	NUMBER	制御ファイル内の対応する行のレコード ID。
STAMP	NUMBER	制御ファイル内の行のタイムスタンプ。(制御ファイル・レコードは再利用されるため、タイムスタンプとレコード ID を組み合わせ、RC_RMAN_STATUS 内のすべてのレコードで一意的な値を取得する必要があります。)
SITE_KEY	NUMBER	このリストア・ポイントと関連付けられている Data Guard データベースの主キー。Data Guard 環境にある各データベースの SITE_KEY の値は一意的です。SITE_KEY を使用して RC_SITE ビューと結合すると、データベースの DB_UNIQUE_NAME を取得できます。
NAME	VARCHAR2 (128)	リストア・ポイントの名前。
RESTORE_POINT_TIME	DATE	リストア・ポイント SCN に対応するデータベースの時刻。
CREATION_TIME	DATE	リストア・ポイントを作成したときの時刻。
SCN	NUMBER	リストア・ポイントを作成したときのデータベース SCN。
LONG_TERM	VARCHAR2 (3)	KEEP オプションを指定して作成されたバックアップに対応したリストア・ポイントの場合は YES、そうでない場合は NO。
PRESERVED	VARCHAR2 (3)	リストア・ポイントを明示的に削除する必要がある場合は YES、そうでない場合は NO。
GUARANTEE_FLASHBACK_DATABASE	VARCHAR2 (3)	この時点までフラッシュバックできるようにフラッシュバック・ログを保持しておく必要がある場合は YES、そうでない場合は NO。

---

## RC\_RESYNC

このビューは、リカバリ・カタログの再同期化に関する情報をリストします。すべての完全再同期化では、ターゲット・データベースの制御ファイルのスナップショットが作成され、そのスナップショットからリカバリ・カタログが再同期化されます。

列	データ型	説明
DB_KEY	NUMBER	ターゲット・データベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	ターゲット・データベースのインカーネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカーネーションの DB_NAME。
RESYNC_KEY	NUMBER	再同期化の主キー。
CONTROLFILE_CHANGE#	NUMBER	カタログの再同期化が開始された制御ファイル・チェックポイント SCN。
CONTROLFILE_TIME	DATE	カタログの再同期化が開始された制御ファイル・チェックポイントのタイムスタンプ。
CONTROLFILE_SEQUENCE#	NUMBER	制御ファイルの順序番号。
CONTROLFILE_VERSION	DATE	カタログが再同期化された元の制御ファイルのバージョンに対する作成時刻。
RESYNC_TYPE	VARCHAR2 (7)	再同期化のタイプ。FULL (全体) または PARTIAL (部分)。
DB_STATUS	VARCHAR2 (7)	ターゲット・データベースの状態。OPEN (オープン) または MOUNTED (マウント済)。
RESYNC_TIME	DATE	再同期化の時刻。

## RC\_RMAN\_BACKUP\_JOB\_DETAILS

RC\_RMAN\_BACKUP\_JOB\_DETAILS は、Recovery Manager バックアップ・ジョブに関する詳細情報を表示します。Recovery Manager ジョブは、Recovery Manager セッション内で実行される一連のコマンドです。Recovery Manager バックアップ・ジョブは、Recovery Manager ジョブ内で実行される一連の BACKUP コマンドです。たとえば、同じ Recovery Manager ジョブ内で実行される BACKUP DATABASE コマンドと BACKUP ARCHIVELOG ALL コマンドは、1 つの Recovery Manager バックアップ・ジョブを構成します。

このビューでは、同じセッションで複数の BACKUP コマンドが実行される場合でも、Recovery Manager セッションごとに 1 行が含まれます。SESSION\_KEY 列は、バックアップ・ジョブが実行された Recovery Manager セッションに対する一意のキーです。Recovery Manager セッション中に実行された操作の詳細は、[RC\\_RMAN\\_BACKUP\\_SUBJOB\\_DETAILS](#) ビューを参照してください。

このビューは、主に Enterprise Manager で内部的に使用します。

列	データ型	説明
DB_KEY	NUMBER	このデータベースに対する主キー。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカンネーションの DB_NAME。
SESSION_KEY	NUMBER	Recovery Manager セッションの識別子。RC_RMAN_OUTPUT および RC_RMAN_BACKUP_JOB_DETAILS との結合に使用します。
SESSION_RECID	NUMBER	SESSION_KEY および SESSION_STAMP とともに使用して、ジョブの出力を RC_RMAN_OUTPUT から一意に識別します。
SESSION_STAMP	NUMBER	SESSION_RECID および SESSION_KEY とともに使用して、ジョブの出力を RC_RMAN_OUTPUT から一意に識別します。
COMMAND_ID	VARCHAR2 (33)	SET COMMAND ID を使用して設定されたユーザー指定の値、または Recovery Manager で生成された一意のコマンド ID のいずれか。
START_TIME	DATE	Recovery Manager ジョブ内の最初のバックアップ・コマンドの開始時刻。
END_TIME	DATE	Recovery Manager ジョブ内の最後のバックアップ・コマンドの終了時刻。
INPUT_BYTES	NUMBER	この Recovery Manager ジョブ中にバックアップされたすべての入力ファイルの合計サイズ。
OUTPUT_BYTES	NUMBER	この Recovery Manager ジョブで生成されたすべての出力バックアップ・ピースの合計サイズ。
STATUS_WEIGHT	NUMBER	Enterprise Manager 内部で使用。
OPTIMIZED_WEIGHT	NUMBER	Enterprise Manager 内部で使用。
INPUT_TYPE_WEIGHT	NUMBER	Enterprise Manager 内部で使用。
OUTPUT_DEVICE_TYPE	VARCHAR2 (17)	DISK、SBT_TAPE または *。アスタリスクは、その出力が複数のデバイス・タイプに書き込まれたバックアップ・ジョブであることを示します。
AUTOBACKUP_COUNT	NUMBER	この Recovery Manager ジョブによって実行された自動バックアップの数。
BACKED_BY_OSB	VARCHAR2 (3)	バックアップ・ピースが Oracle Secure Backup にバックアップされている (YES) か、バックアップされていない (NO) かを示します。
AUTOBACKUP_DONE	VARCHAR2 (3)	YES または NO (制御ファイルの自動バックアップがこのバックアップ・ジョブの一部として実行されたかどうかによって異なる)。

列	データ型	説明
STATUS	VARCHAR2 (23)	次のいずれかの値になります。RUNNING WITH WARNINGS、RUNNING WITH ERRORS、COMPLETED、COMPLETED WITH WARNINGS、COMPLETED WITH ERRORS、FAILED。  ジョブが失敗した場合でも、バックアップ・セットが作成されなかったとはかぎりません。Recovery Manager がバックアップ・セットをいくつか作成した後に、ジョブが失敗した可能性もあります。したがって、RC_BACKUP_SET_DETAILS ビューには、バックアップ・ジョブで正しく作成されたバックアップ・セットを示す行が含まれている場合があります。このビューを RC_BACKUP_SET_DETAILS と結合すると、失敗したバックアップ・ジョブに関する詳細情報を取得できます。
INPUT_TYPE	VARCHAR2 (13)	このバックアップの入力タイプを示す値が表示されます。可能な値は、RC_RMAN_BACKUP_TYPE ビューを参照してください。  これらの値のうち複数に該当するバックアップがジョブに含まれる場合は、複数の行がビューに表示されます。これらの行は各入力タイプごとに異なる INPUT_TYPE 値に対応しており、INPUT_BYTES、OUTPUT_BYTES、INPUT_BYTES_DISPLAY、OUTPUT_BYTES_DISPLAY の各フィールドには対応する値が含まれます。
OPTIMIZED	VARCHAR2 (3)	バックアップ最適化が適用されたかどうかによって YES または NO になります。
ELAPSED_SECONDS	NUMBER	このバックアップ・ジョブの実行で経過した秒数。
COMPRESSION_RATIO	NUMBER	このバックアップの圧縮率。
INPUT_BYTES_PER_SEC	NUMBER	入力読み取り速度 (バイト / 秒)。
OUTPUT_BYTES_PER_SEC	NUMBER	出力書き込み速度 (バイト / 秒)。
INPUT_BYTES_DISPLAY	VARCHAR2 (4000)	INPUT_BYTES と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。
OUTPUT_BYTES_DISPLAY	VARCHAR2 (4000)	OUTPUT_BYTES と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。
INPUT_BYTES_PER_SEC_DISPLAY	VARCHAR2 (4000)	INPUT_BYTES_PER_SEC と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。
OUTPUT_BYTES_PER_SEC_DISPLAY	VARCHAR2 (4000)	OUTPUT_BYTES_PER_SEC と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。
TIME_TAKEN_DISPLAY	VARCHAR2 (4000)	ユーザーが表示可能な書式 (hh:mm:ss など) に変換された Recovery Manager ジョブの合計時間。



## RC\_RMAN\_BACKUP\_SUBJOB\_DETAILS

RC\_RMAN\_BACKUP\_SUBJOB\_DETAILS ビューは Recovery Manager セッション内で類似する操作グループの詳細を表示します。

このビューは、主に Enterprise Manager で内部的に使用します。

列	データ型	説明
DB_KEY	NUMBER	このデータベースに対する主キー。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカネーションの DB_NAME。
SESSION_KEY	NUMBER	セッションの識別子。RC_RMAN_OUTPUT および RC_RMAN_BACKUP_JOB_DETAILS との結合に使用します。
SESSION_RECID	NUMBER	SESSION_KEY および SESSION_STAMP とともに使用して、ジョブの出力を RC_RMAN_OUTPUT から一意に識別します。
SESSION_STAMP	NUMBER	SESSION_RECID および SESSION_KEY とともに使用して、ジョブの出力を RC_RMAN_OUTPUT から一意に識別します。
OPERATION	VARCHAR2 (33)	可能な値は BACKUP、ROLLFORWARD、VALIDATE または BACKUPSET です。このビューではセッションの各操作タイプに対して 1 つずつ行があり、各行にはセッション中に実行されるそのタイプのすべての操作を示すそれぞれの値が含まれます。
COMMAND_ID	VARCHAR2 (33)	SET COMMAND ID を使用して設定されたユーザー指定の値、または Recovery Manager で生成された一意のコマンド ID のいずれか。
START_TIME	DATE	ジョブ内の最初のバックアップ・コマンドの開始時刻。
END_TIME	DATE	ジョブ内の最後のバックアップ・ジョブの終了時刻。
INPUT_BYTES	NUMBER	このジョブ中にバックアップされたすべての入力ファイルの合計サイズ。
OUTPUT_BYTES	NUMBER	このジョブで生成されたすべての出力ピースの合計サイズ。
STATUS_WEIGHT	NUMBER	Enterprise Manager 内部で使用。
OPTIMIZED_WEIGHT	NUMBER	Enterprise Manager 内部で使用。
INPUT_TYPE_WEIGHT	NUMBER	Enterprise Manager 内部で使用。
OUTPUT_DEVICE_TYPE	VARCHAR2 (17)	出力デバイスのタイプ。DISK、SBT_TAPE または *。アスタリスク (*) は、その出力が複数のデバイス・タイプに書き込まれたバックアップ・ジョブであることを示します。
BACKED_BY_OS	VARCHAR2 (3)	バックアップ・ピースが Oracle Secure Backup にバックアップされている (YES) か、バックアップされていない (NO) かを示します。
AUTOBACKUP_DONE	VARCHAR2 (3)	制御ファイルの自動バックアップがこのジョブの一部として実行された (YES) か、実行されなかった (NO) かを示します。
STATUS	VARCHAR2 (23)	次のいずれかの値になります。RUNNING WITH WARNINGS、RUNNING WITH ERRORS、COMPLETED、COMPLETED WITH WARNINGS、COMPLETED WITH ERRORS、FAILED。
INPUT_TYPE	VARCHAR2 (13)	次のいずれかの値が含まれます。DATABASE FULL、RECOVERY AREA、DATABASE INCR、DATAFILE FULL、DATAFILE INCR、ARCHIVELOG、CONTROLFILE、SPFILE。  これらの値のうち複数の該当するバックアップがサブジョブに含まれる場合は、複数の行がビューに表示されます。これらの行は各入力タイプごとに異なる INPUT_TYPE 値に対応しており、INPUT_BYTES、OUTPUT_BYTES、INPUT_BYTES_DISPLAY、OUTPUT_BYTES_DISPLAY の各フィールドには対応する値が含まれます。
OPTIMIZED	VARCHAR2 (3)	OPERATION 列の値が BACKUP の場合、バックアップ最適化が適用されたかどうかによって、OPTIMIZED が YES または NO になります。
AUTOBACKUP_COUNT	NUMBER	このジョブによって実行された自動バックアップの数。

---

列	データ型	説明
COMPRESSION_RATIO	NUMBER	このバックアップの圧縮率。
INPUT_BYTES_DISPLAY	VARCHAR2 (4000)	INPUT_BYTES と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。
OUTPUT_BYTES_DISPLAY	VARCHAR2 (4000)	OUTPUT_BYTES と値は同じですが、ユーザーが表示可能な書式 (798.01M または 5.25G など) に変換されます。

---

---

## RC\_RMAN\_BACKUP\_TYPE

このビューは Enterprise Manager 内部で使⽤します。

このビューは、特定のバックアップ・タイプに関するレポート生成時にその他の Enterprise Manager ビューのフィルタ処理で使⽤します。

列	データ型	説明
WEIGHT	NUMBER	Enterprise Manager 内部で、異なるバックアップ・タイプの優先順位をレポートに設定する⽤に使⽤します。
INPUT_TYPE	VARCHAR2 (13)	Enterprise Manager 内部で、さまざまなレポート画面の作成で使⽤される可能なフィルタを示す⽤に使⽤します。

## RC\_RMAN\_CONFIGURATION

このビューは、Recovery Manager の永続構成の設定情報をリストします。このビューは V\$RMAN\_CONFIGURATION ビューに対応しています。

列	データ型	説明
DB_KEY	NUMBER	この構成に対応するターゲット・データベースの主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
CONF#	NUMBER	この構成レコードを、所有するターゲット・データベース内で識別する一意キー。
NAME	VARCHAR2 (65)	構成のタイプ。CONFIGURE コマンドのほとんどのオプションは有効なタイプです。ただし、CONFIGURE EXCLUDE (RC_TABLESPACE の説明を参照)、CONFIGURE AUXNAME (RC_DATAFILE の説明を参照) および CONFIGURE SNAPSHOT CONTROLFILE (制御ファイルにのみ格納) は、無効です。
VALUE	VARCHAR2 (1025)	CONFIGURE コマンドの設定。たとえば、RETENTION POLICY TO RECOVERY WINDOW OF 1 DAYS などです。
DB_UNIQUE_NAME	VARCHAR2 (512)	このレコードが属しているデータベース・インカネーションの DB_UNIQUE_NAME。Data Guard 環境のすべてのデータベースで同一の DBID が共有されていますが、DB_UNIQUE_NAME の値は異なります。特定のファイルに対するデータベース名が不明の場合、この列の値は NULL です。たとえば、Oracle Database 11g より前のバージョンの Recovery Manager で管理されているデータベースの行は NULL です。
SITE_KEY	NUMBER	この構成と関連付けられている Data Guard データベースの主キー。Data Guard 環境にある各データベースの SITE_KEY の値は一意です。SITE_KEY を使用して RC_SITE ビューと結合すると、データベースの DB_UNIQUE_NAME を取得できます。

---

## RC\_RMAN\_OUTPUT

RC\_RMAN\_OUTPUT ビューは、制御ファイル・ビュー V\$RMAN\_OUTPUT に対応しています。  
このビューは、主に Enterprise Manager 内部で使用します。

列	データ型	説明
DB_KEY	NUMBER	ターゲット・データベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
RSR_KEY	NUMBER	この出力を作成したジョブに対応する RC_RMAN_STATUS の行に対する一意キー。
SESSION_KEY	NUMBER	セッションの識別子。RC_RMAN_OUTPUT および RC_RMAN_BACKUP_JOB_DETAILS との結合に使用します。
RECID	NUMBER	このデータベースに対する V\$RMAN_OUTPUT.RECID に表示される値が含まれます。
STAMP	NUMBER	このライン・アウト出力の行が追加されたときのスタンプ（順序付けに使用）。
OUTPUT	VARCHAR2 (130)	Recovery Manager の出力テキスト。

## RC\_RMAN\_STATUS

このビューには、このリカバリ・カタログに対応付けられているすべてのデータベースでの Recovery Manager 操作の履歴情報が含まれます。このビューには、基本的に V\$RMAN\_STATUS と同じ情報が含まれます。ただし、現行のセッションに関する情報は含まれません。

すべての Recovery Manager 操作（バックアップ、リストア、バックアップの削除など）が、この表に記録されます。この表は、各 Recovery Manager セッション（Recovery Manager クライアントが終了するまでに実行されたすべての操作を含む、Recovery Manager クライアントの実行）の状態、セッション中に実行された操作、および再帰的操作を表示するように構成されています。

また、RC\_RMAN\_STATUS には、V\$RMAN\_STATUS には含まれていない RSR\_KEY 列、PARENT\_KEY 列および SESSION\_KEY 列が含まれます。

列	データ型	説明
DB_KEY	NUMBER	データベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	データベース・インカネーションに対する主キー。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカネーションの DB_NAME。
RECID	NUMBER	制御ファイル内の対応する行のレコード ID。
STAMP	NUMBER	制御ファイル内の行のタイムスタンプ。(制御ファイル・レコードは再利用されるため、タイムスタンプとレコード ID を組み合わせて、RC_RMAN_STATUS 内のすべてのレコードで一意的な値を取得する必要があります。)
RSR_KEY	NUMBER	この行に対する一意のキー。
PARENT_KEY	NUMBER	この行の親行に対する RSR_KEY の値。
SESSION_KEY	NUMBER	この行に対応付けられたセッション行に対する RSR_KEY の値。RC_RMAN_BACKUP_JOB_DETAILS との結合に使用します。
ROW_TYPE	VARCHAR2 (33)	この行で表される操作のタイプ。可能な値は次のとおりです。 <ul style="list-style-type: none"> <li>■ レベル 0 の行の場合は SESSION</li> <li>■ レベル 1 の行の場合は COMMAND</li> <li>■ レベル 2 以上の行の場合は RECURSIVE OPERATION</li> </ul>
ROW_LEVEL	NUMBER	この行のレベル。 <ul style="list-style-type: none"> <li>■ 0 の場合、これはセッション行です。つまり、ROW_TYPE は SESSION となっており、この行は、Recovery Manager クライアントが起動されたことを表します。</li> <li>■ 1 の場合、この行は Recovery Manager クライアントにコマンドを入力して実行したことを表します。レベル 1 の行の場合、ROW_TYPE は COMMAND です。</li> <li>■ 2 以上の場合、この行は再帰的操作であることを表します。再帰的操作とは、Recovery Manager コマンドの補助操作（データベース・バックアップに関連して実行される制御ファイルの自動バックアップなど）のことです。レベル 2 以上の行の場合、ROW_TYPE は RECURSIVE OPERATION です。</li> </ul>
OPERATION	VARCHAR2 (33)	この行で表される操作の名前。SESSION 操作の場合、この列は RMAN に設定されます。COMMAND 操作の場合、この列は実行されたコマンド (BACKUP、RESTORE、CONFIGURE、REPORT など) を表します。
STATUS	VARCHAR2 (33)	この行で表される操作の状態。可能な値は、COMPLETED、COMPLETED WITH WARNINGS、COMPLETED WITH ERRORS および FAILED です。
COMMAND_ID	VARCHAR2 (33)	ユーザー指定のコマンド ID。コマンド ID は、Recovery Manager で SET COMMAND ID 構文を使用して変更できます。デフォルトでは、コマンド ID は、Recovery Manager の起動時に ISO 規格の書式で設定されます。

列	データ型	説明
MBYTES_PROCESSED	NUMBER	この行で表される操作でデータ転送（データのバックアップやリストアなど）を実行した場合、この列には操作で処理された MB 数が含まれます。それ以外の場合、この列には NULL が含まれます。
START_TIME	DATE	この行で表される操作の開始時刻。
END_TIME	DATE	この行で表される操作の終了時刻。
JOB_KEY	NUMBER	Recovery Manager セッションのキー。SESSION_KEY と同じです。
INPUT_BYTES	NUMBER	読み取られた入力バイト数。
OUTPUT_BYTES	NUMBER	書き込まれた入力バイト数。
OPTIMIZED	VARCHAR2 (3)	バックアップ・ジョブ時にバックアップの最適化が適用される場合は YES。それ以外の場合は NO。
OBJECT_TYPE	VARCHAR2 (80)	次のいずれかの値が含まれます。DATABASE FULL、RECOVERY AREA、DATABASE INCR、DATAFILE FULL、DATAFILE INCR、ARCHIVELOG、CONTROLFILE、SPFILE。
SESSION_RECID	NUMBER	ROW_TYPE が SESSION（つまり、この行は親を持たず、Recovery Manager セッションを表す）の場合、この行には NULL が含まれます。それ以外の場合は、この行に対応付けられたセッションを表す行のレコード ID が含まれます。
SESSION_STAMP	NUMBER	ROW_TYPE が SESSION（つまり、この行は親を持たず、Recovery Manager セッションを表す）の場合、この行には NULL が含まれます。それ以外の場合は、この行に対応付けられたセッションを表す行のタイムスタンプが含まれます。
OUTPUT_DEVICE_TYPE	VARCHAR2 (17)	出力デバイスのタイプ。DISK、SBT_TAPE または *。アスタリスク (*) は、複数のデバイス・タイプに出力が書き込まれたことを示します。
SITE_KEY	NUMBER	Recovery Manager のステータス情報と関連付けられている Data Guard データベースの主キー。Data Guard 環境にある各データベースの SITE_KEY の値は一意です。SITE_KEY を使用して RC_SITE ビューと結合すると、データベースの DB_UNIQUE_NAME を取得できます。
OSB_ALLOCATED	VARCHAR2 (3)	このセッションが Oracle Secure Backup に対して SBT チャンネルを割り当てた場合は YES、割り当てていない場合は NO。

---

## RC\_SITE

このビューは、リカバリ・カタログで認識されている Data Guard 環境のすべてのデータベースに関する情報を表示します。このビューを使用すると、DB\_UNIQUE\_NAME 列を持たないビューに対して、その値を取得できます。

列	データ型	説明
SITE_KEY	NUMBER	このデータベースの一意のキー。RC_SITE.SITE_KEY 列をその他のビューの RC_SITE 列と結合すると、バックアップに関連付けられている DB_UNIQUE_NAME を特定できます。
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DATABASE_ROLE	VARCHAR2(7)	Data Guard 環境でのデータベースのロール。
CF_CREATE_TIME	DATE	制御ファイルの作成日。
DB_UNIQUE_NAME	VARCHAR2(30)	データベースの DB_UNIQUE_NAME。Data Guard 環境のすべてのデータベースで同一の DBID が共有されていますが、DB_UNIQUE_NAME の値は異なります。



---

## RC\_STORED\_SCRIPT

このビューは、リカバリ・カタログに格納されているスクリプトに関する情報をリストします。このビューでは、1 ストアド・スクリプトに対して 1 行の説明があります。Recovery Manager のスクリプト管理コマンド (LIST SCRIPT NAMES、LIST SCRIPT など) を使用すると、この情報をより簡単に表示できます。

列	データ型	説明
DB_KEY	NUMBER	このストアド・スクリプトを所有しているデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカネーションの DB_NAME。
SCRIPT_NAME	VARCHAR2 (100)	ストアド・スクリプトの名前。
SCRIPT_COMMENT	VARCHAR2 (255)	このスクリプトの作成時にユーザーが入力したコメント。コメントが入力されなかった場合は NULL。

---

## RC\_STORED\_SCRIPT\_LINE

このビューは、リカバリ・カタログ内のストアド・スクリプトの個々の行に関する情報をリストします。このビューでは、1 ストアド・スクリプトに対して 1 行の説明があります。

列	データ型	説明
DB_KEY	NUMBER	このストアド・スクリプトを所有しているデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
SCRIPT_NAME	VARCHAR2(100)	ストアド・スクリプトの名前。
LINE	NUMBER	ストアド・スクリプトの行番号。スクリプトの行は、SCRIPT_NAME および LINE によって一意に識別されます。
TEXT	VARCHAR2(1024)	ストアド・スクリプトの行のテキスト。

## RC\_TABLESPACE

このビューは、リカバリ・カタログに登録されているすべての表領域、すべての削除済表領域および古いインカーネーションに属している表領域をリストします。このビューは、V\$TABLESPACE ビューに対応しています。表領域属性の現行の値を表示します。

列	データ型	説明
DB_KEY	NUMBER	ターゲット・データベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	ターゲット・データベースのインカーネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカーネーションの DB_NAME。
TS#	NUMBER	ターゲット・データベースでの表領域 ID。表領域を削除し、再作成した場合は、同じインカーネーションに複数の TS# が存在します。
NAME	VARCHAR2 (30)	表領域名。表領域を削除し、再作成した場合は、同じインカーネーションにその名前が複数存在している可能性があります。
CREATION_CHANGE#	NUMBER	作成時の SCN (最初のデータファイルから)。
CREATION_TIME	DATE	表領域の作成時刻。制御ファイルを作成後にオフラインの表領域は NULL。
DROP_CHANGE#	NUMBER	表領域を削除したときに記録された SCN。同じ TS# を持つ新規の表領域が発見された場合、DROP_CHANGE# はその表領域の CREATION_CHANGE# に設定されます。それ以外の場合、値は RC_CHECKPOINT.CKP_SCN に設定されます。
DROP_TIME	DATE	表領域を削除したときの時刻。
INCLUDED_IN_DATABASE_BACKUP	VARCHAR2 (3)	この表領域がデータベース全体のバックアップに含まれるかどうかを YES または NO で示します。NO 値が発生するのは、このデータファイルを所有する表領域に対して CONFIGURE EXCLUDE が実行された場合のみです。
BIGFILE	VARCHAR2 (3)	この表領域が BIGFILE オプションを使用して作成された表領域かどうかを示します。
TEMPORARY	VARCHAR2 (3)	データベースがローカル管理の一時表領域の場合は YES。それ以外の場合は NO。
ENCRYPT_IN_BACKUP	VARCHAR2 (3)	可能な値は次のとおりです。 <ul style="list-style-type: none"> <li>■ ON: 表領域レベルで暗号化を有効にします。</li> <li>■ OFF: 表領域レベルで暗号化を無効にします。</li> <li>■ NULL: 表領域レベルでの暗号化の有効化または無効化を明示的に行いません (デフォルト、または CLEARED の場合)。</li> </ul>

## RC\_TEMPFILE

このビューは、リカバリ・カタログに登録されているすべての一時ファイルに関する情報をリストします。このビューは、V\$tempfile ビューに対応しています。一時ファイルは、その表領域が削除された場合、削除されたものとして表示されます。

列	データ型	説明
DB_KEY	NUMBER	ターゲット・データベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
DBINC_KEY	NUMBER	ターゲット・データベースのインカーネーションに対する主キー。この列を使用して、RC_DATABASE_INCARNATION との結合を構成します。
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカーネーションの DB_NAME。
TS#	NUMBER	ターゲット・データベースでの表領域 ID。表領域を削除し、再作成した場合は、同じインカーネーションに複数の TS# が存在します。
TABLESPACE_NAME	VARCHAR2 (30)	表領域名。表領域を削除し、再作成した場合は、同じインカーネーションにその名前が複数存在している可能性があります。
FILE#	NUMBER	一時ファイルの絶対ファイル番号。削除および再作成された一時ファイルのインカーネーションと同じインカーネーションに同一の一時ファイル番号が存在している可能性があります。
CREATION_CHANGE#	NUMBER	一時ファイルが作成されたときの SCN。
CREATION_TIME	DATE	一時ファイルが作成された時刻。
DROP_CHANGE#	NUMBER	一時ファイルを削除したときに記録された SCN。同じ FILE# を持つ新規の一時ファイルが発見された場合、DROP_CHANGE# はその一時ファイルの CREATION_CHANGE# に設定されます。それ以外の場合、値は RC_CHECKPOINT.CKP_SCN に設定されます。
DROP_TIME	DATE	一時ファイルを削除したときの時刻。同じ FILE# を持つ新規の一時ファイルが発見された場合、DROP_TIME はその一時ファイルの CREATION_TIME に設定されます。それ以外の場合、値は RC_CHECKPOINT.CKP_TIME に設定されます。
BYTES	NUMBER	一時ファイルのサイズ (バイト)。
BLOCKS	NUMBER	ファイルのサイズ (ブロック数)。
BLOCK_SIZE	NUMBER	一時ファイルのブロック・サイズ (バイト)。
NAME	VARCHAR2 (1024)	一時ファイル名。
RFILE#	NUMBER	表領域内でのこの一時ファイルの相対ファイル番号。
AUTOEXTEND	VARCHAR2 (3)	一時ファイルが自動拡張可能な場合は ON。それ以外の場合は OFF。
MAXSIZE	NUMBER	ファイルを拡張可能な最大ファイル・サイズ (ブロック数)。AUTOEXTEND が ON の場合にのみ有効です。AUTOEXTEND が OFF の場合は常に 0 です。
NEXTSIZE	NUMBER	拡張可能なファイルの増分サイズの量 (ブロック数)。AUTOEXTEND が ON の場合にのみ有効です。AUTOEXTEND が OFF の場合は常に 0 です。
BIGFILE	VARCHAR2 (3)	表領域が BIGFILE 表領域である場合は YES、そうでない場合は NO。
SITE_KEY	NUMBER	このファイルと関連付けられている Data Guard データベースの主キー。Data Guard 環境にある各データベースの SITE_KEY の値は一意です。SITE_KEY を使用して RC_SITE ビューと結合すると、データベースの DB_UNIQUE_NAME を取得できます。
TABLESPACE_CREATION_CHANGE#	NUMBER	この表領域が作成されたときの SCN。
TABLESPACE_CREATION_TIME	DATE	この表領域が作成されたときのタイムスタンプ。

---

列	データ型	説明
TABLESPACE_DROP_CHANGE#	NUMBER	この表領域が削除されたときの SCN。
TABLESPACE_DROP_TIME	DATE	この表領域が削除されたときのタイムスタンプ。
DB_UNIQUE_NAME	VARCHAR2 (512)	このレコードが属しているデータベース・インカンネーションの DB_UNIQUE_NAME。Data Guard 環境のすべてのデータベースで同一の DBID が共有されていますが、DB_UNIQUE_NAME の値は異なります。特定のファイルに対するデータベース名が不明の場合、この列の値は NULL です。たとえば、Oracle Database 11g より前のバージョンの Recovery Manager で管理されているデータベースの行は NULL です。

---

## RC\_UNUSABLE\_BACKUPFILE\_DETAILS

このビューには、UNAVAILABLE または EXPIRED とマークされているすべてのバックアップ・ファイル（バックアップ・ピース、プロキシ・コピーまたはイメージ・コピー）が表示されます。ユーザーは、行のいずれかを選択し、BTYPE\_KEY または FILETYPE\_KEY を使用して、バックアップ・セットまたは特定のファイルのステータスを AVAILABLE に変更できます。

このビューは、主に Enterprise Manager で内部的に使用します。

列	データ型	説明
DB_NAME	VARCHAR2 (8)	このレコードが属しているデータベース・インカネーションの DB_NAME。
DB_KEY	NUMBER	リカバリ・カタログ内のこのデータベースに対する主キー。この列を使用して、ほとんどのカタログ・ビューとの結合を構成します。
SESSION_KEY	NUMBER	セッションの識別子。RC_RMAN_OUTPUT および RC_RMAN_BACKUP_JOB_DETAILS との結合に使用します。
RSR_KEY	NUMBER	このファイルを作成したジョブに対応する RC_RMAN_STATUS の行に対する一意キー。
BTYPE	CHAR (9)	バックアップ・タイプ・コンテナ (BACKUPSET、IMAGECOPY または PROXYCOPY)。
BTYPE_KEY	NUMBER	バックアップ・タイプの一意の識別子。BS_KEY/COPY_KEY です。
ID1	NUMBER	バックアップ・セットとして取られるバックアップでは、この列には SET_STAMP が含まれます。プロキシ・コピーまたはイメージ・コピーのバックアップでは、この列には制御ファイルからの RECID が含まれます。
ID2	NUMBER	バックアップ・セットとして取られるバックアップでは、ID2 には SET_COUNT が含まれます。イメージ・コピーおよびプロキシ・コピーのバックアップでは、ID2 には STAMP が含まれます。
FILETYPE	VARCHAR2 (15)	このバックアップ・ファイルのタイプ。可能な値は BACKUPPIECE、COPY または PROXYCOPY です。
FILETYPE_KEY	NUMBER	ファイルがバックアップ・ピースの場合はバックアップ・ピースのキー、それ以外の場合は COPY_KEY です。このキーを使用して、ファイルのステータスを使用可能に直接変更することができます。
STATUS	VARCHAR2 (1)	U (使用不可能なバックアップ) または X (期限切れバックアップ) のいずれか。
FILESIZE	NUMBER	使用不可能なバックアップ・ファイルのサイズ (バイト)。
DEVICE_TYPE	VARCHAR2 (255)	この使用不可能なバックアップ・ファイルを格納するデバイス・タイプ。可能な値は DISK および SBT_TAPE です。
FILENAME	VARCHAR2 (1024)	ファイル名。
MEDIA	VARCHAR2 (80)	このバックアップを作成したメディア・マネージャについての詳細コメント。
MEDIA_POOL	NUMBER	バックアップを格納するメディア・プールの数。

## 使用不可になった Recovery Manager コマンド

この付録では、使用不可になった Recovery Manager 構文と、優先される構文がある場合は、その構文について説明します。

使用不可になった Recovery Manager 構文は、下位互換性のために今後のリリースでも引き続きサポートされます。たとえば、Oracle8i では SET CLONENAME コマンドが SET AUXNAME コマンドに置き換えられ、さらに、Oracle9i では SET AUXNAME コマンドが CONFIGURE AUXNAME に置き換えられました。ただし、SET CLONENAME も SET AUXNAME も Recovery Manger のすべての後続リリースで引き続き実行できます。

表 A-1 使用不可になった Recovery Manager 構文

使用不可になったリリース	使用不可になった構文	優先される現行の構文
11.1.0	CONVERT ON TARGET PLATFORM	CONVERT ON DESTINATION PLATFORM
11.1.0	UNTIL RESTORE POINT	TO RESTORE POINT
11.1.0	BACKUP ... AS STANDBY	N/A
11.1.0	... KEEP [LOGS   NOLOGS]	... KEEP
11.1.0	BLOCKRECOVER	RECOVER
10.0.1	BACKUP ... INCREMENTAL LEVEL [2,3,4]	0 および 1 以外のレベルは使用不可です。
10.0.1	BACKUP ... PARMS	CONFIGURE CHANNEL ... PARMS
10.0.1	COPY	BACKUP AS COPY
10.0.1	CREATE CATALOG TABLESPACE	CREATE CATALOG
10.0.1	LIST ... BY BACKUP [SUMMARY]	N/A
10.0.1	LIST ... VERBOSE	N/A
10.0.1	RESTORE ... PARMS	CONFIGURE CHANNEL ... PARMS
10.0.1	SEND ... PARMS	CONFIGURE CHANNEL ... PARMS
9.2	REPLICATE	RESTORE CONTROLFILE FROM ...
9.2	SET AUTOLOCATE	デフォルトで有効化されます。
9.0.1	ALLOCATE CHANNEL FOR DELETE	N/A
9.0.1	ALLOCATE CHANNEL ... TYPE	CONFIGURE CHANNEL ... DEVICE TYPE
9.0.1	ALLOCATE CHANNEL ... KBYTES	CONFIGURE CHANNEL ... MAXPIECESIZE
9.0.1	ALLOCATE CHANNEL ... READRATE	CONFIGURE CHANNEL ... RATE
9.0.1	... ARCHIVELOG ... LOGSEQ	... ARCHIVELOG ... SEQUENCE

表 A-1 使用不可になった Recovery Manager 構文 (続き)

使用不可になったリリース	使用不可になった構文	優先される現行の構文
9.0.1	BACKUP ... SETSIZE	BACKUP ... MAXSETSIZE
9.0.1	CHANGE ... CROSSCHECK	CROSSCHECK
9.0.1	CHANGE ... DELETE	DELETE
9.0.1	REPORT ... AT LOGSEQ	REPORT ... AT SEQUENCE
9.0.1	SET AUXNAME	CONFIGURE AUXNAME
9.0.1	SET DUPLEX	SET BACKUP COPIES CONFIGURE BACKUP COPIES
9.0.1	SET LIMIT CHANNEL ...	ALLOCATE CHANNEL ... CONFIGURE CHANNEL ...
9.0.1	SET SNAPSHOT	CONFIGURE SNAPSHOT
9.0.1	UNTIL LOGSEQ (3-38 ページの「 <i>untilClause</i> 」を参照)	UNTIL SEQUENCE (3-38 ページの「 <i>untilClause</i> 」を参照)
8.1.7	CONFIGURE COMPATIBLE	N/A
8.1.5	ALLOCATE CHANNEL CLONE	CONFIGURE AUXILIARY CHANNEL
8.1.5	CHANGE ... VALIDATE	CROSSCHECK
8.1.5	CLONE (「RMAN」を参照)	AUXILIARY (「RMAN」を参照)
8.1.5	CONFIGURE CLONE	CONFIGURE AUXILIARY
8.1.5	MSGLOG (「RMAN」を参照)	LOG (「RMAN」を参照)
8.1.5	RCVCAT (「RMAN」を参照)	CATALOG (「RMAN」を参照)



---

---

## Recovery Manager の互換性

この付録では、Recovery Manager (RMAN) 環境の異なるコンポーネント間の互換性の要件について説明します。内容は次のとおりです。

- Recovery Manager の互換性について
- リカバリ・カタログ・スキーマのリリースの確認
- Recovery Manager の互換性マトリックス
- リカバリ・カタログのエクスポートとインポートに関するバージョン間の互換性
- Recovery Manager の互換性: 使用例

## Recovery Manager の互換性について

表 B-1 に、Recovery Manager 環境のコンポーネントを示します。各コンポーネントには、対応するリリース番号が付いています。

**表 B-1 Recovery Manager 環境のコンポーネント**

コンポーネント	リリース番号の照会
Recovery Manager クライアント	Recovery Manager クライアントのリリース (Recovery Manager 起動時に表示される)
リカバリ・カタログ・データベース	Oracle データベースのリリース
リカバリ・カタログ・データベースにあるリカバリ・カタログ・スキーマ	リカバリ・カタログの作成に使用される Recovery Manager クライアントのリリース
ターゲット・データベース	Oracle データベースのリリース
補助データベース	Oracle データベースのリリース

たとえば、リリース 1 (9.0.1) の Recovery Manager クライアントは、次のコンポーネントとともに使用できます。

- リリース 1 (9.0.1) のターゲット・データベース
- リリース 1 (9.0.1) の補助データベース
- カタログ表が Recovery Manager リリース 1 (9.0.1) で作成されている、リリース 8.1.7 のリカバリ・カタログ・データベース

## リカバリ・カタログ・スキーマのリリースの確認

カタログ・スキーマの現行のリリースを確認するには、SQL 問合せを実行します。

1. SQL\*Plus を使用して、カタログ所有者としてリカバリ・カタログ・データベースに接続します。次に例を示します。

```
% sqlplus rman@catdb
```

2. rcver カタログ表を問い合わせます。この問合せは、次のように実行します。

```
SQL> SELECT * FROM rcver;
```

```
VERSION
-----
09.00.01.00
10.02.01.00
11.01.00.03
```

複数のリリースが表示された場合、最後の行が現行のリリースを示し、その前の行は、以前のリリースを示します。前述の例では、リカバリ・カタログ・スキーマの現行のリリースは 11.1 で、以前のリリースは 10.2 です。

リリース 10.2 以降では、rcver 出力にあるリリース番号の最後の 2 桁がパッチ・レベルを表していることに注意してください。以前のリリースでは、この 2 桁は常にゼロになります。

## Recovery Manager の互換性マトリックス

通常、Recovery Manager の互換性ルールは、次のとおりです。

- 8.X または 9.X のリカバリ・カタログ・スキーマは、リリース 8.1.X 以上の Oracle データベースで作成できます。また、10.0.1 以上のリカバリ・カタログ・スキーマは、リリース 9.0.1 以上の Oracle データベースで作成できます。
- リカバリ・カタログ・スキーマのリリースは、Recovery Manager クライアントのリリース以上である必要があります。
- リカバリ・カタログが**仮想プライベート・カタログ**（「[CREATE CATALOG](#)」を参照）の場合、それに接続する Recovery Manager クライアントのパッチ・レベルは、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。このバージョン制限は、Oracle Database 11g の基本リカバリ・カタログに接続する Recovery Manager クライアントには適用されません。基本カタログに仮想プライベート・カタログのユーザーがいる場合も同様です。
- Recovery Manager クライアントとターゲット・データベースのリリースは、同じにすることを推奨します（他の有効な組合せについては、[表 B-2](#) を参照してください）。Recovery Manager クライアントは、ターゲット・データベースまたは補助データベースより上位のリリースにすることはできません。
- Oracle9i の Recovery Manager クライアントを使用して Oracle Database 10g 以上をバックアップする場合、COMPATIBLE=10.0.0 で作成された制御ファイルをデータファイルのバックアップ・セットに含めることはできません。この状況に対処するには、制御ファイルの自動バックアップを ON に設定します。
- 補助データベース・インスタンスのリリースは、Recovery Manager クライアントのリリースと同じである必要があります。

[表 B-2](#) に、Recovery Manager コンポーネントのリリース要件を示します。リリースの前にある記号 >= は、このリリース以上のすべての Oracle データベースのリリースおよびそれらのパッチを示します。

**表 B-2 Recovery Manager の互換性一覧**

ターゲット / 補助データベース	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
8.0.6	8.0.6	>=8.1.7	>=8.0.6
8.1.7	8.0.6.1	>=8.1.7	>=8.1.7
8.1.7	8.1.7	>=8.1.7	>=Recovery Manager クライアント
8.1.7.4	8.1.7.4	>=8.1.7	8.1.7.4
8.1.7.4	8.1.7.4	>=8.1.7	>= 9.0.1.4
9.0.1	9.0.1	>=8.1.7	>=Recovery Manager クライアント
9.2.0	>=9.0.1.3 および <= ターゲット・データベースの実行可能ファイル	>=8.1.7	>=Recovery Manager クライアント
>=10.1.0	>=9.0.1.3 および <= ターゲット・データベースの実行可能ファイル	>=9.0.1	>=Recovery Manager クライアント

新しいリリースのデータベースで、古いリリースの Recovery Manager クライアントを使用する場合、新しいリリースの機能は使用できません。たとえば、Oracle9i の Recovery Manager クライアントを使用して Oracle Database 10g をバックアップする場合、フラッシュ・リカバリ領域、フラッシュバック・データベース、Recovery Manager で管理される補助インスタンスを使用した TSPITR、RESETLOGS を使用したリカバリなどの機能にはアクセスできません。

## リカバリ・カタログのエクスポートとインポートに関するバージョン間の互換性

リカバリ・カタログのデータ・ポンプ・エクスポートは、データベースの内容をバックアップする方法としてよく利用されます。データ・ポンプ・エクスポートを使用してリカバリ・カタログの論理バックアップを行う予定の場合、Oracle データベースの複数バージョンをまたいでエクスポートする場合の互換性の詳細は、『Oracle Database ユーティリティ』を参照してください。

新しいバージョンのデータベースからエクスポートした内容を、古いバージョンで稼働するデータベースにインポートすることはできません。リカバリ・カタログが必要とされる古いバージョンの Oracle データベースのリカバリ・カタログ・データは、エクスポート・ユーティリティを使用してエクスポートする必要があります。

たとえば、リリース 9.2.0.5 データベースからリカバリ・カタログのデータをエクスポートして、これを障害時リカバリの目的でリリース 8.1.7.4 の Oracle にインポートすることが予想される場合、リリース 8.1.7.4 のエクスポート・ユーティリティを指定してエクスポート操作を行う必要があります。このリリースを指定しない場合、インポート操作は失敗します。

### Recovery Manager の互換性 : 使用例

次のリリースの本番データベースをメンテナンスするとします。

- 9.2.0
- 10.2.0
- 11.1.0

これらのデータベースに関する Recovery Manager リポジトリ・データを単一のリカバリ・カタログ・データベースに記録する必要があります。B-3 ページの表 B-2 に基づいて、すべてのターゲット・データベースに対して、リリース 11.1.0 の単一のリカバリ・カタログ・データベースをリリース 11.1.0 のカタログ・スキーマとともに使用できます。各ターゲット・データベースのバックアップに使用する Recovery Manager クライアントのリリースが、次の要件に一致することを確認します。

- リリース 9.2.0 のデータベースのバックアップには、実行可能なリリース 9.2.0 の Recovery Manager を使用します。
- リリース 10.2.0 のデータベースのバックアップには、実行可能なリリース 9.2.0 または 10.2.0 の Recovery Manager を使用します。
- リリース 11.1.0 のデータベースのバックアップには、すべての実行可能な Recovery Manager を使用できます。

## 記号

---

? 記号

Recovery Manager の引用符付き文字列内, 1-3

@@ コマンド, 2-4

@ コマンド, 2-2

Recovery Manager の引用符付き文字列内, 1-3

## A

---

ADVISE FAILURE コマンド, 2-6

ALLOCATE CHANNEL コマンド, 2-10, 3-2

FOR MAINTENANCE オプション, 2-13

共有サーバー, 2-13

ALTER DATABASE コマンド, 2-17

archivelogRecordSpecifier 句, 3-6, 3-19

## B

---

BACKUP コマンド, 2-19

## C

---

CATALOG コマンド, 2-55

CHANGE コマンド, 2-60

completedTimeSpec 句, 3-10

CONFIGURE コマンド, 2-68

connectStringSpec 句, 3-12

CONNECT コマンド, 2-88

CONVERT コマンド, 2-91

CREATE CATALOG コマンド, 2-103

CREATE SCRIPT コマンド, 2-106

CROSSCHECK コマンド, 2-110

## D

---

DELETE SCRIPT コマンド, 2-120

DELETE コマンド, 2-114

DROP CATALOG コマンド, 2-122

DROP DATABASE コマンド, 2-124

DUPLICATE コマンド, 2-125

## E

---

EXECUTE SCRIPT コマンド, 2-143

EXIT コマンド, 2-146

## F

---

fileNameConversionSpec 句, 3-16

FLASHBACK コマンド, 2-147

## G

---

GRANT コマンド, 2-153

## H

---

HOST コマンド, 2-156, 2-158

## L

---

listObjList 句, 3-27

LIST コマンド, 2-161

## M

---

MAXCORRUPT パラメータ

SET コマンド, 2-258

MAXSETSIZE パラメータ

BACKUP コマンド, 2-34, 2-292

## P

---

PRINT SCRIPT コマンド, 2-183

## Q

---

QUIT コマンド, 2-185

## R

---

RC\_ARCHIVED\_LOG ビュー, 4-5

RC\_BACKUP\_ARCHIVELOG\_DETAILS ビュー, 4-7

RC\_BACKUP\_ARCHIVELOG\_SUMMARY ビュー, 4-8

RC\_BACKUP\_CONTROLFILE\_DETAILS ビュー, 4-11

RC\_BACKUP\_CONTROLFILE\_SUMMARY ビュー,  
4-12

RC\_BACKUP\_CONTROLFILE ビュー, 4-9

RC\_BACKUP\_COPY\_DETAILS ビュー, 4-13

RC\_BACKUP\_COPY\_SUMMARY ビュー, 4-15

RC\_BACKUP\_CORRUPTION ビュー, 4-16

RC\_BACKUP\_DATAFILE\_DETAILS ビュー, 4-19

RC\_BACKUP\_DATAFILE\_SUMMARY ビュー, 4-20

RC\_BACKUP\_DATAFILE ビュー, 4-17

RC\_BACKUP\_PIECE ビュー, 4-24  
 RC\_BACKUP\_REDOLOG ビュー, 4-28  
 RC\_BACKUP\_SET ビュー, 4-30  
 RC\_BACKUP\_SPFILE ビュー, 4-35  
 RC\_CHECKPOINT ビュー, 4-38  
 RC\_CONTROLFILE\_COPY ビュー, 4-39  
 RC\_COPY\_CORRUPTION ビュー, 4-41  
 RC\_DATABASE\_BLOCK\_CORRUPTION ビュー, 4-43  
 RC\_DATABASE\_INCARNATION ビュー, 2-260, 4-44  
 RC\_DATABASE ビュー, 2-260  
 RC\_DATABASE リカバリ・カタログ・ビュー, 4-42  
 RC\_DATAFILE\_COPY ビュー, 4-47  
 RC\_DATAFILE ビュー, 4-45  
 RC\_LOG\_HISTORY ビュー, 4-50  
 RC\_OFFLINE\_RANGE ビュー, 4-51  
 RC\_PROXY\_CONTROLFILE ビュー, 4-56  
 RC\_PROXY\_DATAFILE ビュー, 4-61  
 RC\_REDO\_LOG ビュー, 4-63, 4-74  
 RC\_REDO\_THREAD ビュー, 4-64  
 RC\_RESYNC ビュー, 4-66  
 RC\_RMAN\_CONFIGURATION ビュー, 4-72  
 RC\_SITE ビュー, 4-76  
 RC\_RESTORE\_POINT ビュー, 4-65  
 RC\_STORED\_SCRIPT\_LINE ビュー, 4-78  
 RC\_STORED\_SCRIPT ビュー, 4-77  
 RC\_TABLESPACE ビュー, 4-79, 4-80  
 recordSpec 句, 3-35  
 Recovery Manager  
   構文規則, 1-2  
   互換性, B-2  
   コマンド  
     @@, 2-4  
     ALLOCATE CHANNEL, 2-10, 3-2  
     ALLOCATE CHANNEL FOR MAINTENANCE,  
       2-13  
     ALTER DATABASE, 2-17  
     archiveLogRecordSpecifier 句, 3-6, 3-19  
     BACKUP, 2-19  
     CATALOG, 2-55  
     CHANGE, 2-60  
     completedTimeSpec, 3-10  
     CONFIGURE, 2-68  
     CONNECT, 2-88  
     connectStringSpec, 3-12  
     CREATE CATALOG, 2-103  
     CREATE SCRIPT, 2-106  
     CROSSCHECK, 2-110  
     DELETE, 2-114  
     DELETE SCRIPT, 2-120  
     DROP CATALOG, 2-122  
     DROP DATABASE, 2-124  
     DUPLICATE, 2-125  
     EXECUTE SCRIPT, 2-143  
     EXIT, 2-146  
     fileNameConversionSpec, 3-16  
     FLASHBACK, 2-147  
     GRANT, 2-153  
     HOST, 2-156, 2-158  
     LIST, 2-161  
     listObjList, 3-27  
     PRINT SCRIPT, 2-183  
     QUIT, 2-185

recordSpec, 3-35  
 RECOVER, 2-186  
 REGISTER, 2-200  
 RELEASE CHANNEL, 2-202  
 REPLACE SCRIPT, 2-208  
 REPORT, 2-211  
 RESET DATABASE, 2-219  
 RESTORE, 2-221  
 RESYNC CATALOG, 2-238  
 REVOKE, 2-242  
 RUN, 2-250  
 SEND, 2-252  
 SET, 2-254  
 SHOW, 2-263  
 SHUTDOWN, 2-266  
 SPOOL, 2-268  
 SQL, 2-269  
 STARTUP, 2-271  
 SWITCH, 2-273  
 TRANSPORT TABLESPACE, 2-277  
 UNREGISTER DATABASE, 2-282  
 untilClause, 3-38  
 UPGRADE CATALOG, 2-285  
 VALIDATE, 2-287  
 コマンドの日付, 3-39  
 バックアップ  
   制御ファイルの自動バックアップ, 2-80  
 Recovery Manager コマンドのコメント, 1-4  
 RECOVER コマンド, 2-186  
 REGISTER コマンド, 2-200  
 RELEASE CHANNEL コマンド (Recovery Manager),  
   2-202  
 REPAIR FAILURE コマンド, 2-204  
 REPLACE SCRIPT コマンド, 2-208  
 REPORT コマンド, 2-211  
 RESET DATABASE コマンド, 2-219  
 RESTORE コマンド, 2-221  
 RESYNC CATALOG コマンド, 2-238  
 REVOKE コマンド, 2-242  
 RMAN コマンド, 2-244  
 RUN コマンド, 2-250

## S

---

SECTION SIZE パラメータ  
   BACKUP コマンド, 2-36  
 SEND コマンド, 2-252  
 SET コマンド, 2-254  
 SHOW コマンド, 2-263  
 SHUTDOWN コマンド, 2-266  
 SPOOL コマンド, 2-268  
 SQL コマンド, 2-269  
 STARTUP コマンド, 2-271  
 Structured Query Language (SQL)  
   構文, 1-2  
 SWITCH コマンド, 2-273

## T

---

TRANSPORT TABLESPACE コマンド, 2-277

## U

---

UNREGISTER DATABASE コマンド, 2-282  
untilClause, 3-38  
UPGRADE CATALOG コマンド, 2-285

## V

---

V\$ARCHIVED\_LOG ビュー, 2-176, 3-7, 3-8  
V\$BACKUP\_DEVICE ビュー, 2-11, 2-14  
V\$BACKUP\_PIECE ビュー, 2-169, 2-170, 2-172  
V\$BACKUP\_SET ビュー, 2-168, 2-172, 2-173, 2-174  
V\$DATABASE\_BLOCK\_CORRUPTION ビュー, 2-192,  
2-193, 2-194, 2-198, 2-227, 2-291  
V\$DATABASE ビュー, 2-260  
V\$DATAFILE\_COPY ビュー, 2-175, 2-176, 3-14  
V\$DATAFILE\_HEADER ビュー, 3-14  
V\$DATAFILE ビュー, 3-14, 3-37  
V\$PROXY\_DATAFILE ビュー, 2-171  
V\$SESSION ビュー, 2-260  
VALIDATE コマンド, 2-287

## い

---

引用符付き文字列  
環境変数, 1-3

## か

---

環境変数  
Recovery Manager 文字列内, 1-3

## き

---

キーワード  
構文図, 1-3  
共有サーバー  
チャンネルの割当て, 2-13

## こ

---

構文規則、Recovery Manager, 1-2  
構文図  
キーワード, 1-3  
パラメータ, 1-3  
互換性  
Recovery Manager, B-2  
リカバリ・カタログ, B-2  
コマンドライン  
Recovery Manager の引数, 2-244  
コマンド、Recovery Manager  
@, 2-2  
@@, 2-4  
ADVISE FAILURE, 2-6  
ALLOCATE CHANNEL, 2-10, 3-2  
ALLOCATE CHANNEL FOR MAINTENANCE,  
2-13  
ALTER DATABASE, 2-17  
archivelogRecordSpecifier 句, 3-6, 3-19  
BACKUP, 2-19  
CATALOG, 2-55  
CHANGE, 2-60

completedTimeSpec 句, 3-10  
CONFIGURE, 2-68  
CONNECT, 2-88  
connectStringSpec 句, 3-12  
CONVERT, 2-91  
CREATE CATALOG, 2-103  
CREATE SCRIPT, 2-106  
CROSSCHECK, 2-110  
DELETE, 2-114  
DELETE SCRIPT, 2-120  
DROP CATALOG, 2-122  
DROP DATABASE, 2-124  
DUPLICATE, 2-125  
EXECUTE SCRIPT, 2-143  
EXIT, 2-146  
fileNameConversionSpec 句, 3-16  
FLASHBACK, 2-147  
GRANT, 2-153  
HOST, 2-156, 2-158  
LIST, 2-161  
listObjList 句, 3-27  
PRINT SCRIPT, 2-183  
QUIT, 2-185  
recordSpec, 3-35  
RECOVER, 2-186  
REGISTER, 2-200  
RELEASE CHANNEL, 2-202  
REPAIR FAILURE, 2-204  
REPLACE SCRIPT, 2-208  
REPORT, 2-211  
RESET DATABASE, 2-219  
RESTORE, 2-221  
RESYNC CATALOG, 2-238  
REVOKE, 2-242  
RMAN, 2-244  
RUN, 2-250  
SEND, 2-252  
SET, 2-254  
SHOW, 2-263  
SHUTDOWN, 2-266  
SPOOL, 2-268  
SQL, 2-269  
STARTUP, 2-271  
SWITCH, 2-273  
TRANSPORT TABLESPACE, 2-277  
UNREGISTER DATABASE, 2-282  
untilClause, 3-38  
UPGRADE CATALOG, 2-285  
VALIDATE, 2-287  
概要, 1-6, 4-2  
使用できない, A-1

## し

---

自動バックアップ  
制御ファイル, 2-80  
使用できないコマンド  
Recovery Manager, A-1  
使用例、Recovery Manager  
RESETLOG より前のバックアップのリカバリ, 2-198

## せ

---

制御ファイル

自動バックアップ, 2-80

## ち

---

チャンネル

共有サーバー・セッションへの割当て, 2-13

## て

---

データファイル

バックアップ, 2-43

未使用ブロックの圧縮, 2-43

## は

---

破損データファイル・ブロック

最大許容数, 2-258

破損の検出

SET MAXCORRUPT コマンドを使用, 2-258

バックアップ

データファイル, 2-43

未使用ブロックの圧縮, 2-43

バイナリ圧縮, 2-20, 2-44

バックアップ・セット

バイナリ圧縮, 2-20, 2-44

未使用ブロックの圧縮, 2-43

バックアップ・セットのバイナリ圧縮, 2-20, 2-44

パラメータ

構文図, 1-3

## ひ

---

日付

Recovery Manager コマンドでの指定, 3-39

ビュー、リカバリ・カタログ, 4-1

## み

---

未使用ブロックの圧縮, 2-43

## り

---

リカバリ

データベース

NOARCHIVELOG モード, 2-198

リカバリ・カタログ

ビュー, 4-1